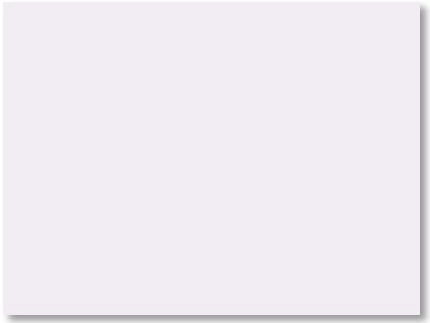
Learning high-level reasoning in vision, language and robotics

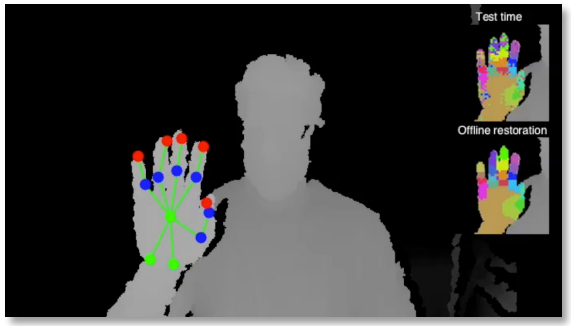
Christian Wolf
September 15th, 2021

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

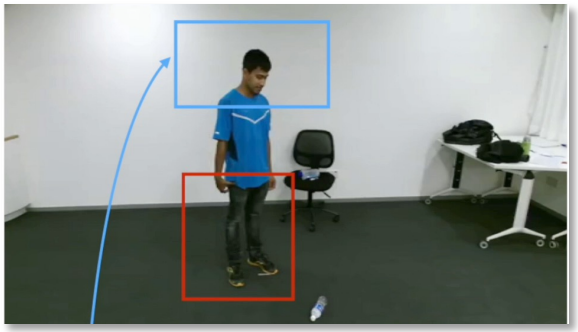
Learning vision & robotics

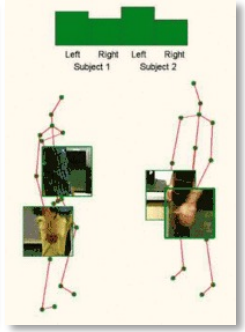


Gesture recognition

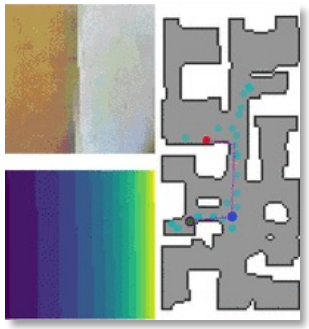


Pose estimation



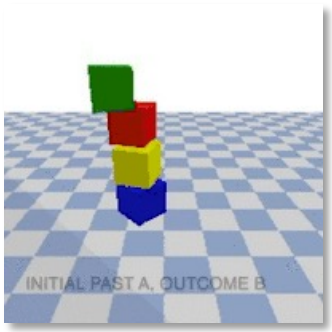


Activity Recognition

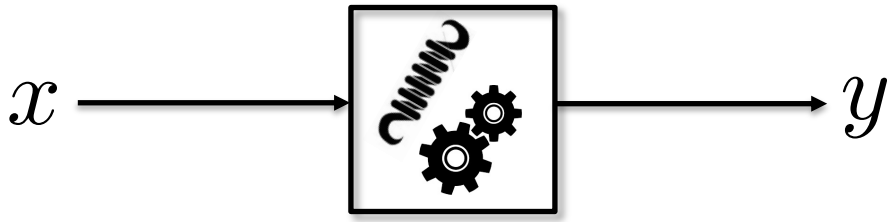


Robot Perception and Navigation

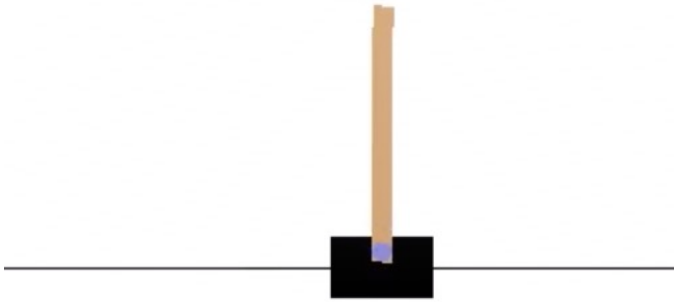
H-C Interaction



Physics

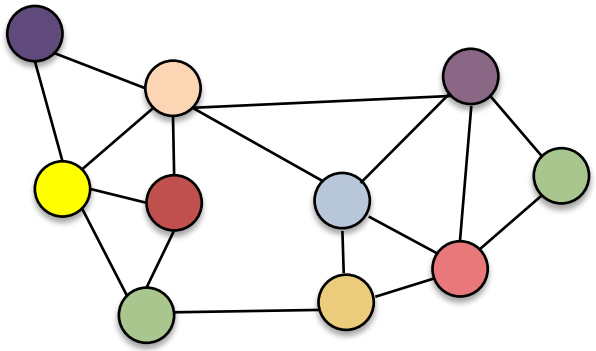


(Physical) model



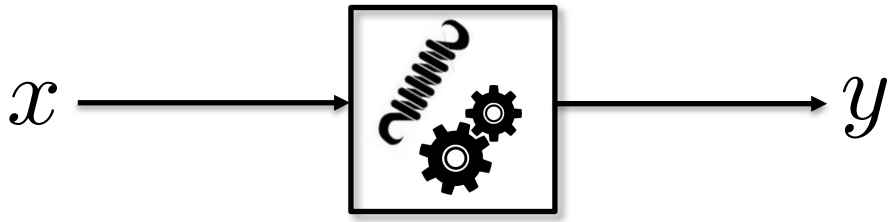
$$\ddot{\theta}_t = \frac{g \sin\theta_t + \cos\theta_t \left[\frac{-F_t - ml\dot{\theta}_t^2 \sin\theta_t + \mu_c \operatorname{sgn}(\dot{x}_t)}{m_c + m} \right] - \frac{\mu_p \dot{\theta}_t}{ml}}{l\left[\frac{4}{3} - \frac{m\cos^2\theta_t}{m_c + m} \right]}$$

$$\ddot{x}_t = \frac{F_t + ml\left[\dot{\theta}_t^2 \sin\theta_t - \ddot{\theta}_t \cos\theta_t\right] - \mu_c \operatorname{sgn}(\dot{x}_t)}{m_c + m}$$

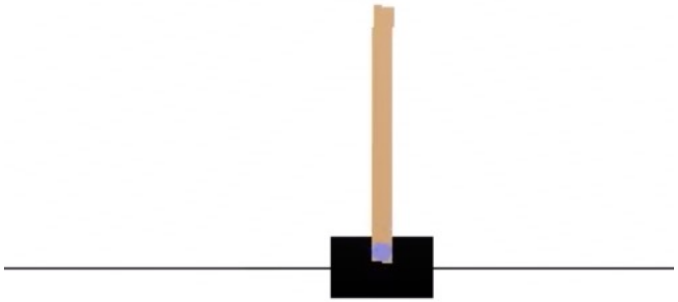


Planing/shortest path

- **Dijkstra**
- **A***
- **Front Propagation**

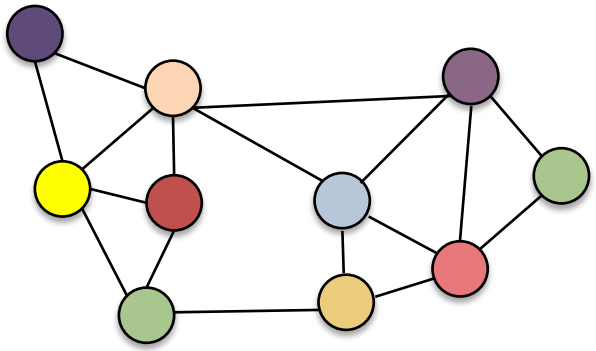


(Physical) model



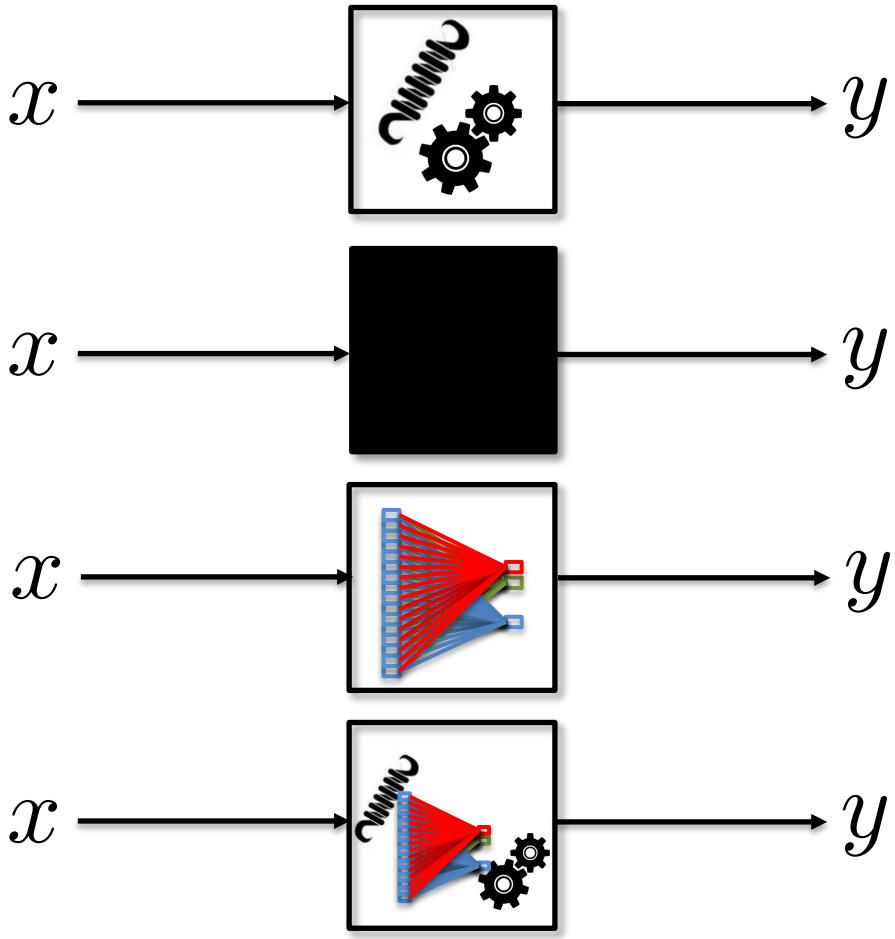
$$\ddot{\theta}_t = \frac{g \sin\theta_t + \cos\theta_t \left[\frac{-F_t - ml\dot{\theta}_t^2 \sin\theta_t + \mu_c \operatorname{sgn}(\dot{x}_t)}{m_c + m} \right] - \frac{\mu_p \dot{\theta}_t}{ml}}{l\left[\frac{4}{3} - \frac{m\cos^2\theta_t}{m_c + m} \right]}$$

$$\ddot{x}_t = \frac{F_t + ml\left[\dot{\theta}_t^2 \sin\theta_t - \ddot{\theta}_t \cos\theta_t\right] - \mu_c \operatorname{sgn}(\dot{x}_t)}{m_c + m}$$



Plannifcation/shortest path

- **Dijkstra**
- **A***
- **Front Propagation**



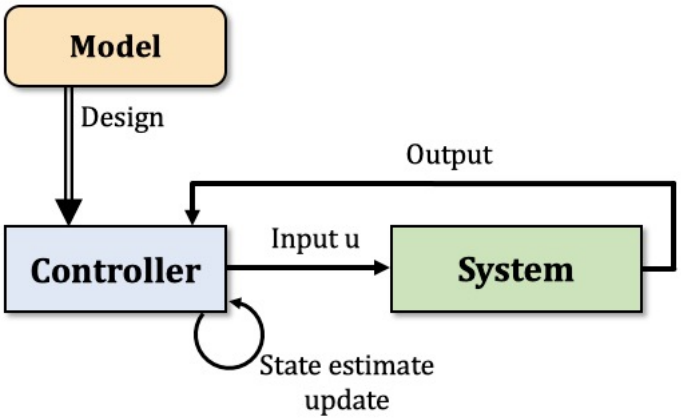
(Physical) model

Black box (unknown function)

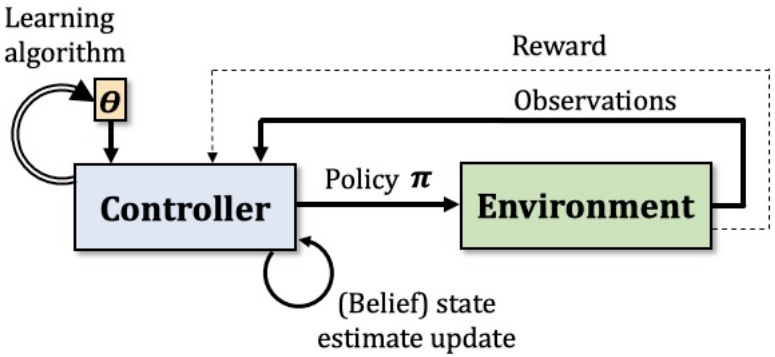
White box (learned but known complex function)

Hybrid model

ML + Control



(a) The control theory viewpoint



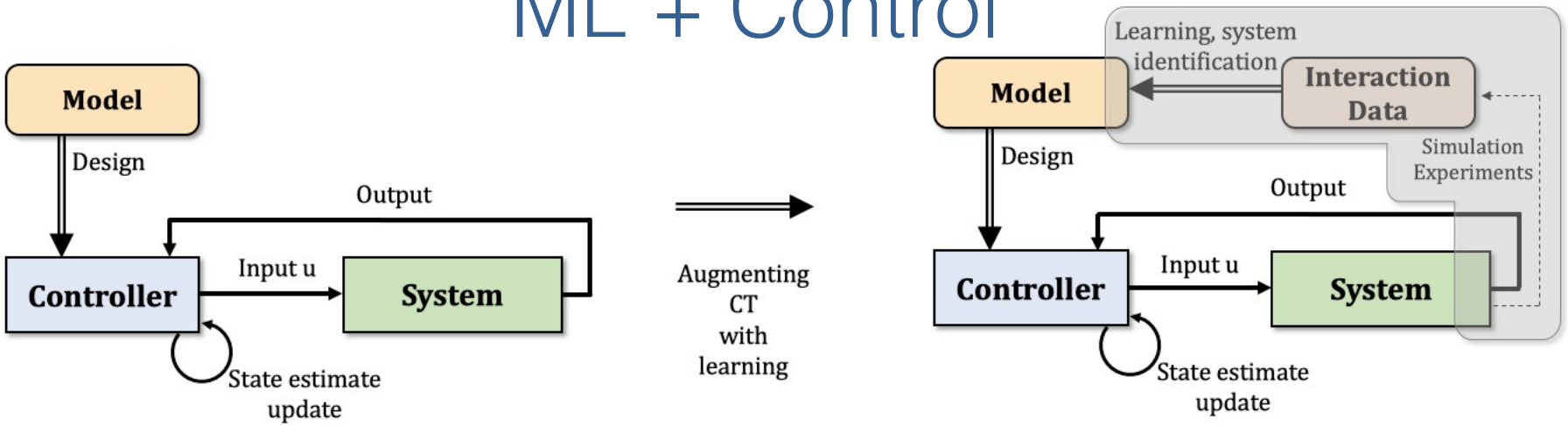
(b) The machine learning viewpoint

+ Optimality,
+ Stability guarantees

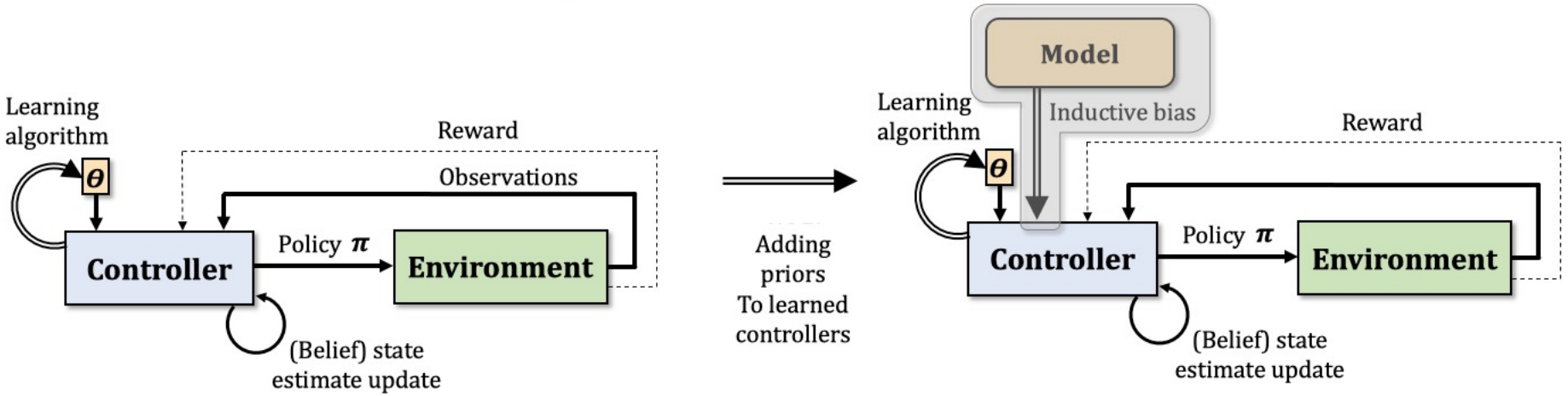
+ Learning from interactions,
no need for physical model

