Deep Learning: models and algorithms

Christian Wolf

July 8th, 2021

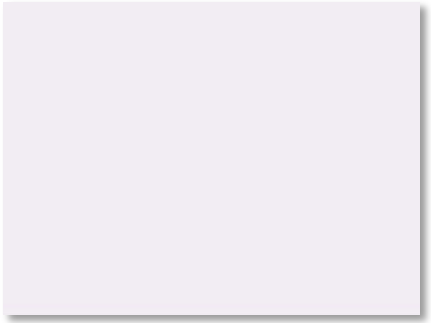
INSA LIRIS CITI

Our group

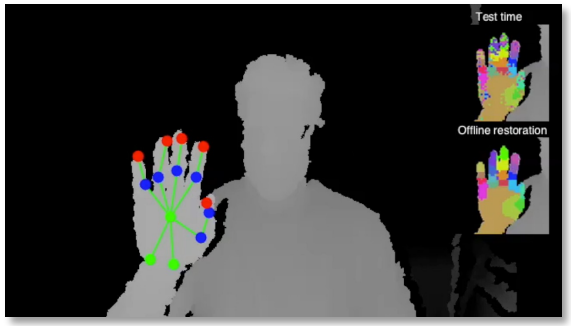
Christian Wolf
INSA-Lyon, LIRIS UMR CNRS 5205
liris.cnrs.fr/christian.wolf

The group in Feb. 2020: Corentin Kervadec, Steeven Janny, Edward Beeching, Fabien Baradel, Théo Jaunet, Quentin Possamaï.

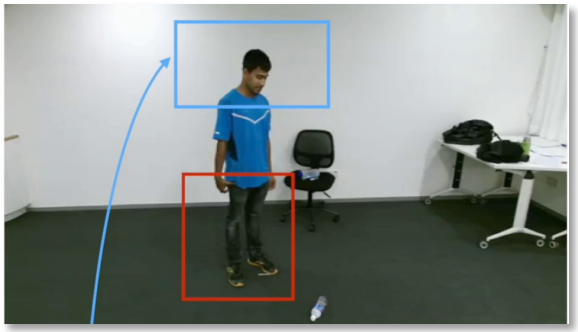
Our group: learning vision & robotics

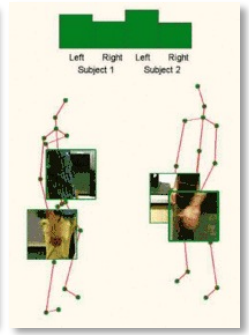


Gesture recognition

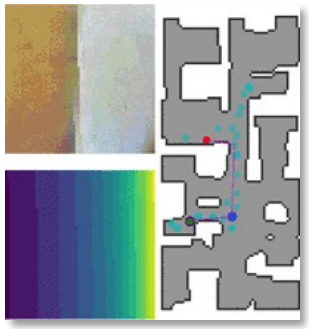


Pose estimation



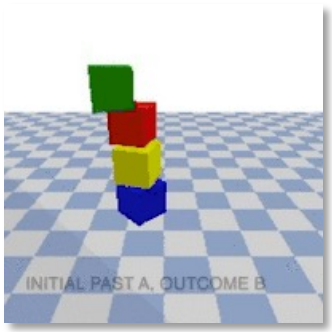


Activity Recognition



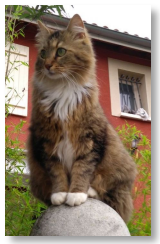
Robot Perception and Navigation

H-C Interaction



Physics

Taking decisions



→ {dog, **cat**, avocado, chair, …}

→ {0, 1, … 26, 27, **28**…, 98, 99, …}

→ {Left, right, forward, backward, …}

→ Motor control

"A blue parrot with a yellow belly sitting on a branch in a forest" →

The 3 problems of Machine Learning

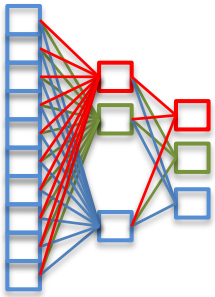
1. Expressivity
   - What is the complexity of the functions my model can represent?

2. Trainability
   - How easy is training of my model (i.e. solving the optimization problem)?

3. Generalization
   - How does my model behave on unseen data?
   - In presence of a shift in distributions?

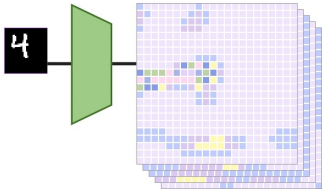
(D'après Eric Jang & Jascha Sohl-Dickstein)

The Deep Toolbox



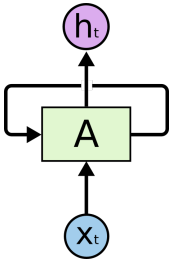
MLP

Nothing (vector space)



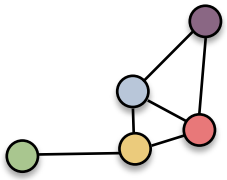
CNN / Convolutions

Translation equivariance



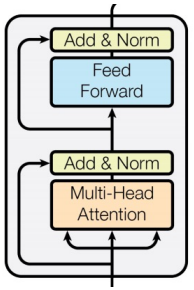
RNN / Recurrence

Sequential data, Markov property



GN, GCN / Graphs, geometry

Graph structured data

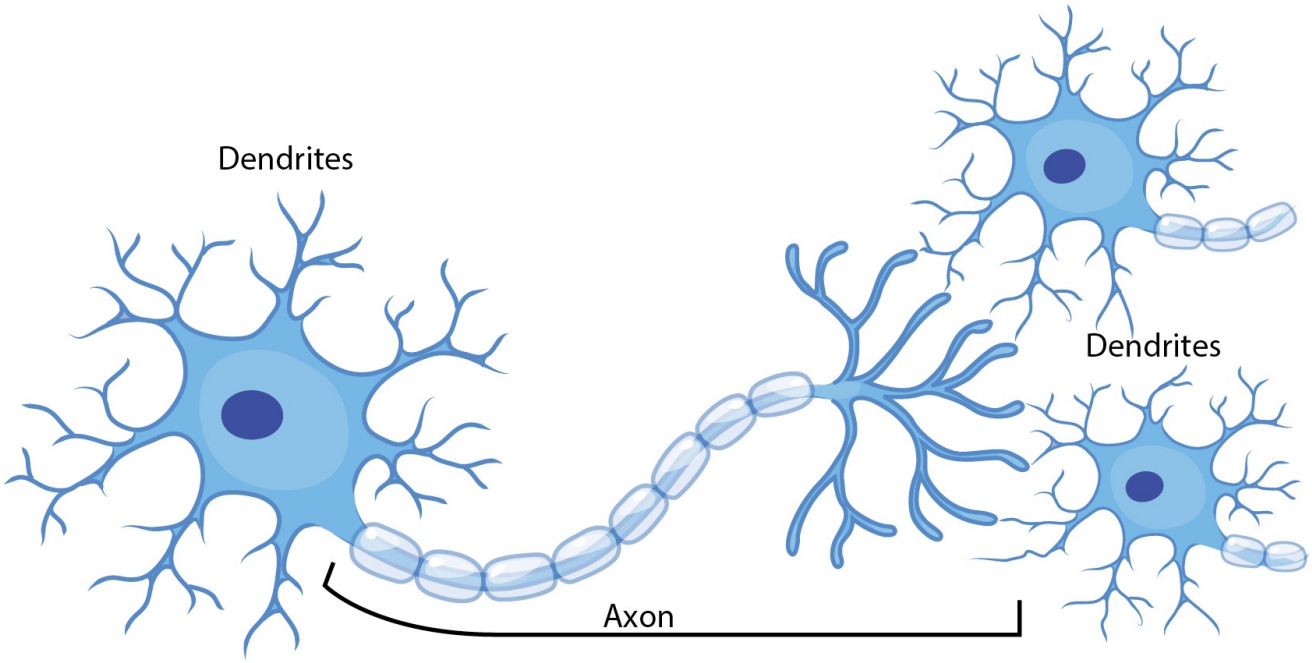


Transformers / Self-attention

Permutation equivariance

What do I know about the data and the task?

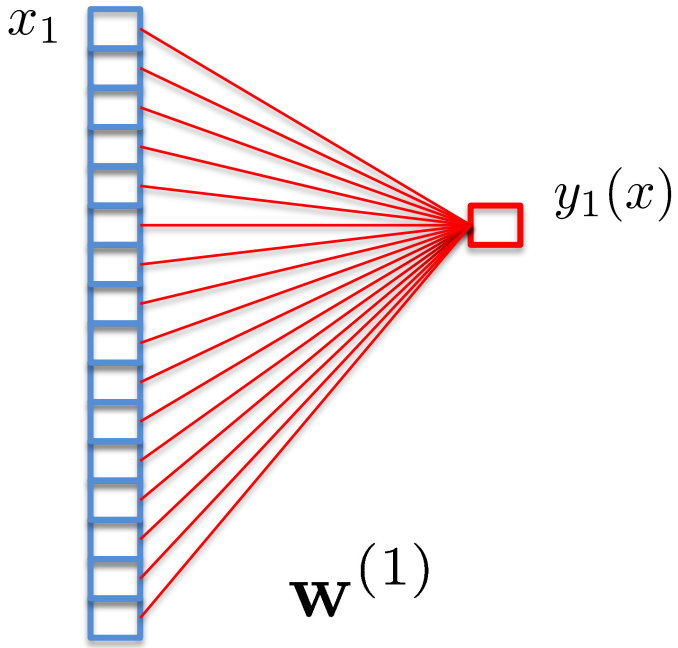
Biological neurons



Devin K. Phillips

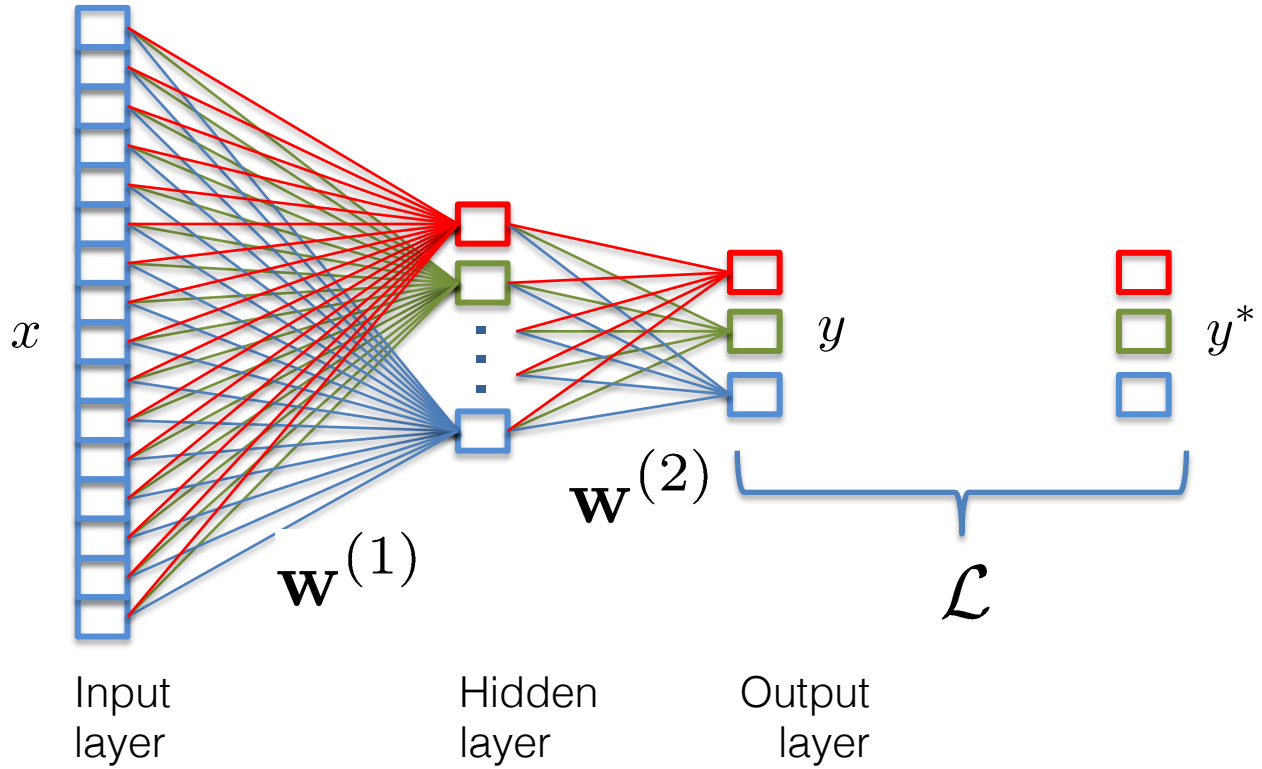
Neural networks

« Perceptron »



$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=0}^{D} \boldsymbol{w}_i \boldsymbol{x}_i$$

Deep neural networks



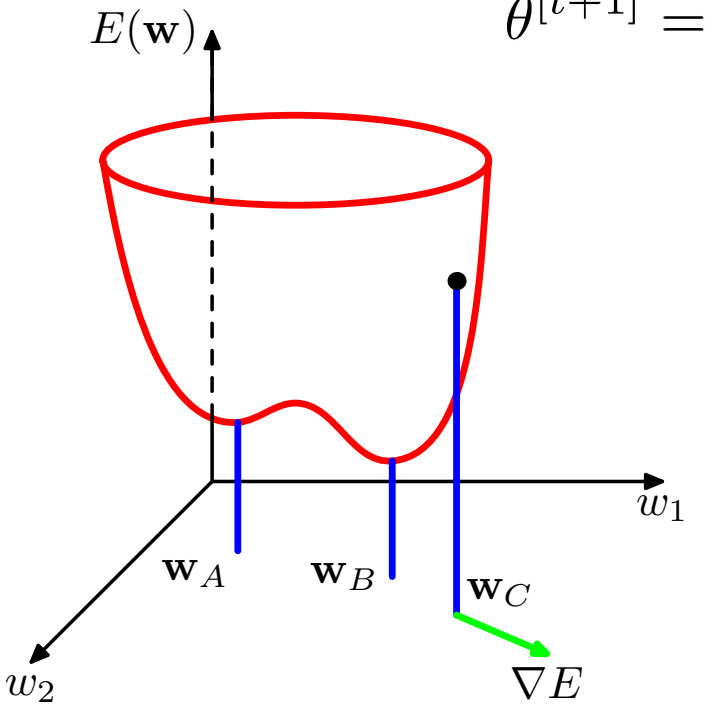
Gradient descent

Minimize the error on known data

"Empirical Risk Minimization"

$$\theta^{[t+1]} = \theta^{[t]} + \nu \nabla \mathcal{L}(h(x, \theta), y^*)$$

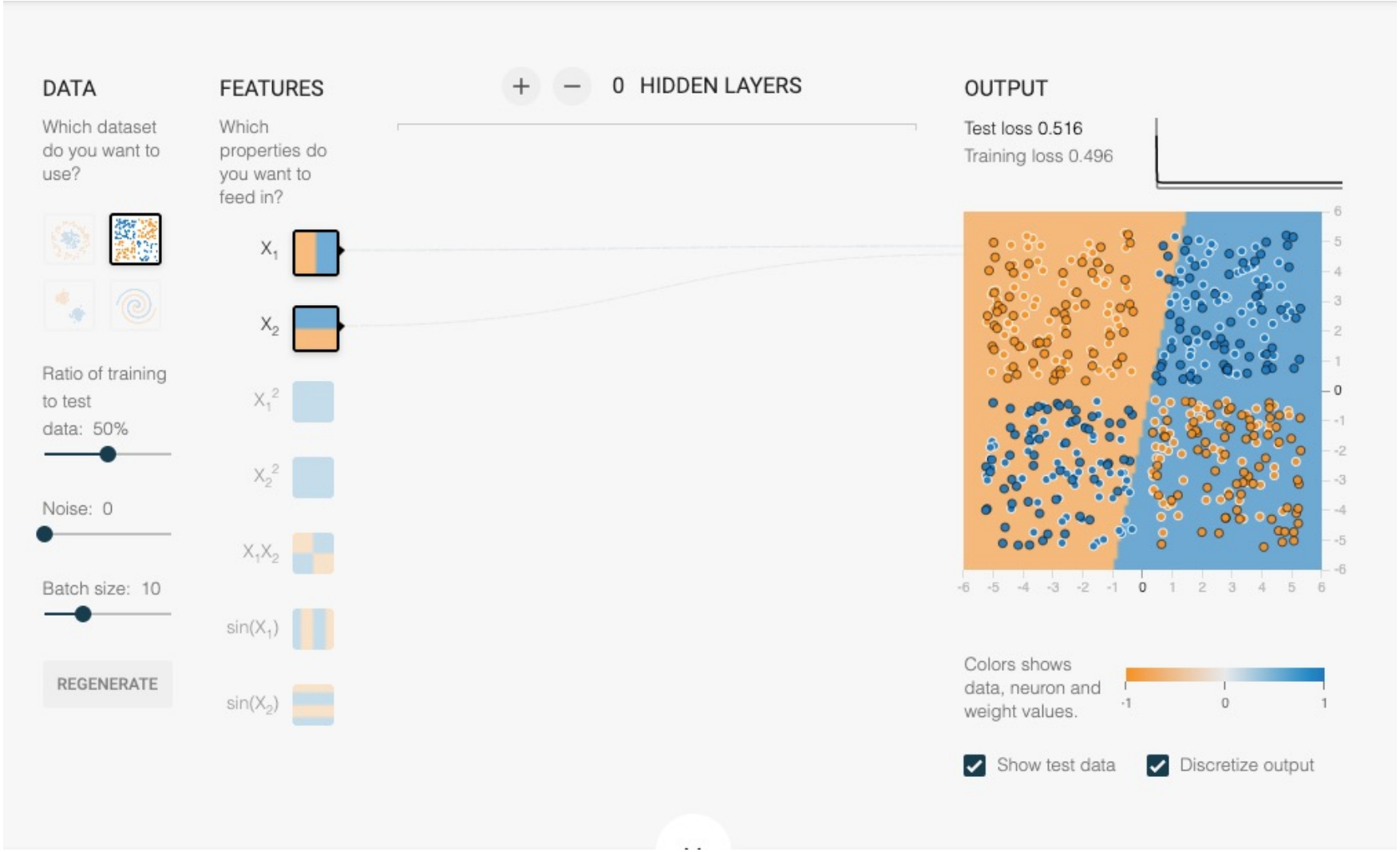
Learning rate



Can be blocked in a local minimum

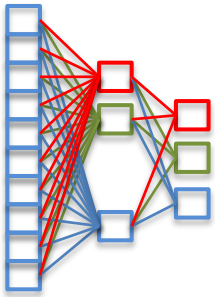
[C. Bishop, Pattern recognition and Machine learning, 2006]

Tensorflow Playground

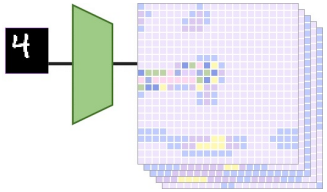


https://playground.tensorflow.org

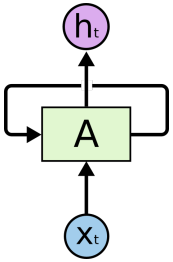
The Deep Toolbox



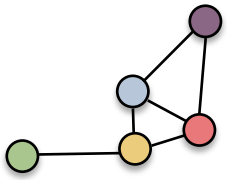
MLP



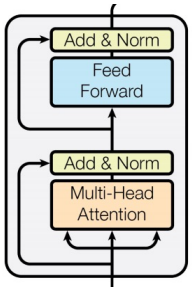
CNN / Convolutions



RNN / Recurrence



GN, GCN / Graphs, geometry



Transformers / Self-attention

What do I know about the data and the task?

Nothing (vector space)

Translation equivariance

Sequential data, Markov property

Graph structured data

Permutation equivariance

Fully connected layers might be harmful

400x400 pixels,
3 color channels

Flattened into
480 000 values

500 units

*500 * 480 000 =*
240 million parameters (for a single layer!)

Fully connected layers might be harmful

Parameters learned for one part of the image do not generalize to other parts of the image.

Translation invariance

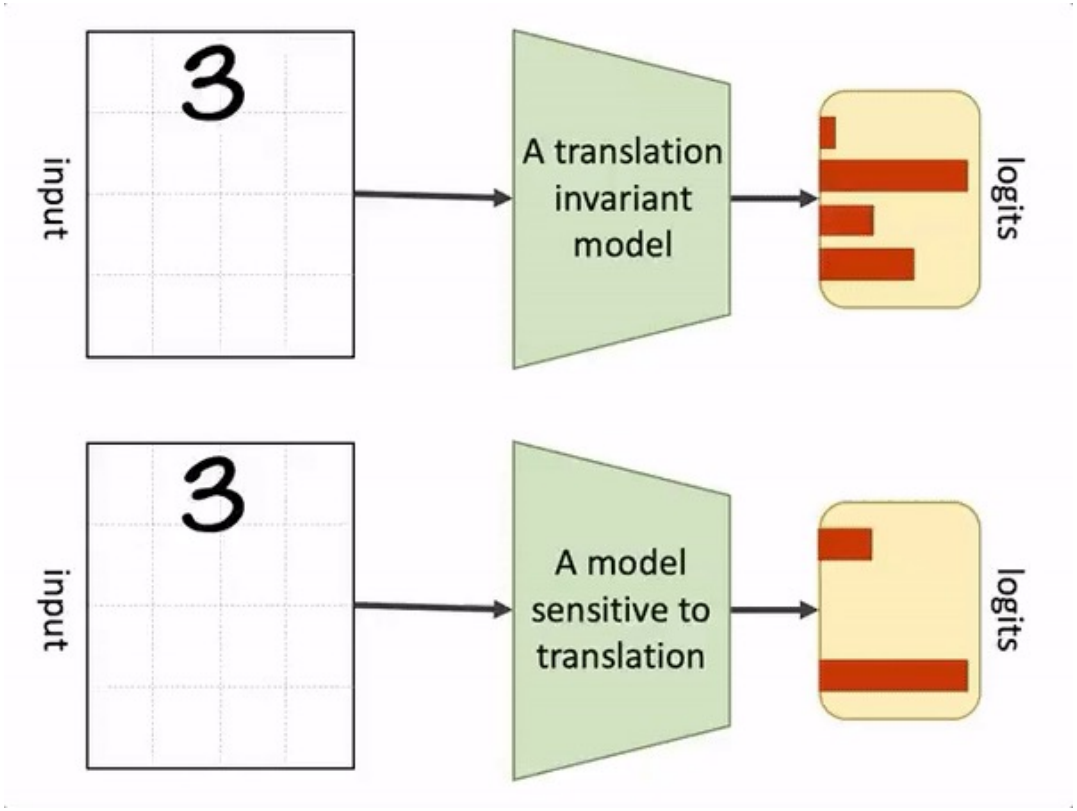
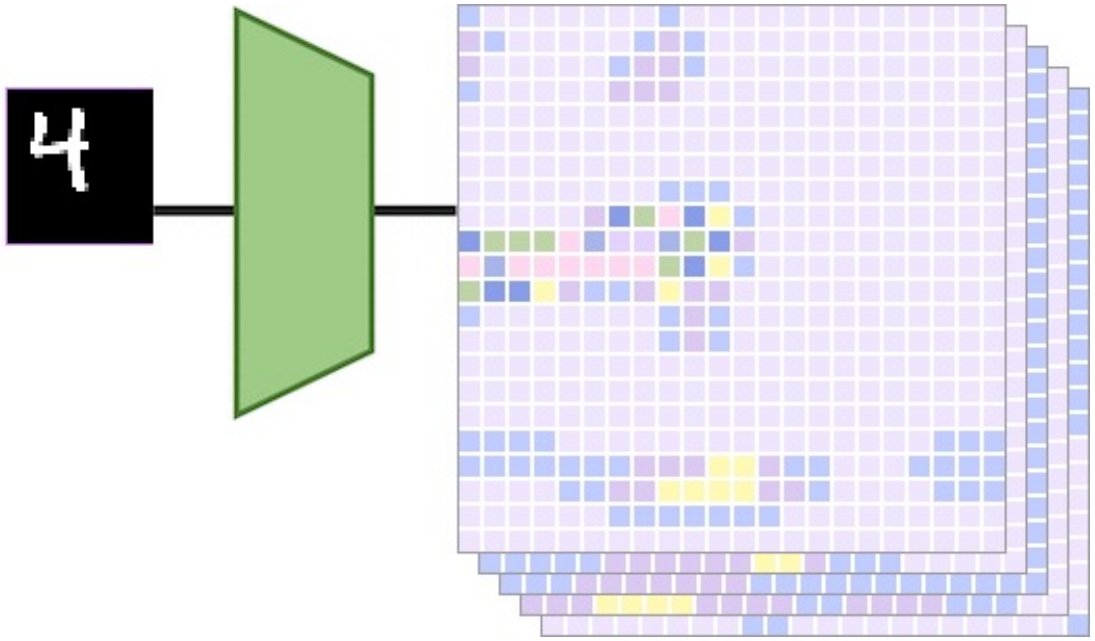


