Universiteit van Amsterdam

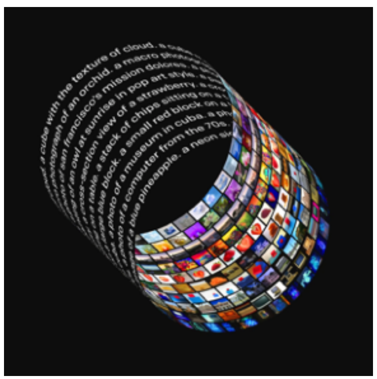
AMLAB Amsterdam Machine Learning Lab

Modular learning for improving AI assistants

Herke van Hoof

AI milestones

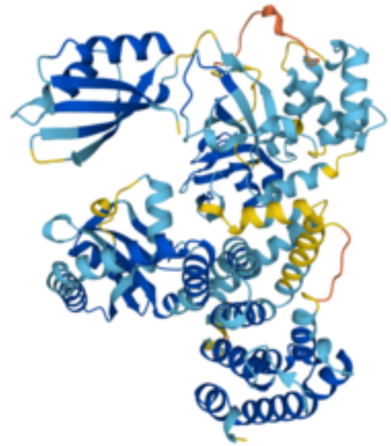
ChatGPT



openai.com / Dall-E

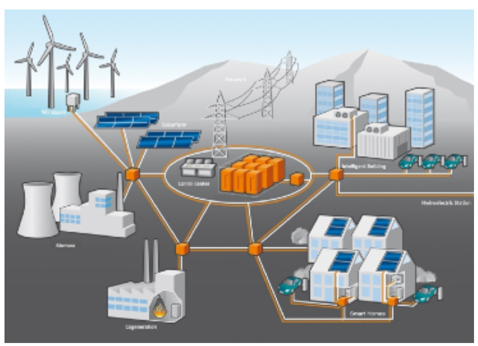
Challenges for AI assistants

Wikipedia / University of Toronto (CC BY-SA 4.0)

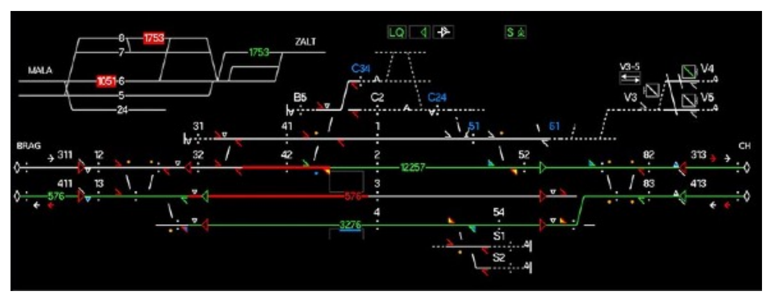


Wikipedia/DeepMind (CC BY 4.0)

Wikipedia / Luka Peternel (CC BY-SA 4.0)



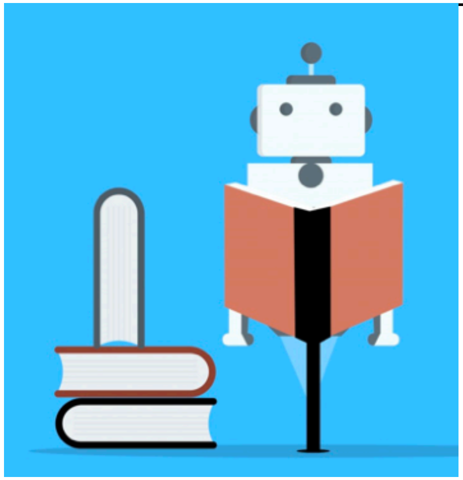
Wikipedia / Miloš Křivan (CC0)



Flickr/kecko (CC BY 2.0)

What makes these tasks hard?

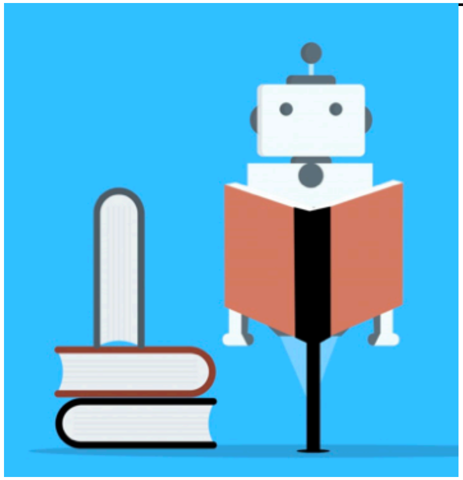
What makes these tasks hard?



pxhere.com /
mohamed hassan (CC0)

Data availability

What makes these tasks hard?