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR
ANR

ML + Control



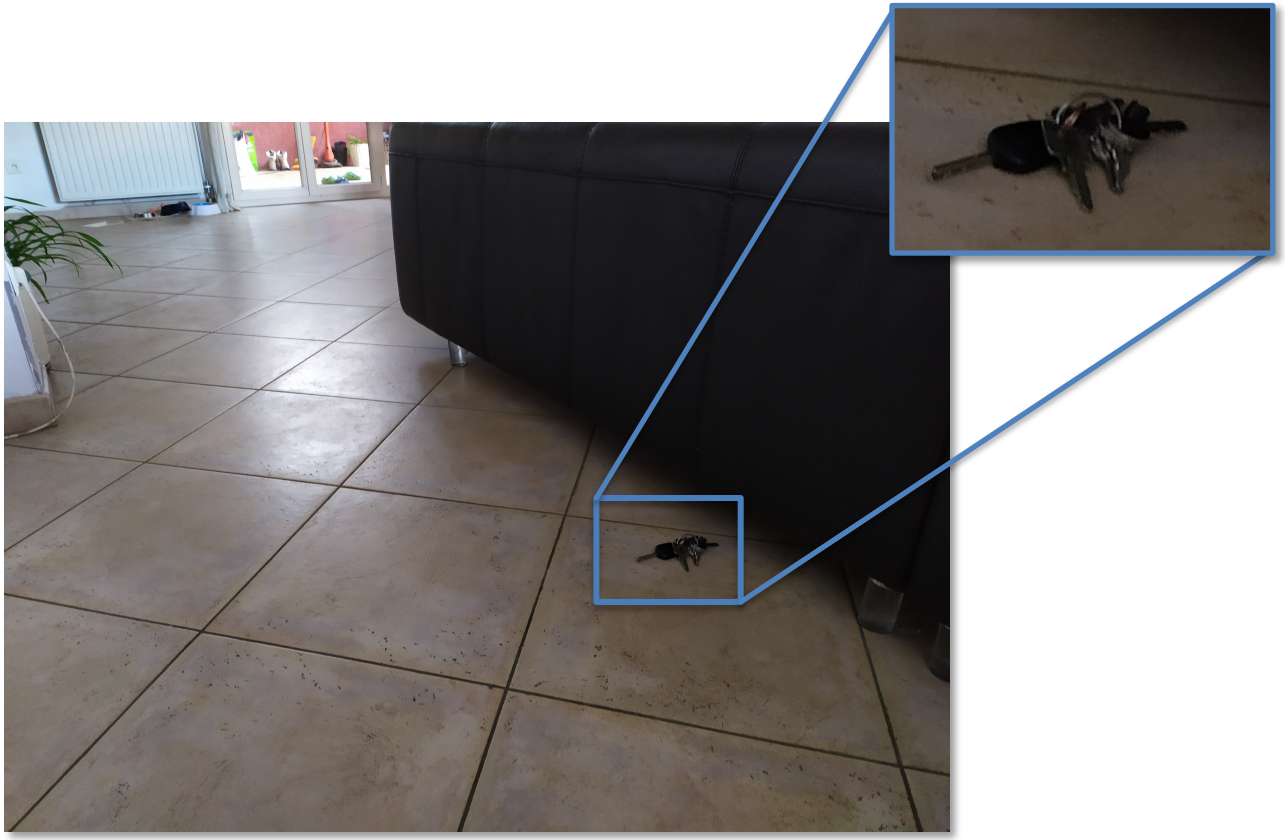
(a) The control theory viewpoint



(b) The machine learning viewpoint

<u>Goal:</u> train a physical agent (robot) to be spatially and semantically aware and to perform high-level reasoning tasks (eg assisting humans) in a possibly large environment, generalizing to unseen environments.

Discovering object affordances



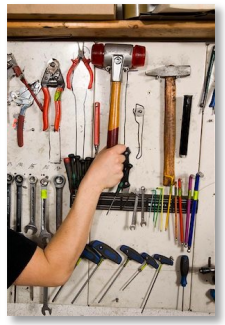
Reasoning and shortcuts in learning

Train for classification including
a class "to nail"

Test on data including new classes
"remove a nail", "store a hammer »



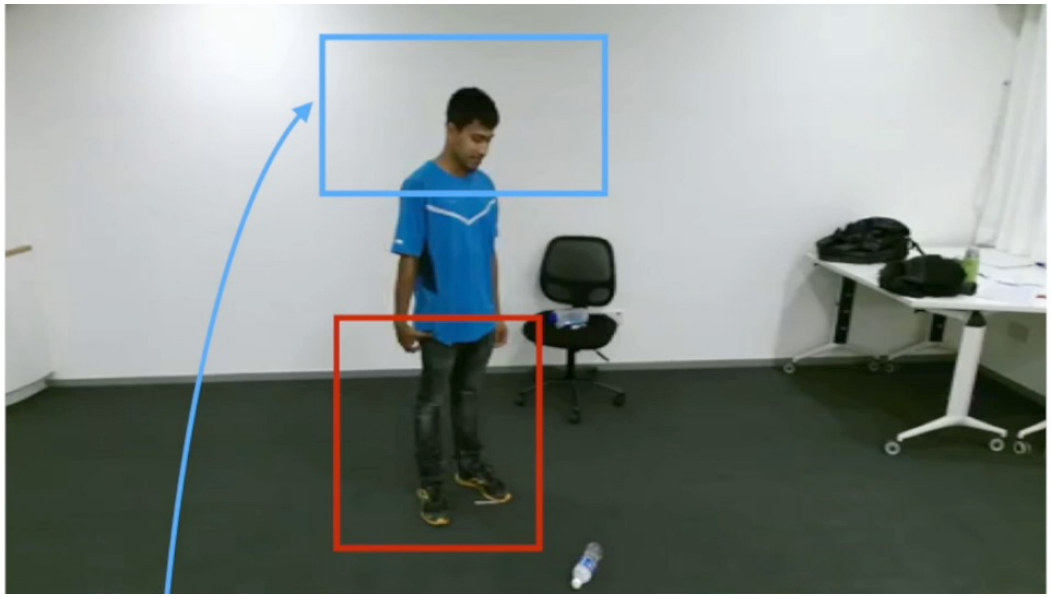




WYGISNWYE:

What you get is not what you expected!

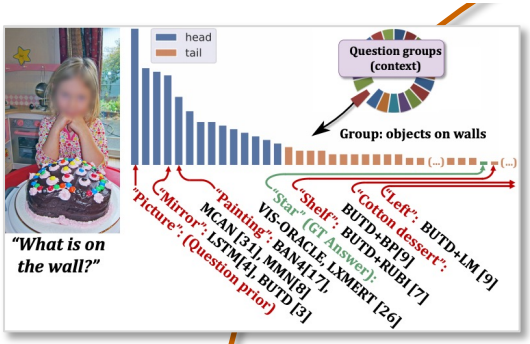
WYGISNWYE



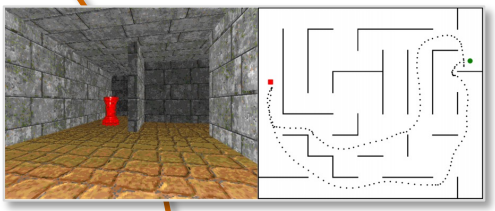
[Baradel, Wolf, Mille, Taylor, CVPR 2018]

Learning to reason

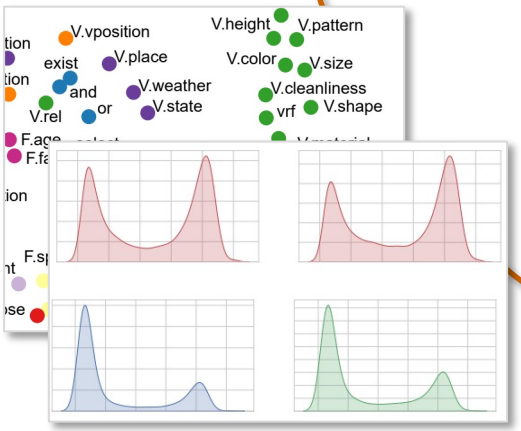
How can we evaluate biases in learning? (CVPR 2021a)



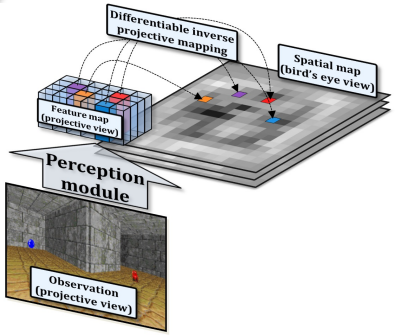
Which tasks favor emergence of reasoning? (ICLR 2020, ECML-PKDD 2020)



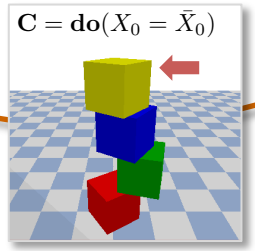
How can we visualize and transfer reasoning? (CVPR 2021b, VIS 2021)



How can we create inductive bias for reasoning? (ECCV 2018, ECCV 2020, CVPR 2021c, ECAI 2020, ECML-PKDD 2020)



What are the causal links in the data? (ICLR 2020)



Visual navigation and spatial reasoning

Office space

Homes

Hospitals

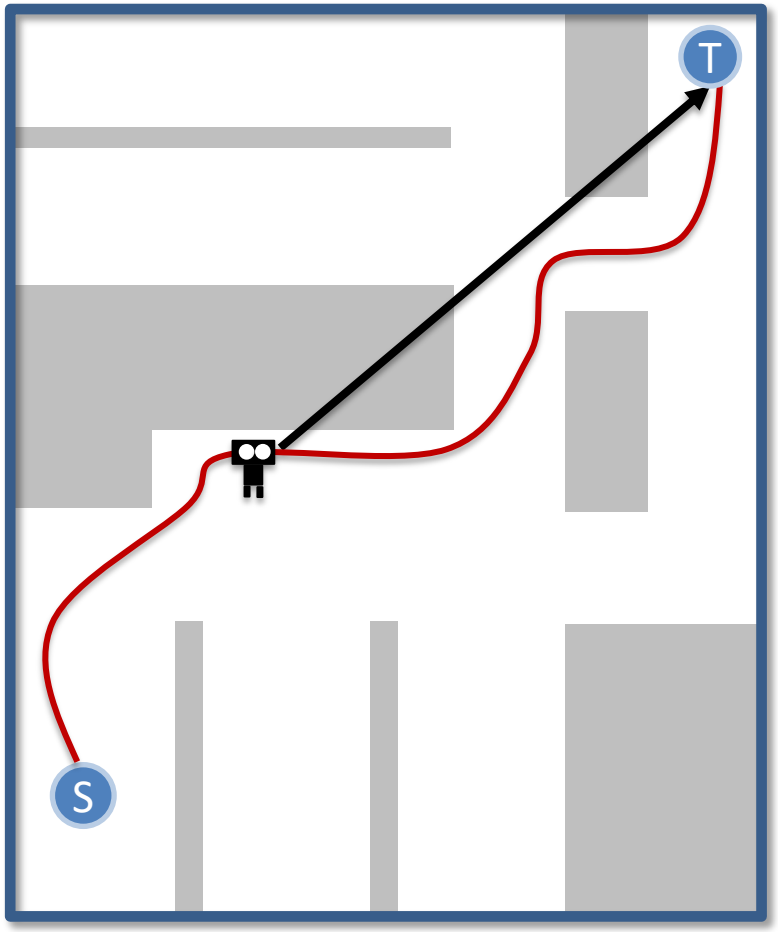
Edward Beeching

Jilles Dibangoye

Olivier Simonin

Christian Wolf

Task: PointGoal (+GPS)



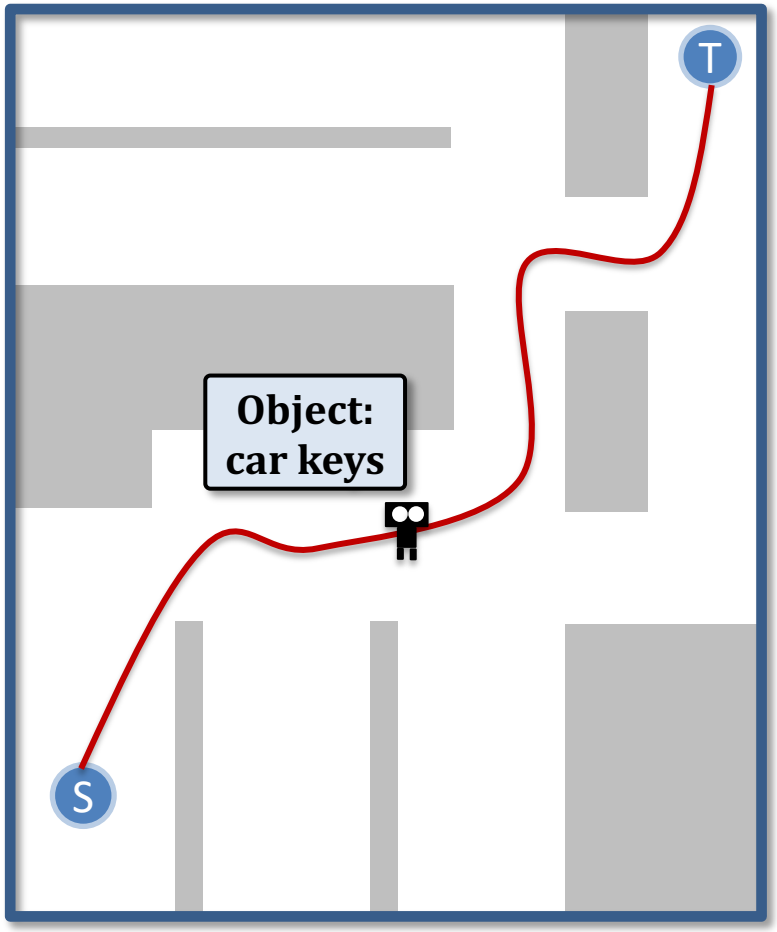
Task:

Find a target given coordinates, receive a direction vector et each step

Required reasoning:

- Recognize free navigation space
- Follow the direction vector
- Learn how to overcome obstacles (difference between Euclidean and geodesic path)

Task: ObjectNav



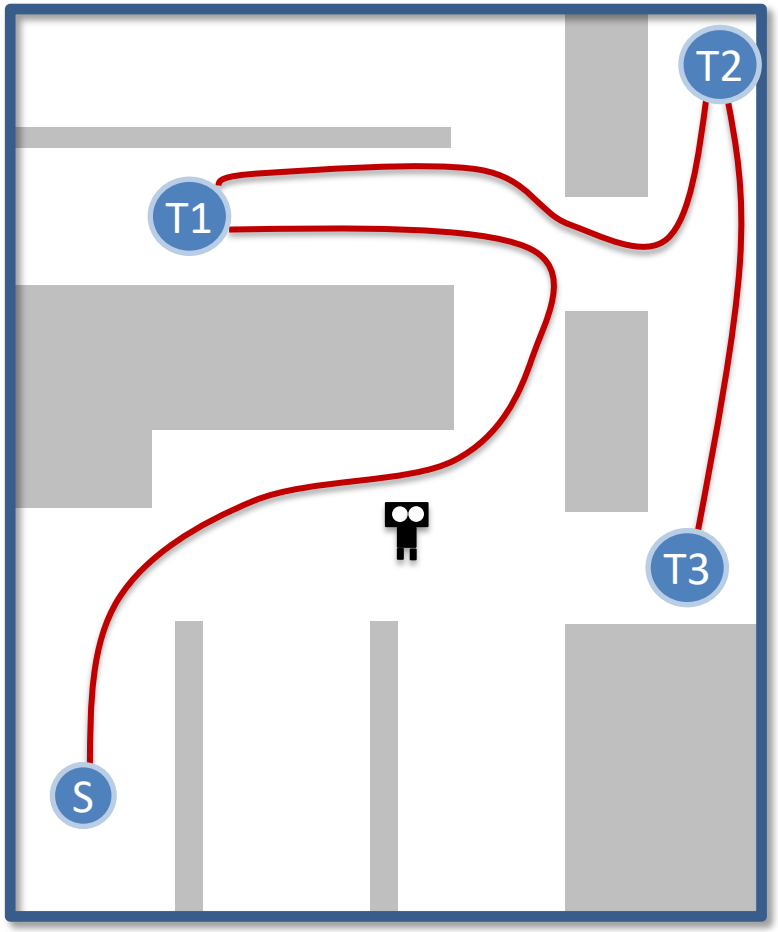
Task:

Find a target object given its object class.

Required reasoning:

- Recognize free navigation space
- *Explore the environment*
- *Recognize the object when seen and move towards it.*

Task: K-item scenario



Task:

Navigate to a list of objects sequentially in the right order.

Required reasoning:

- Recognize free navigation space
- Explore the environment
- *Map an object if I need to find it later (!)*
- Recognize the object when seen and move towards it.

[Beeching, Dibangoye, Simonin, Wolf, ECML-PKDD 2020]

[Beeching, Dibangoye, Simonin, Wolf, ICPR 2020]

What do we want the model to learn?

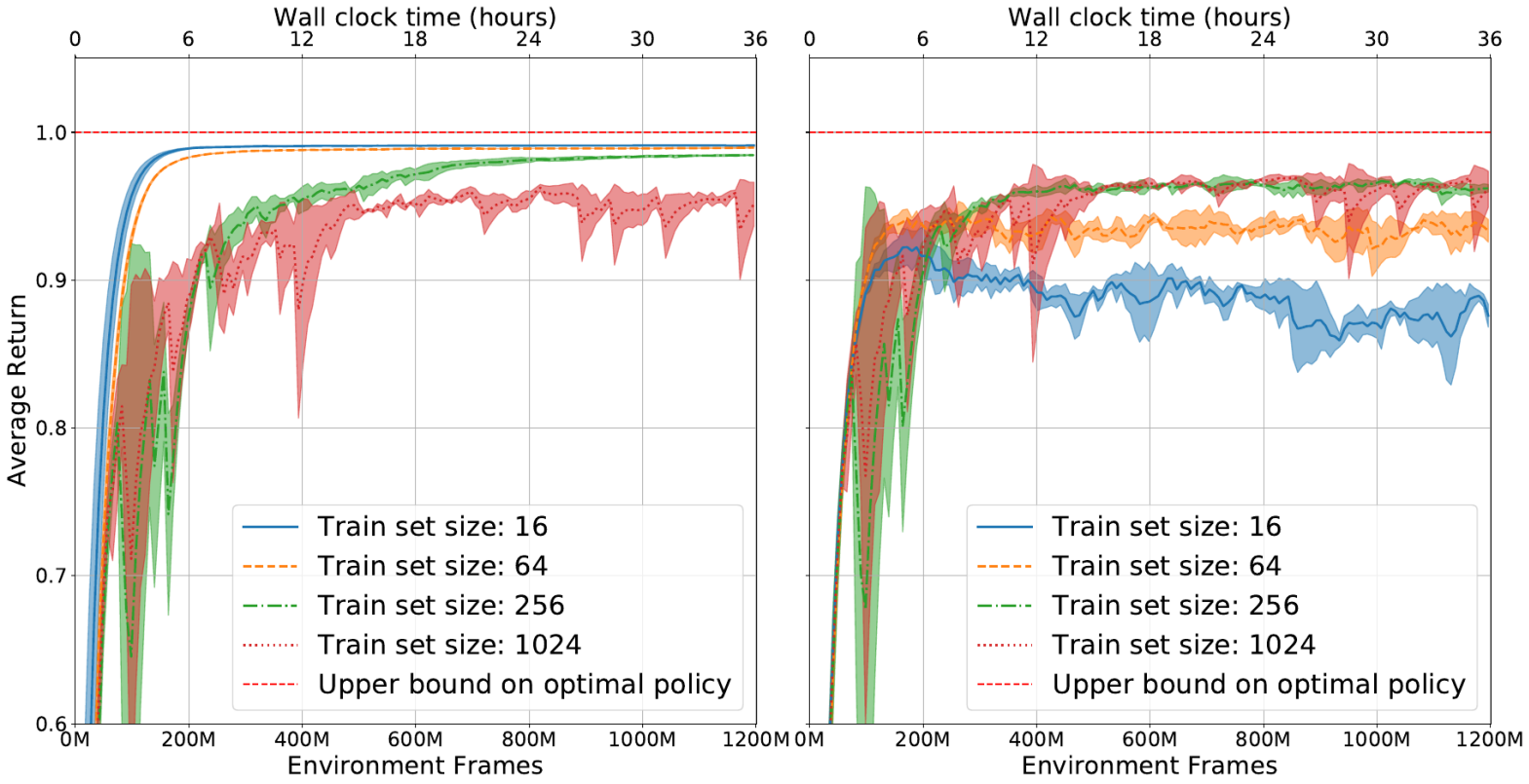
Generalization to new environments requires:

	Agent memory	Network parameters
Information on the (seen) environment, e.g. position of a couch	✓	✗
Positions of objects placed in the environment	✓	✗
Regularities of the environment (eg. bath tubs are in bathrooms; toilets are accessible from an aisle, not the living room)	✗	✓
Object affordances	✓	✓

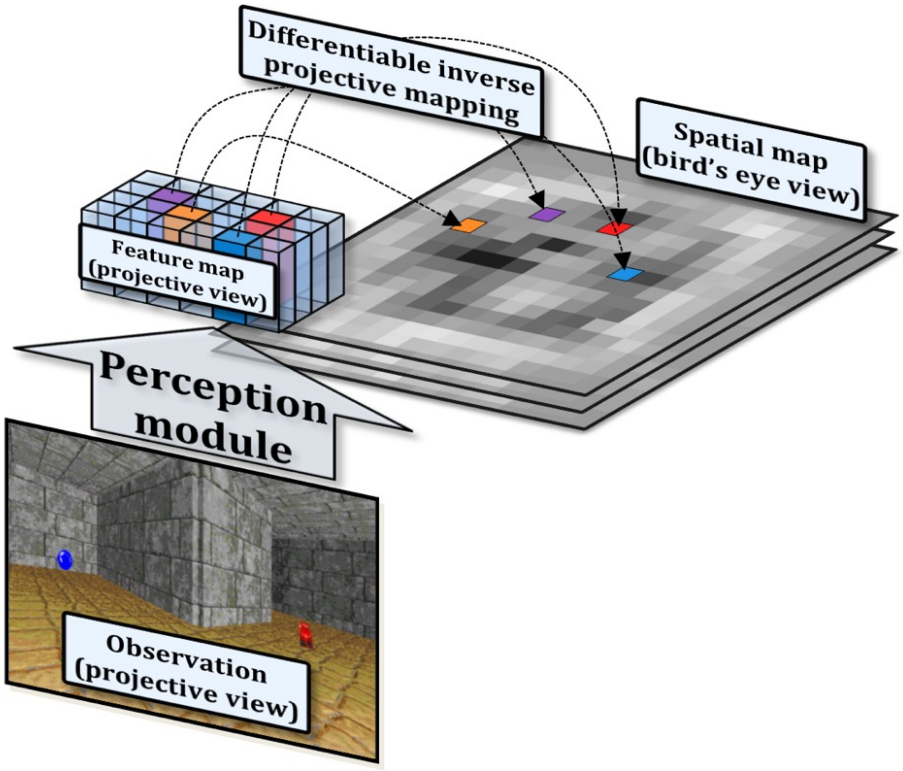
The task formulation decides how learned information is stored!

Tasks, regularities and generalization

What does my network learn:
A reasoning strategy, or the environment?

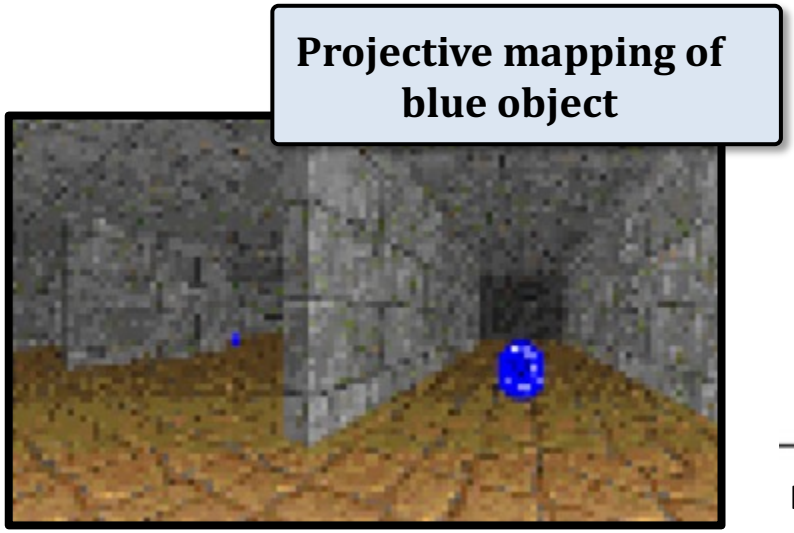


Inductive bias for projective mapping

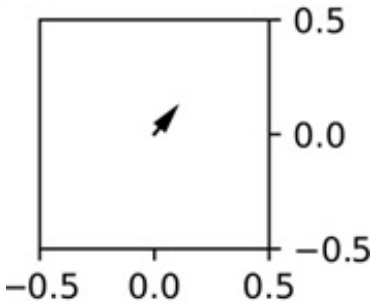


[Beeching, Dibangoye, Simonin, Wolf, ECML-PKDD 2020]