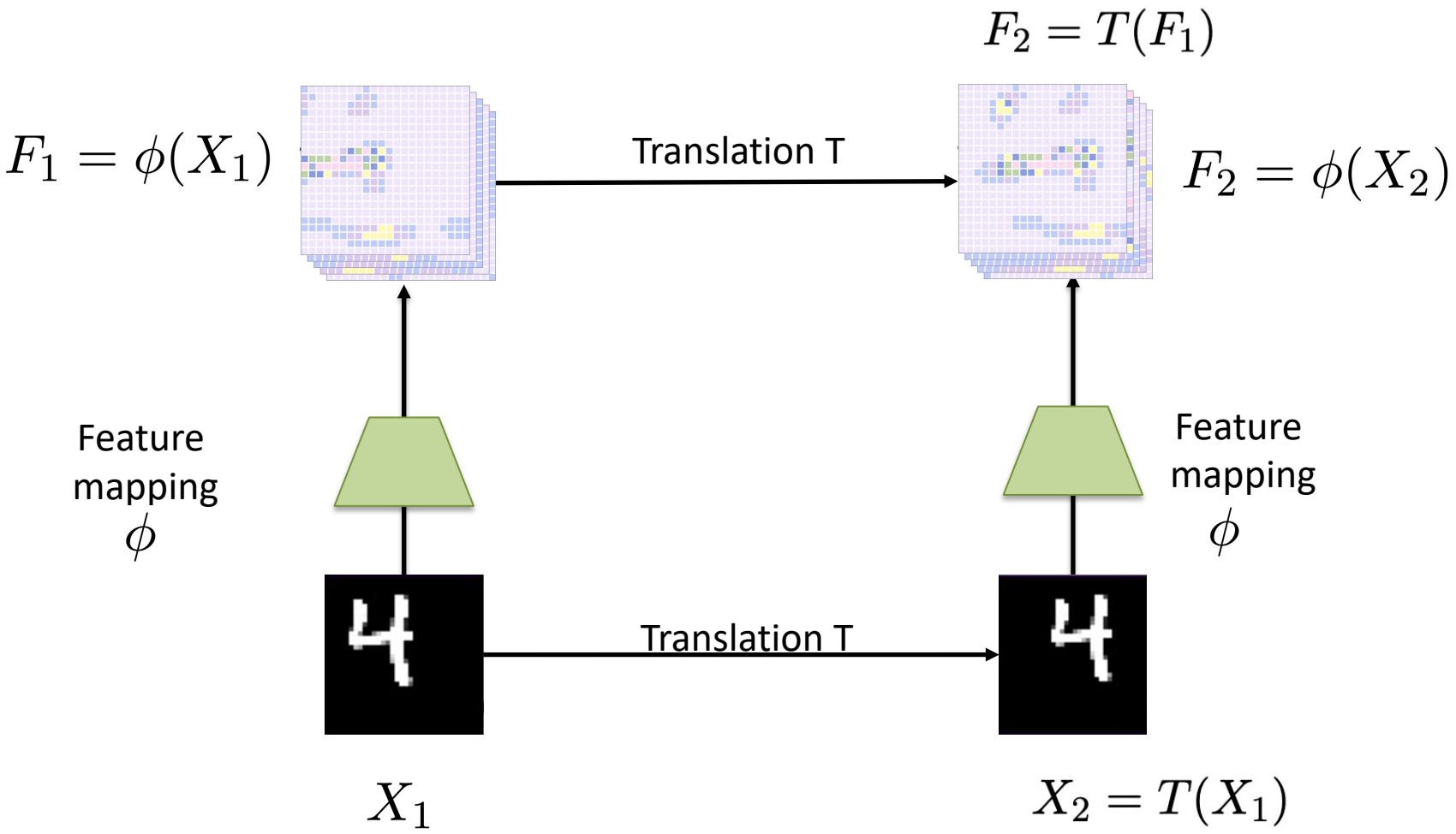
Figure by Samira Abnar (U. of Amsterdam)
https://samiraabnar.github.io/

Translation equivariance

An equivariant mapping preserves the algebric structure of the transformation.



Translation Equivariance



Shift-equivariant linear operations

Classical MLPs are sequences of linear layers followed by point-wise non-linearities. What kind of linear and shift-equivariant operators can there exist?

Let's consider an operator $\varnothing$ with following properties:

Linearity:

$$\phi\left(\alpha f + \beta f'\right) = \alpha\phi(f) + \beta\phi\left(f'\right)$$

Shift-equivariance:

$$\phi\left({}^{m,n}S(f)\right) = {}^{m,n}\,S(\phi(f))$$

where ${}^{m,n}S(f)$ shifts a signal by ^{m,n}.

Impulse response h:

$$h = \phi({}^{0,0}p)$$

where ${}^{0,0}p$ is a Dirac impulse centered at $0, 0$.

Shift-equivariant linear operations

We decompose the signal f into a series of Diracs ${}^{m,n}p$:

$$[\phi(f)](x,y) = \left[\phi\left(\sum_{m=-M/2}^{M/2}\sum_{n=-N/2}^{N/2} f(m,n)\,{}^{m,n}p\right)\right](x,y)$$

Dirac at position m,n

We use **linearity**:

$$= \left[\sum_{m=-M/2}^{M/2}\sum_{n=-N/2}^{N/2} f(m,n)\,\phi\left({}^{m,n}p\right)\right](x,y)$$

We use **shift-equivariance**:

$$= \left[\sum_{m=-M/2}^{M/2}\sum_{n=-N/2}^{N/2} f(m,n)\,\phi\left({}^{m,n}S\left({}^{0,0}p\right)\right)\right](x,y)$$

We use **linearity** again:

$$= \left[\sum_{m=-M/2}^{M/2}\sum_{n=-N/2}^{N/2} f(m,n)\,{}^{m,n}S\left(\phi\left({}^{0,0}p\right)\right)\right](x,y)$$

Shift-equivariant linear operations

Change in notation: we replace $\phi({}^{0,0}p)$ by h (through definition):

$$[\phi(f)](x,y) = \left[\sum_{m=-M/2}^{M/2} \sum_{n=-N/2}^{N/2} f(m,n)^{m,n} S(h) \right] (x,y)$$

Change in notation: make the shift-operator explicit:

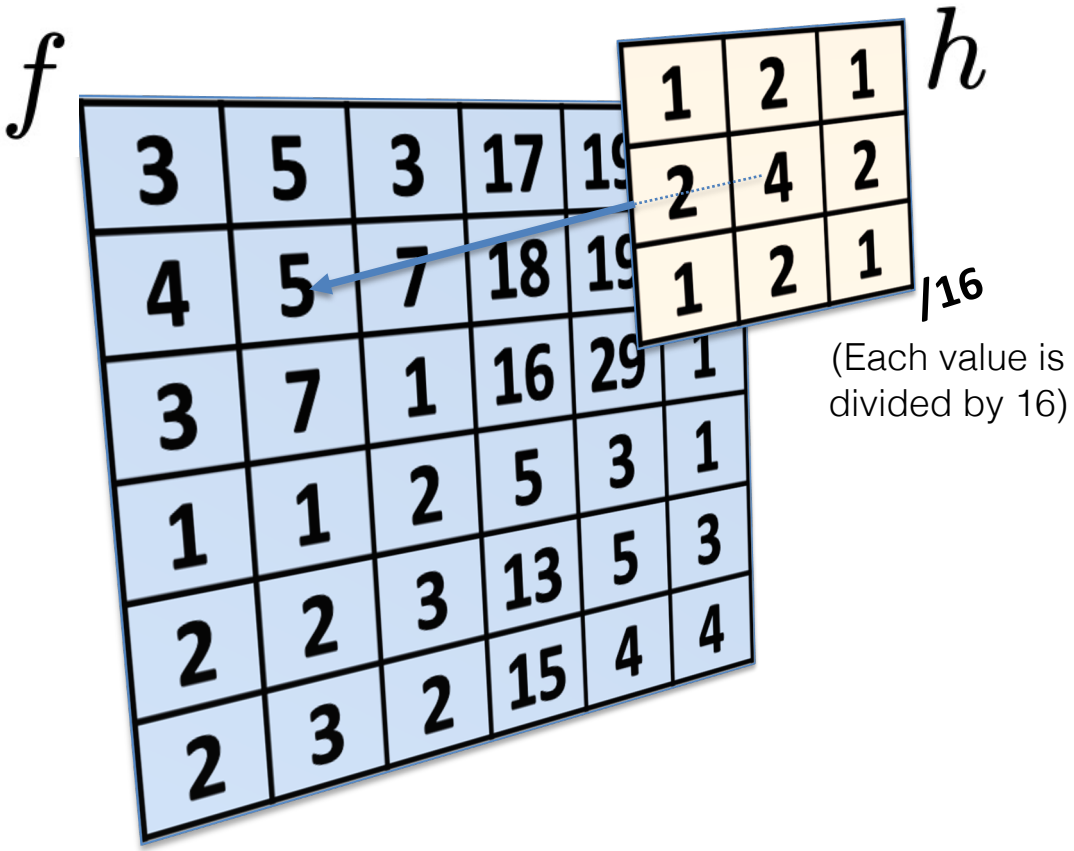
$$= \sum_{m=-M/2}^{M/2} \sum_{n=-N/2}^{N/2} f(m,n) h(x-m, y-n)$$

Change of variables:

$$= \sum_{m'=-M/2}^{M/2} \sum_{n'=-N/2}^{N/2} f\left(x-m', y-n'\right) h\left(m', n'\right)$$

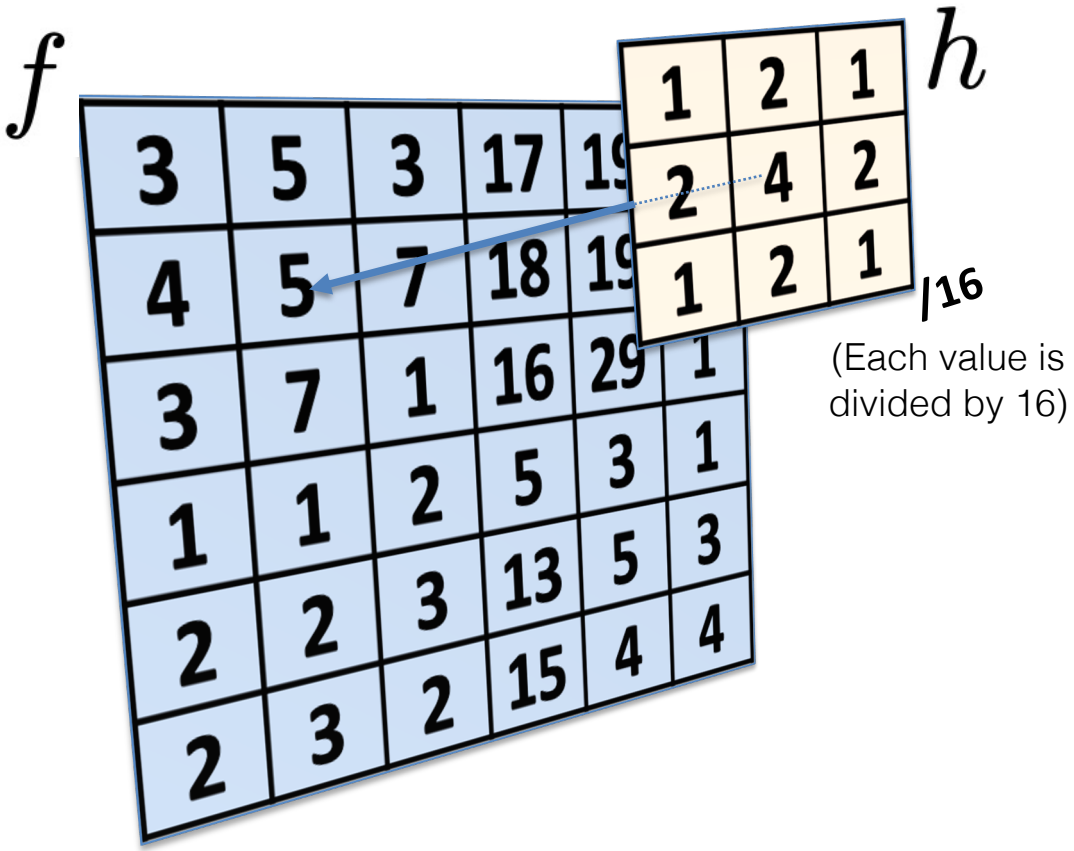
$(m' = x - m, n' = y - n) \Rightarrow$ we get a convolution!

Convolutions

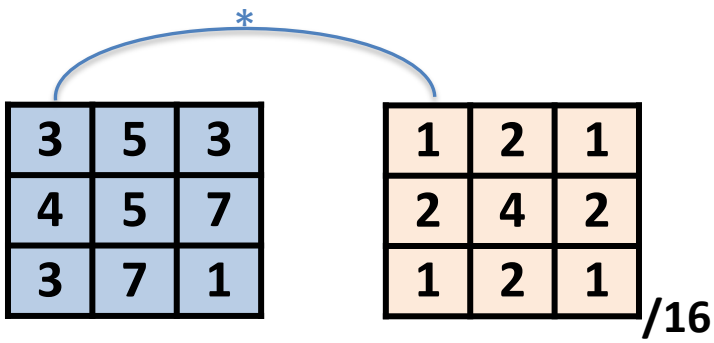


(Each value is divided by 16)

Convolutions

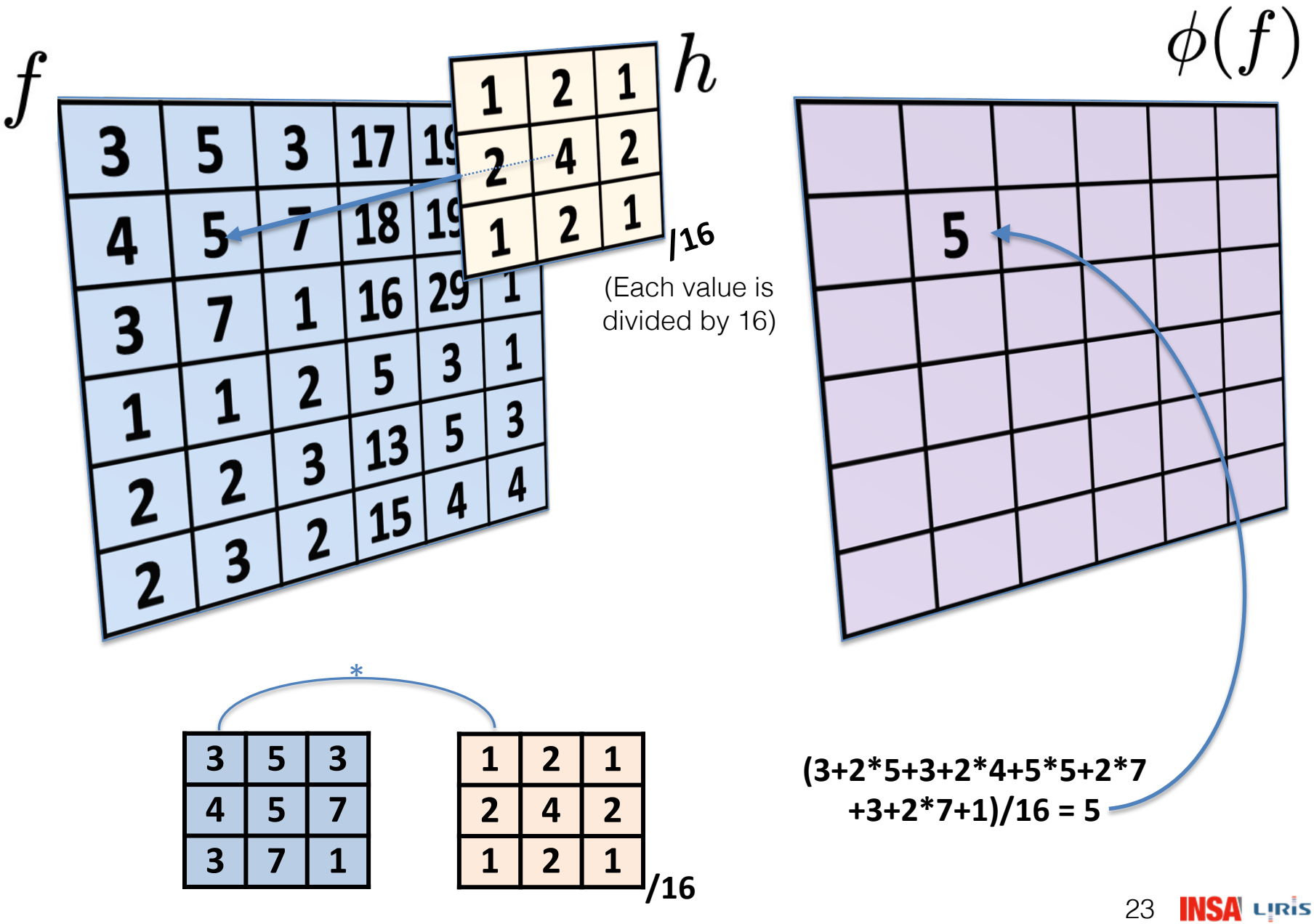


(Each value is divided by 16)