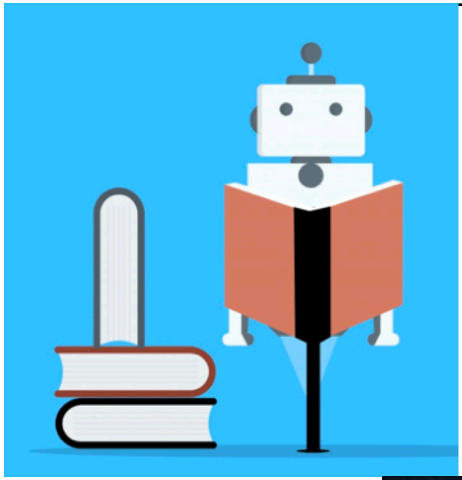
pxhere.com /
mohamed hassan (CC0)

Data availability

Instruction following

Wikipedia / Ocarina188 (CC BY-SA 4.0)

What makes these tasks hard?



pxhere.com /
mohamed hassan (CC0)

Data availability

Instruction following

Lack of explicit planning

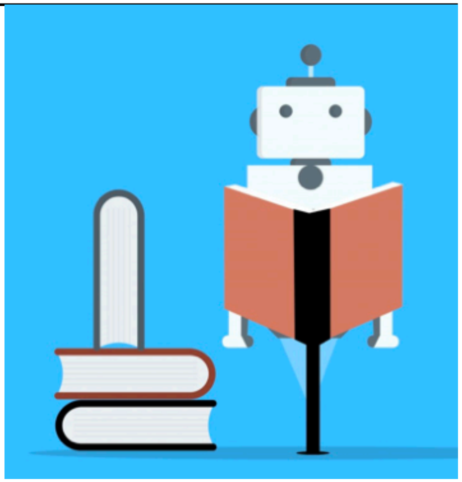
Wikipedia / Ocarina188 (CC BY-SA 4.0)

Flicker / Tengrain

What makes these tasks hard

Data availability

- Often small datasets for each specific task
- LLM / VLM pre-training targets text/visual domains and non-specialists tasks
- Current (deep) RL needs much more data



pxhere.com / mohamed hassan (CC0)

What makes these tasks hard

Instruction following

- Capable artificial agents should be able to reach different goals, under different preferences and constraints, each time
- Instructions usually more abstract than direct sensory-motor signals

Wikipedia / Ocarina188 (CC BY-SA 4.0)

LLMs/VLMs are strong at instruction following

(Deep) RL methods usually don't take complex instructions

What makes these tasks hard

Explicit reasoning or planning needed or desired

- Generalize in predictable ways, provide transparancy
- Use prior knowledge or constraints: ‘business logic’, fairness

LLMs notoriously bad in multi-step reasoning (e.g., Sudoku)

Some deep RL methods do look ahead (e.g. AlphaGo)

Flicker / Tengrain

What makes these tasks hard

Requirements for capable AI agents

Requirement	LLMs / VLMs	Deep RL
Handle modest 'niche' datasets	✗	✗
Explicit reasoning or planning	✗	✔
Instruction following	✔	✗

A modular point of view

A *modular* point of view can help address these issues

Modular: solutions are (possibly recursively) composed of smaller units that can be shared within and/or between tasks

Example:

- Modules form a hierarchy with one high-level policy that chooses to activate one of several low-level policies

A modular point of view

Modular strategies can help address the main challenges:

- Modular strategies can generalize in a predictable and structured manner, efficiently learning from relatively *small amounts of data*.
- Modular strategies can help bridge between low-level and high-level behavior, aiding *instruction following, reasoning* and *planning.*

Our work on modular learning so far

	Data efficiency	*Learning & Planning*	*Instruction following*
Daniel et al., 2016	✓		
Smith et al., 2018	✓		
Woehlke et al., 2021	✓	✓	½
Kool et al., 2022		✓	
Höpner et al., 2022	✓		
Woehlke et al., 2022	✓	✓	½
Woehlke et al., 2023		✓	½
Kuric et al., 2023	✓		
Kuric et al., 2024	✓	✓	✓
Höpner et al., 2025	✓	½	✓
Macfarlane et al., 2025			✓

Our work on modular learning so far

	Data efficiency	*Learning & Planning*	*Instruction following*
Daniel et al., 2016	✓		
Smith et al., 2018	✓		
Woehlke et al., 2021	✓	✓	½
Kool et al., 2022		✓	
Höpner et al., 2022	✓		
Woehlke et al., 2022	✓	✓	½
Woehlke et al., 2023		✓	½
Kuric et al., 2023	✓		
Kuric et al., 2024	✓	✓	✓
Höpner et al., 2025	✓	½	✓
Macfarlane et al., 2025			✓