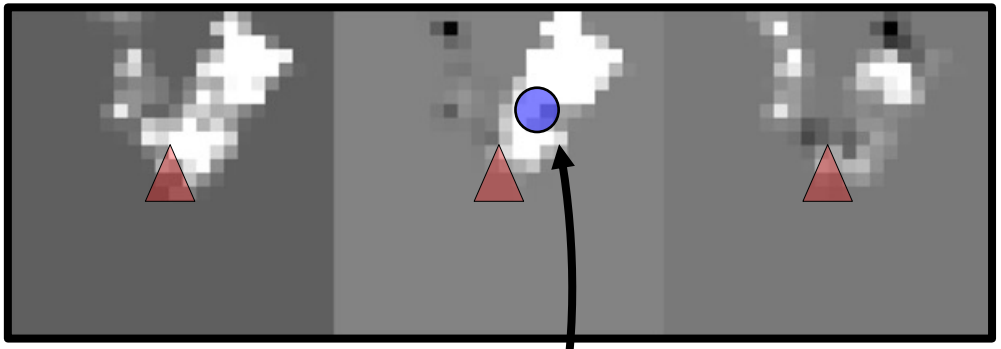
6 item scenario: time-step 005



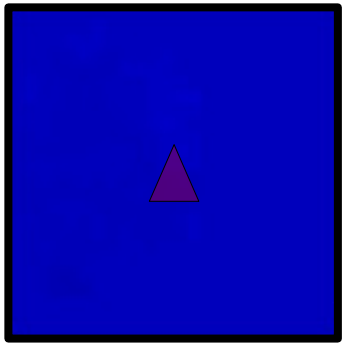
Projective mapping of blue object



EgoMap: Query position



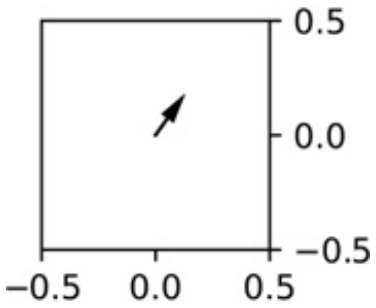
EgoMap: 3 Largest principal components



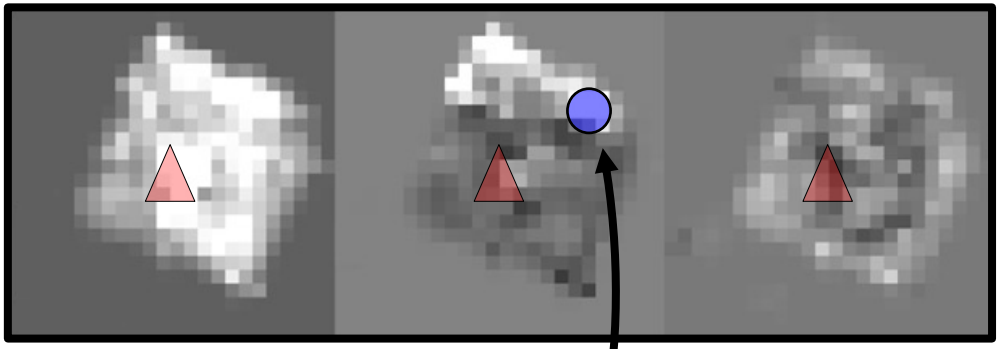
EgoMap: Attention

Object features retained in map

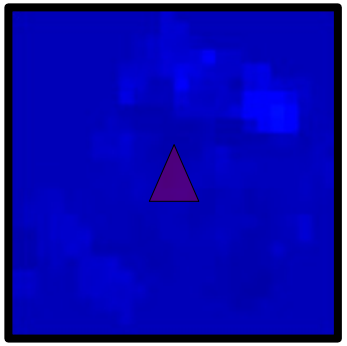
6 item scenario: time-step 105



EgoMap: Query position



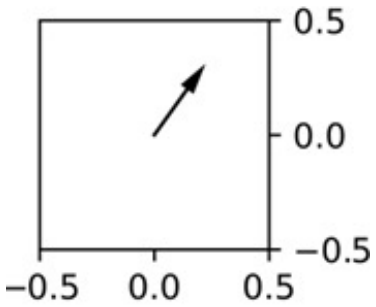
EgoMap: 3 Largest principal components



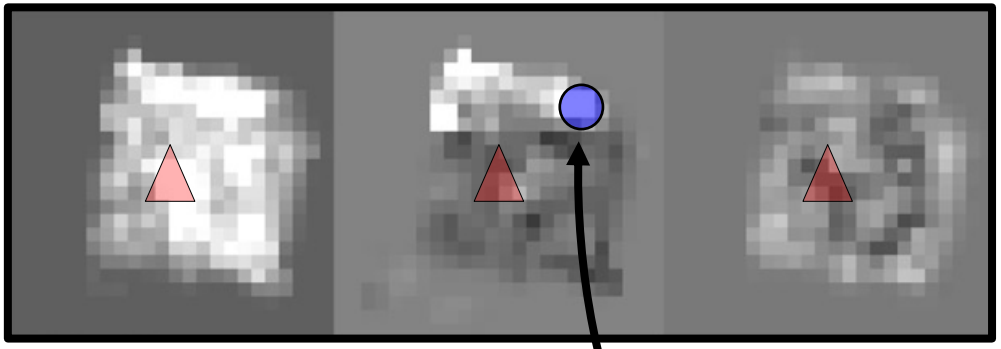
EgoMap: Attention

Object features retained in map

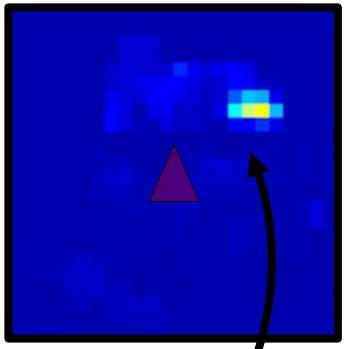
6 item scenario: time-step 108



EgoMap: Query position



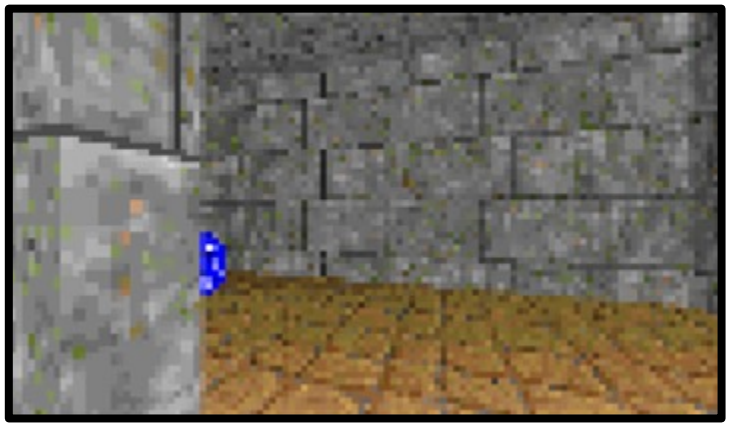
EgoMap: 3 Largest principal components

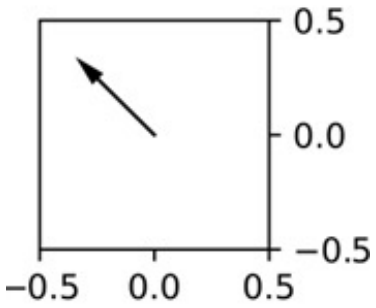


EgoMap: Attention

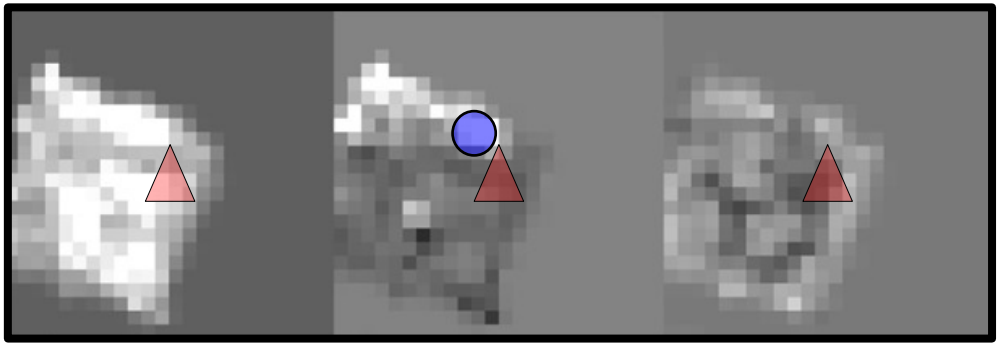
Collection of object n-1 triggers attention to object n

6 item scenario: time-step 134

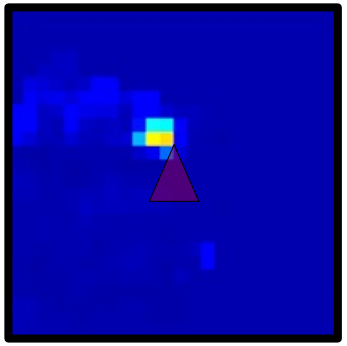




EgoMap: Query position

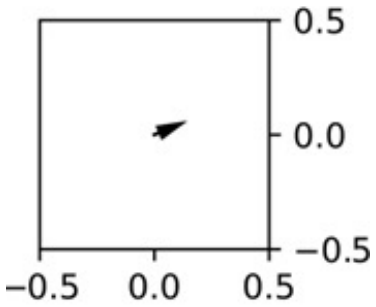


EgoMap: 3 Largest principal components

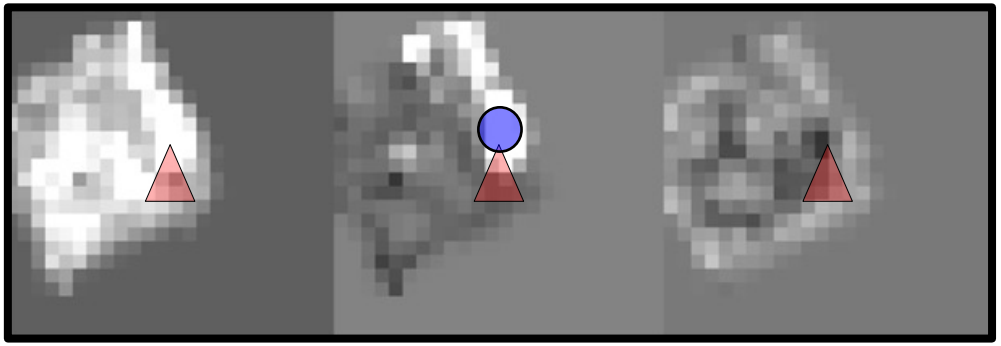


EgoMap: Attention

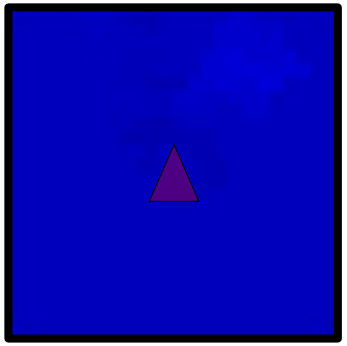
6 item scenario: time-step 140



EgoMap: Query position



EgoMap: 3 Largest principal components



EgoMap: Attention

When the object is not occluded, the agent does not attend to it

Self-supervision for Deep-RL

1st ranked at Multi-ON Challenge (CVPR 2021)

Rank	Team	Progress	PPL	Success	SPL
1	Lyon	67	44	55	35
2	SGoLAM	64	38	52	32
3	VIMP	57	36	41	26

1st place: Team Lyon, *Pierre Marza, Laetitia Matignon, Olivier Simonin, Christian Wolf*, Learning mapping and spatial reasoning with auxiliary tasks.

[Marza, Matignon, Simonin, Wolf, Submitted to ICRA 2021]

How about real robots?

NAVER LABS EUROPE

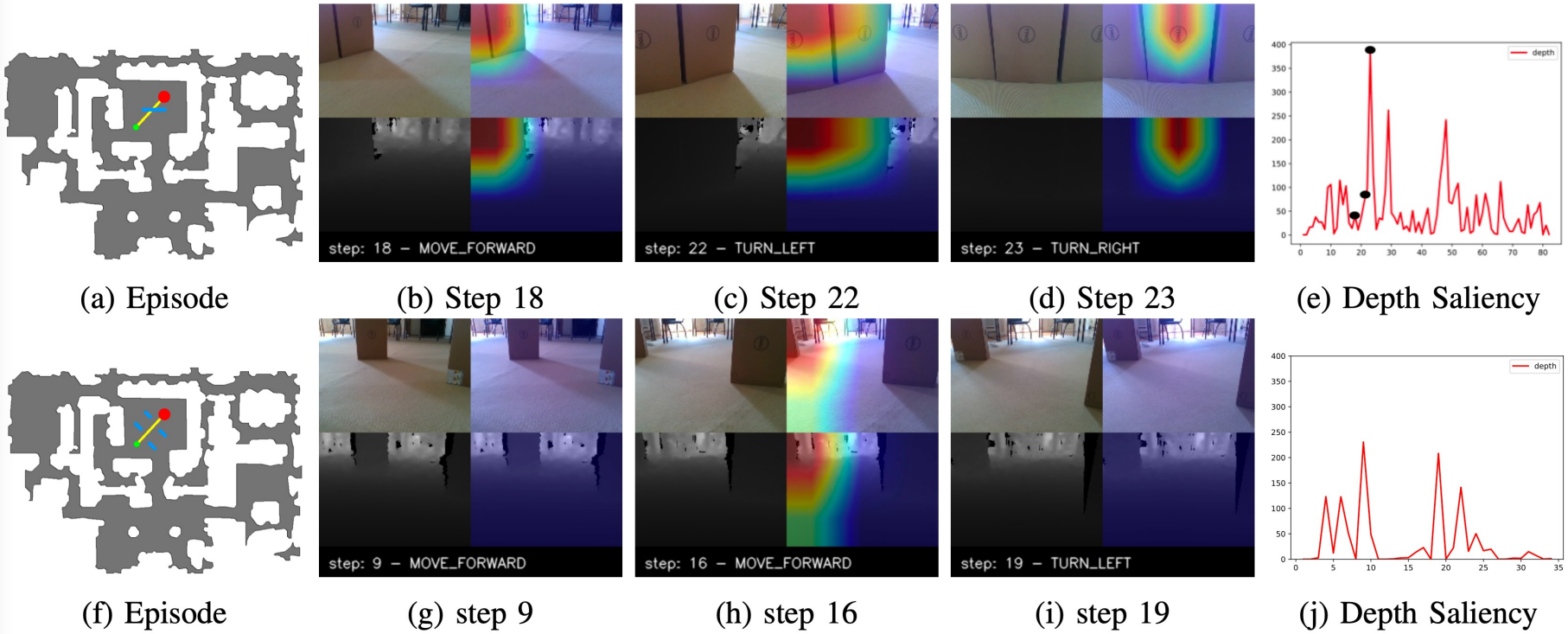

Assem Sadek


Guillaume Bono


Boris Chidlovskii


Christian Wolf

How about real robots?

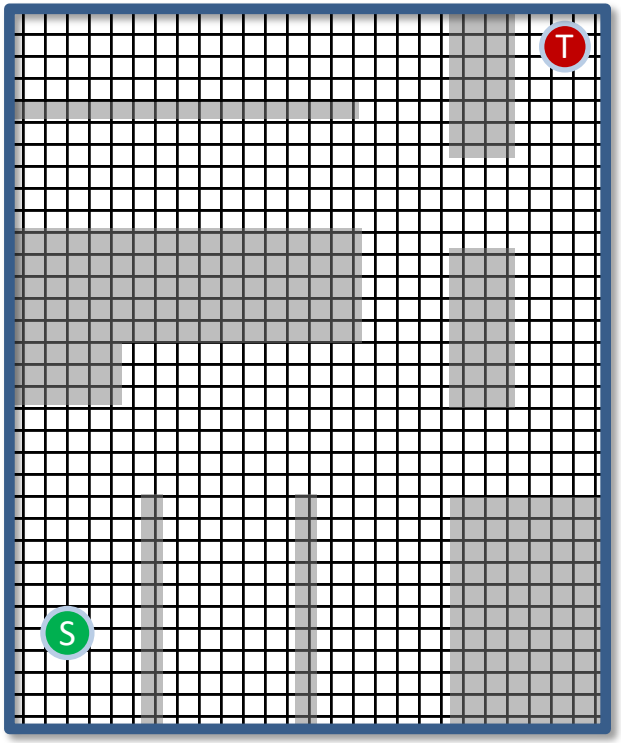


(a) Episode (b) Step 18 (c) Step 22 (d) Step 23 (e) Depth Saliency

(f) Episode (g) step 9 (h) step 16 (i) step 19 (j) Depth Saliency

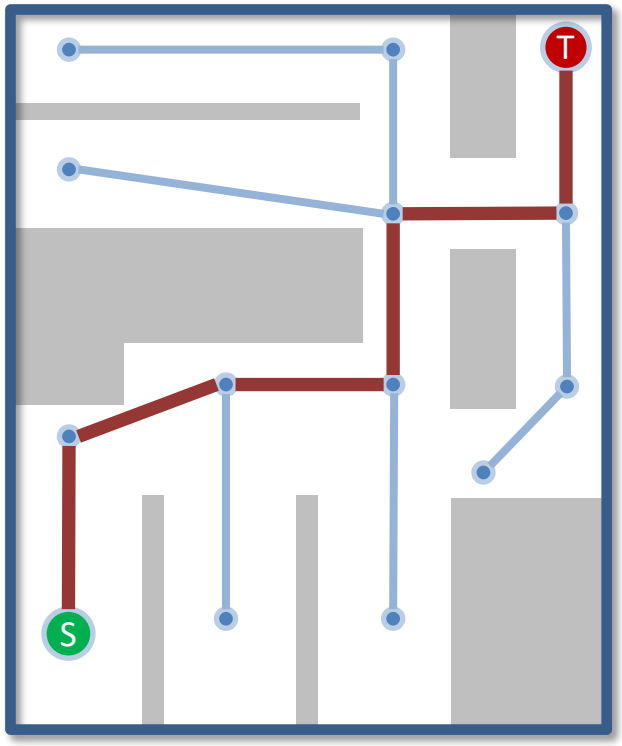
[Sadek, Bono, Chidlovskii, Wolf, Submitted to ICRA 2021]

Spatial maps in robotics



Metric map
(=2D or 3D Grid)