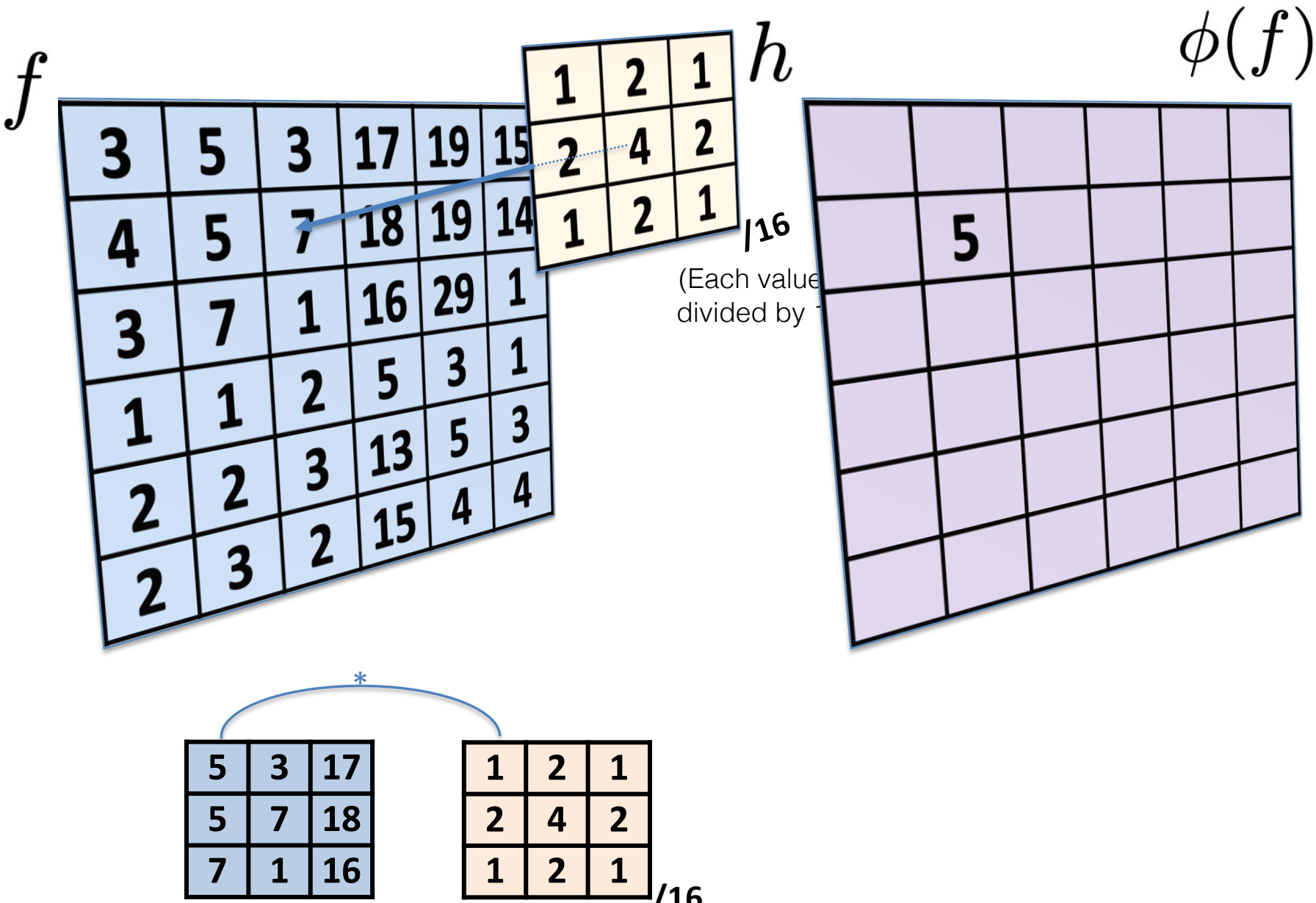


(3+2*5+3+2*4+5*5+2*7 +3+2*7+1)/16 = 5

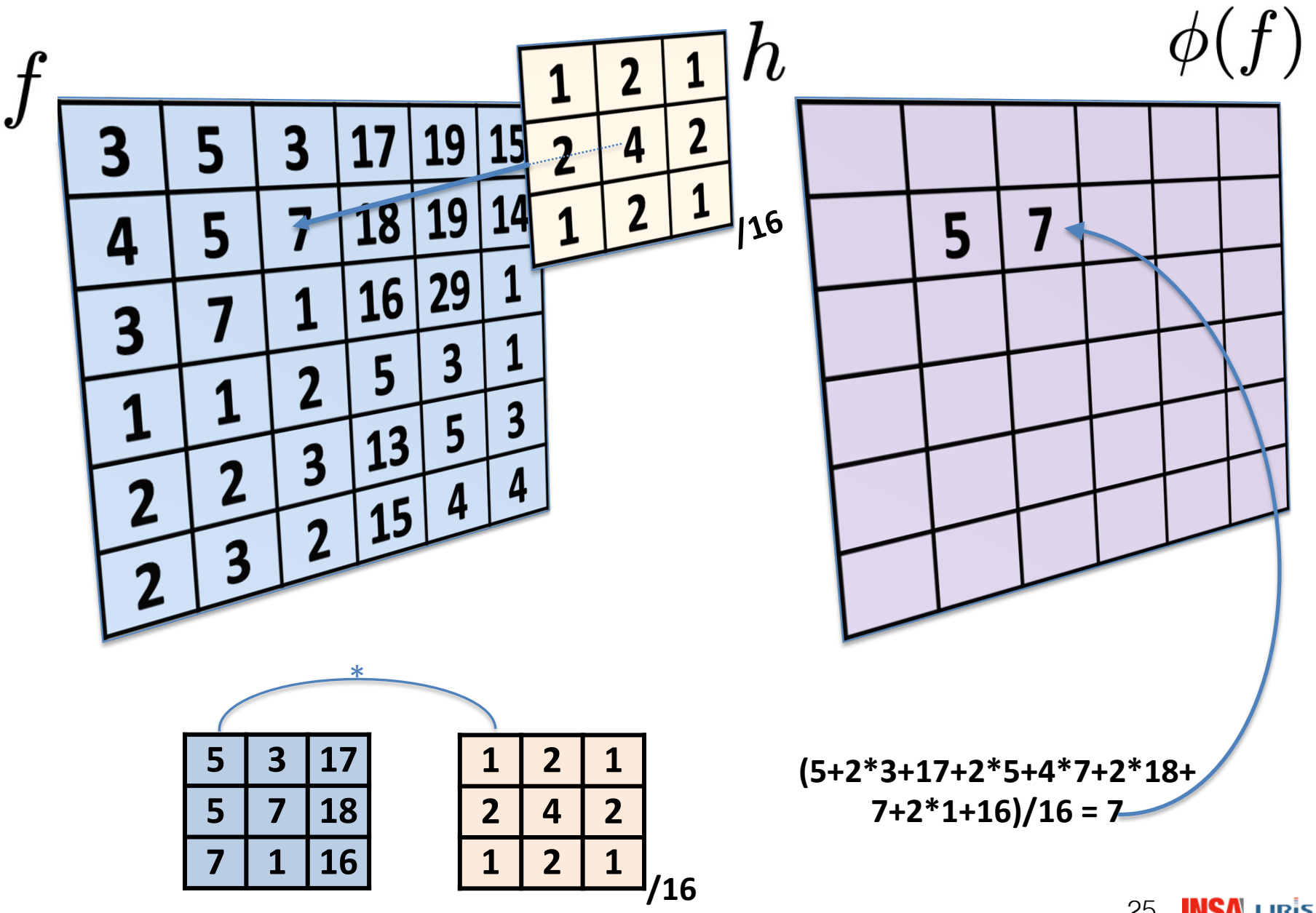
Convolutions



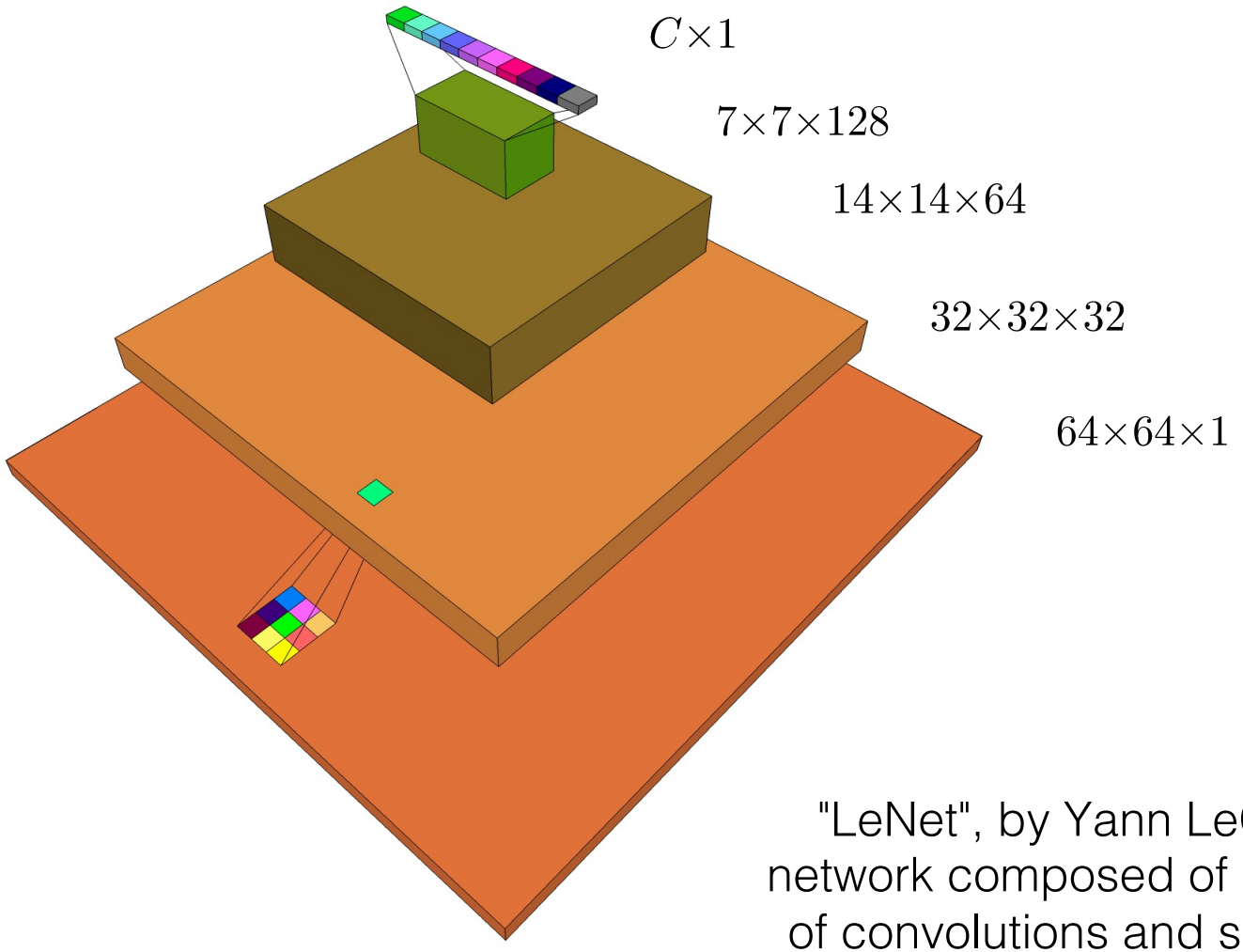
Convolutions



Convolutions

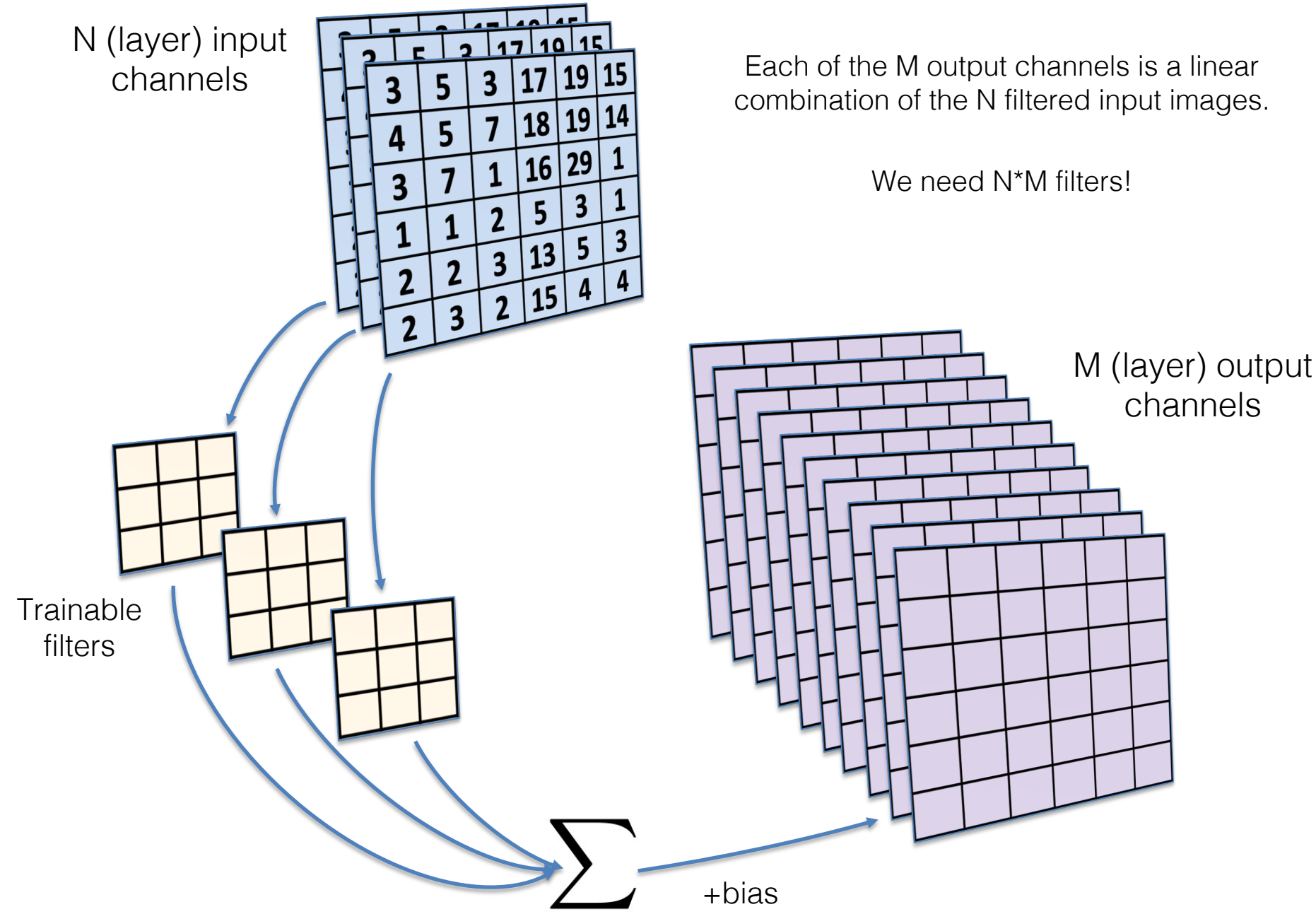


« LeNet »



"LeNet", by Yann LeCun, is a network composed of sequences of convolutions and spatial size reductions (« pooling »).

[LeCun et al., *1998*]


N (layer) input channels
3 5 3 17 19 15
4 5 7 18 19 14
3 7 1 16 29 1
1 1 2 5 3 1
2 2 3 13 5 3
2 3 2 15 4 4
Each of the M output channels is a linear combination of the N filtered input images.
We need N*M filters!
M (layer) output channels
Trainable filters
+bias

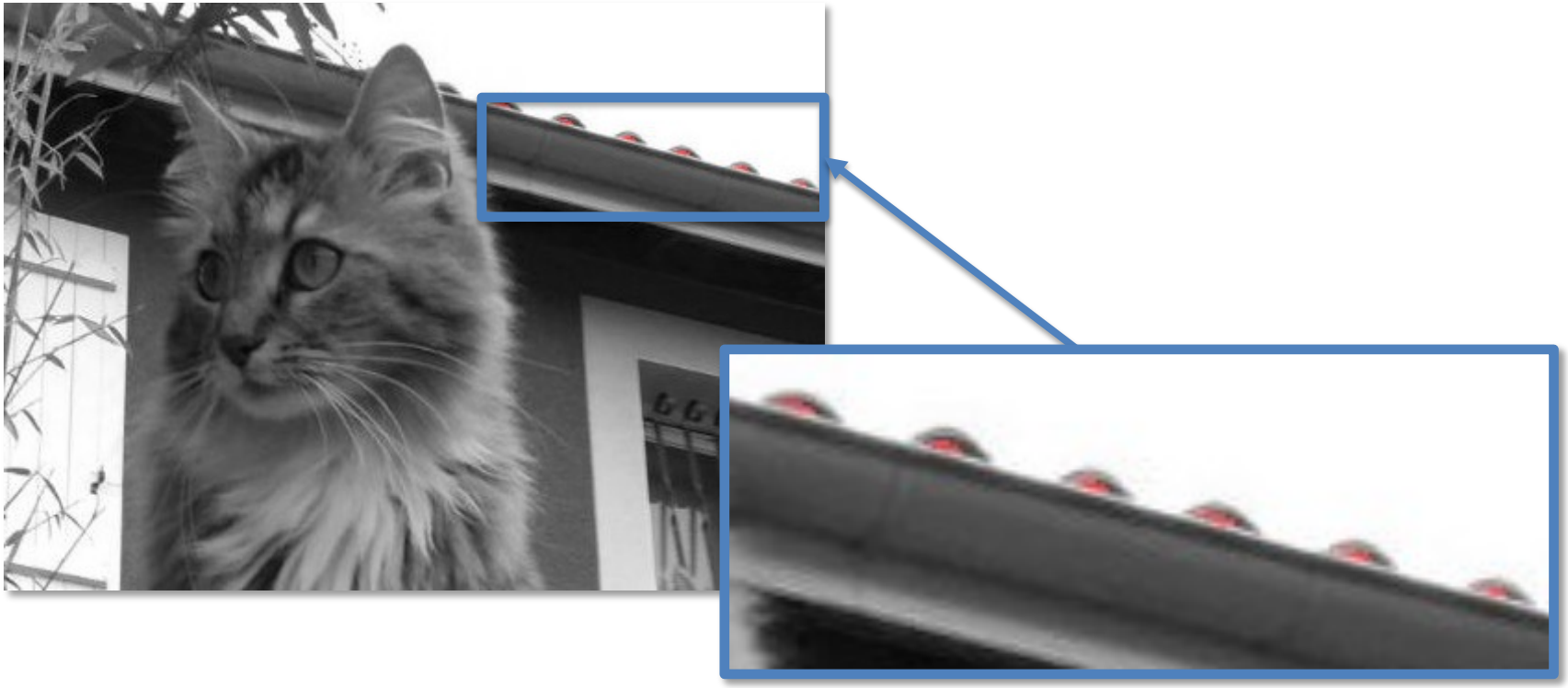
Template matching

Input image:

Filter kernel

- We normalize input and kernel (subtract mean, divide by standard deviation).
- We convolve the input with the kernel
- We threshold response and superimpose it on the red channel.

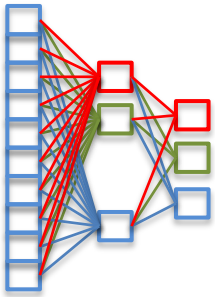
Template matching



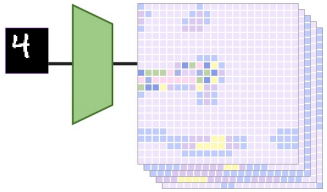
Convolutions can match patterns:

- either in input space
- or in intermediate network layers (which still have a spatial organization).

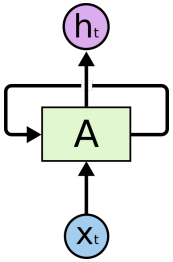
The Deep Toolbox



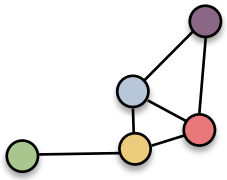
MLP



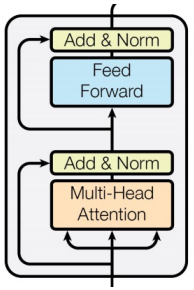
CNN / Convolutions



RNN / Recurrence



GN, GCN / Graphs, geometry



Transformers / Self-attention

What do I know about the data and the task?

Nothing (vector space)

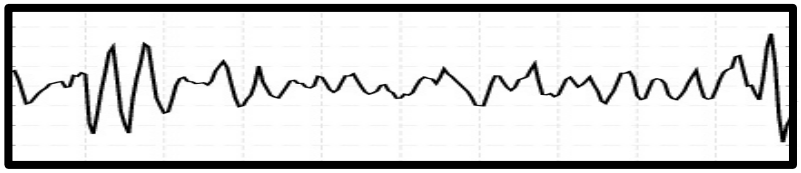
Translation equivariance

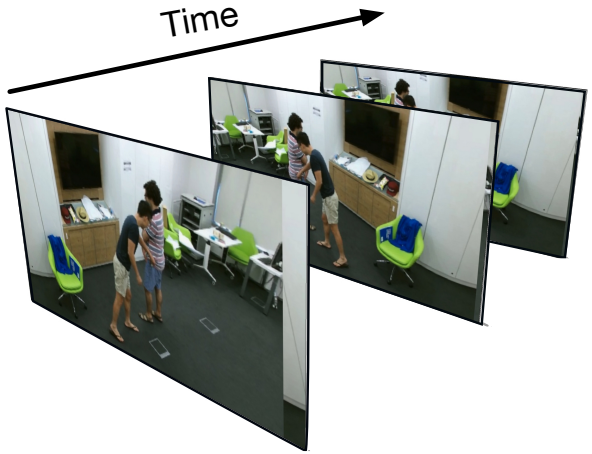
Sequential data, Markov property

Graph structured data

Permutation equivariance

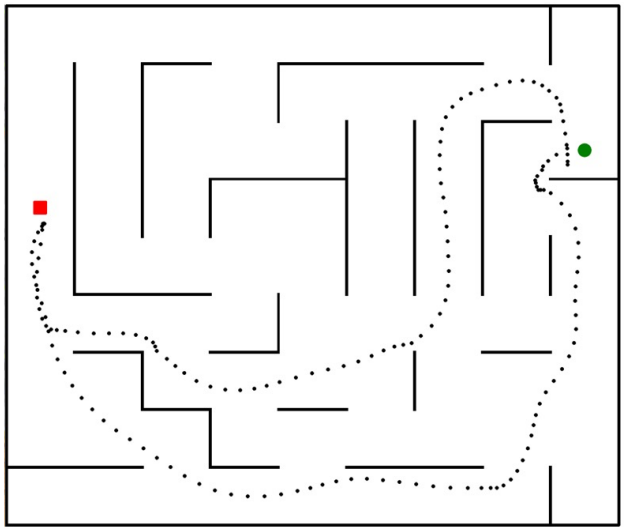
Problem: dealing with sequences





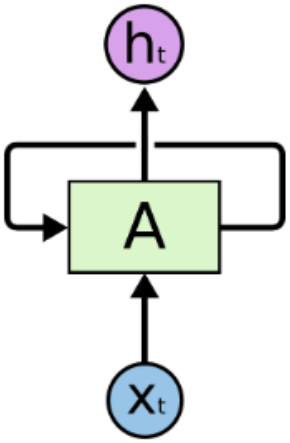
Toutes les familles heureuses se ressemblent. Chaque famille malheureuse, au contraire, l'est à sa façon.

Happy families are all alike. Every unhappy family is unhappy in its own way.



Recurrent Neural Networks (RNNs)

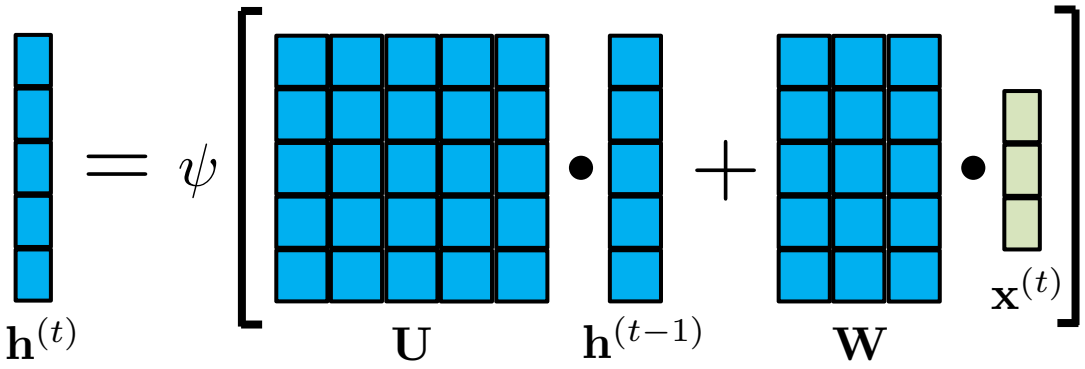
As feed-forward networks, Recurrent Neural Networks (RNNs) predict some output from a given input.
However, they also pass information over time, from instant $(t-1)$ to (t):

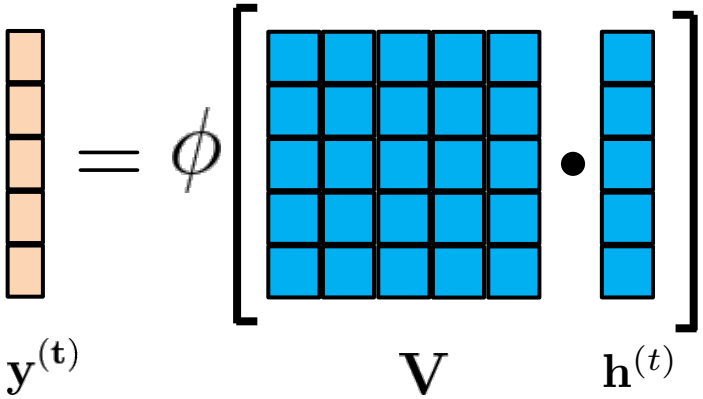


Here, we write h_t for the output, since these networks can be stacked into multiple layers, i.e. h_t is input into a new layer.

`Chris Olah, https://colah.github.io/posts/2015-08-Understanding-LSTMs/`

From state h to output y





Toy example with handcrafted parameters

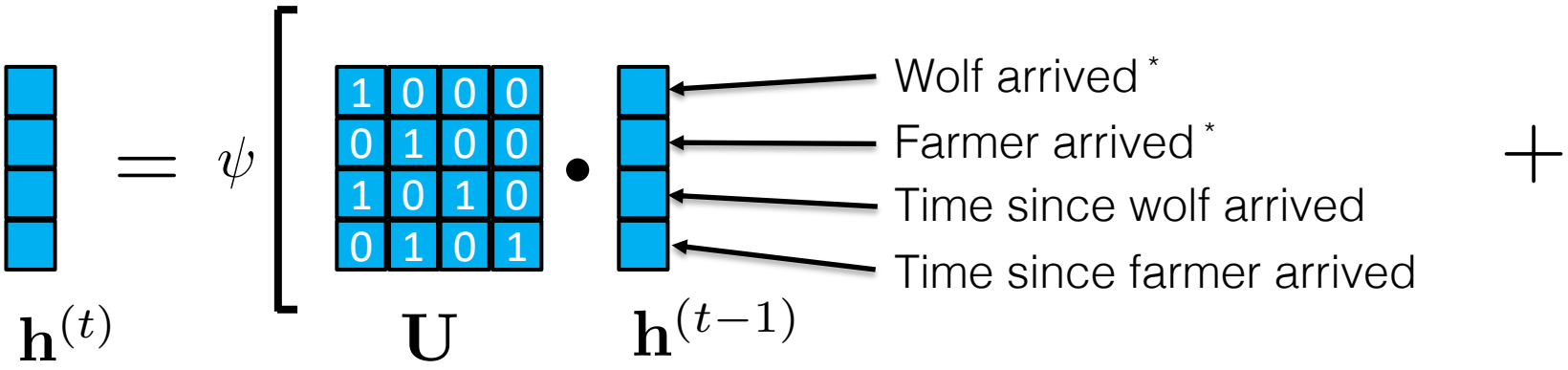
In a sequential problem, we surveil a farm and watch for the appearance of objects. At each instant t we oberve a vector 🎚 which can indicate the apparence on the scene of

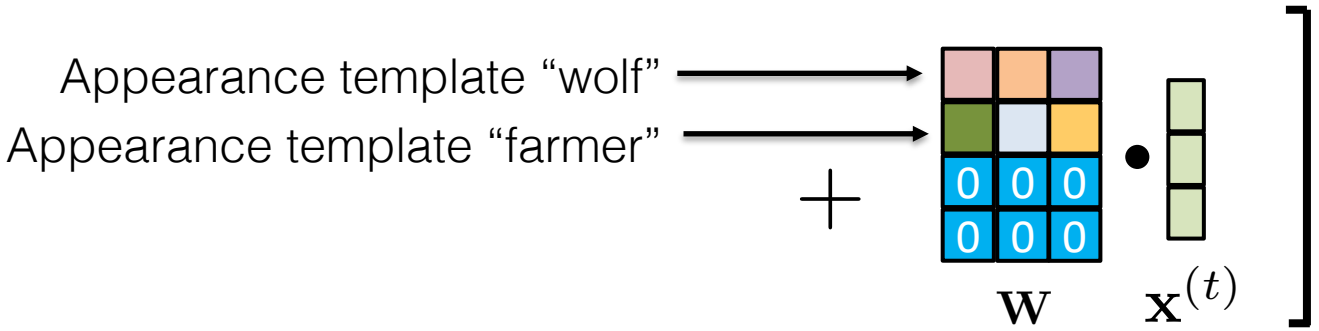
- a wolf 
- a farmer 

The objective is to output an estimate of danger, i.e.

- presence of the wolf w/o the farmer, or
- presence of both, arrival of the wolf before the farmer.