Our work on modular learning so far

	Data efficiency	*Learning & Planning*	*Instruction following*
Daniel et al., 2016			
Smith et al., 2018			
Woehlke et al., 2021			
Kool et al., 2022			
Höpner et al., 2022	✓		
Woehlke et al., 2022	✓	✓	½
Woehlke et al., 2023		✓	½
Kuric et al., 2023	✓		
Kuric et al., 2024	✓	✓	✓
Höpner et al., 2025	✓	½	✓
Macfarlane et al., 2025			✓

Our work on this topic focuses on the underlying principles, and not (yet) on scaling up to complex applications

Modular learning for improving AI assistants

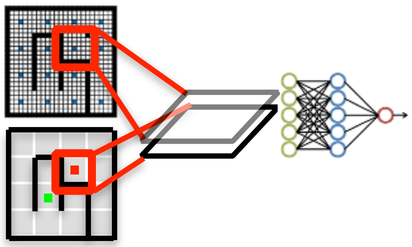
Woehlke et al., 2022	✓	✓	½
Kuric et al., 2024	✓	✓	✓
Macfarlane et al., 2025			✓

Modular learning for improving AI assistants

Woehlke et al., 2022	✓	✓	½
Kuric et al., 2024	✓	✓	✓
Macfarlane et al., 2025			✓

Modular learning for improving AI assistants

Woehlke et al., 2022	✓	✓	½



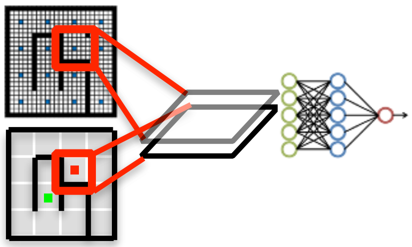
- *Learning and planning* have different advantages
- How can learning and planning modules interact?
- Does this allow tackling large and complex domains?

Kuric et al., 2024	✓	✓	✓

Macfarlane et al., 2025			✓

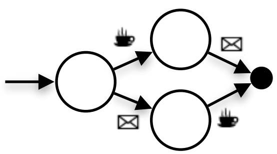
Modular learning for improving AI assistants

Woehlke et al., 2022	✓	✓	½



- *Learning and planning* have different advantages
- How can learning and planning modules interact?
- Does this allow tackling large and complex domains?

Kuric et al., 2024	✓	✓	✓

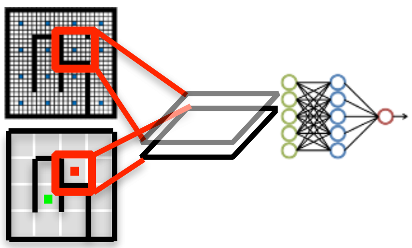


- How to learn for complex & never before seen *instructions*?
- Pre-learn behavior modules for different context
- Use on-the-fly *planning* to combine these modules

Macfarlane et al., 2025			✓

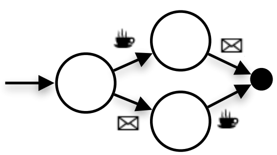
Modular learning for improving AI assistants

Woehlke et al., 2022	✓	✓	½



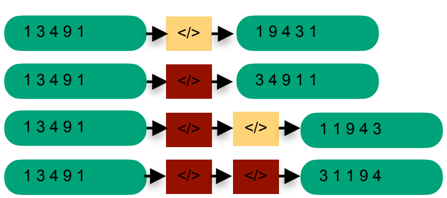
- *Learning and planning* have different advantages
- How can learning and planning modules interact?
- Does this allow tackling large and complex domains?

Kuric et al., 2024	✓	✓	✓



- How to learn for complex & never before seen *instructions*?
- Pre-learn behavior modules for different context
- Use on-the-fly *planning* to combine these modules

Macfarlane et al., 2025			✓



- Learn symbolic ‘language’ to describe mappings
- Allows *compositional generalization* in learned model
- Test-time optimization using differentiable decoder

Planning and learning modules

AI assistants should be capable of quick adaptation to changes in their environment

Pure learning approaches would need large amount of data / experience to react to new changes

Can planning help?

Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

Planning and learning modules

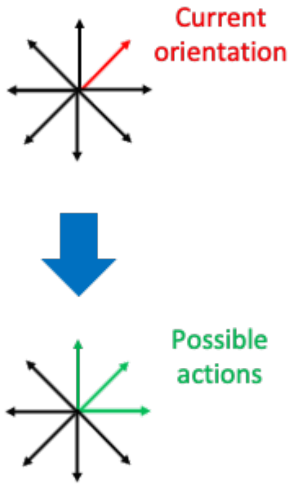
While planning does not *use* data, it requires time, especially for large problems.