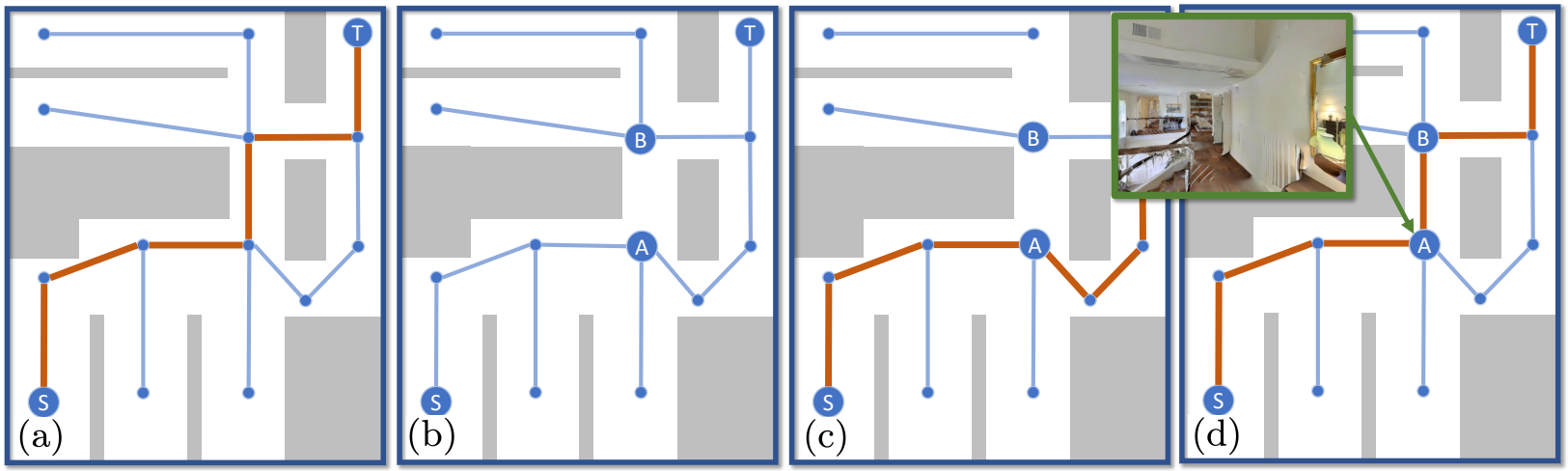
Beeching, Dibangoye, Simonin, Wolf,
ECML-PKKD 2020



Topological map
(=Graph)

Beeching, Dibangoye, Simonin, Wolf,
ECCV 2020

Planning under uncertainty



$D_{i,j}$	*Spatial distance between nodes* i *and* j*.*
$E^*_{i,j}$	*GT connection* i *and* j $(\in \{0,1\})$*.*
$E_{i,j}$	*Estimated connection probability between* i *and* j*.*
$\mathcal{P}$	*Set of all paths of the fully connected graph.*
$\mathcal{P}_{E^*}$	*Set of all paths only using edges s.t.* $E^*_{i,j} = 1$*.*

Shortest path problem:

$$p^* = \arg\min_{p \in \mathcal{P}_{E^*}} \sum_{e \in p} \mathrm{D}_{e_{src}, e_{dst}}$$

Optimal planning under uncertainty

Problem: GT edges $E^*_{i,j}$ are not available!

Integrate distance and uncertainty into a single cost function:

$$\text{cost}(i,j) = \underbrace{\mathbf{D}_{i,j}}_{\text{Spatial dist.}} + \lambda \underbrace{\log \mathbf{E}_{i,j}}_{\text{Conn. prob.}}$$

New shortest path problem:

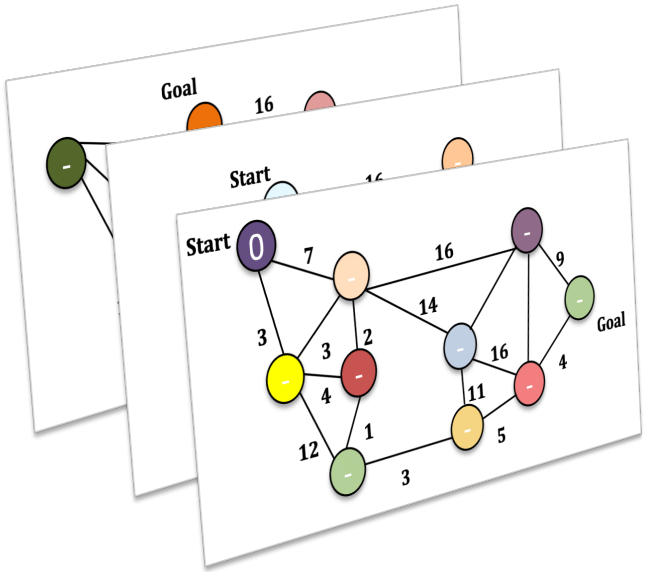
$$p^\dagger = \arg\min_{p \in \mathcal{P}} \sum_{e \in p} \text{cost}(e_{src}, e_{dst})$$

(Can be solved with classical algorithms — Dijkstra, Bellmann-Ford, . . .)

The optimal path $p^\dagger$ is not necessarily the ground truth path p^*!

Learning to plan

Can we exploit regularities in the connection probabilities *over a full set of uncertain training graphs*?

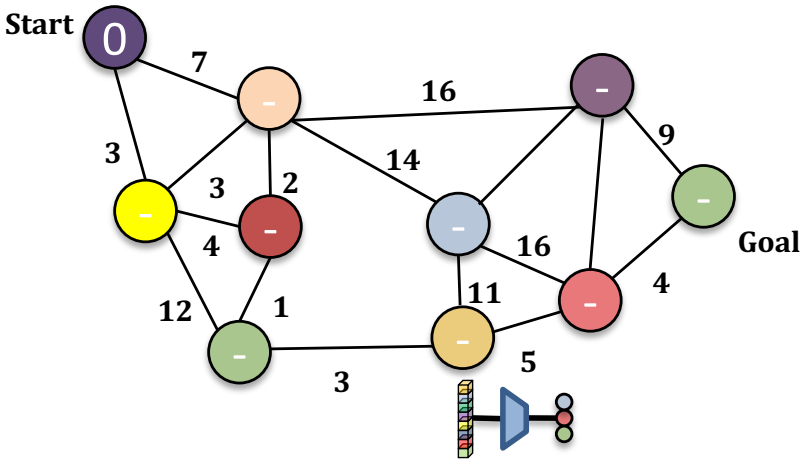


We

- train on a set of instances,
- predict for a given instance.

Learning to plan

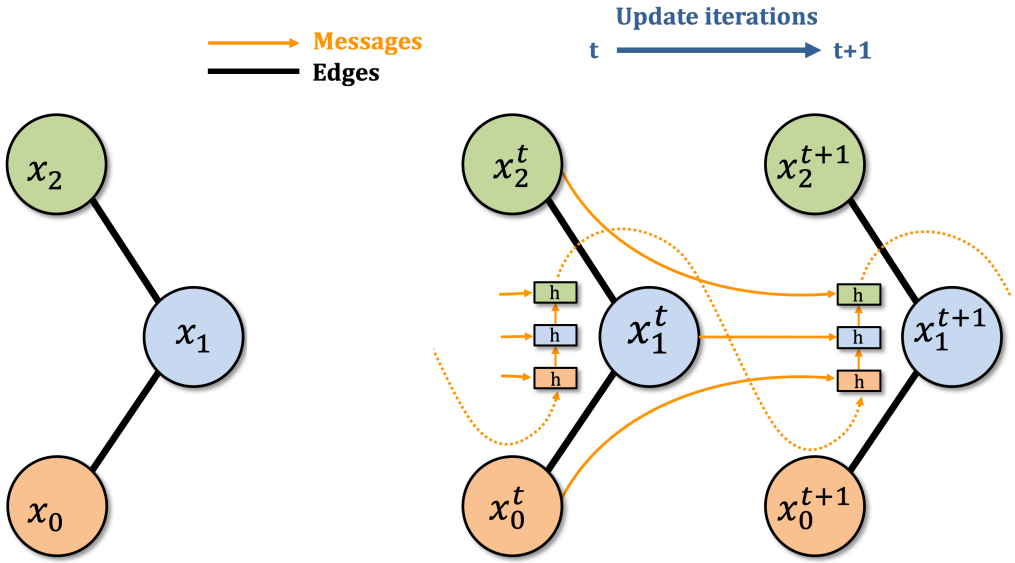
Formulate planning as a classification problem over a graph $\mathcal{G}$, supervise with GT paths p^*:



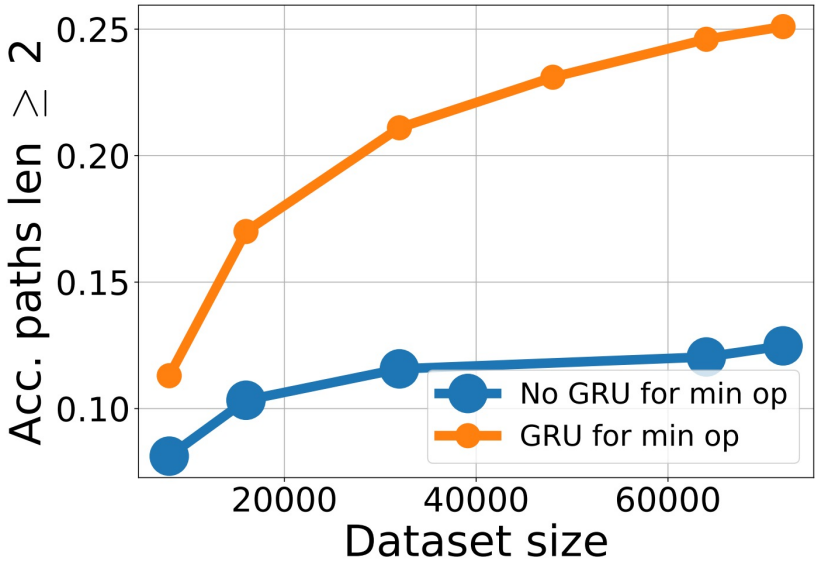
$$\hat{p} = \{(\hat{e}_{src}, \hat{e}_{dst})\}, \quad \hat{e}_{dst} = f(\hat{e}_{src}, \mathcal{G}; \theta)$$

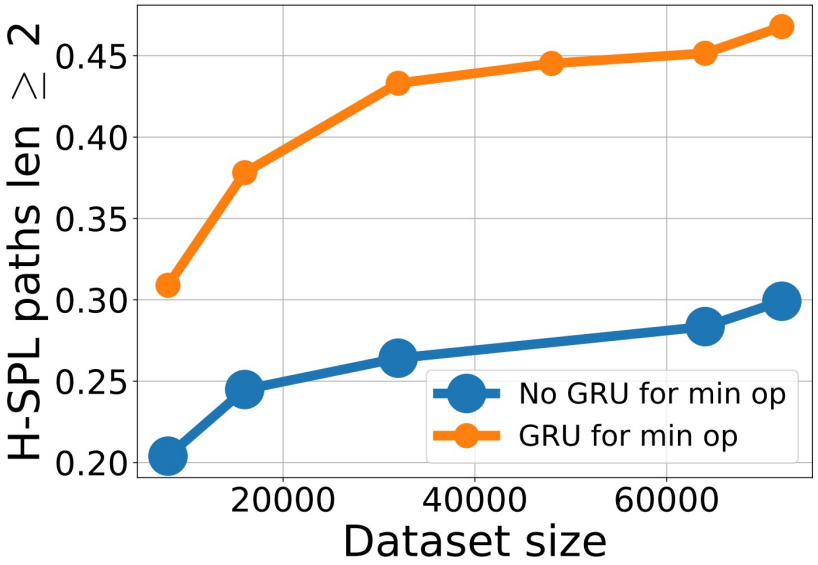
Inductive bias for bound updates

We add a new inductive bias to graph neural networks to better represent bound updates inherent in algorithms like Dijkstra or Bellmann-Ford:



Ablation of inductive bias





Results

Training on 75k graphs (= 75 000 000 problems).
Probabilities estimated by a neural network from visual observations taken from an RL-agent.

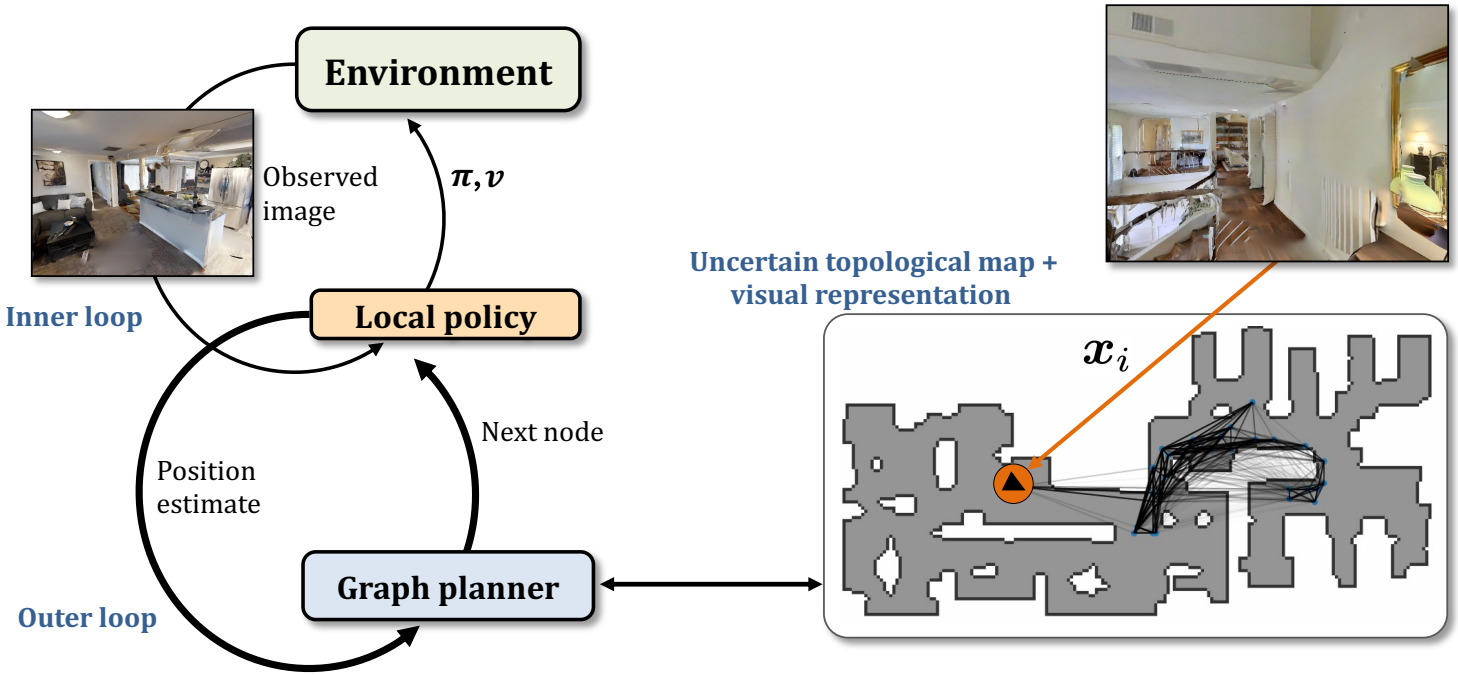
Method	**Acc**	**H-SPL**
Symbolic (threshold)	0.114	0.184
Symbolic (custom cost)	0.115	0.269
Neural (w/o visual)	0.251	0.468
Neural (w visual)	**0.262**	**0.501**