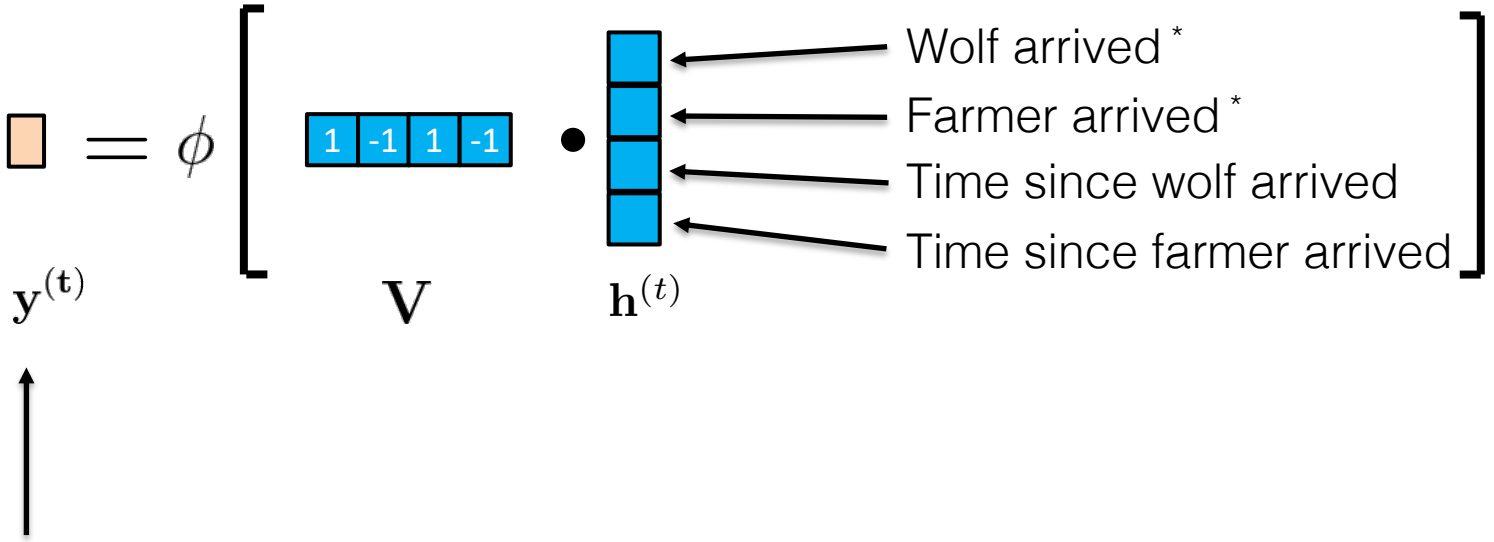
Toy example with handcrafted parameters





*Appropriate activation functions required to normalize state values

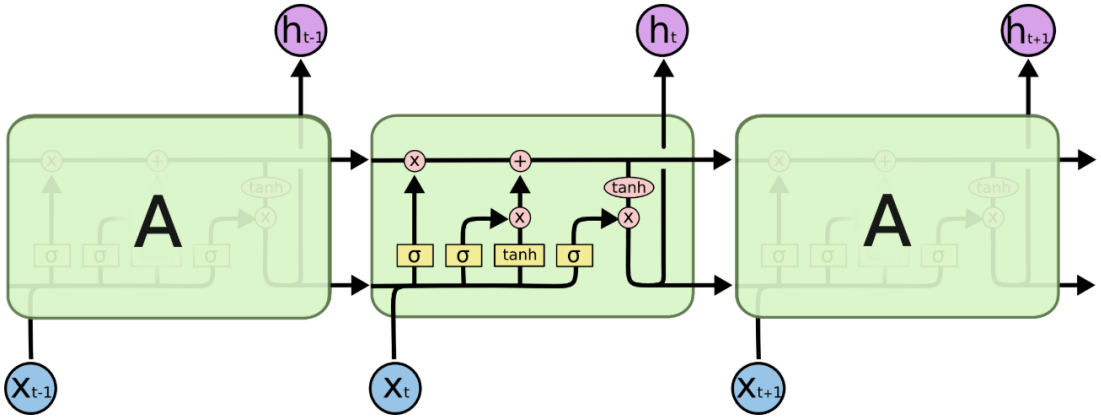
Toy example with handcrafted parameters



Output: danger!

LSTM Networks

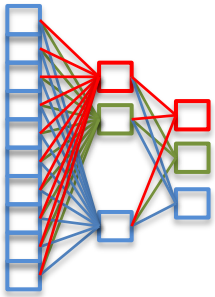
LSTM (=Long-short term Memory) networks use gating mechanisms, which handle information flow in a fully trainable way.



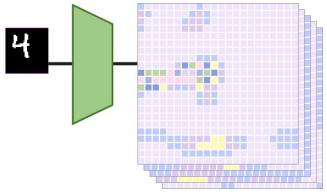
$$
\begin{aligned}
f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \\
i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \\
\tilde{C}_t &= \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \\
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
o_t &= \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right) \\
h_t &= o_t * \tanh\left(C_t\right)
\end{aligned}
$$

Chris Olah, https://colah.github.io/posts/2015-08-Understanding-LSTMs/

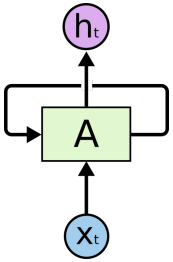
The Deep Toolbox



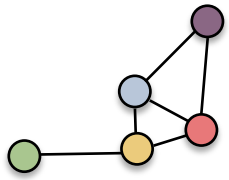
MLP



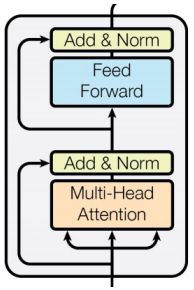
CNN / Convolutions



RNN / Recurrence



GN, GCN / Graphs, geometry



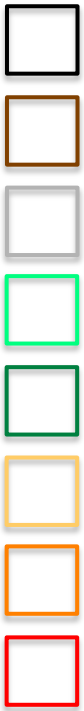
Transformers / Self-attention

What do I know about the data and the task?

Nothing (vector space)	*Translation equivariance*	*Sequential data, Markov property*	*Graph structured data*	*Permutation equivariance*

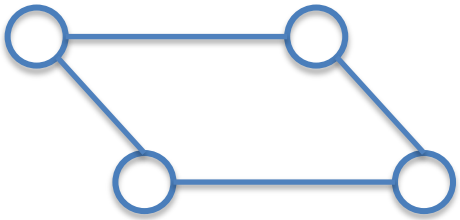
Structured Input and/or structured Output

Predicting for multiple inter-dependent variables

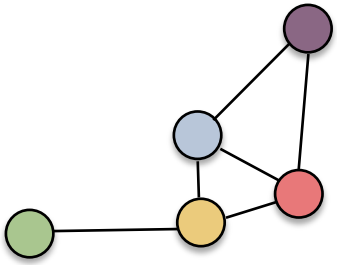


Multi-label Problems

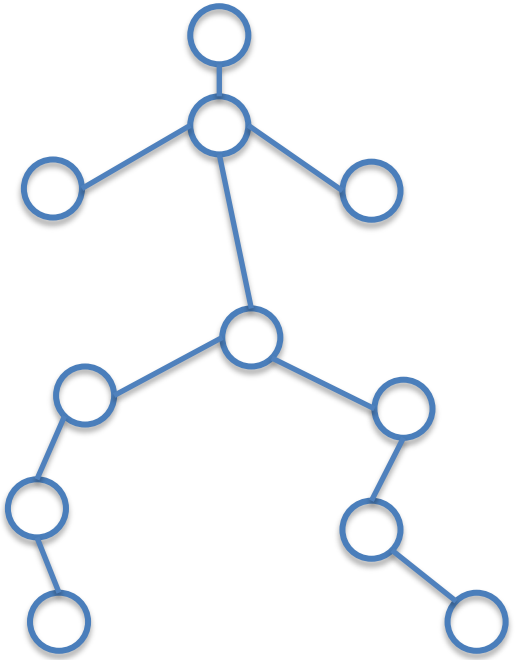
Sequences



Images and other 2D grids



General graphs

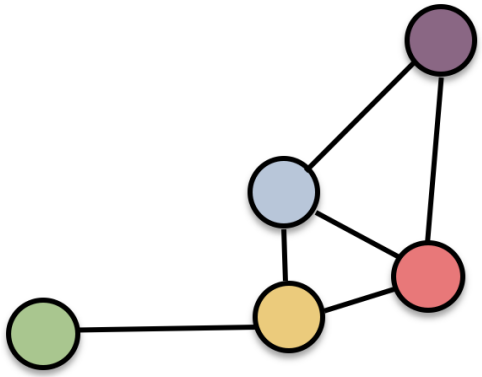


Kinematic trees

Graphs: definition

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of:

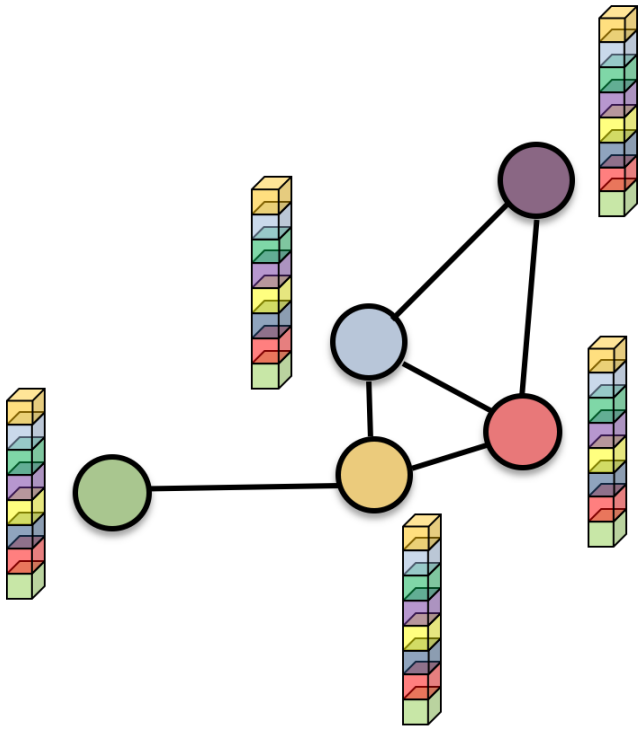
- a set $\mathcal{V}$ of nodes, and
- a set $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ of edges



Attributed Graphs

Attributed graphs also have values for each node: $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathcal{X})$:

- a set $\mathcal{V}$ of nodes, and
- a set $\mathcal{E} \in \mathcal{V}\times\mathcal{V}$ of edges
- a set $\mathcal{X} = \{x_0, \ldots, x_N\}$ of values, each value being associated to a node. In our case we find **embeddings** $x_i \in \mathbb{R}^d$ for each node.



Relational Reasoning

$$g(x_1, x_2, \ldots, x_N) = \max(h(x_1), h(x_2), \ldots, h(x_N))$$

"PointNet" [Qi, Su, Mo, Guibas, CVPR 2017]

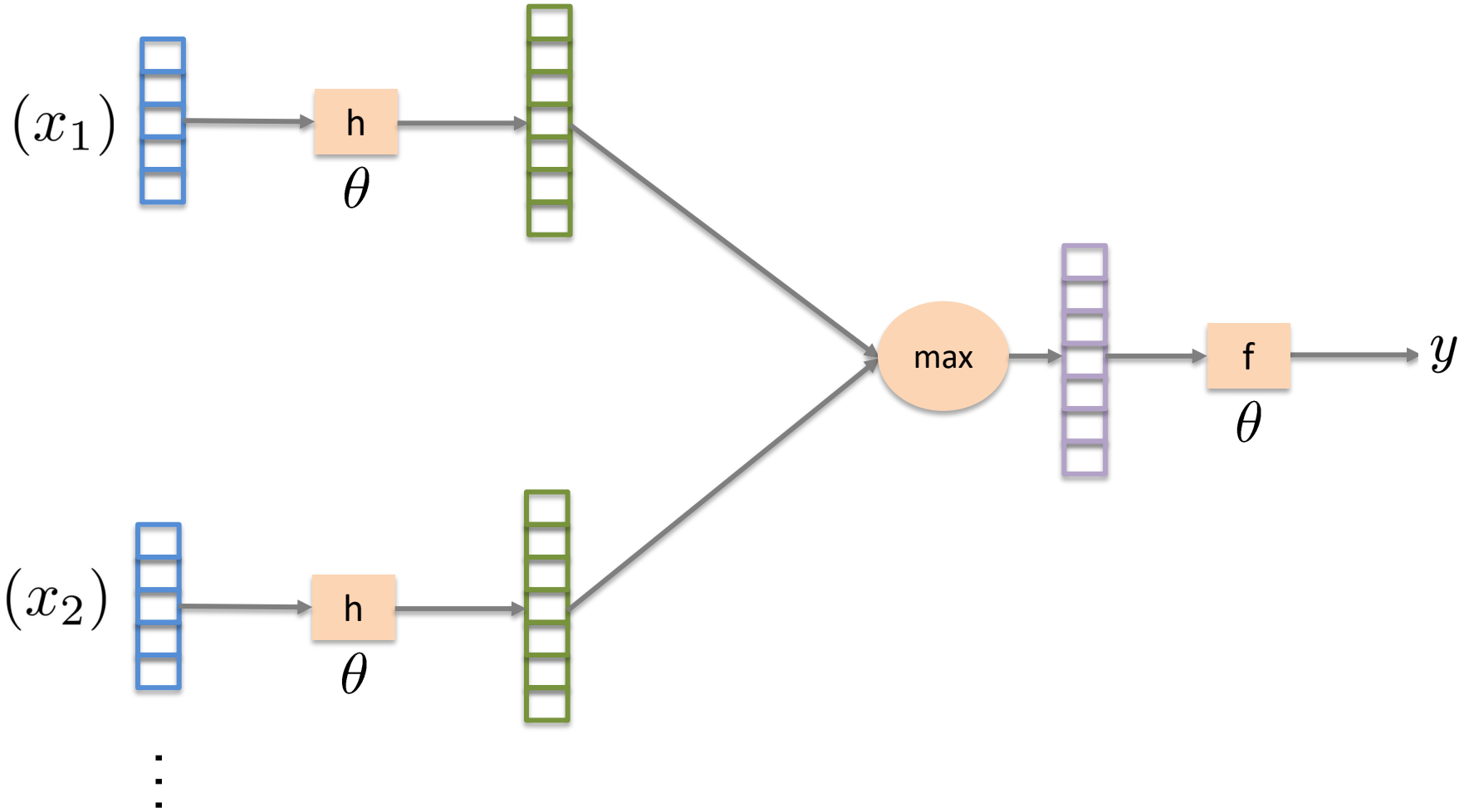
Defined over points of a point cloud

$$g(x_1, x_2, \ldots, x_N) = \sum_{i,j} h(x_i, x_j)$$

"Relational Reasoning" [Santoro et al., NIPS 2017]

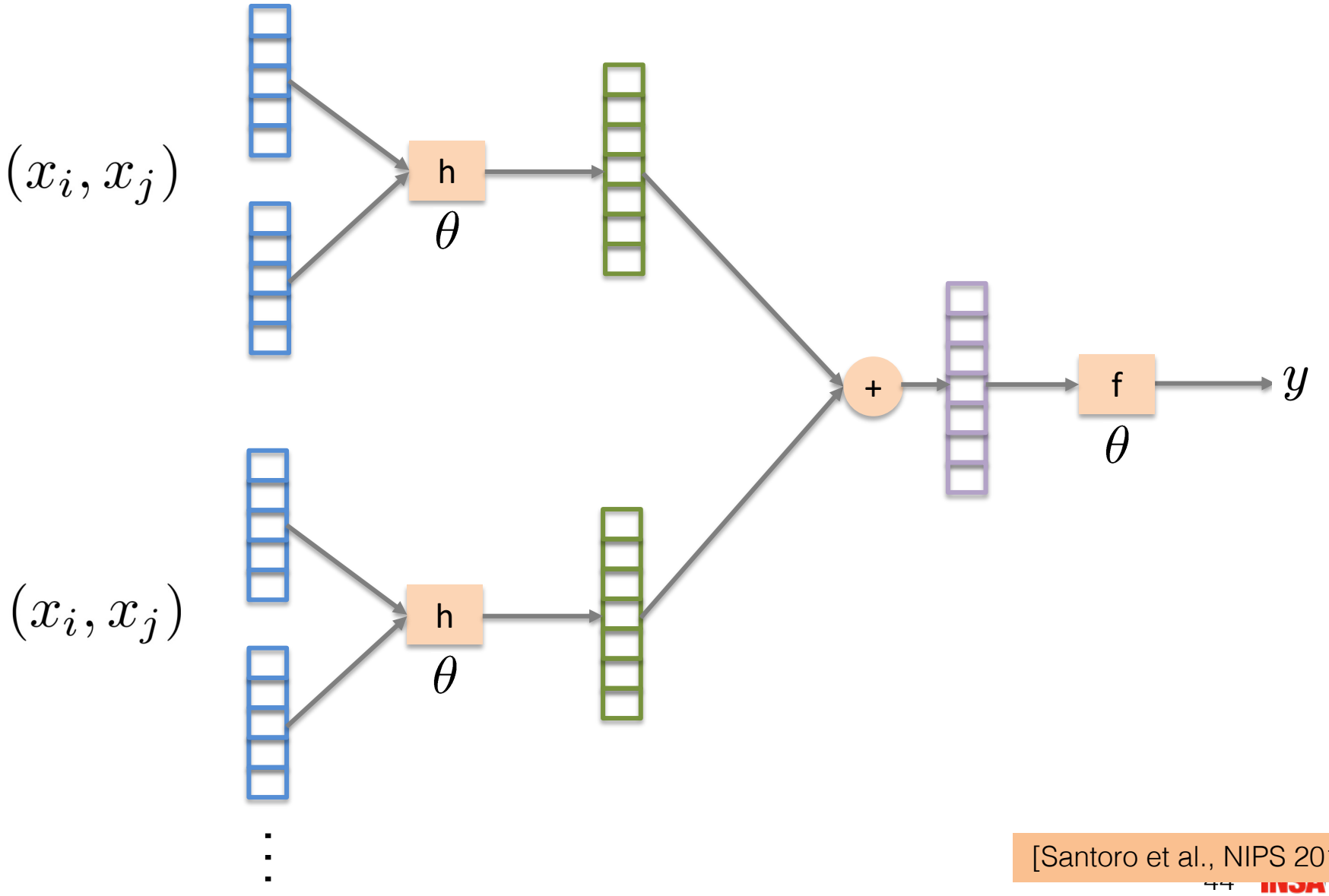
Defined over feature map cells

PointNet



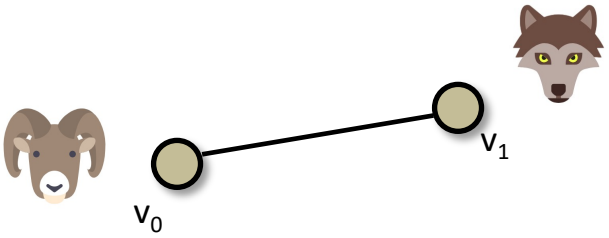
[Qi, Su, Mo, Guibas, CVPR 2017]

Relational Reasoning: pairwise terms

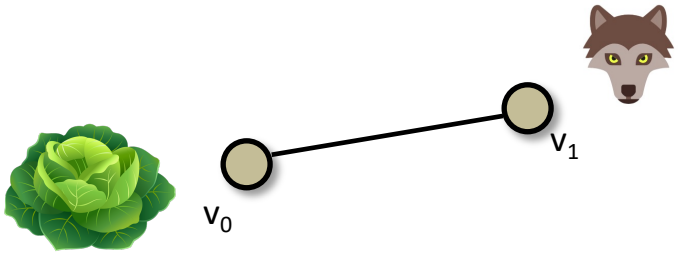


[Santoro et al., NIPS 2017]

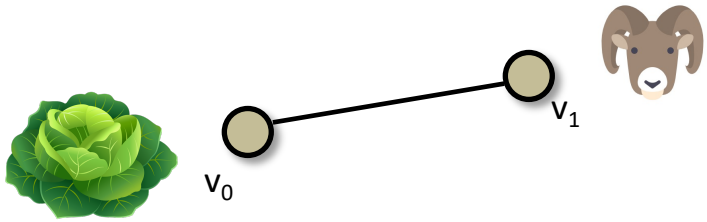
A toy problem: will somebody get eaten?