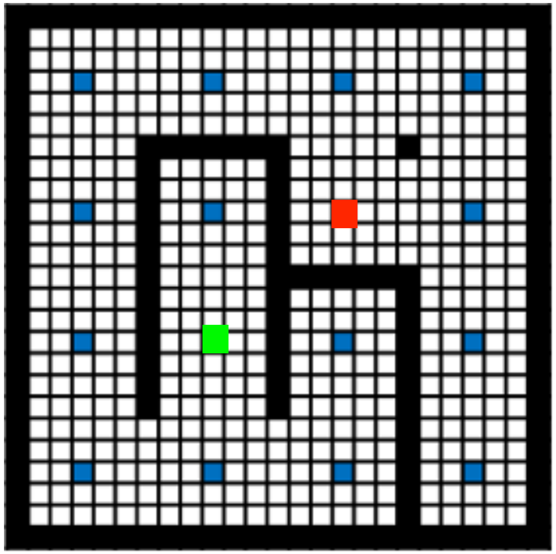
Can we decompose a family of large decision making problem into two modules?

- A local, learned module is learned and can decide fast. A local focus makes the learning problem much easier.
- A global planning module can adapt to changes in environment. Local details can be abstracted

Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

Motivating example: grid world navigation

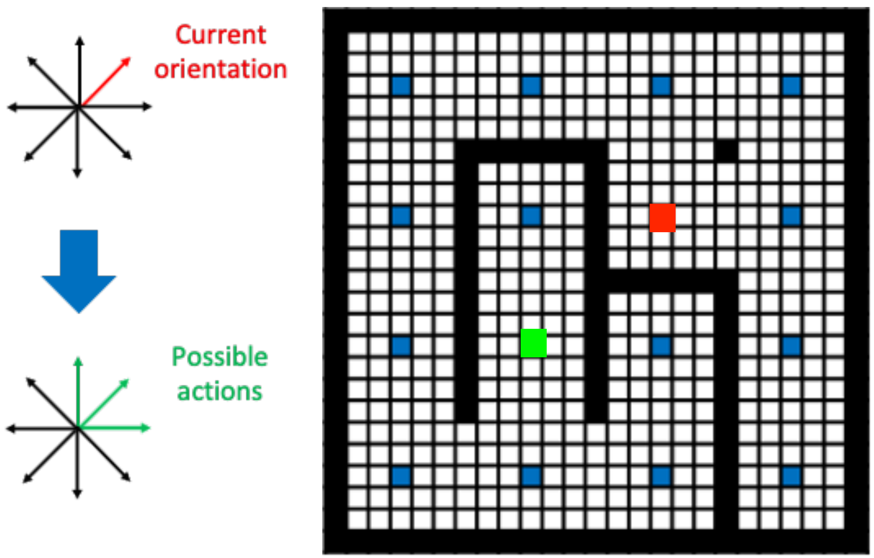




400 positions x 8 orientations
= 3200 states

Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

Motivating example: grid world navigation

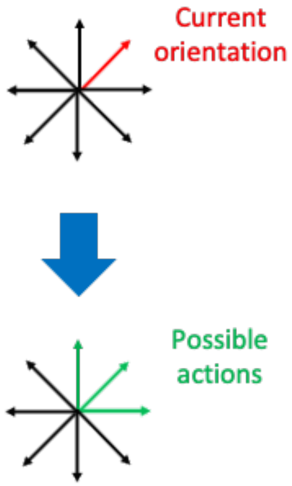


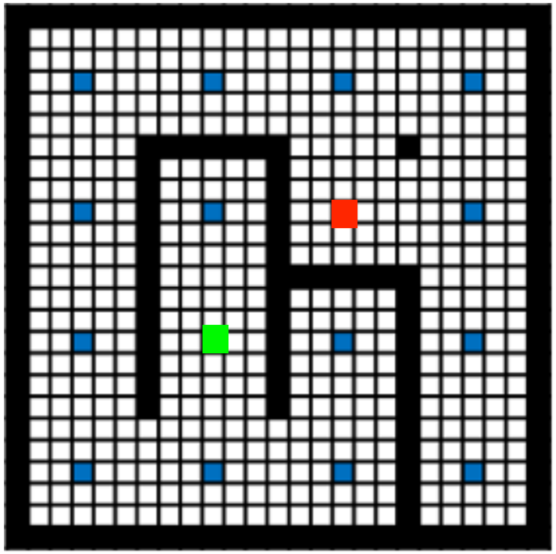
400 positions x 8 orientations
= 3200 states

Global planning in abstracted problem

Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

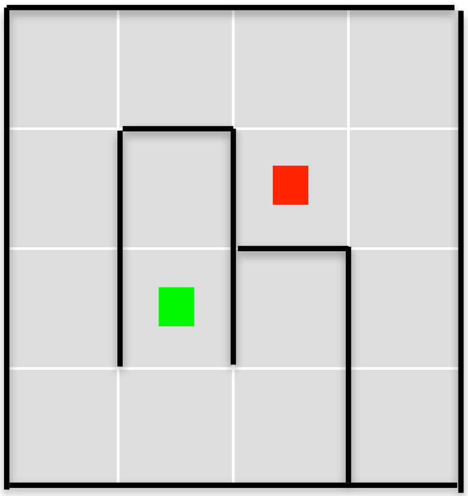
Motivating example: grid world navigation