Uncertain graphs

Method	**Acc**	**H-SPL**
Symbolic (GT)	**1.00**	**1.00**
Neural planner (GT)	0.921	0.983

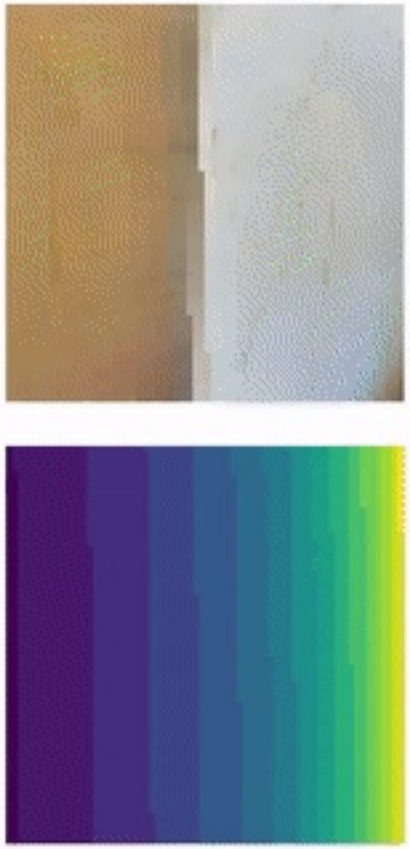
Ground truth graphs

Hierarchical planner



[Beeching, Dibangoye, Simonin, Wolf, ECCV 2020]

Hierarchical planning and control



[Beeching, Dibangoye, Simonin, Wolf, ECCV 2020]

Performance of hierarchical planner

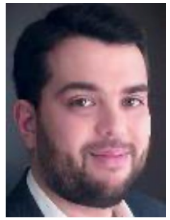
Method: Planner + Local policy	**Success rate**	**SPL**
Graph oracle (optimal point-goals, not comparable)	0.963	0.882
Random	0.152	0.111
Recurrent Image-goal agent	0.548	0.248
Symbolic (threshold)	0.621	0.527
Symbolic (custom cost)	0.707	0.585
Neural planner (sampling)	0.966	0.796
Neural planner (deterministic)	**0.983**	**0.877**

Vision and Language Reasoning

orange™


Corentin Kervadec


Grigory Antipov


Moez Baccouche


Christian Wolf

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

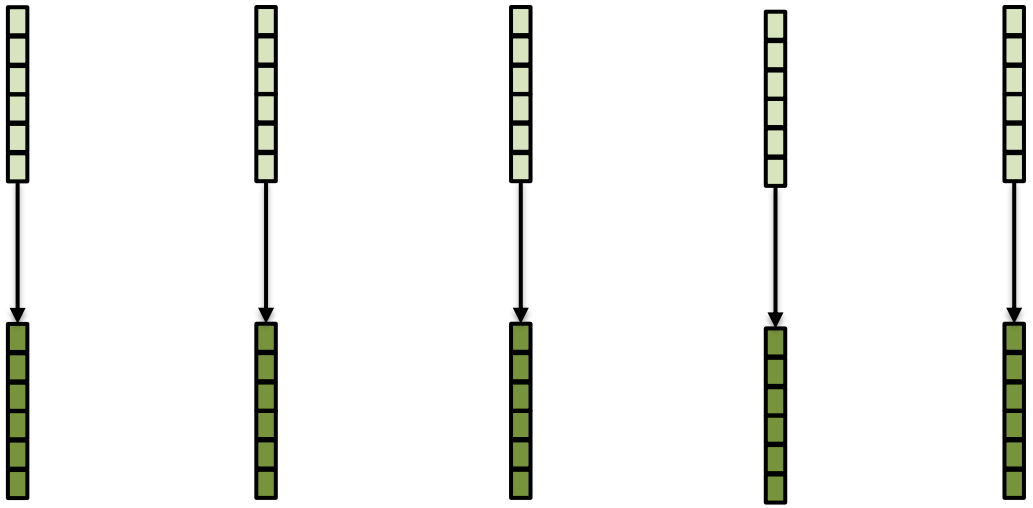
[L. Tolstoy, 1873]

Contextualization

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

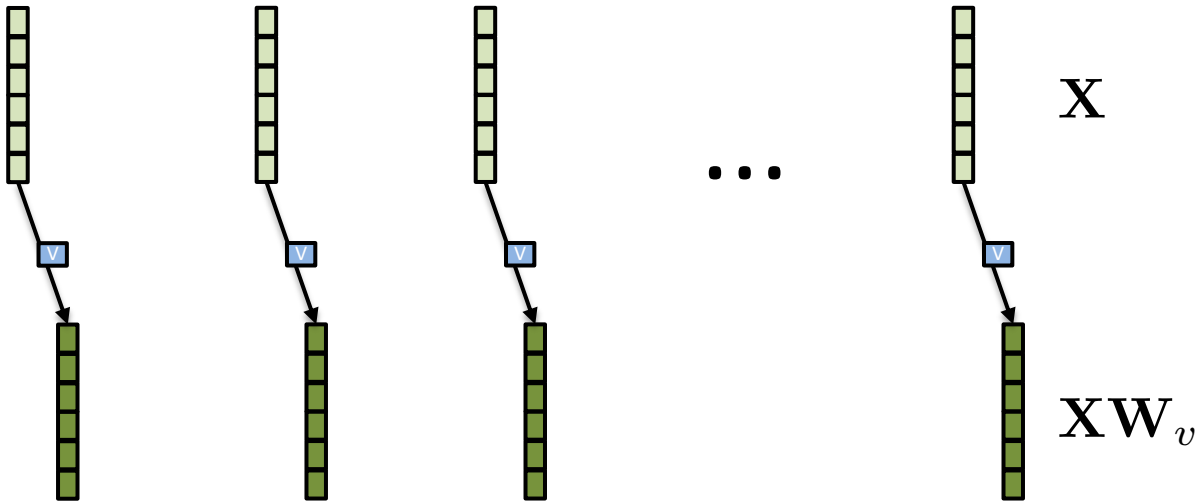
Iteratively "enrich" each item by providing context from the other items

The quick brown fox jumps ....



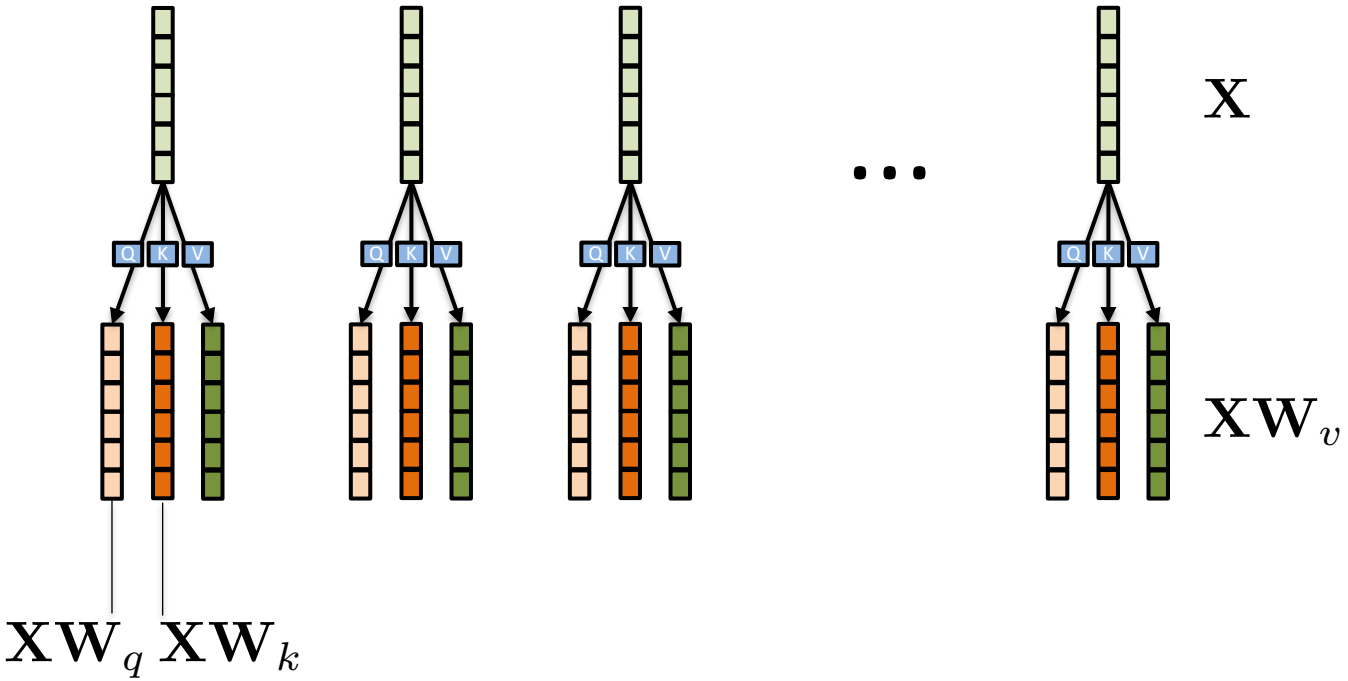
quick brown fox

Transformers: attention is all you need

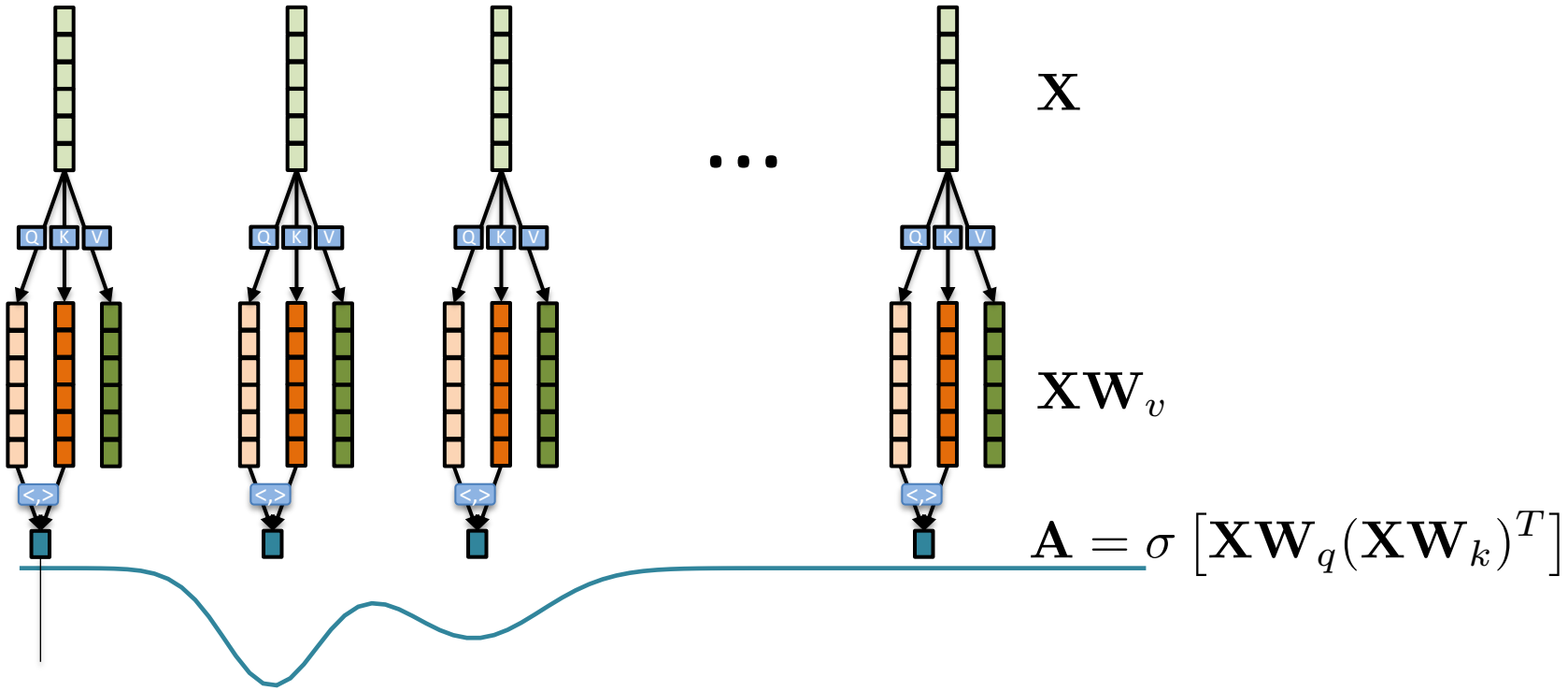


We suppose unordered data (a set) $\mathbf{X} = \{\mathbf{x}_i\}$. Each input item is "transformed" by a linear transform weighted by attention.