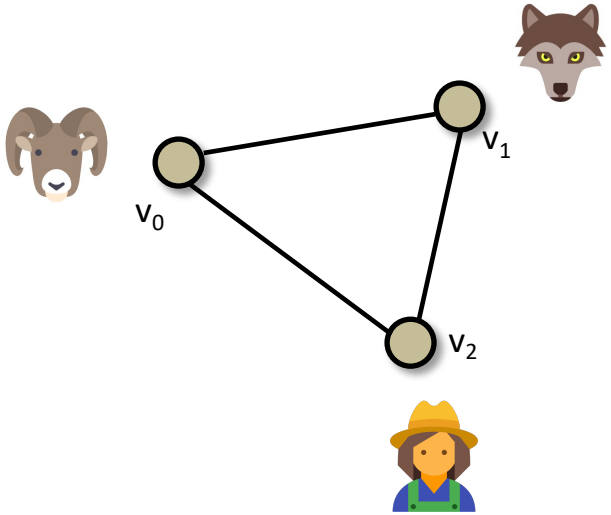
Yes



No

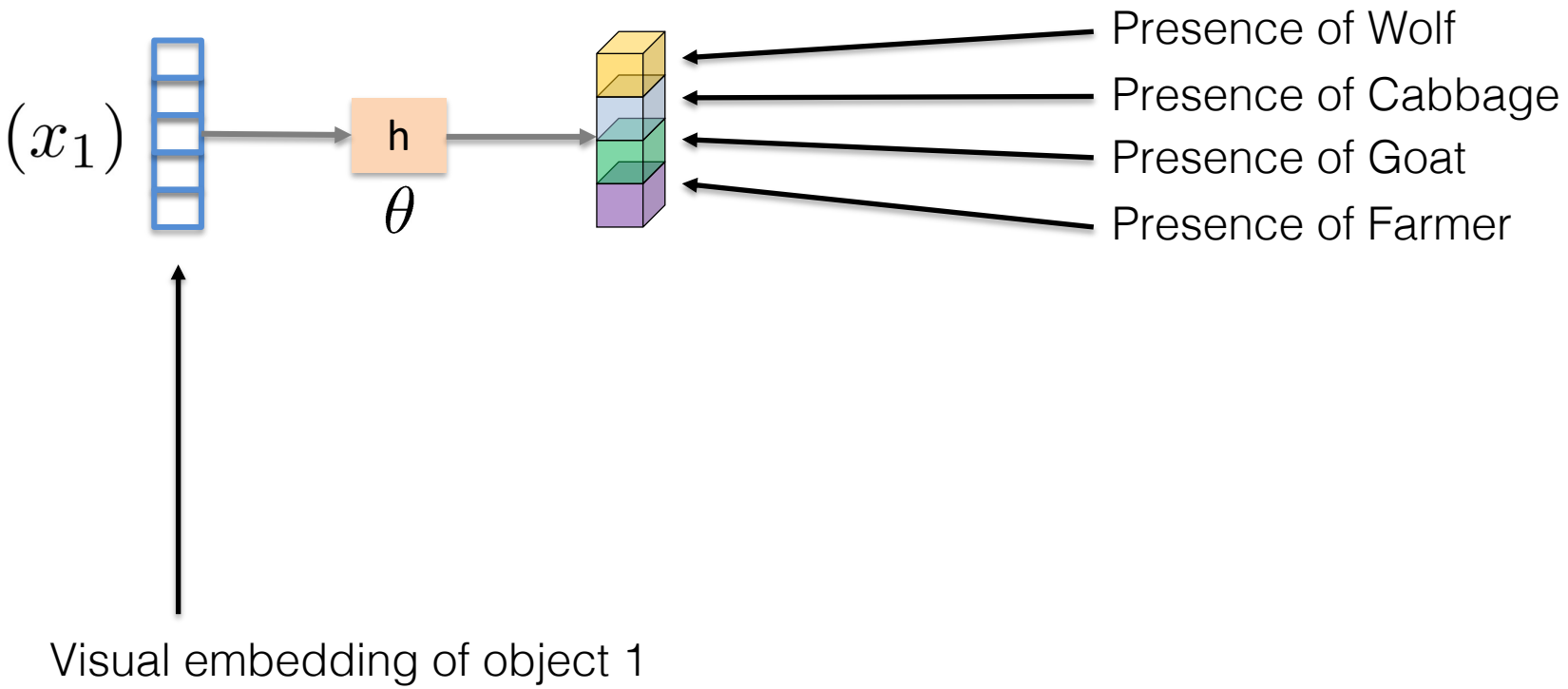


Yes

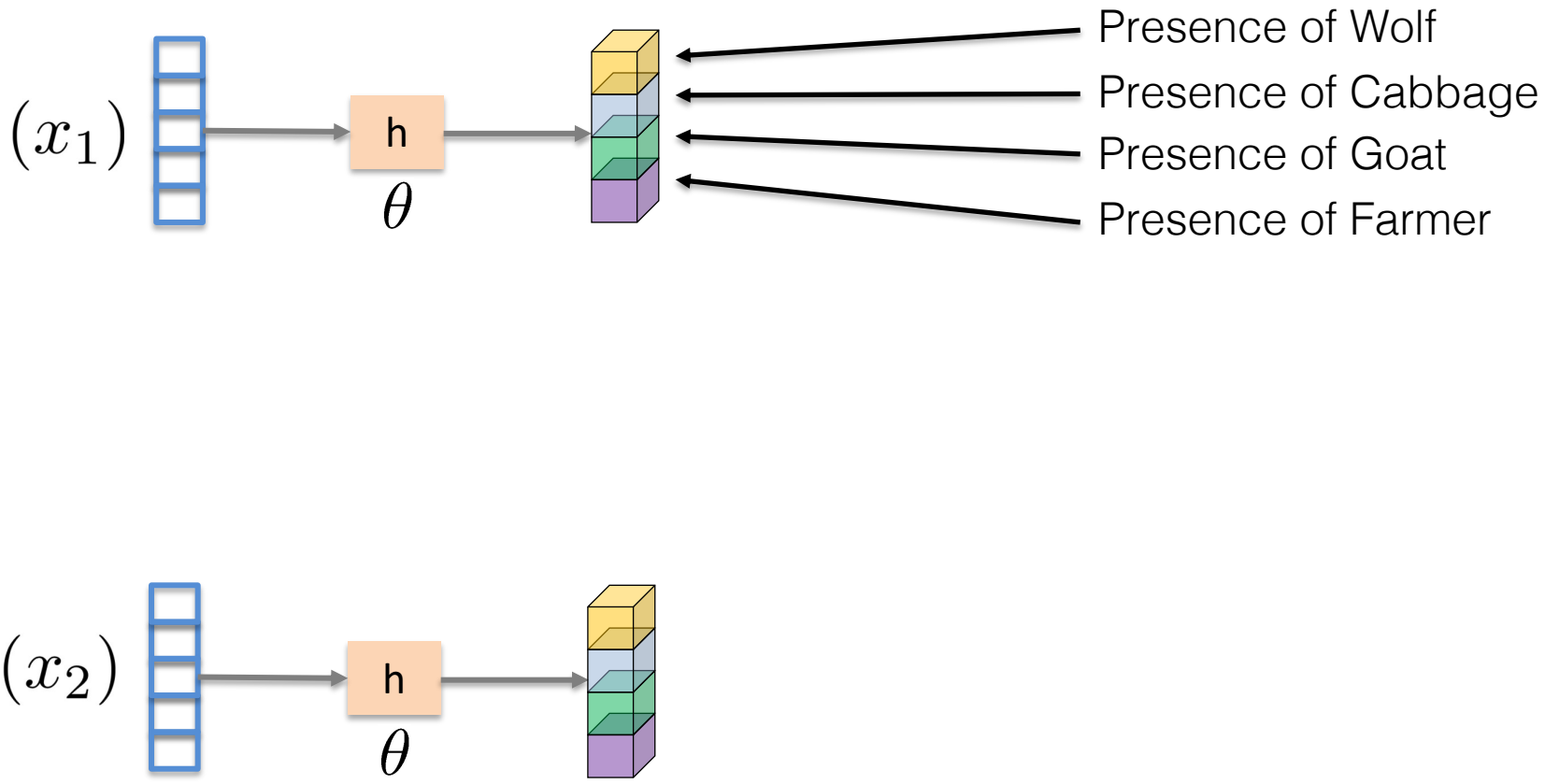


No

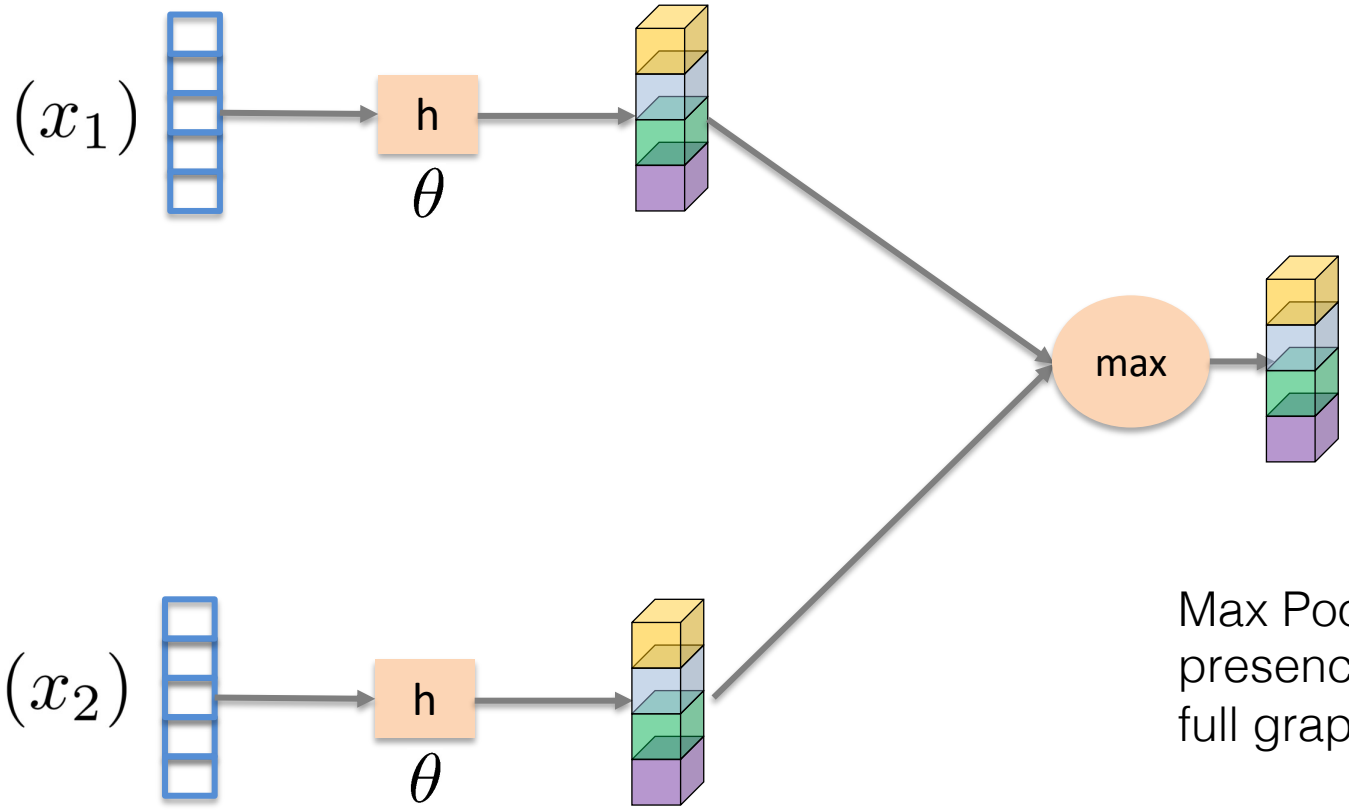
Representation by PointNet



Representation by PointNet

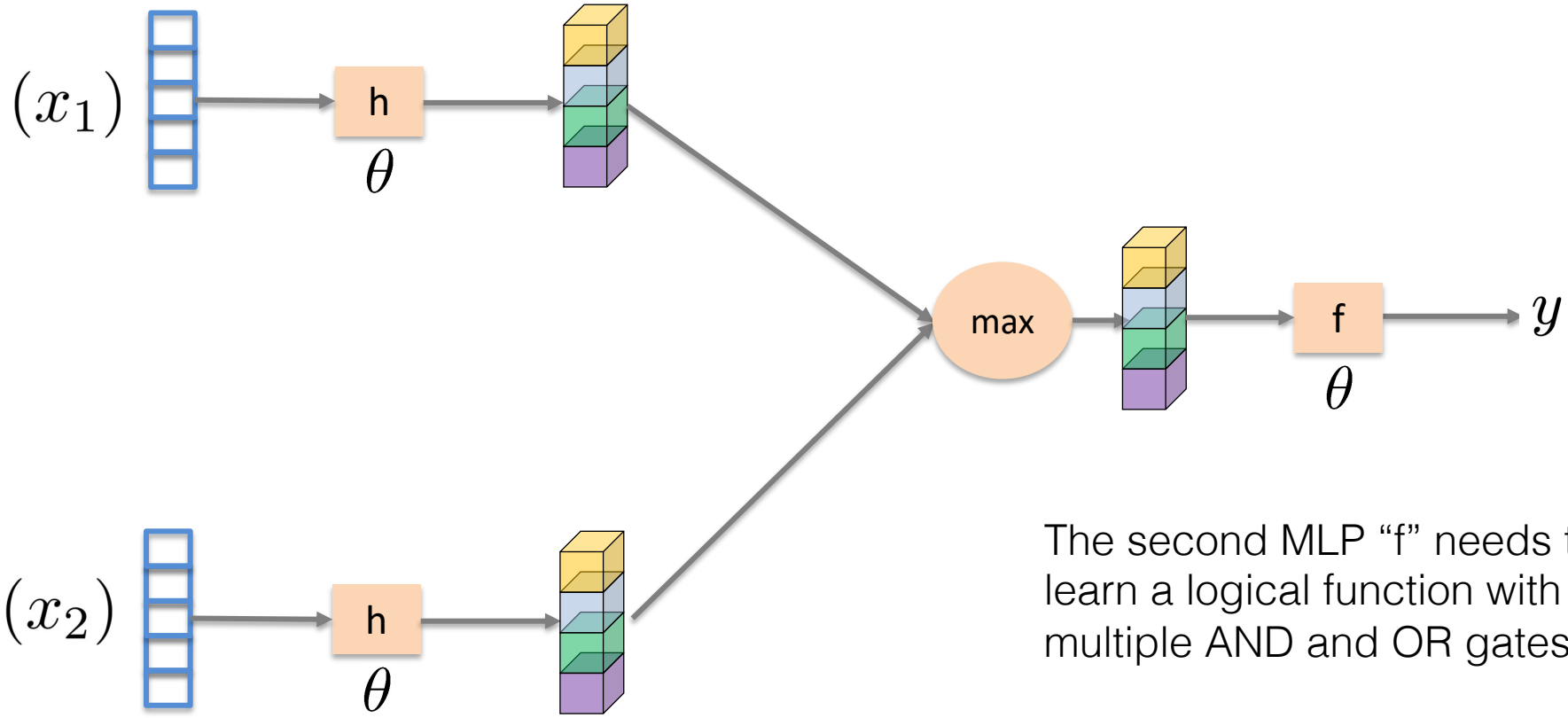


Representation by PointNet



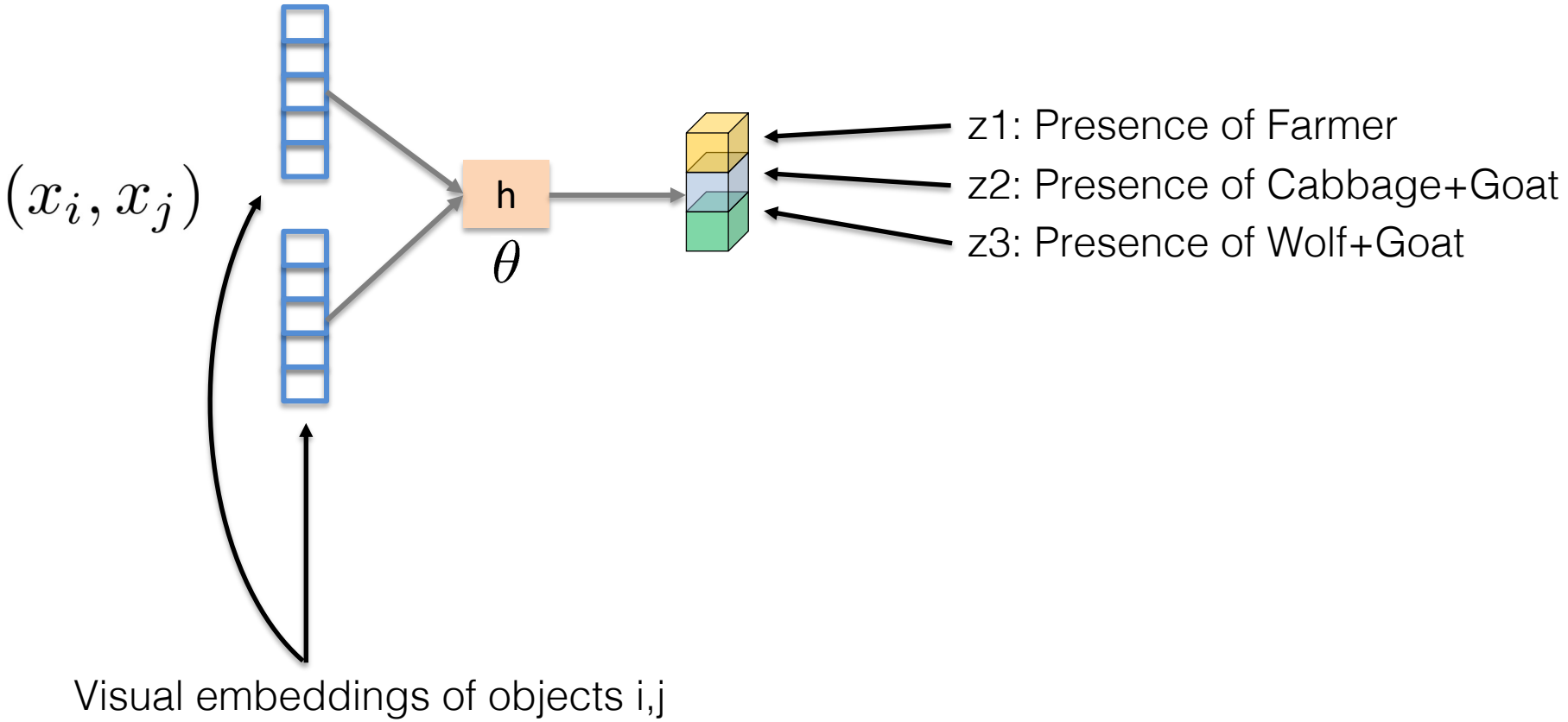
Max Pooling will cumulate the presence information from the full graph into a single vector

Representation by PointNet

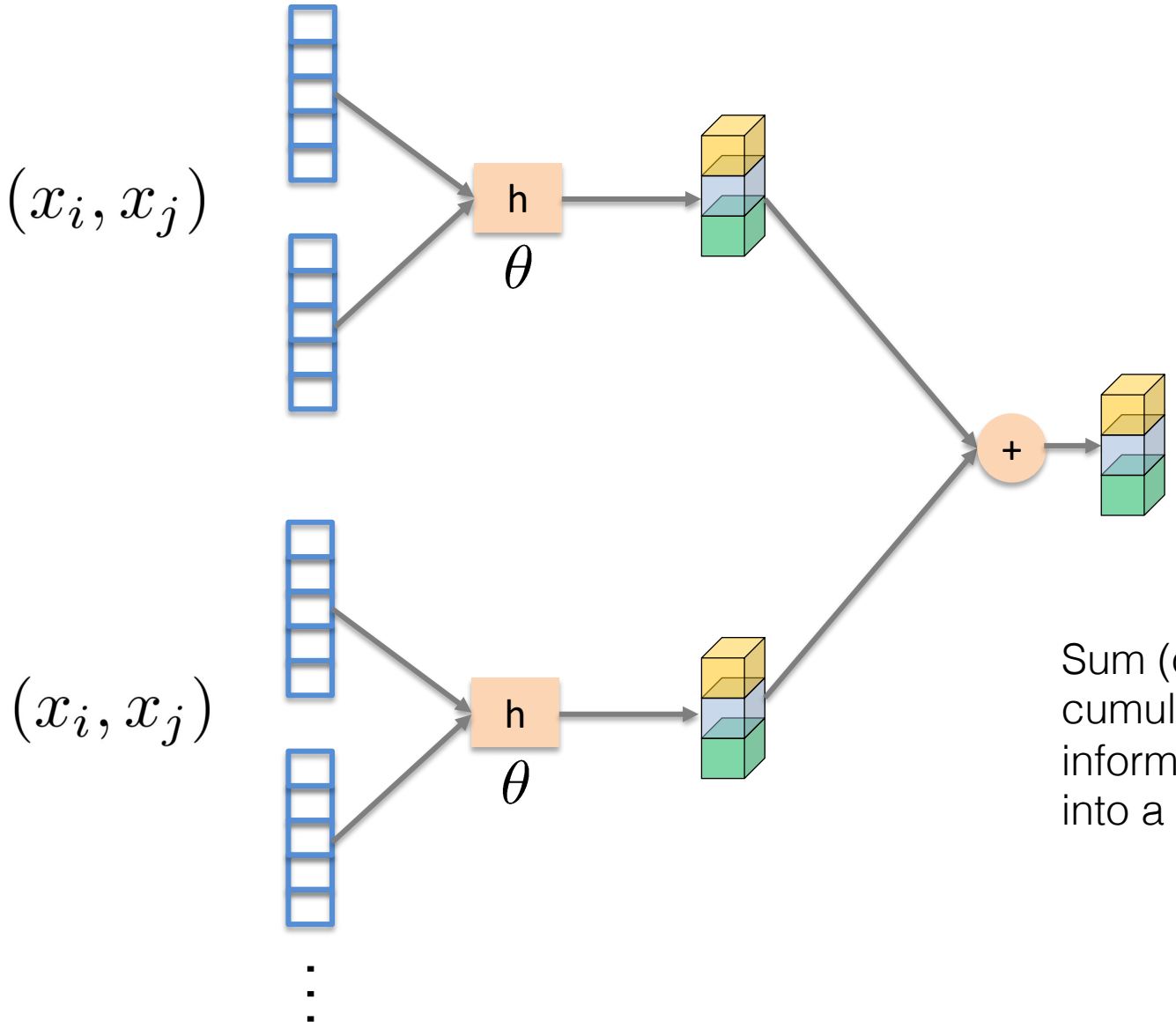


The second MLP “f” needs to learn a logical function with multiple AND and OR gates.

Representation with pairwise terms

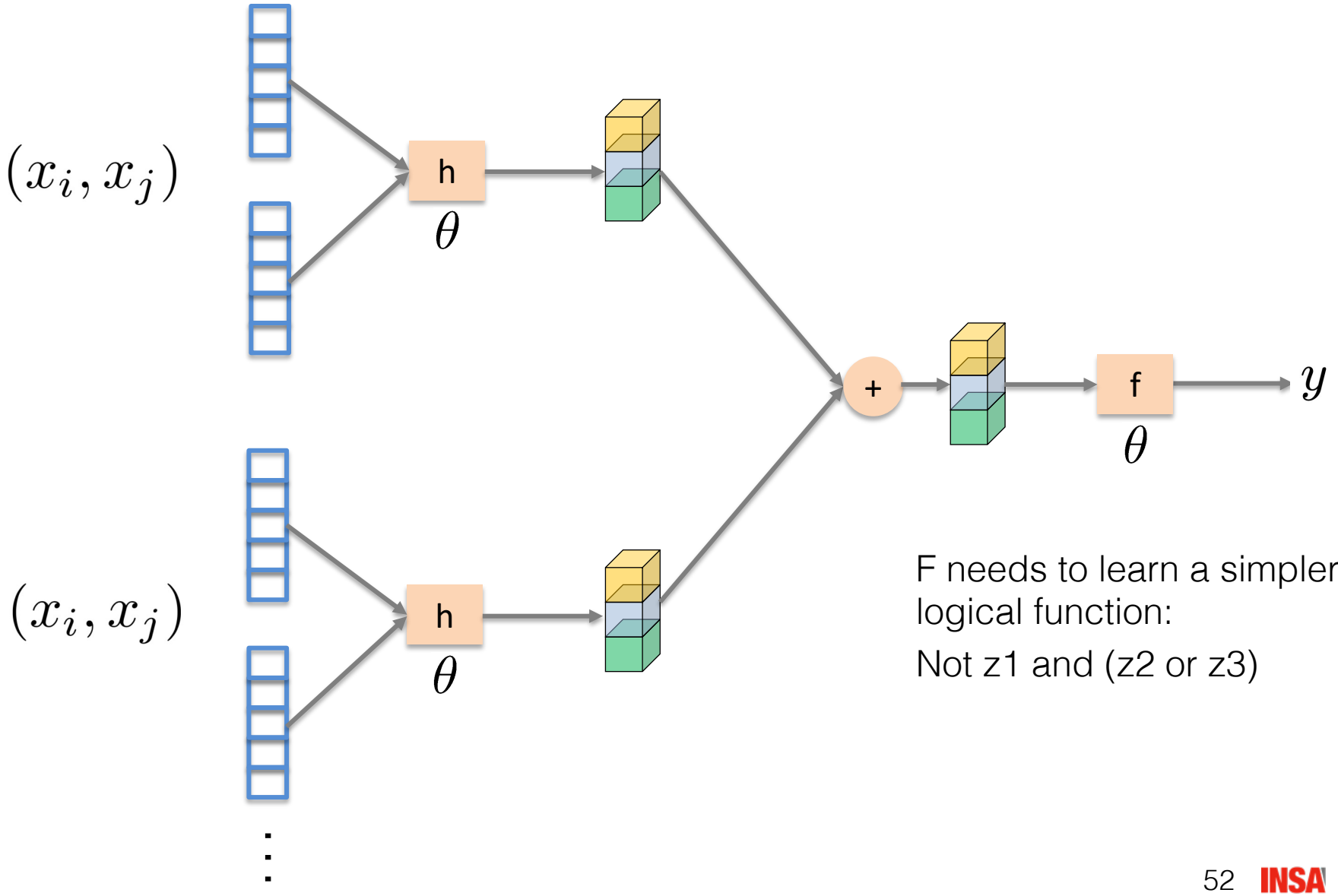


Representation with pairwise terms



Sum (or Max Pooling) will cumulate the logical information from the full graph into a single vector

Representation with pairwise terms

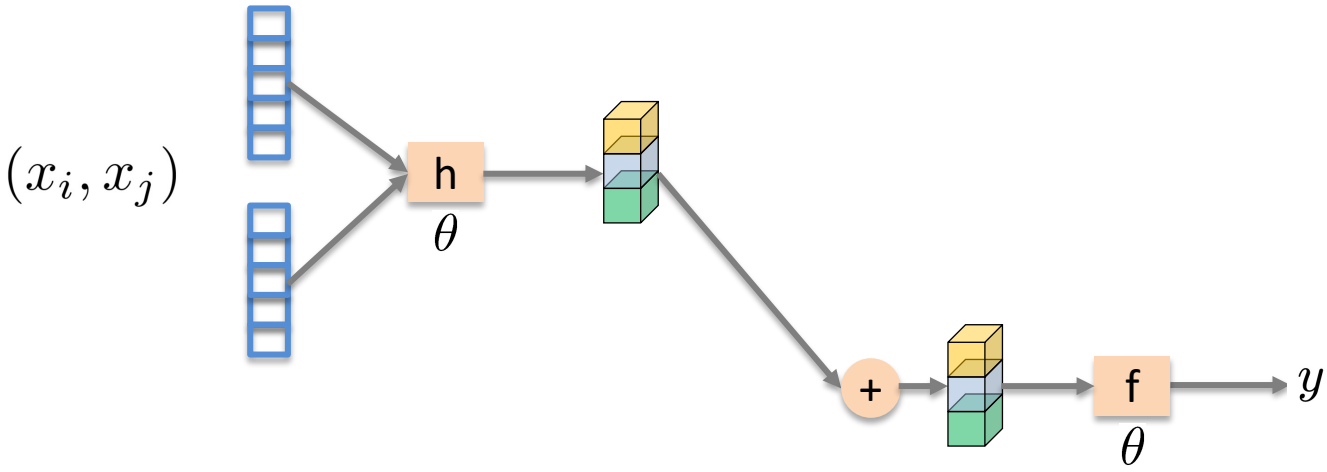


F needs to learn a simpler logical function:

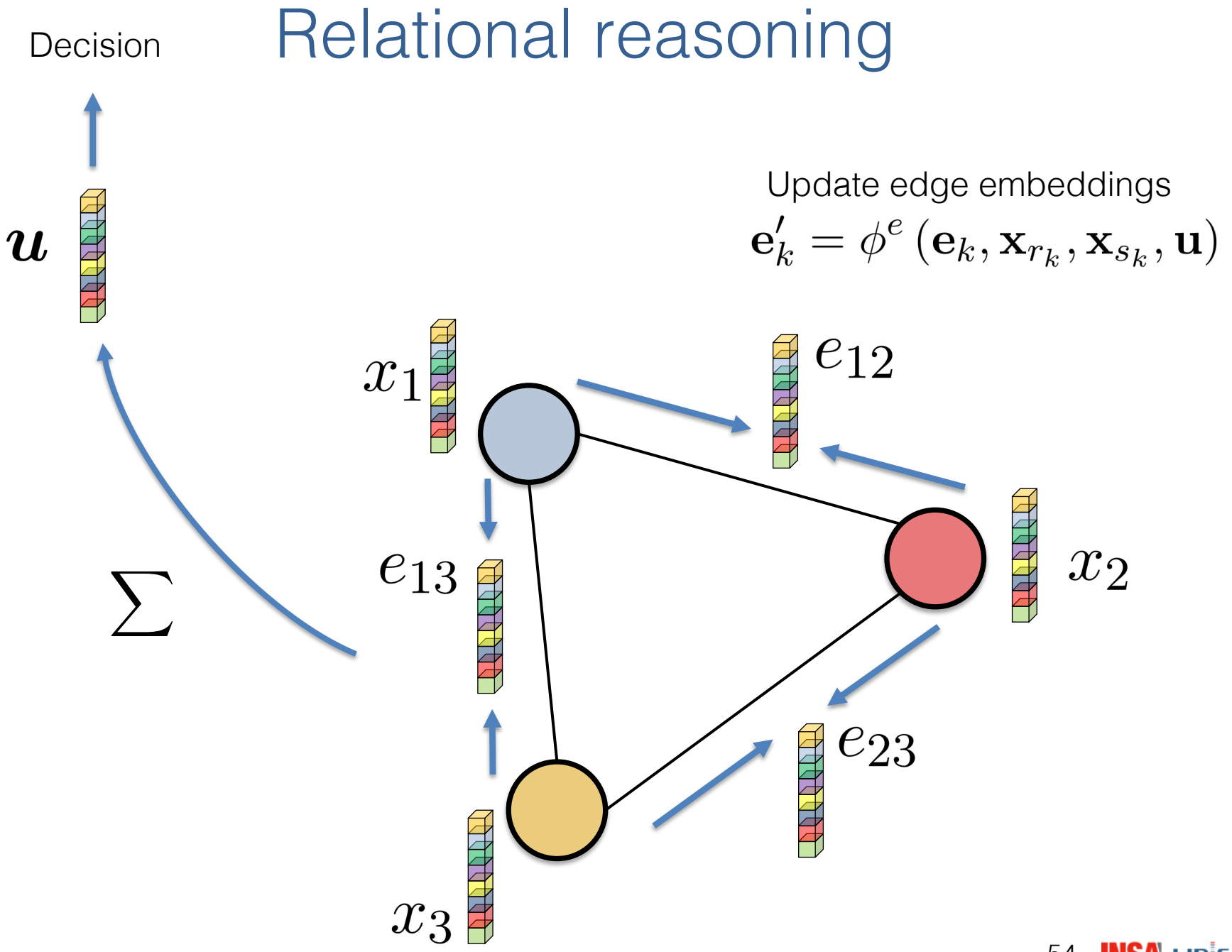
Not z1 and (z2 or z3)

Comparison

- Increasing the complexity of h may allow to decrease the complexity of f.
- There is no known rule which determines the best trad-off between h and f for a given problem.
- Example: there are problems dominated by pairwies relationships in the data where models without pairwise terms work better.