400 positions x 8 orientations
= 3200 states

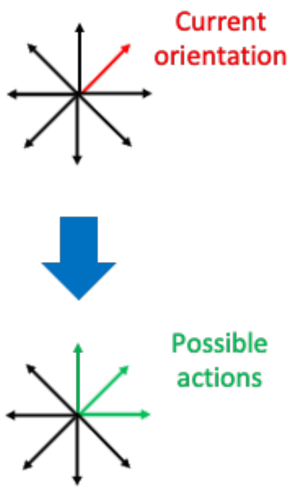
Global planning in abstracted problem

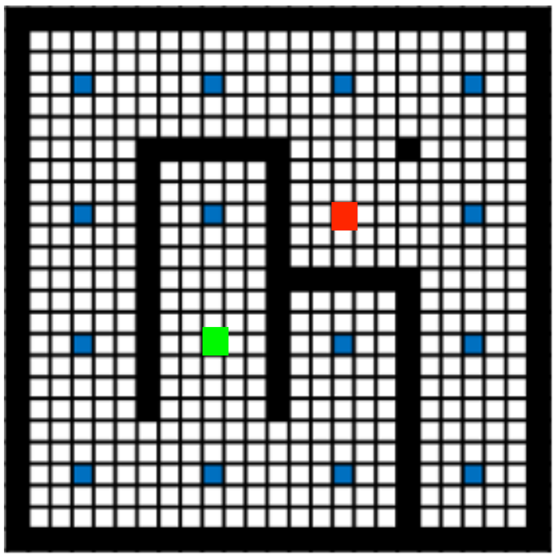


16 positions,
no orientation

Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

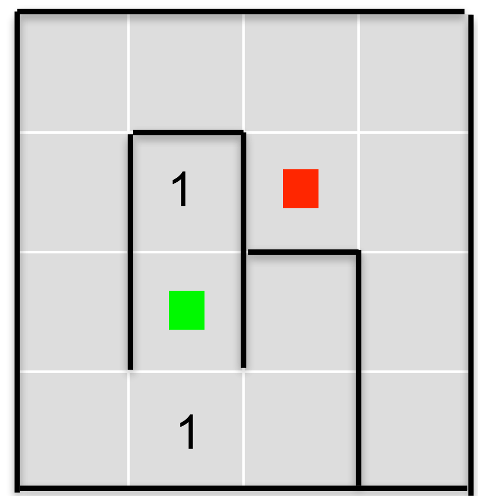
Motivating example: grid world navigation





400 positions x 8 orientations
= 3200 states

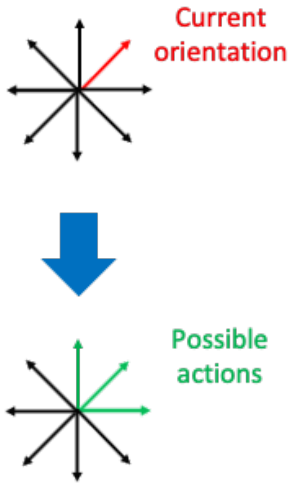
Global planning in abstracted problem

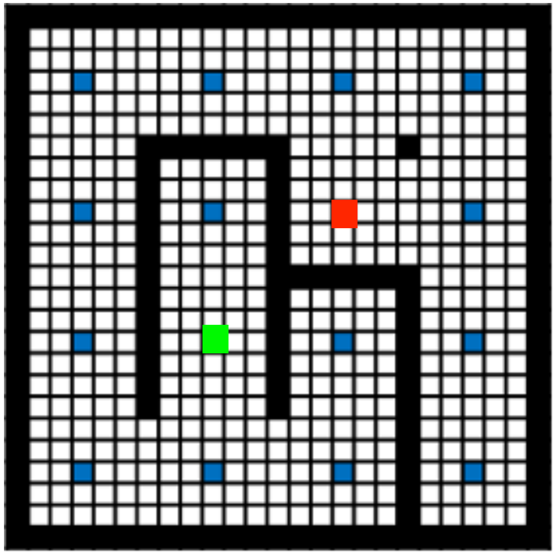


16 positions,
no orientation

Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

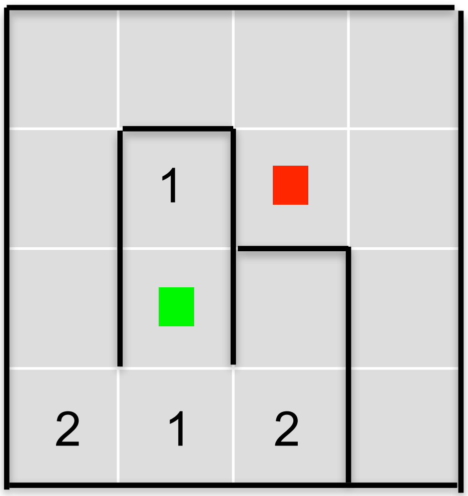
Motivating example: grid world navigation