Transformers: attention is all you need



Transformers: attention is all you need



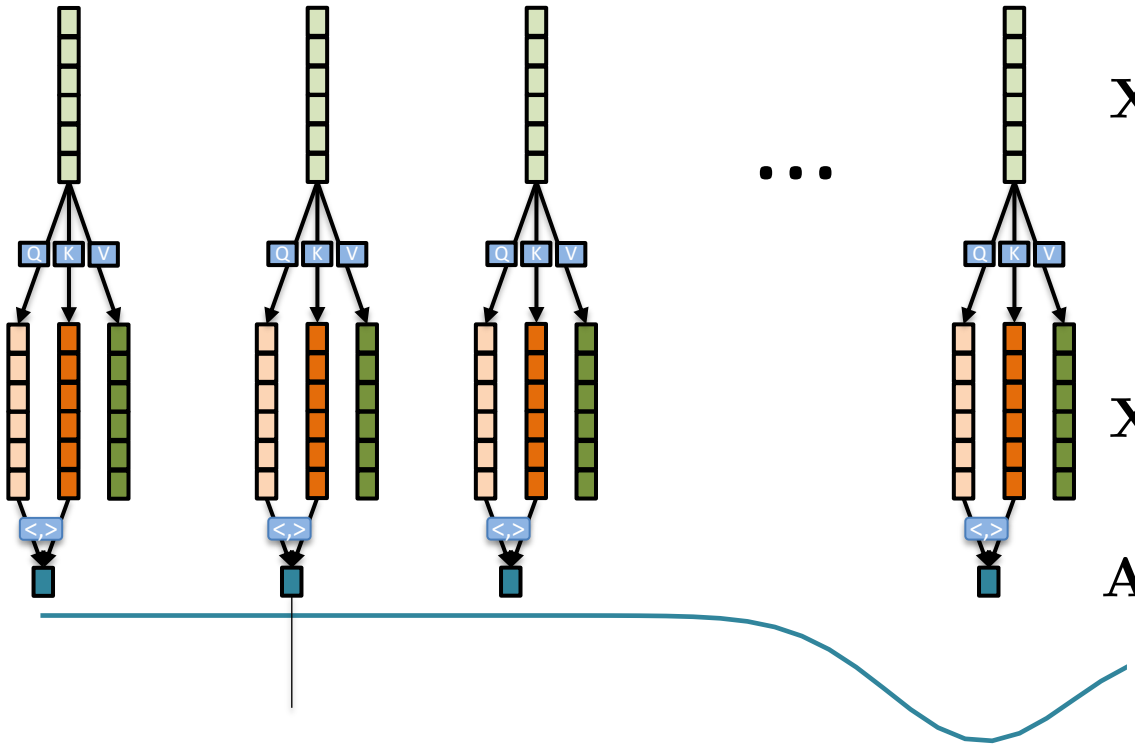
$\mathbf{X}$

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

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

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

Transformers: attention is all you need



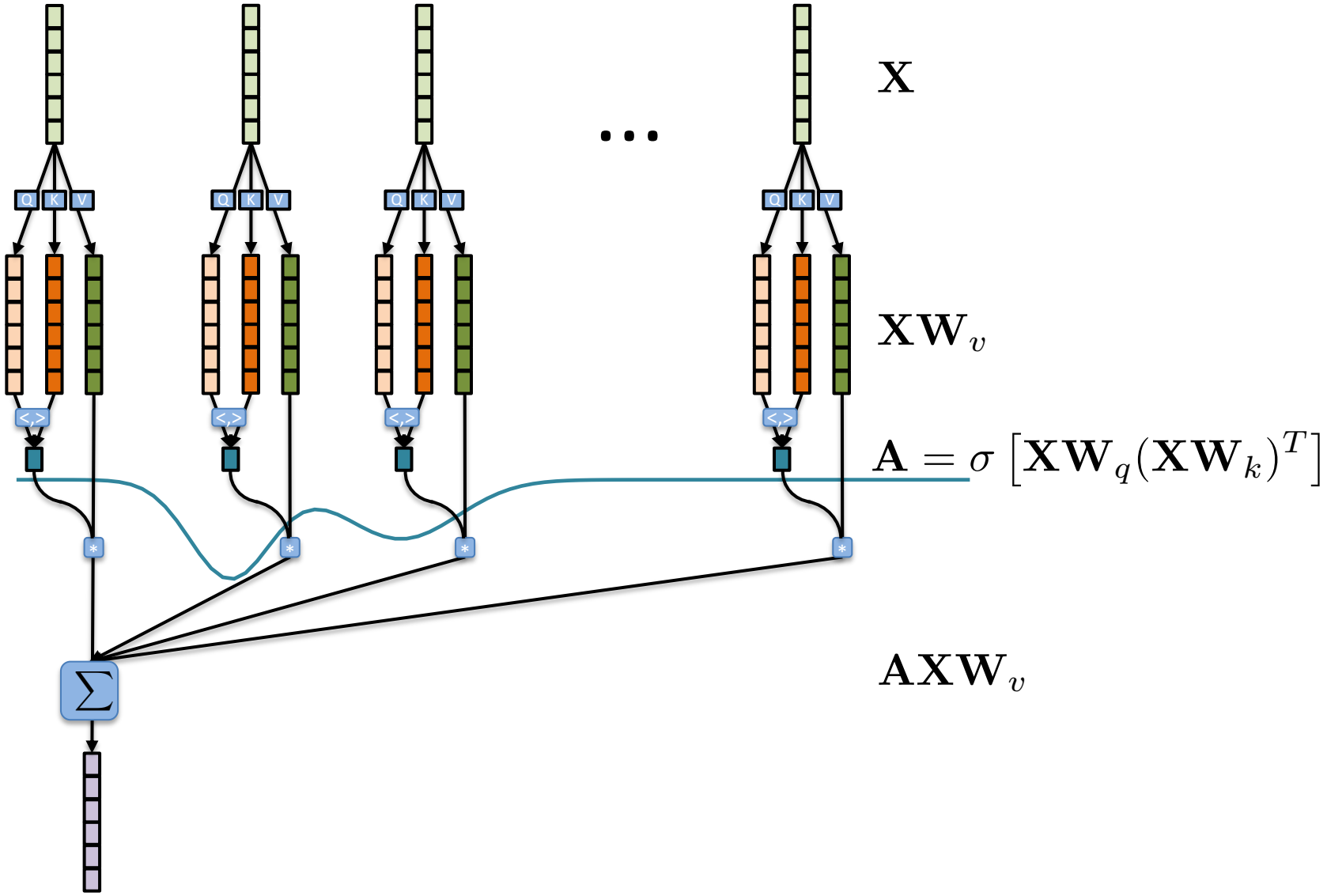
$\mathbf{X}$

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

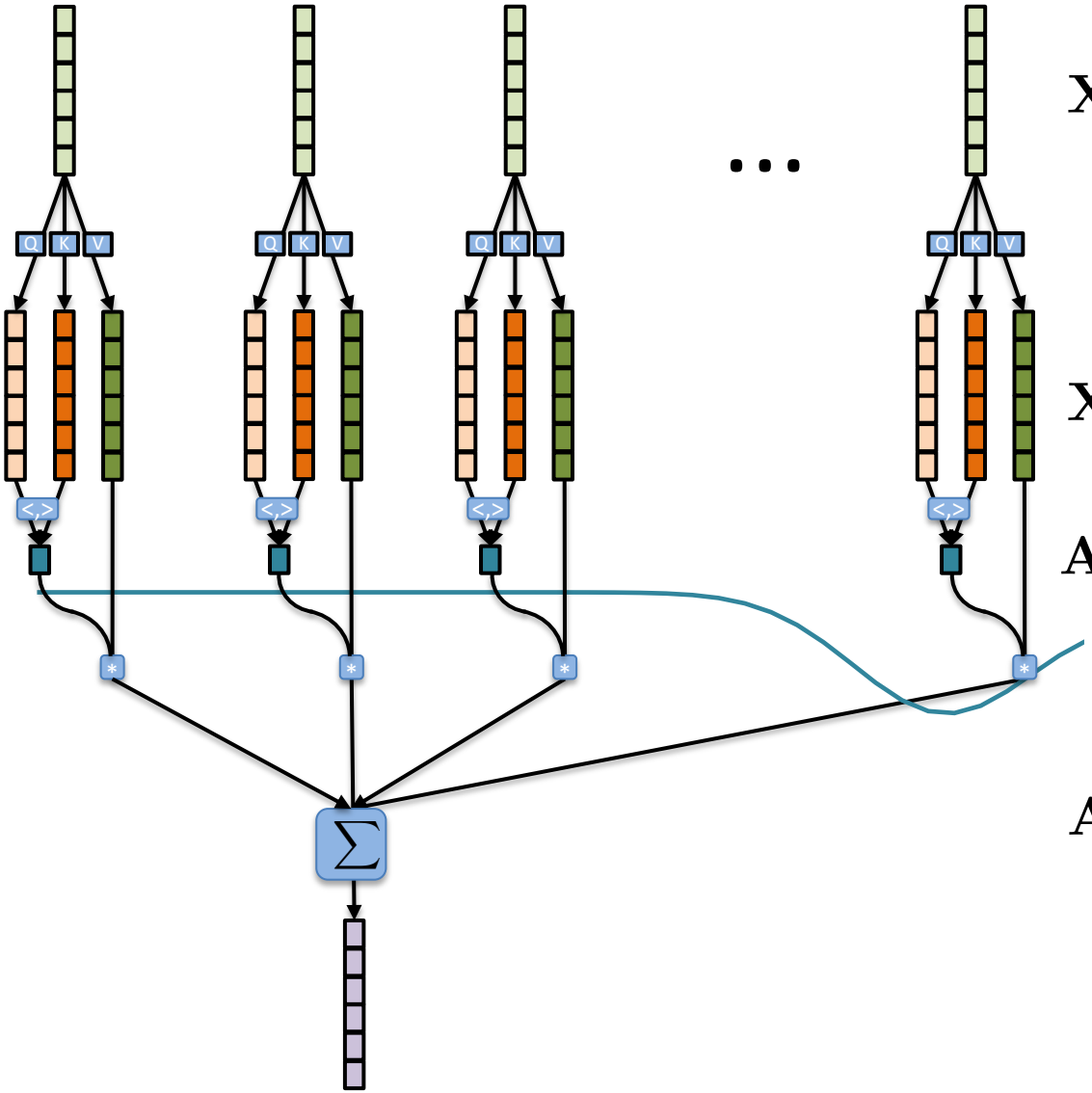
$\mathbf{A} = \sigma\left[\mathbf{X}\mathbf{W}_q(\mathbf{X}\mathbf{W}_k)^T\right]$

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

Transformers: attention is all you need



Transformers: attention is all you need



$\mathbf{X}$

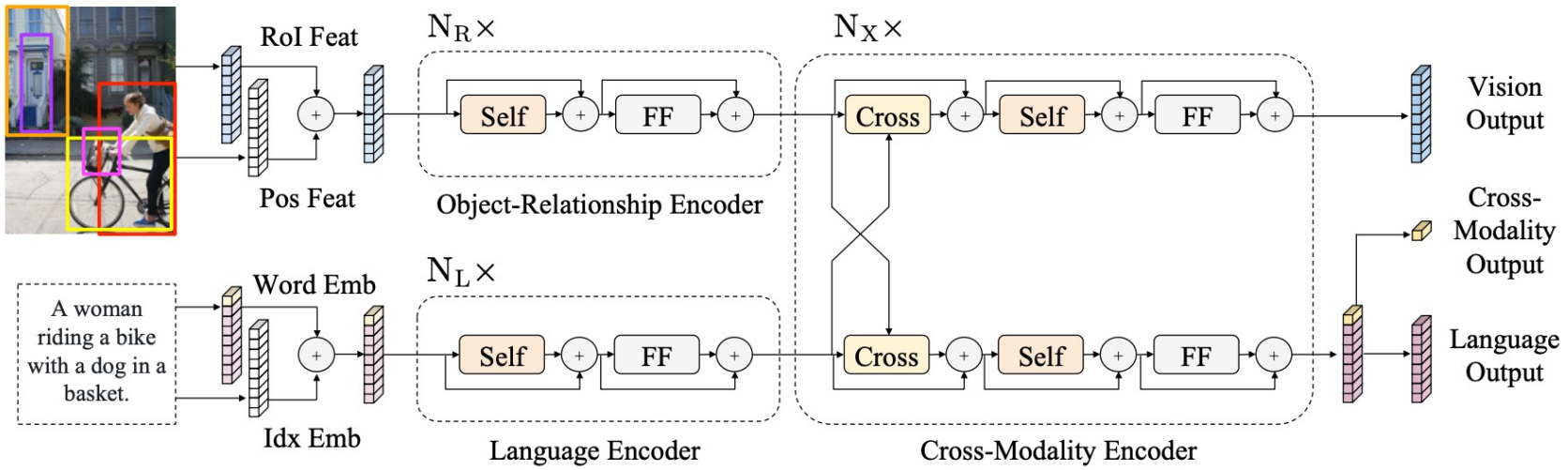
$\mathbf{X}\mathbf{W}_v$

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

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

LXMERT

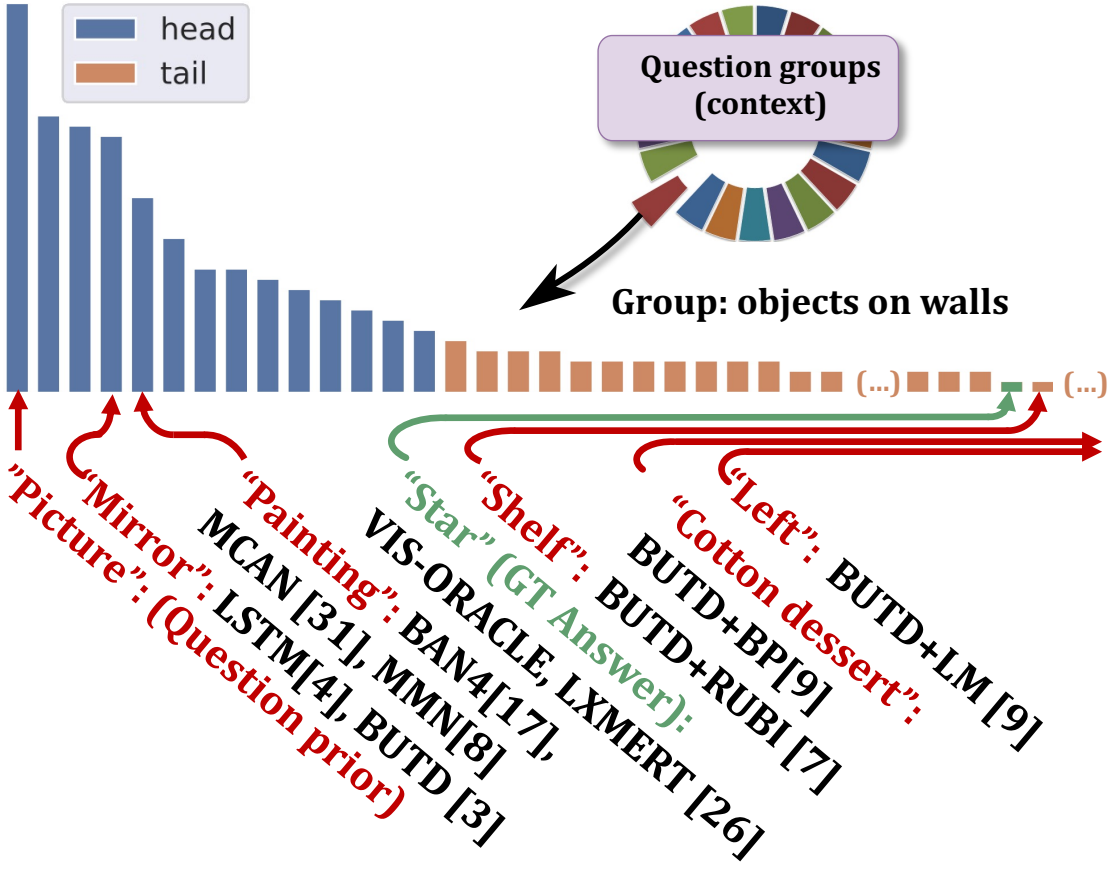
A vision and language encoder with self-attention and cross-attention.



[Tan et Bansal, EMNLP 2019]

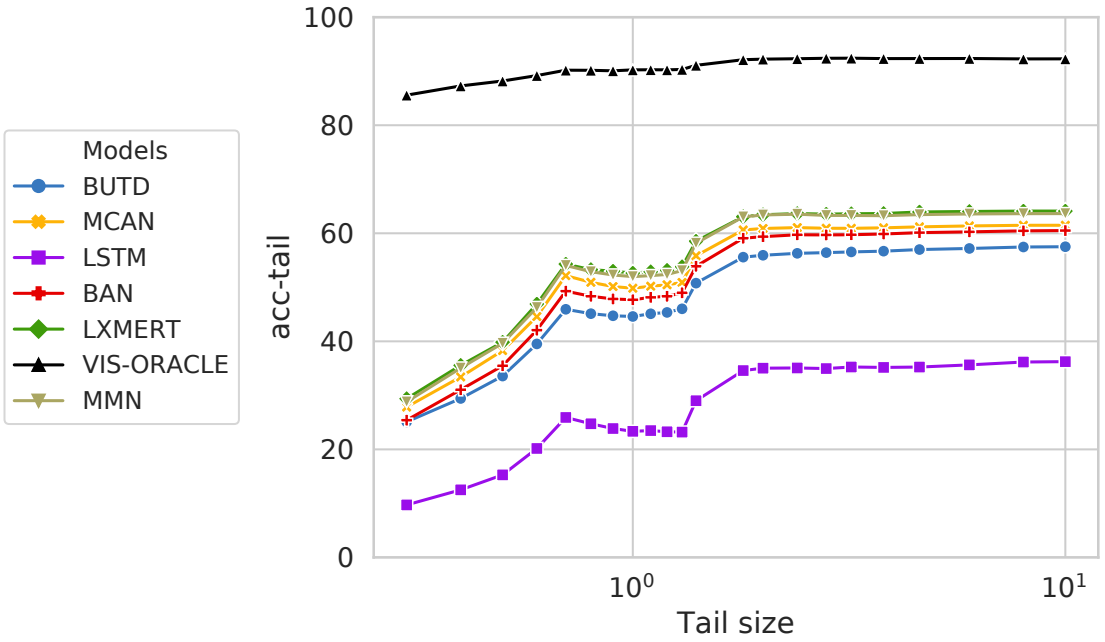
Roses Are Red, Violets Are Blue... but Should VQA Expect Them To?