Relational reasoning

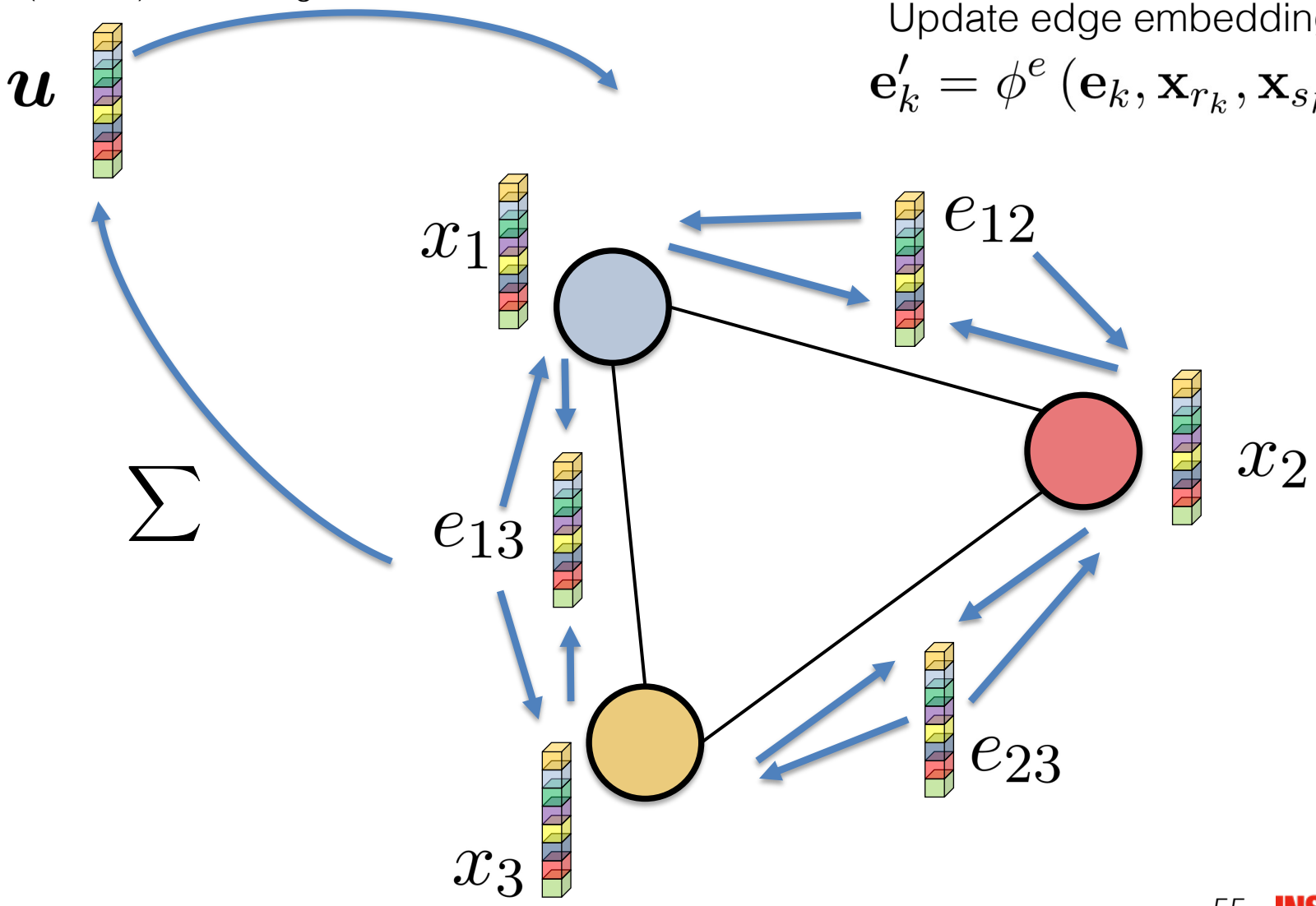


Graph Networks

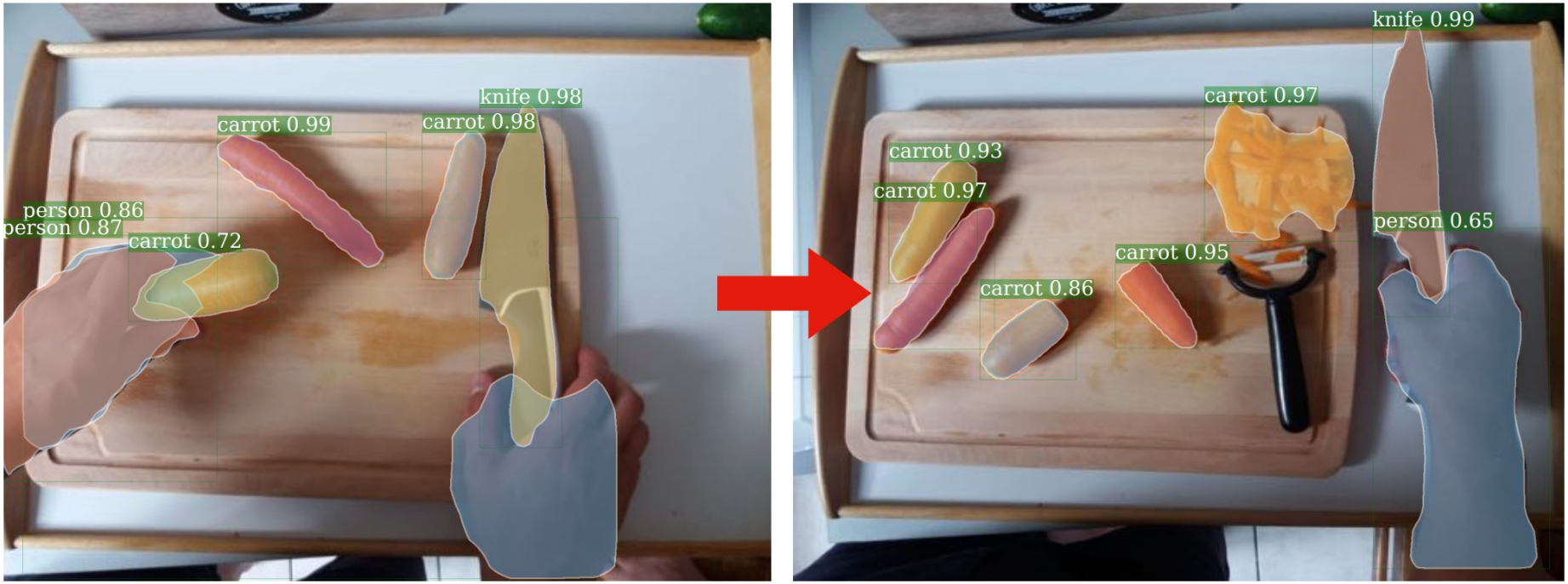
All messages are influenced by the global (context) embedding

Update edge embeddings

$$\mathbf{e}'_k = \phi^e\left(\mathbf{e}_k, \mathbf{x}_{r_k}, \mathbf{x}_{s_k}, \mathbf{u}\right)$$

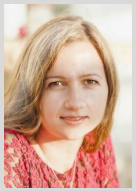


Object level Visual Reasoning



[Baradel, Neverova, Wolf, Mille, Mori, ECCV 2018]

Fabien Baradel
Phd @ LIRIS,
INSA-Lyon



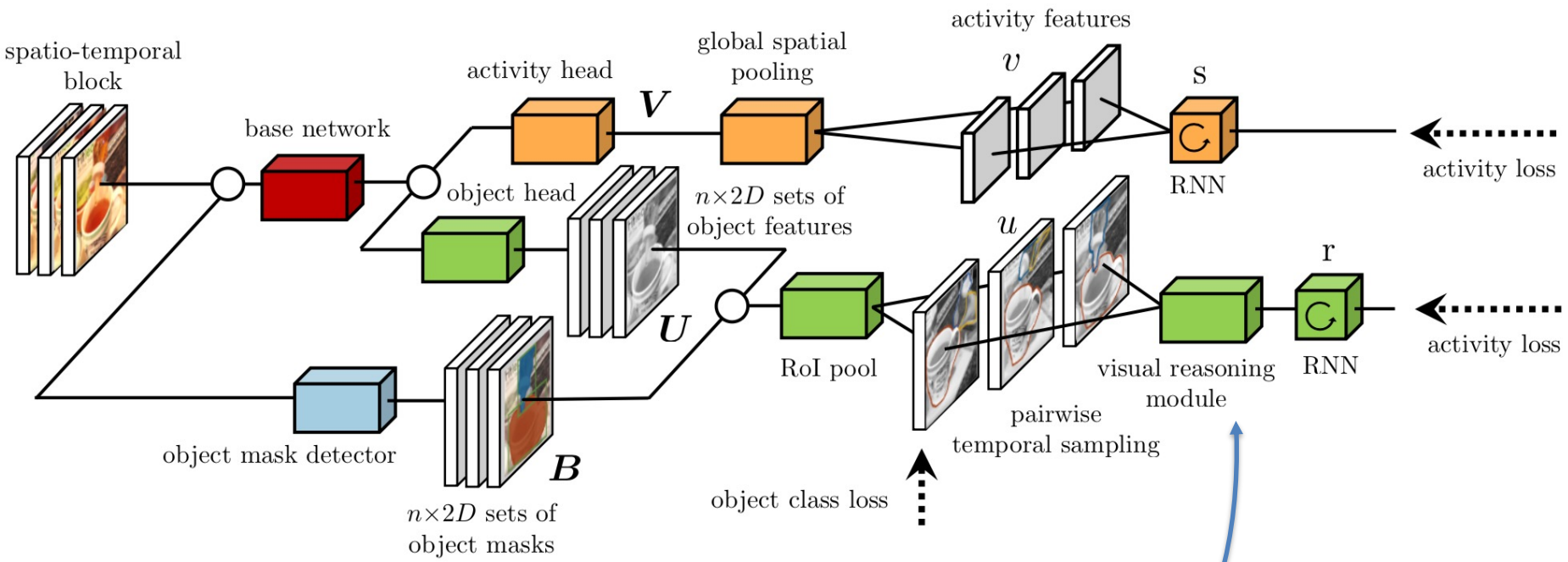
Natalia Neverova
Facebook AI
Research, Paris

Christian Wolf
Insa-Lyon,
LIRIS
INRIA Chroma

Julien Mille
LIFAT,
INSA VdL

Greg Mori
Simon
Fraser
University,
Canada

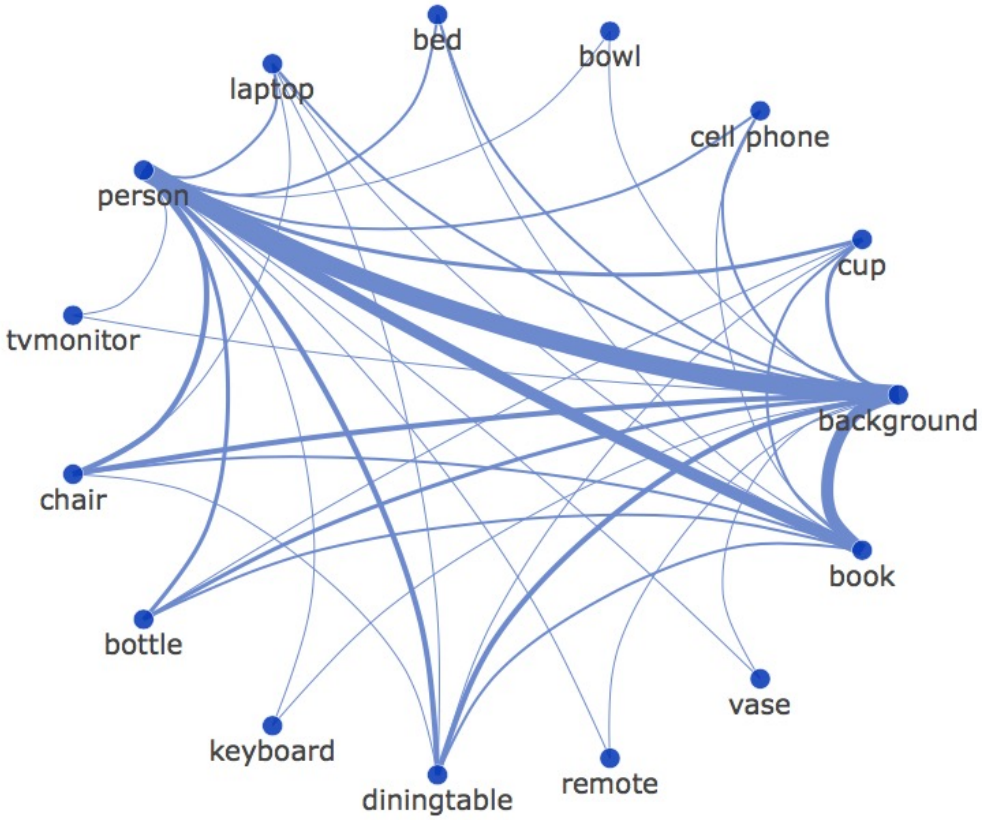
Object level Visual Reasoning

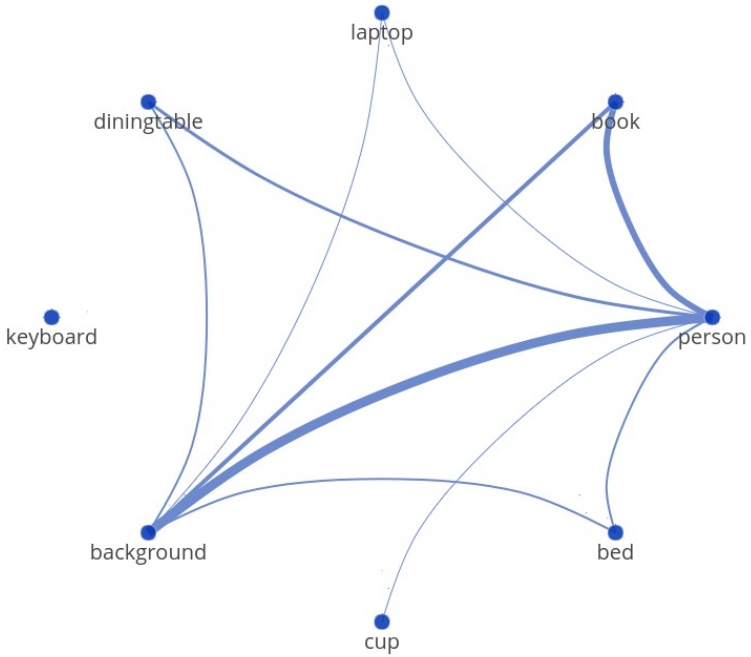


$$\mathbf{g}_t = \sum_{j,k} h_\theta(\mathbf{o}_{t'}^j, \mathbf{o}_t^k)$$

[Baradel, Neverova, Wolf, Mille, Mori, ECCV 2018]

Learned interactions

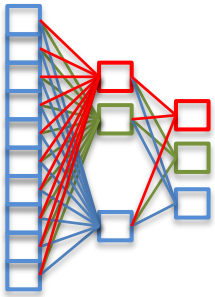




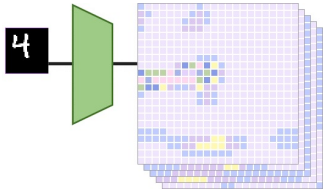
Class: person-book interaction

[Baradel, Neverova, Wolf, Mille, Mori, ECCV 2018]

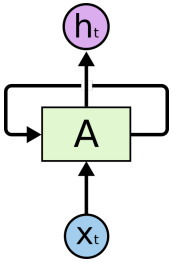
The Deep Toolbox



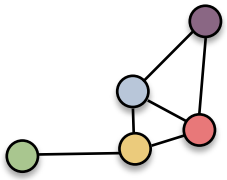
MLP



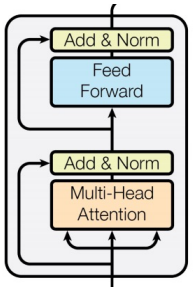
CNN / Convolutions



RNN / Recurrence



GN, GCN / Graphs, geometry



Transformers / Self-attention

What do I know about the data and the task?

Nothing (vector space)

Translation equivariance

Sequential data, Markov property

Graph structured data

Permutation equivariance

Learning language reasoning

Все счастливые семьи похожи друг на друга, каждая несчастливая семья несчастлива по-своему..

Toutes les familles heureuses se ressemblent. Chaque famille malheureuse, au contraire, l'est à sa façon.

Happy families are all alike. Every unhappy family is unhappy in its own way.

Alle glücklichen Familien gleichen einander, jede unglückliche Familie ist auf ihre eigene Weise unglücklich

[L. Tolstoy, 1873]

Examples

VO

If you really want to hear about it, the first thing you'll probably want to know is where I was born, and what my lousy childhood was like, and how my parents were occupied and all before they had me, and all that David Copperfield kind of crap, but I don't feel like going into it, if you want to know the truth. In the first place, that stuff bores me, and in the second place, my parents would have two hemorrhages apiece if I told anything pretty personal about them.

VF Google

Si vous voulez vraiment en entendre parler, la première chose que vous voud probablement savoir est où je suis né, et ce que mon enfance moche était, et comment mes parents étaient occupés et tout ce qu'ils avaient avant moi, et tout ce que David Copperfield, mais je n'ai pas envie d'y aller, si tu veux savoir la vérité. En premier lieu, cela m'ennuie, et en second lieu, mes parents auraient deux hémorragies si je leur racontais quelque chose de très personnel.

VF officielle

Si vous voulez vraiment que je vous dise, alors surement la première chose q aller demander c'est où je suis né, et à quoi ça ressemblait, ma saloperie d'enfance et c'est que faisaient mes parents avant de m'avoir, et toutes ces conneries à la David Copperfield, mais j'ai pas envie de raconter ça et tout. Primo, ce genre de trucs ça me rase, et secundo, mes parents ils auraient chacun une attaque ou même deux chacun, si je me mettais à baratiner sur leur compte quelque chose d'un peu personnel.

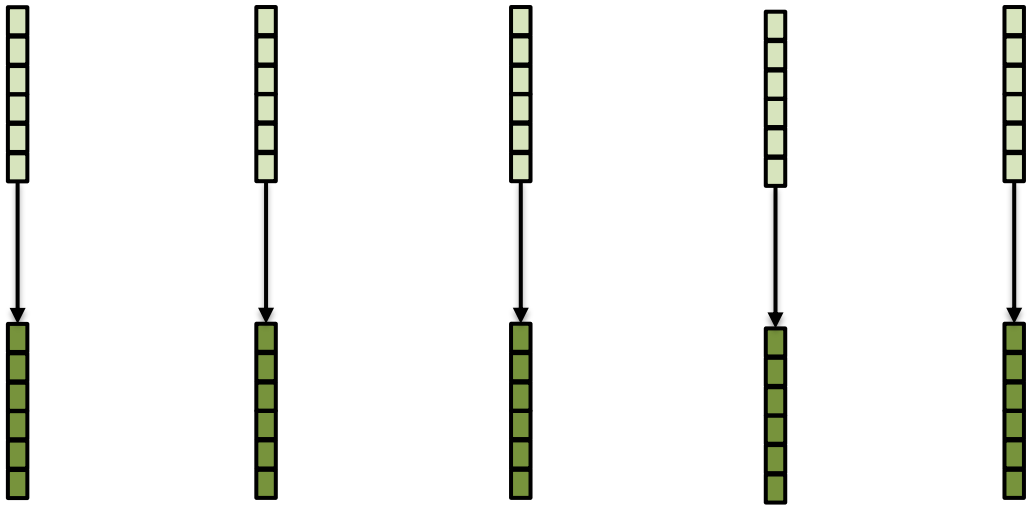
J.D. Salinger, « the catcher in the Rye »

Contextualization

We suppose a set $\mathbf{X} = \{\mathbf{x}_i\}$ of items (vectors).

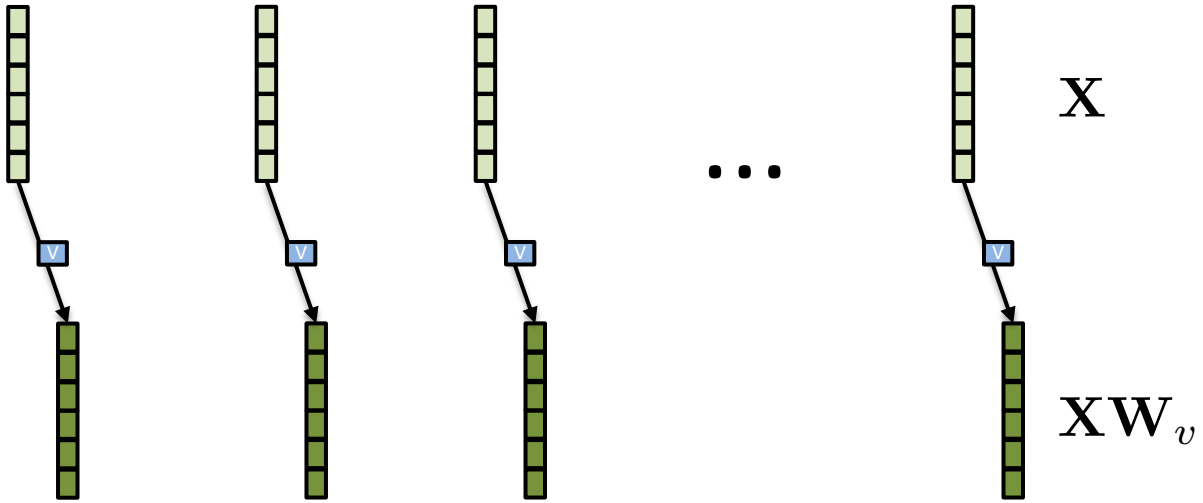
Iteratively "enrich" each item by providing context from the other items

The quick brown fox jumps ....