400 positions x 8 orientations
= 3200 states

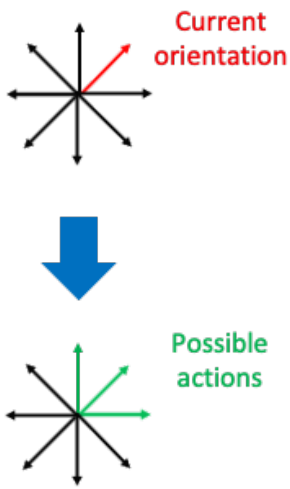
Global planning in abstracted problem

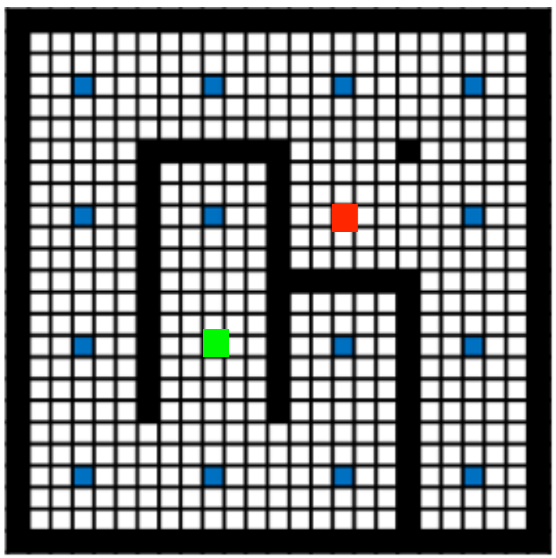


16 positions,
no orientation

Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

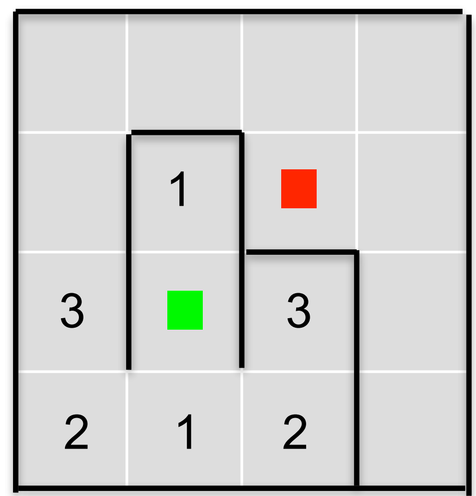
Motivating example: grid world navigation





400 positions x 8 orientations
= 3200 states

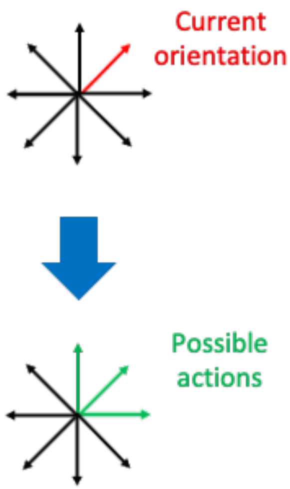
Global planning in abstracted problem

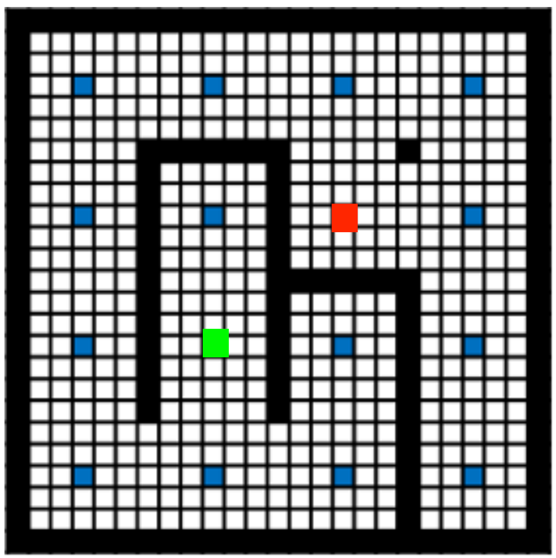


16 positions,
no orientation

Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

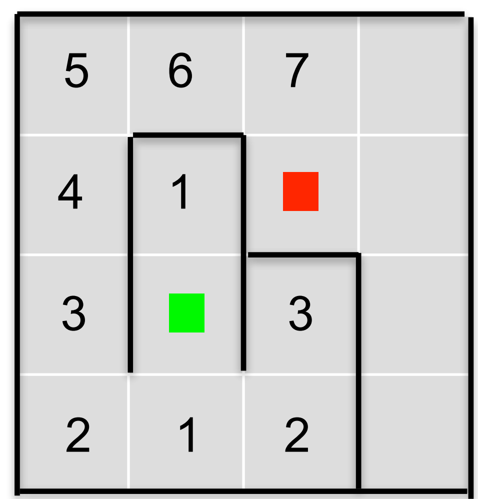
Motivating example: grid world navigation