"What is on the wall?"



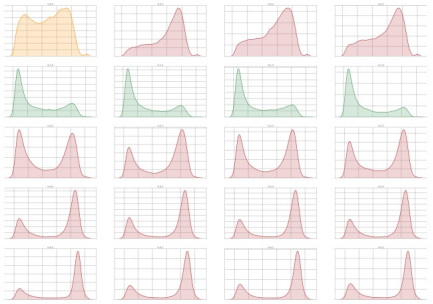
[Kervadec, Antipov, Baccouche, Wolf, CVPR 2021a]

Reasoning vs. bias exploitation

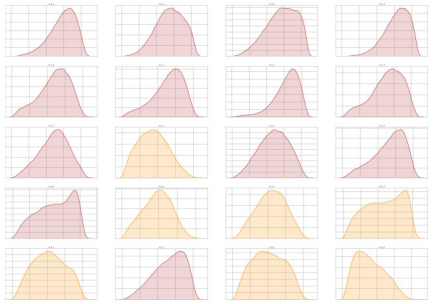


[Kervadec, Antipov, Baccouche, Wolf, CVPR 2021a]

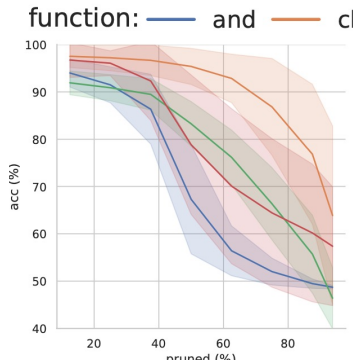
Analyzing reasoning patterns



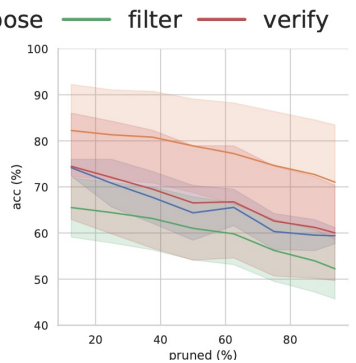
Oracle



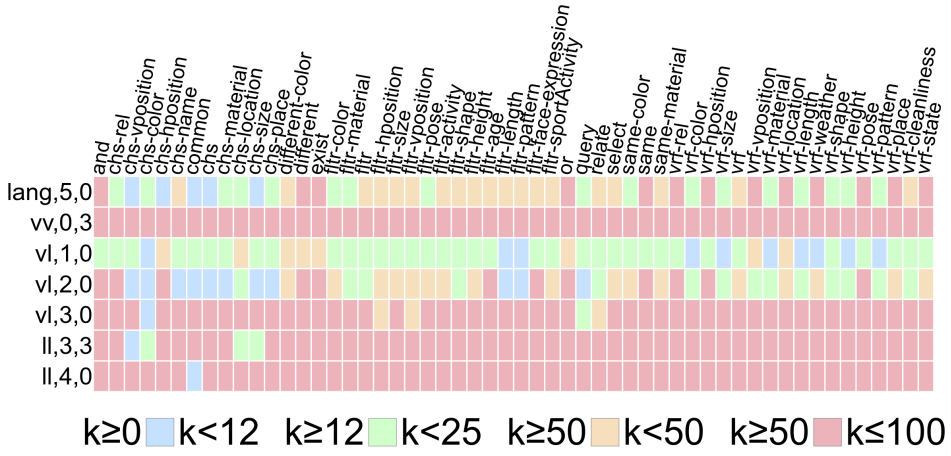
Noisy/deployable



Oracle



Noisy/deployable



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

How transferrable are Reasoning Patterns in VQA?

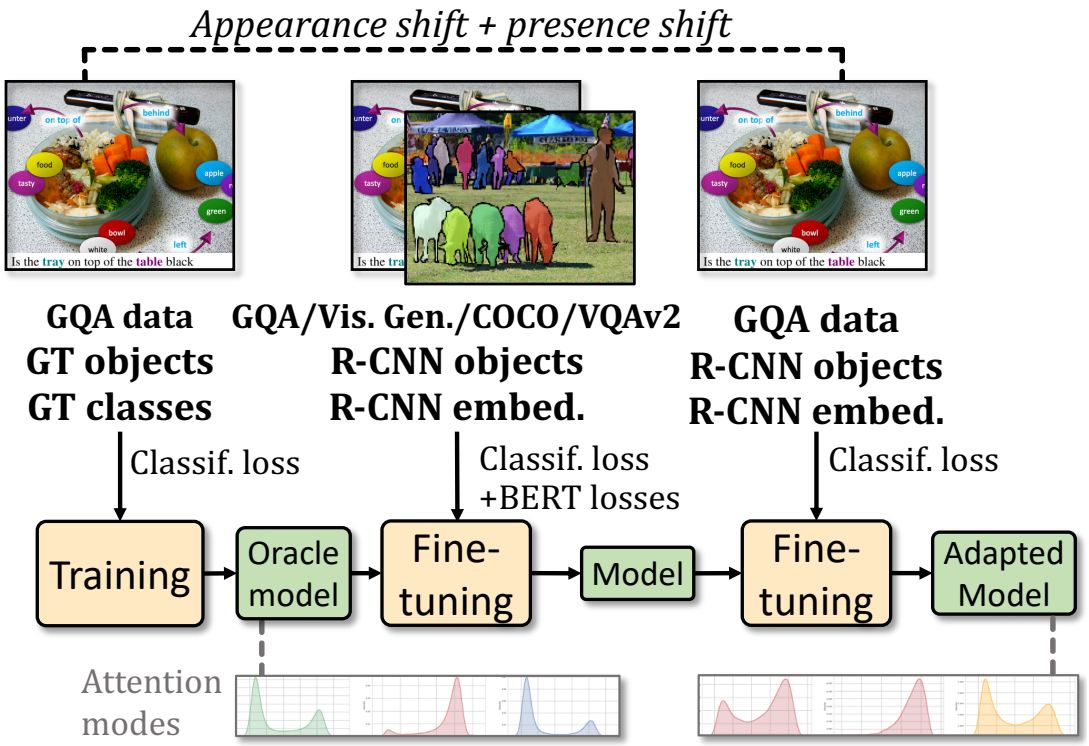
CVPR 2021

Corentin Kervadec Théo Jaunet Grigory Antipov
Moez Baccouche Romain Vuillemot Christian Wolf

LIRIS, INSA-Lyon, Ecole Centrale de Lyon, Orange

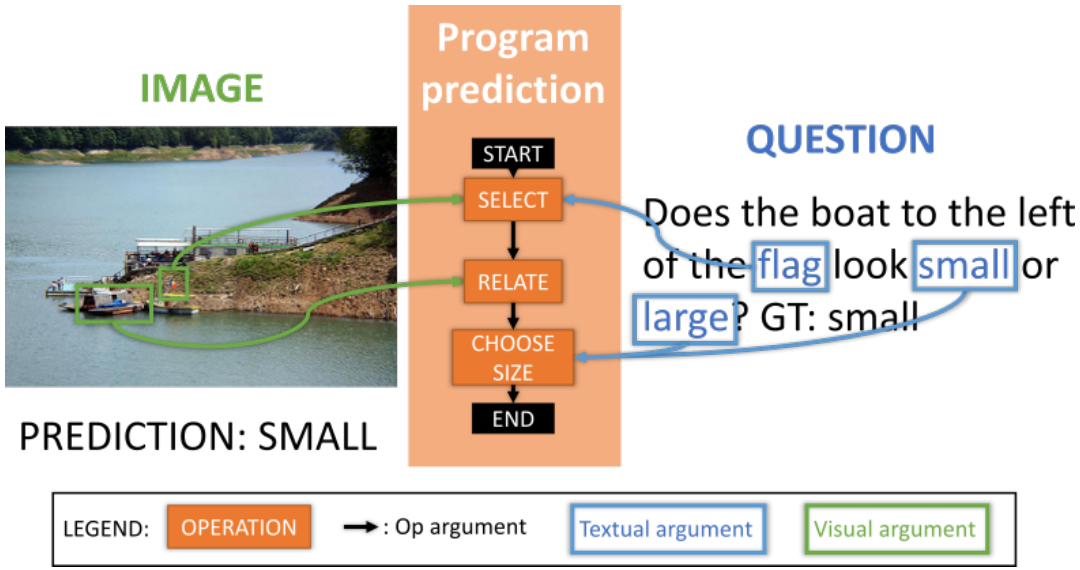
https://visqa.liris.cnrs.fr

Oracle transfer



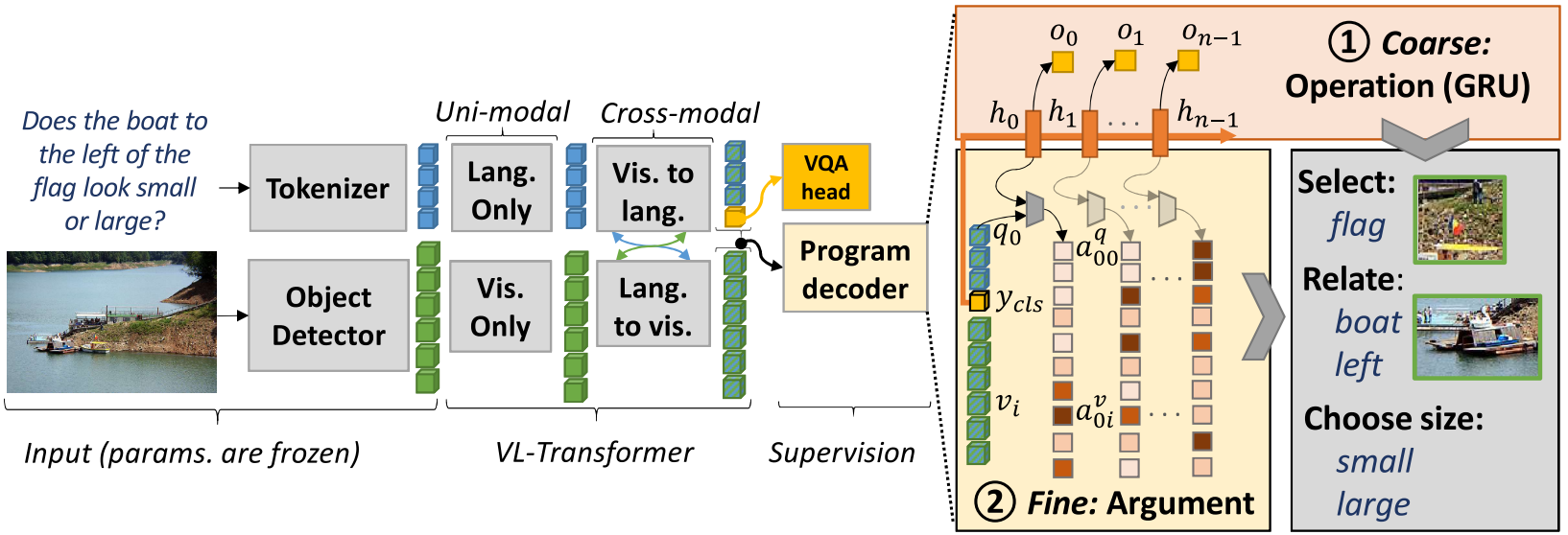
[Kervadec, Jaunet, Antipov, Baccouche, Vuillemot, Wolf, CVPR 2021b]

GT reasoning programs



[Kervadec*, Wolf*, Antipov, Baccouche, Nadri, Submitted to NeurIPS]

Program prediction



[Kervadec*, Wolf*, Antipov, Baccouche, Nadri, Submitted to NeurIPS]

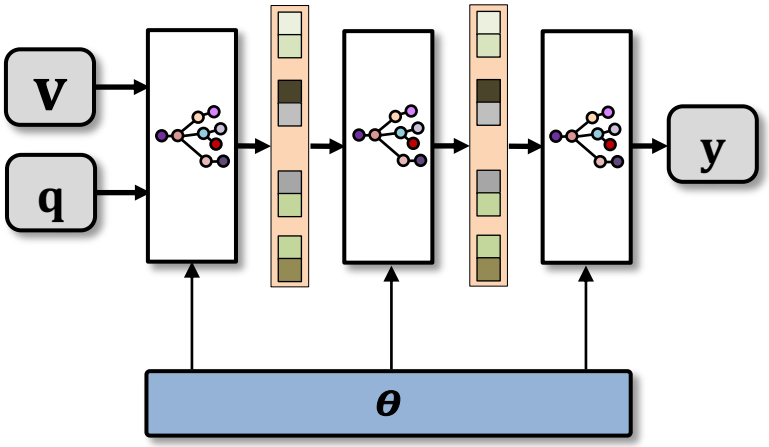
Oracle + program supervision

	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

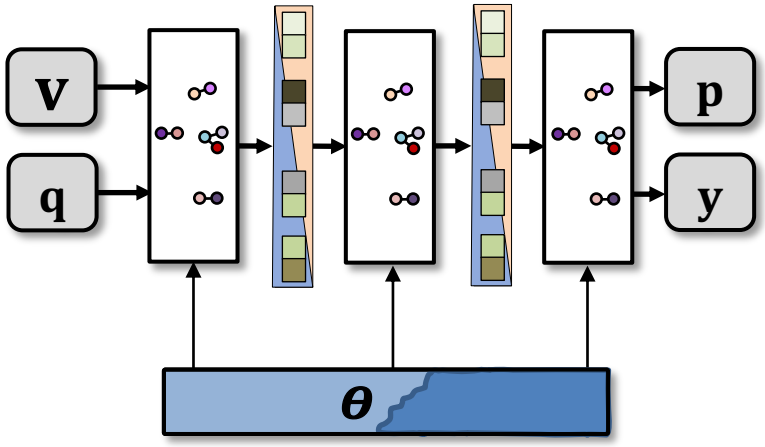
Table 1: Impact of program supervision on *Oracle transfer* [23] for vision-language transformers. LXMERT [36] pre-training is done on the GQA unbalanced training set. We report scores on GQA [17] (*test-dev* and *test-std*) and GQA-OOD (*test*). * binary and open scores are computed on the test-std;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

[Kervadec*, Wolf*, Antipov, Baccouche, Nadri, Submitted to NeurIPS]

Sample complexity



Classical training, CE loss on answers **y**



CE loss on **y** + program prediction **p**

Learned knowledge of the reasoning processes

Latent variables necessary for reasoning over multiple hops (used by reasoning processes)

Decomposition of the underlying (unknown) reasoning function

[Kervadec*, Wolf*, Antipov, Baccouche, Nadri, Submitted to NeurIPS]

Conclusion

- Mid- and long-term goal: create robots, which are
  - Situation aware
  - Capable of high-level reasoning
  - Trained from simulated and real data

- Challenges:
  - Find learning signals
  - Create inductive biases
  - Identify and collect data sources (world models?)
  - Avoid short cuts