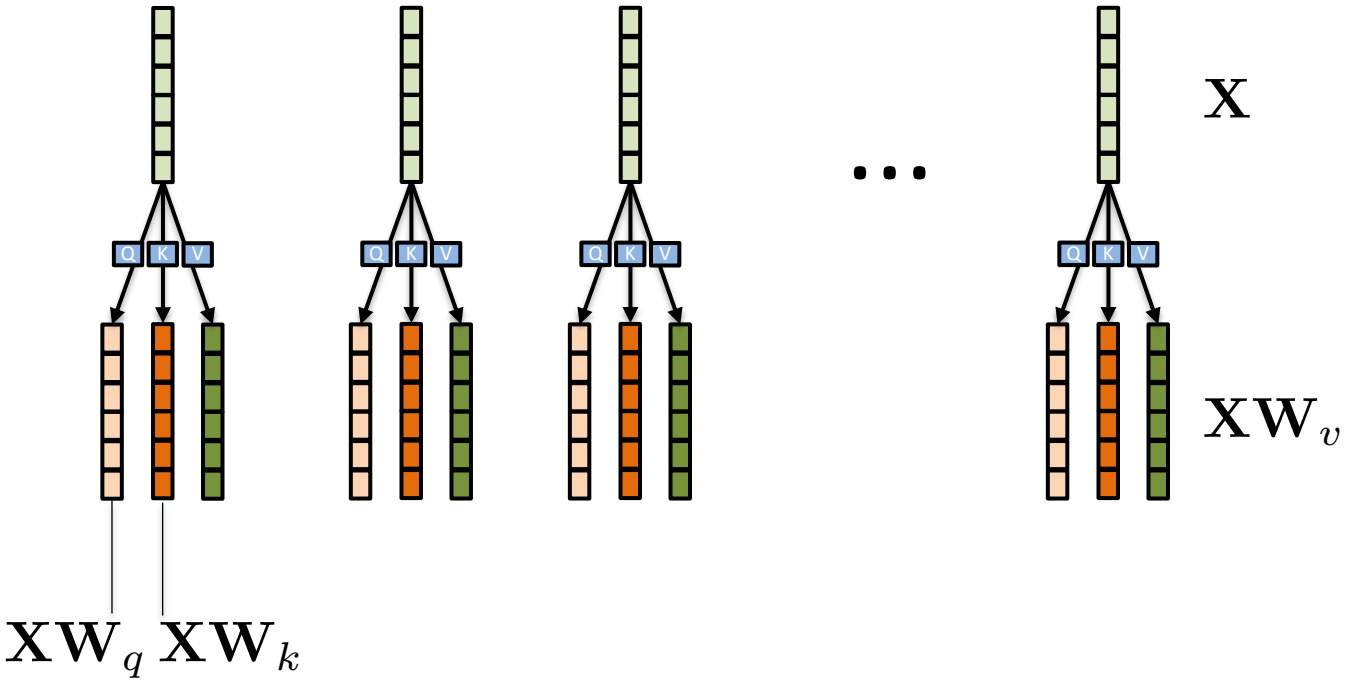
quick brown fox

Transformers: attention is all you need

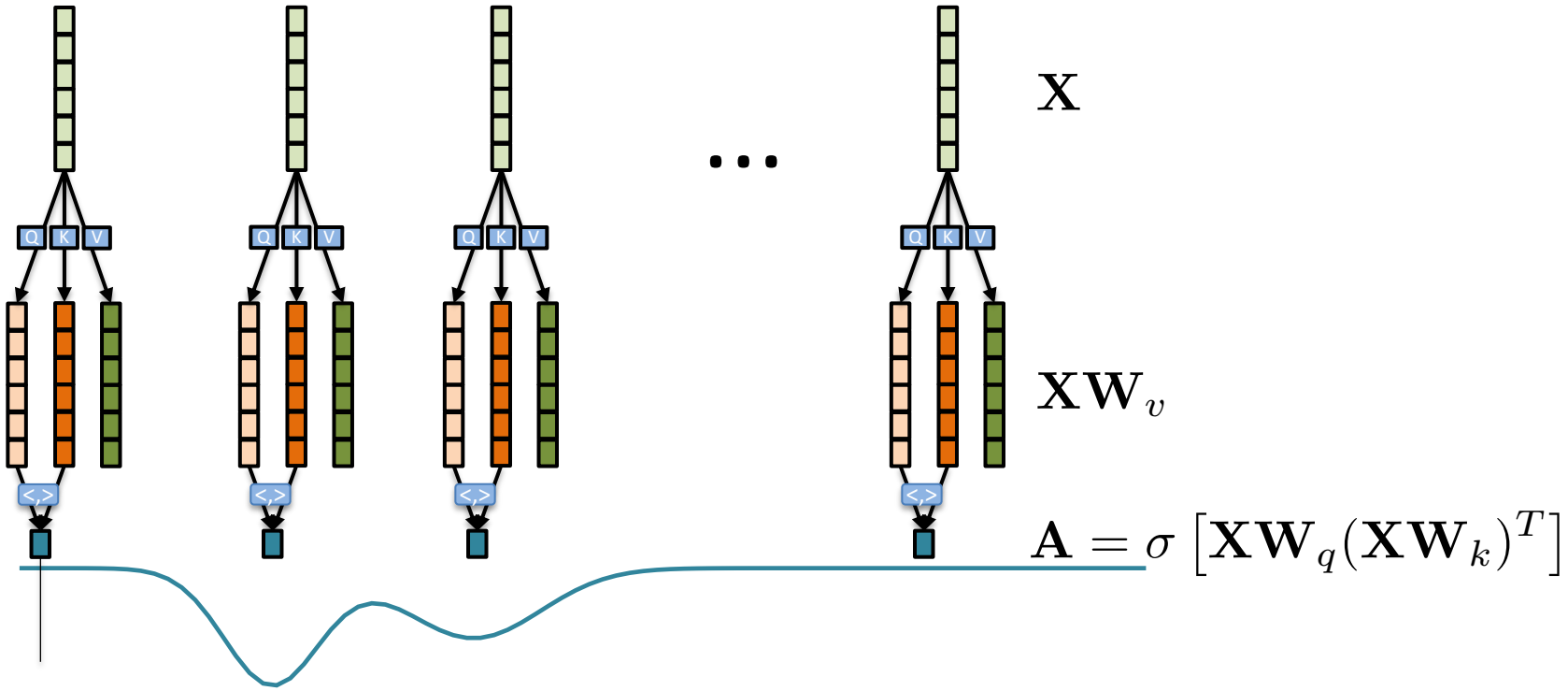


We suppose unordered data (a set) $\mathbf{X} = \{\mathbf{x}_i\}$. Each input item is "transformed" by a linear transform weighted by attention.

Transformers: attention is all you need



Transformers: attention is all you need



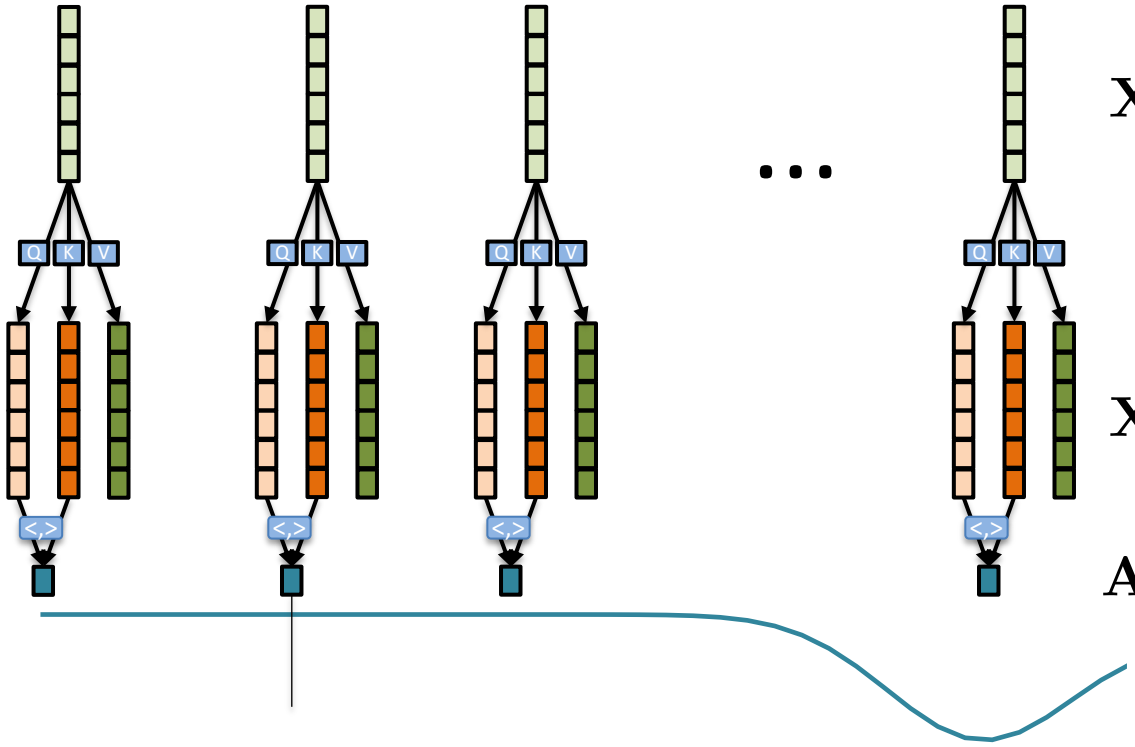
$\mathbf{X}$

$\mathbf{X}\mathbf{W}_v$

$\mathbf{A} = \sigma\left[\mathbf{X}\mathbf{W}_q(\mathbf{X}\mathbf{W}_k)^T\right]$

Each item has its own distribution,
associated with its query vector!

Transformers: attention is all you need



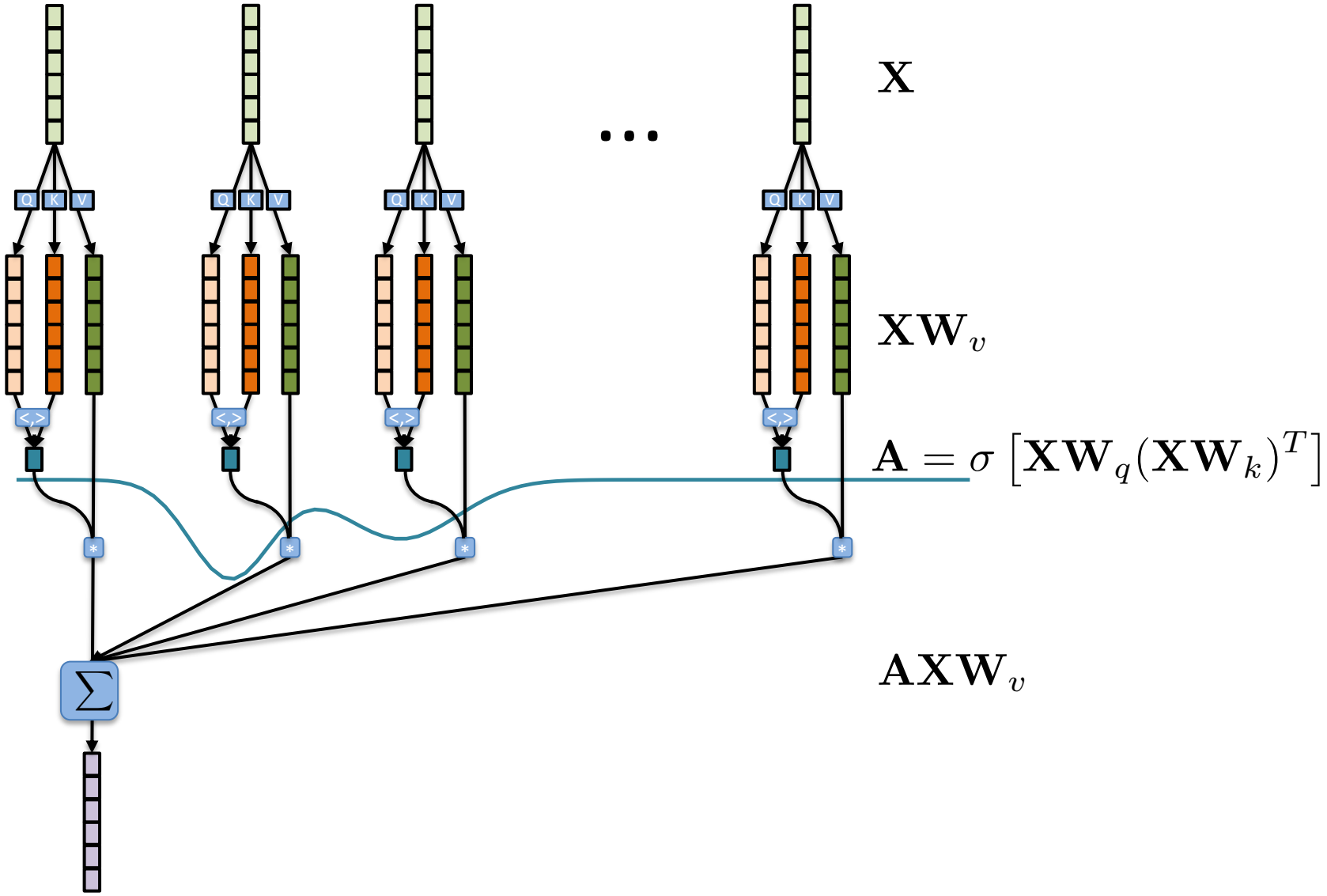
$\mathbf{X}$

$\mathbf{XW}_v$

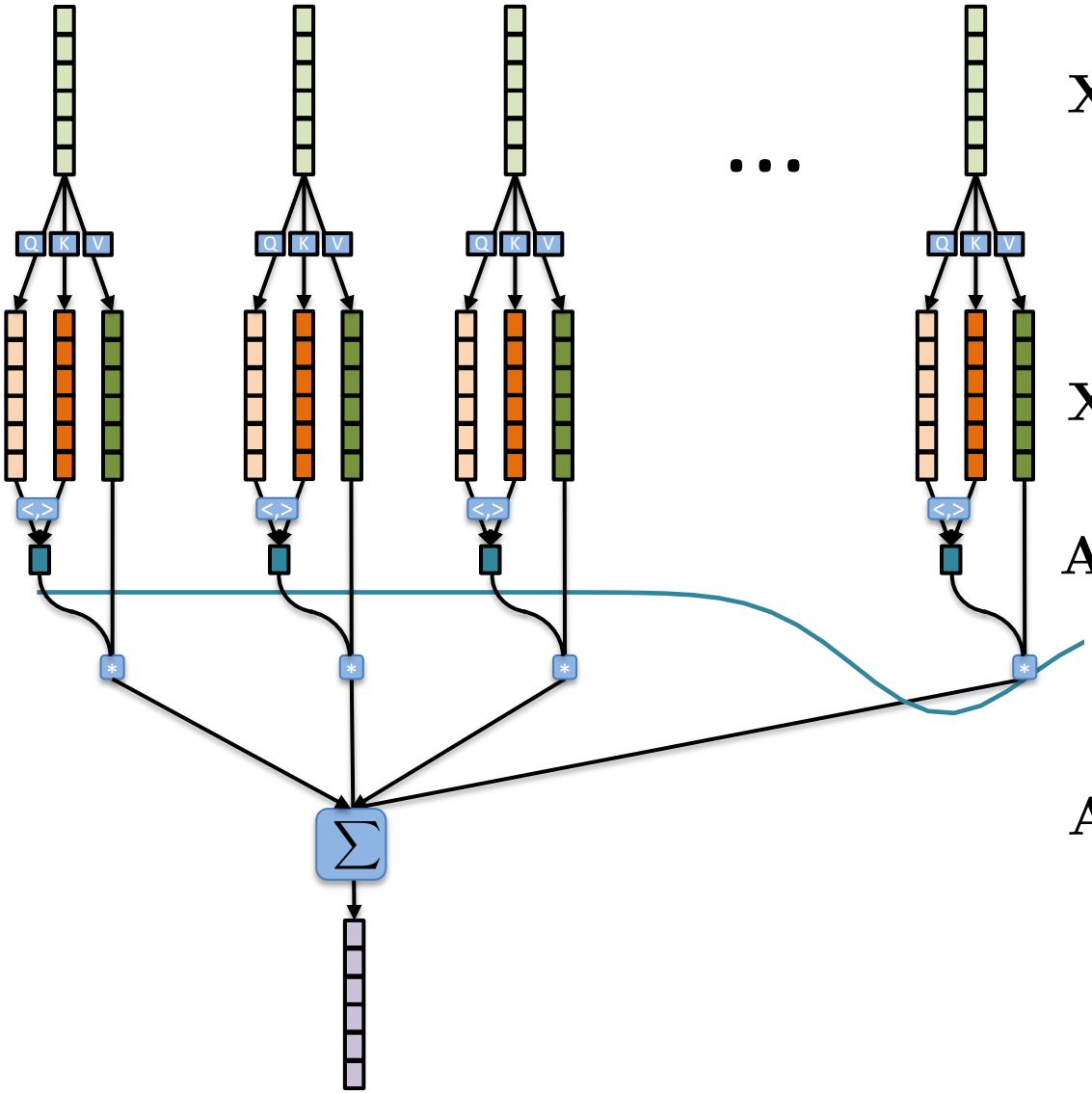
$\mathbf{A} = \sigma\left[\mathbf{XW}_q(\mathbf{XW}_k)^T\right]$

Each item has its own distribution,
associated with its query vector!

Transformers: attention is all you need



Transformers: attention is all you need



$\mathbf{X}$

$\mathbf{X}\mathbf{W}_v$

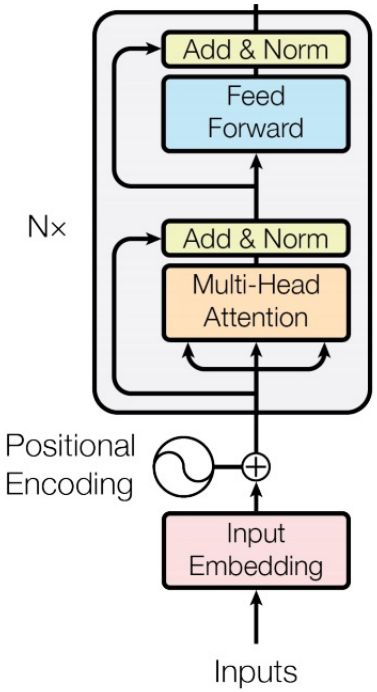
$\mathbf{A} = \sigma\left[\mathbf{X}\mathbf{W}_q(\mathbf{X}\mathbf{W}_k)^T\right]$

$\mathbf{A}\mathbf{X}\mathbf{W}_v$

Integration ...

Toutes les familles heureuses se ressemblent. Chaque famille malheureuse, au contraire, l'est à sa façon.

Happy families are all alike. Every unhappy family is unhappy in its own way.

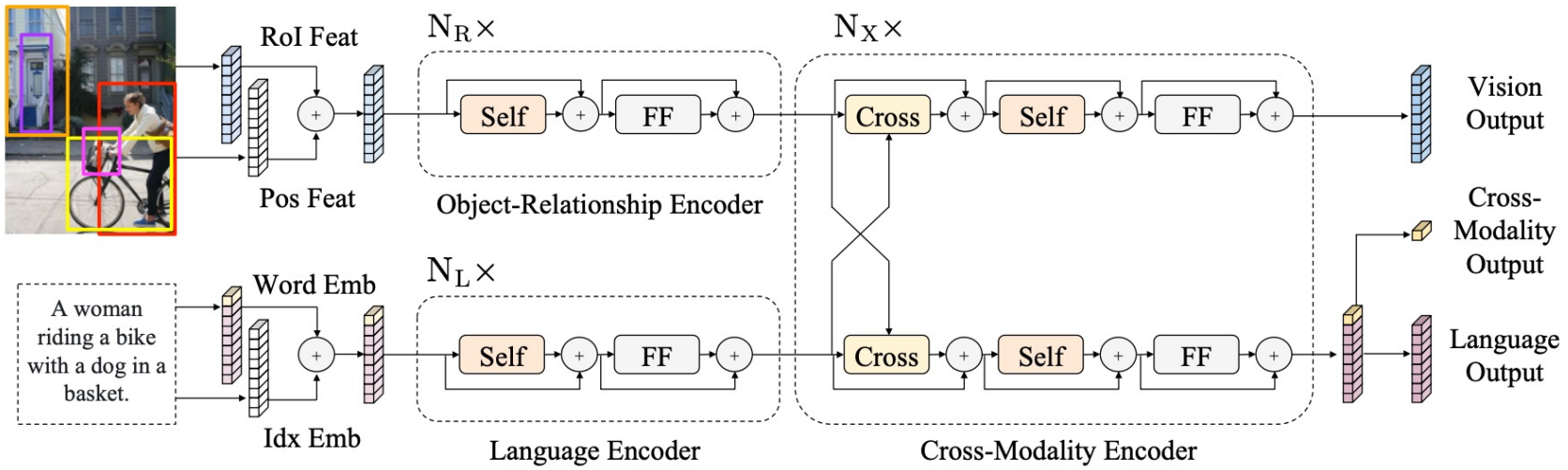


A concrete large-scale application

Vision and Language Reasoning

LXMERT

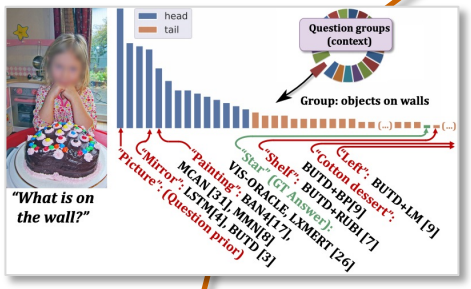
A vision and language encoder with self-attention and cross-attention.