400 positions x 8 orientations = 3200 states

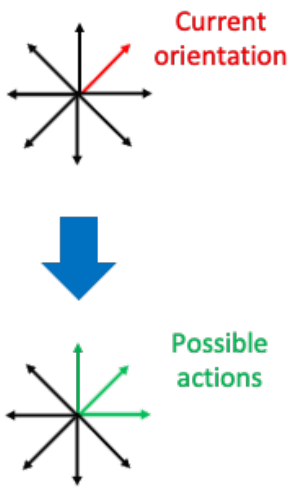
Global planning in abstracted problem

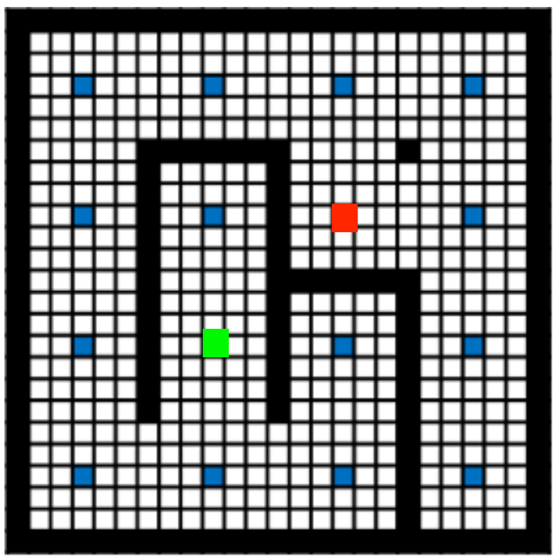


16 positions, no orientation

Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

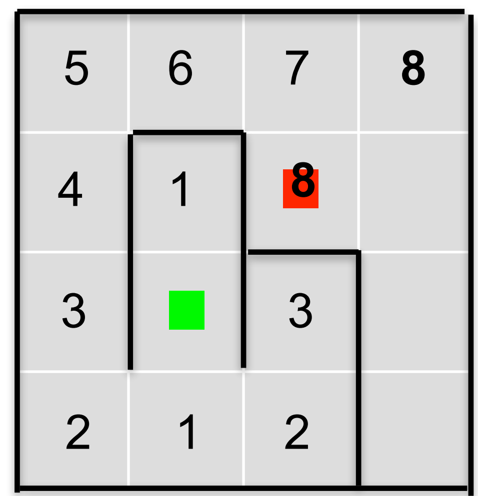
Motivating example: grid world navigation





400 positions x 8 orientations
= 3200 states

Global planning in abstracted problem



16 positions,
no orientation

Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

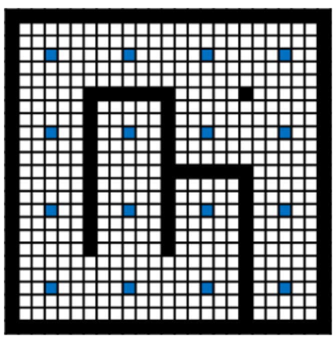
Motivating example: grid world navigation

Local decision-making function

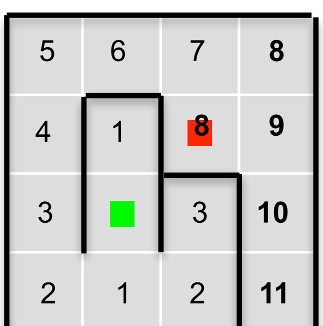
Orientation (repeated)

NE	***NE***	***NE***
NE	***NE***	***NE***
NE	***NE***	***NE***

Map



Coarse values



Motivating example: grid world navigation

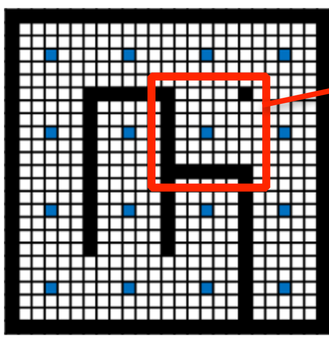
Local decision-making function

Orientation (repeated)