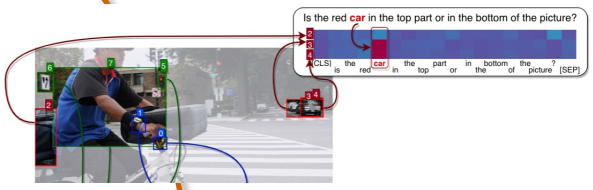
[Tan et Bansal, EMNLP 2019]

VQA

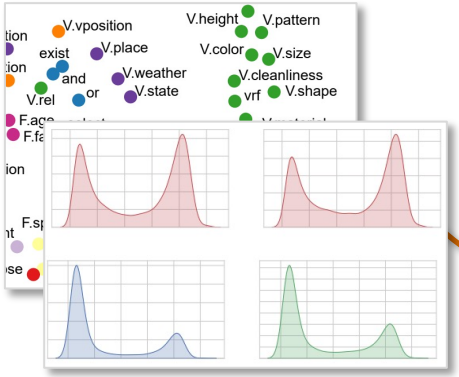
How can we evaluate biases in learning? (CVPR 2021a)



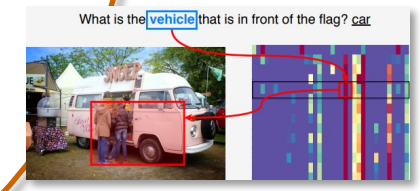
Can we ground object detection through language (in writing)



How can we visualize and transfer reasoning? (CVPR 2021b)



Can we weakly supervise word-object alignment? (ECAI 2020)

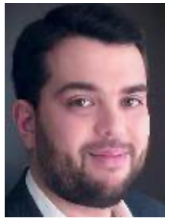


Can we supervise reasoning programs? (arxiv, under review)

orange™


Corentin Kervadec

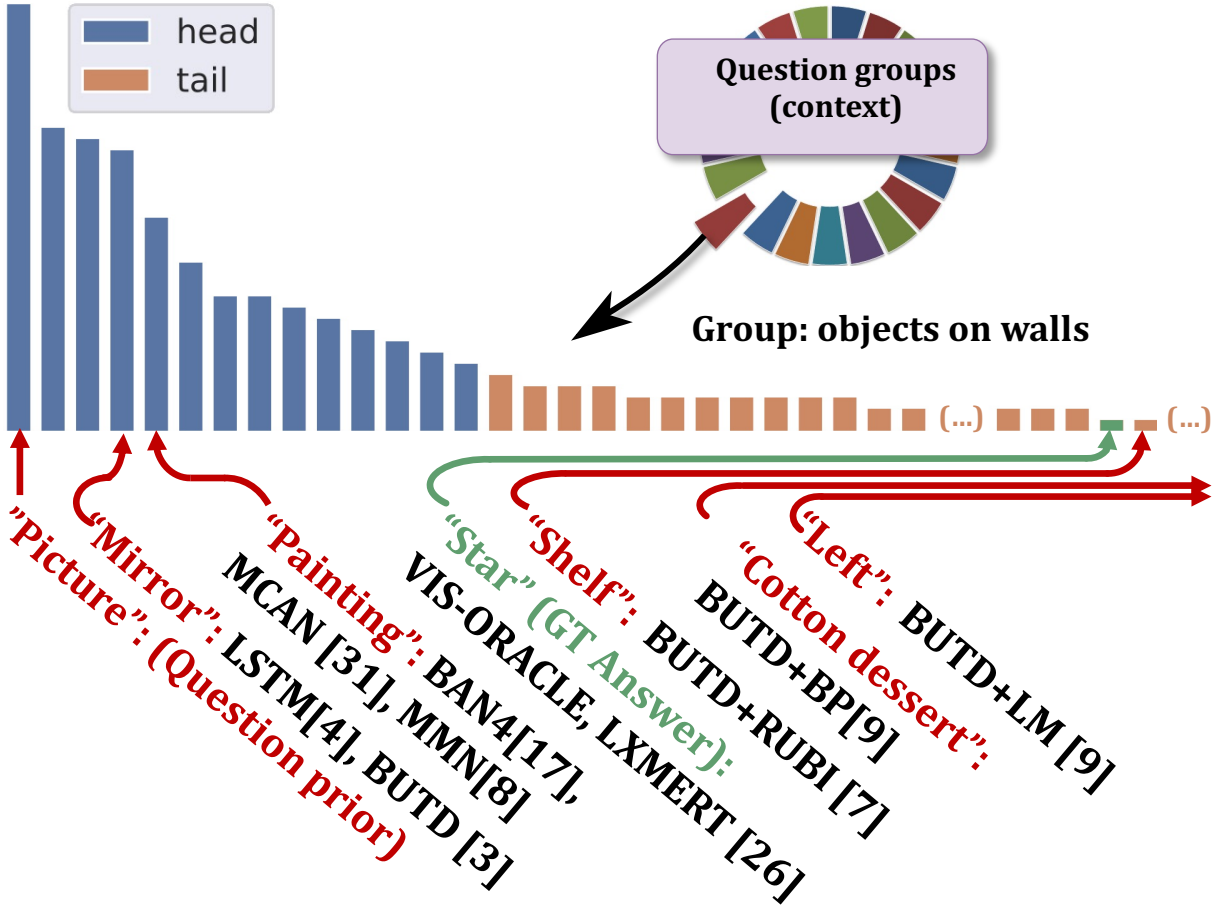

Grigory Antipov


Moez Baccouche


Christian Wolf

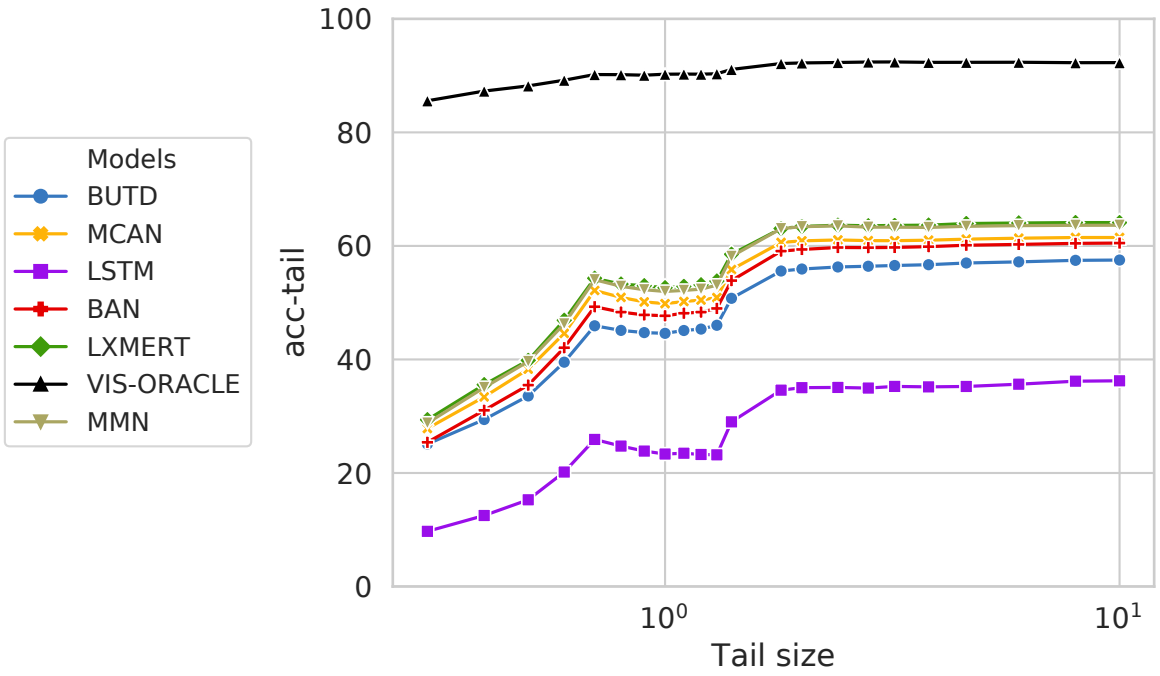
Roses are Red, Violets are blue …
But should VQA expect them to?

"What is on the wall?"



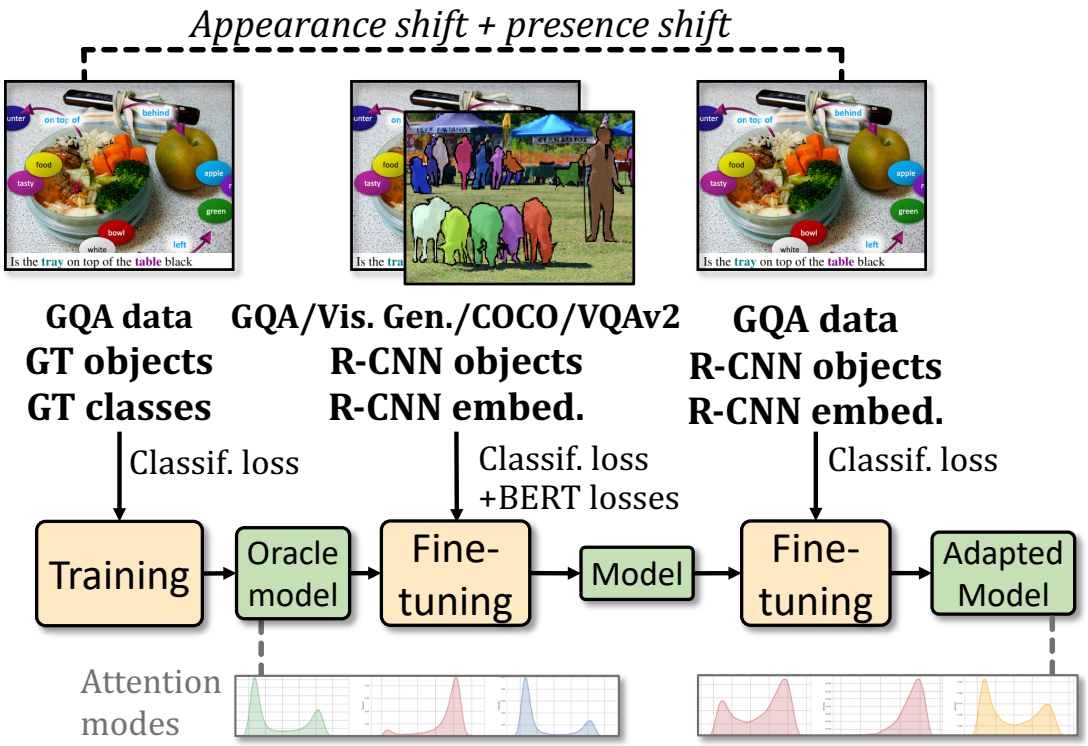
[Kervadec, Antipov, Baccouche, Wolf, CVPR 2001a]

Reasoning vs. bias exploitation



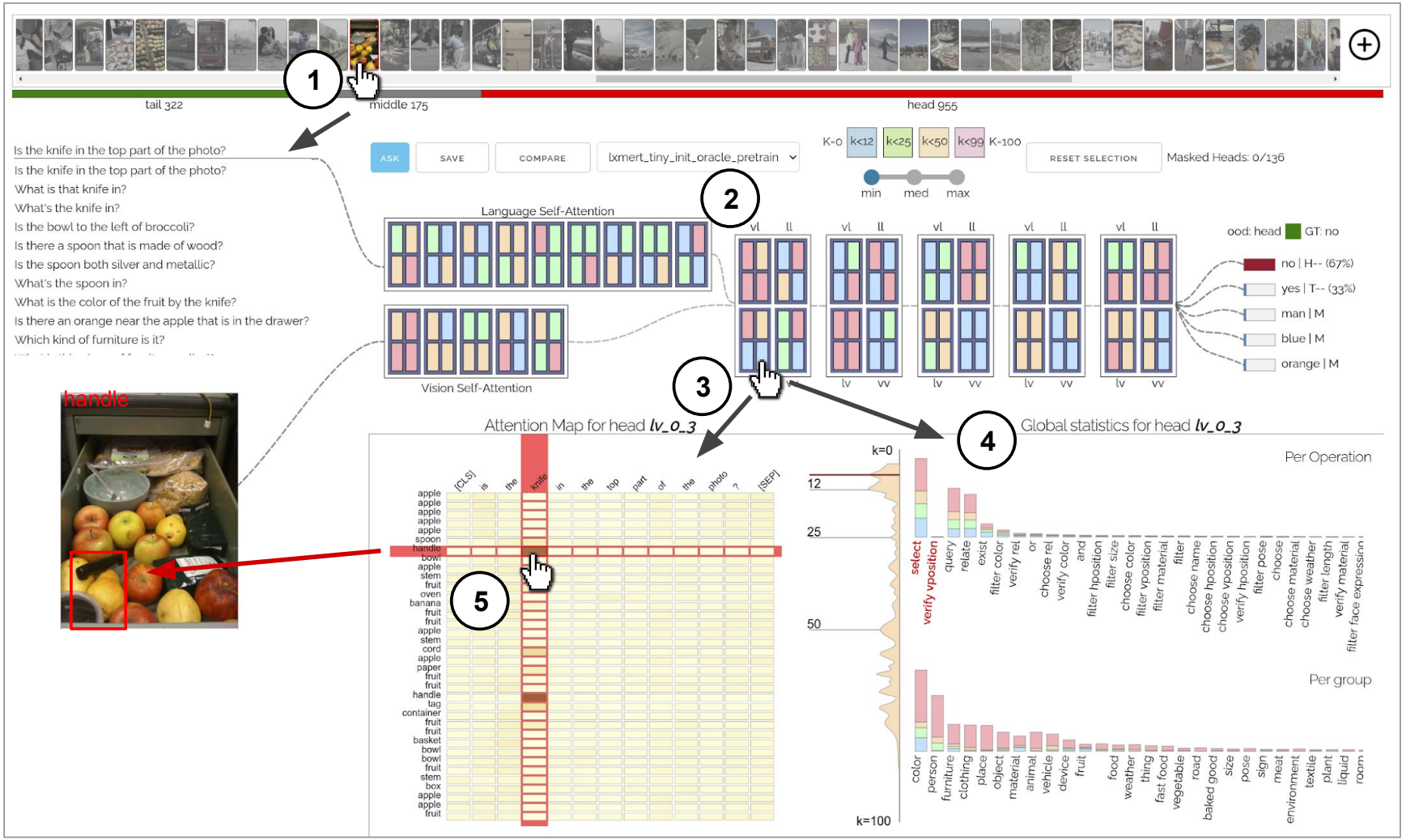
[Kervadec, Antipov, Baccouche, Wolf, CVPR 2021a]

Oracle transfer



[Kervadec, Jaunet, Antipov, Baccouche, Vuilletmot, Wolf, CVPR 2021b]

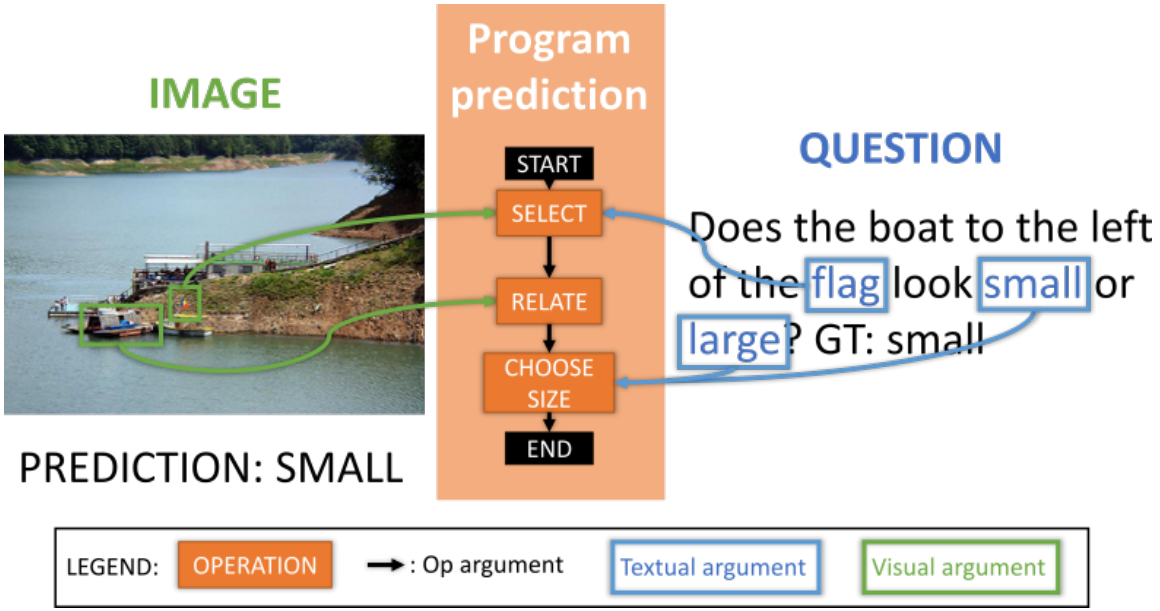
Visualizing Reasoning Patterns



[Jaunet, Kervadec, Vuillemot, Antipov, Baccouche, Wolf, Conditionally accepted to VIS 2021]

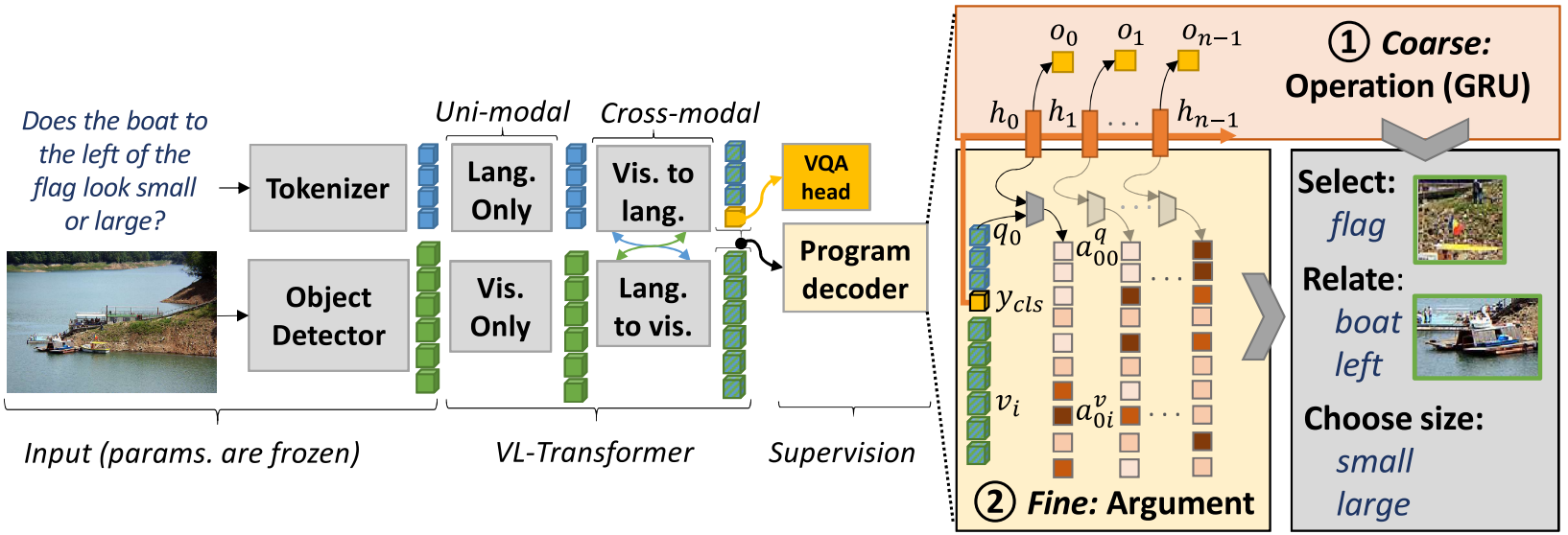
https://visqa.liris.cnrs.fr

GT reasoning programs



[Kervadec, Antipov, Baccouche, Wolf, arxiv / under review]

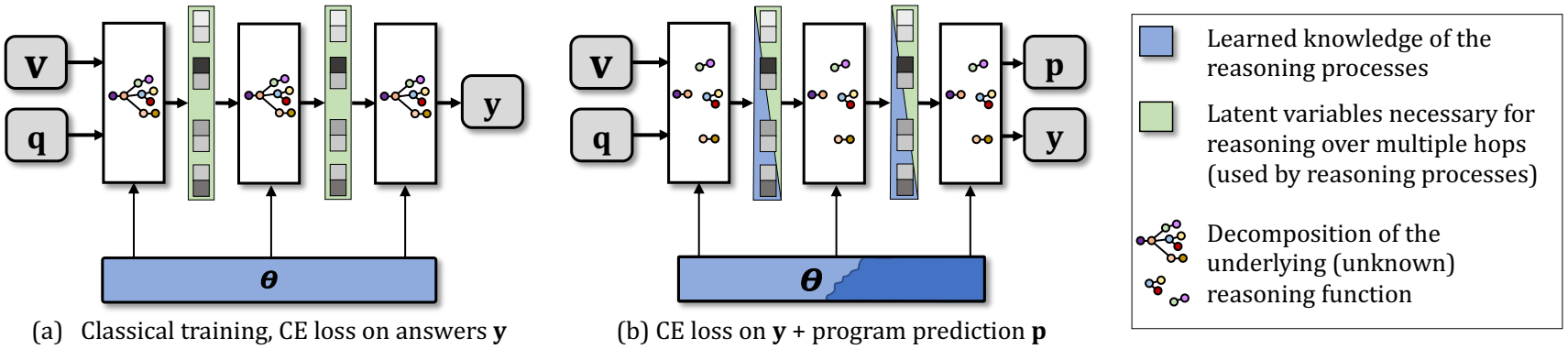
Program prediction



[Kervadec, Antipov, Baccouche, Wolf, arxiv / under review]

Program supervision

We theoretically show through PAC-bounds that program supervision as auxiliary loss can decrease sample complexity under some hypotheses.



(a) Classical training, CE loss on answers **y**

(b) CE loss on **y** + program prediction **p**

[Kervadec, Antipov, Baccouche, Wolf, arxiv / under review]

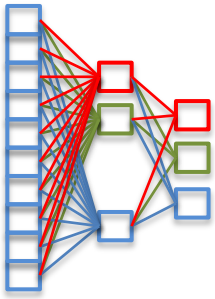
Oracle + program supervision

	Model	Oracle transf.	Prog. sup.	GQA-OOD [20] acc-tail	GQA-OOD [20] acc-head	GQA [17] test-dev	GQA [17] binary*	GQA [17] open*	GQA [17] test-std	AUC† prog.
scratch	(a) Baseline			42.9	49.5	52.4	-	-	-	/
	(b) Oracle transfer	✓		48.2±0.3	54.6±1.1	57.0±0.3	74.5	42.1	57.3	/
	(c) Ours	✓	✓	**48.8**±0.1	**56.1**±0.3	**57.8**±0.2	**75.4**	**43.0**	**58.2**	97.1
+ lxmert	(d) Baseline			47.5	55.2	58.5	-	-	-	/
	(e) Oracle transfer	✓		47.1	54.8	58.4	77.1	42.6	58.8	/
	(f) **Ours**	✓	✓	**48.0**±0.6	**56.6**±0.6	**59.3**±0.3	**77.3**	**44.1**	**59.7**	96.4

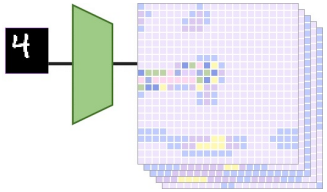
Table 1: Impact of program supervision on *Oracle transfer* [23] for vision-language transformers. LXMERT [36] pre-training is done on the GQA unbalanced training set. We report scores on GQA [17] (*test-dev* and *test-std*) and GQA-OOD (*test*). * binary and open scores are computed on the test-std; † we evaluate visual argument prediction by computing AUC@0.66 on GQA-val.

[Kervadec, Antipov, Baccouche, Wolf, arxiv / under review]

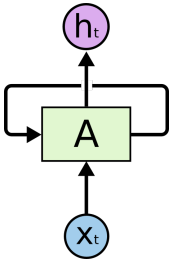
Conclusion (1)



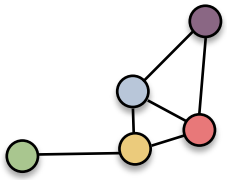
MLP



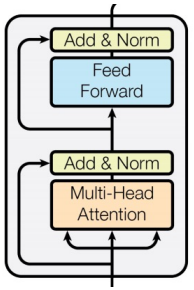
CNN / Convolutions



RNN / Recurrence



GN, GCN / Graphs, geometry



Transformers / Self-attention

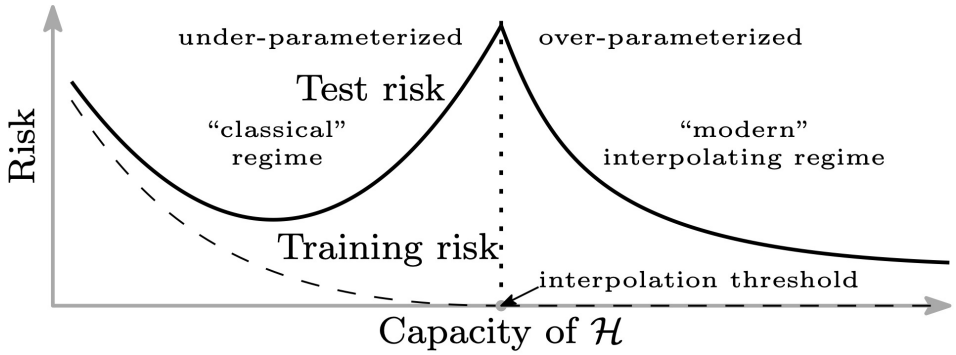
What do I know about the data and the task?

Nothing (vector space)	*Translation equivariance*	*Sequential data, Markov property*	*Graph structured data*	*Permutation equivariance*

Conclusion (2)

What is missing?

- Inductive biases for images
  - Convolutions vs. Transformers vs. MLPs
- ML formulations in deep learning
  - Self-supervised learning
  - Reinforcement Learning
- ML + physics (what do we model?)
- Theoretical models of Deep Learning



[Belkin et al., 2019]