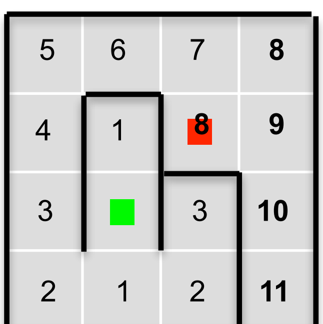
NE	*NE*	*NE*
NE	*NE*	*NE*
NE	*NE*	*NE*

Together describe environment locally
Generalizes to new maps or goals

Map



Coarse values



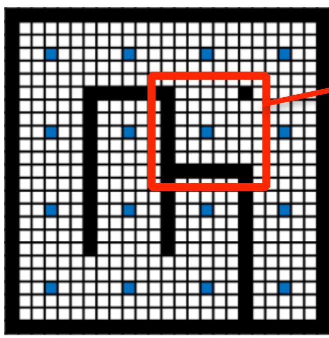
Motivating example: grid world navigation

Local decision-making function

Orientation (repeated)

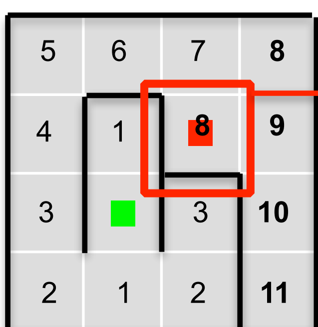
NE	*NE*	*NE*
NE	*NE*	*NE*
NE	*NE*	*NE*

Map



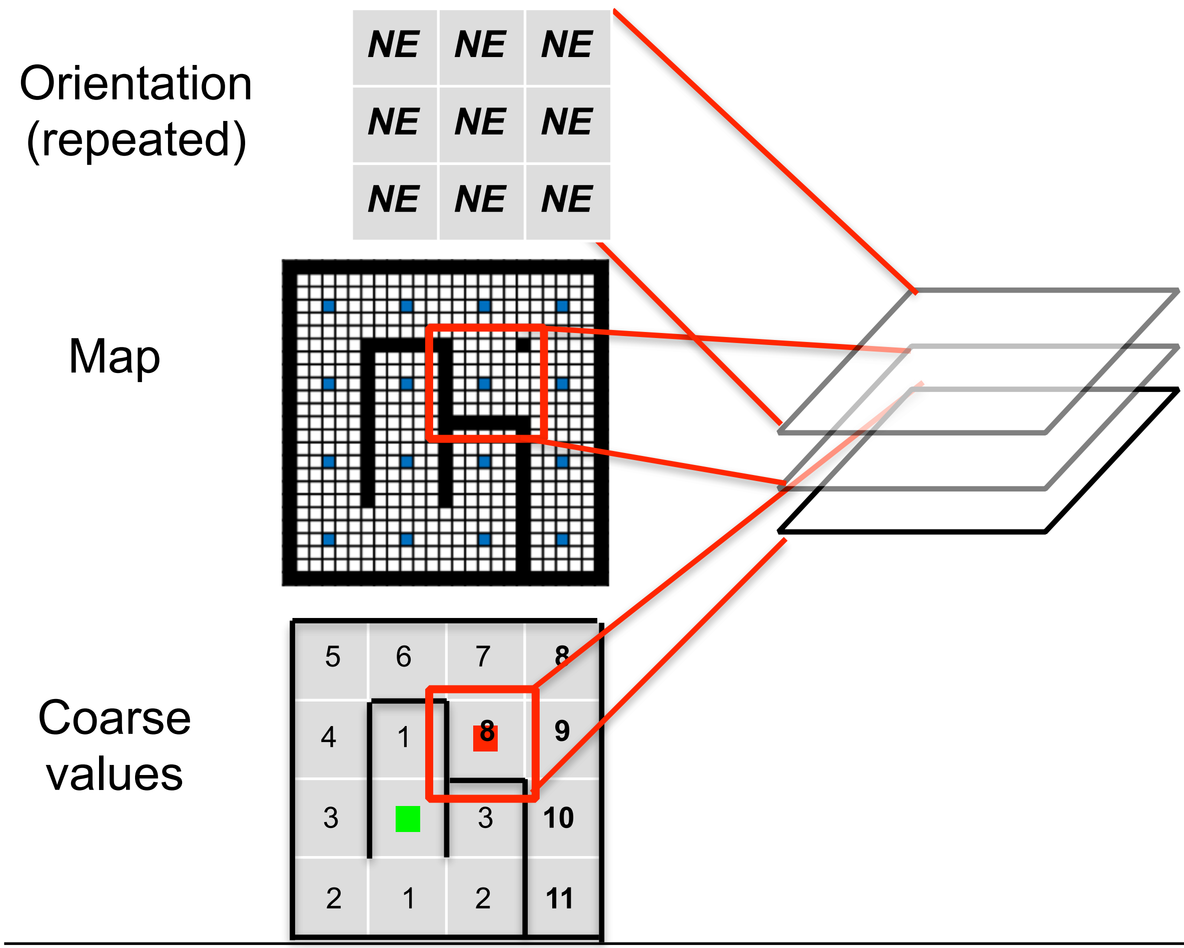
Together describe environment locally
Generalizes to new maps or goals

Coarse values

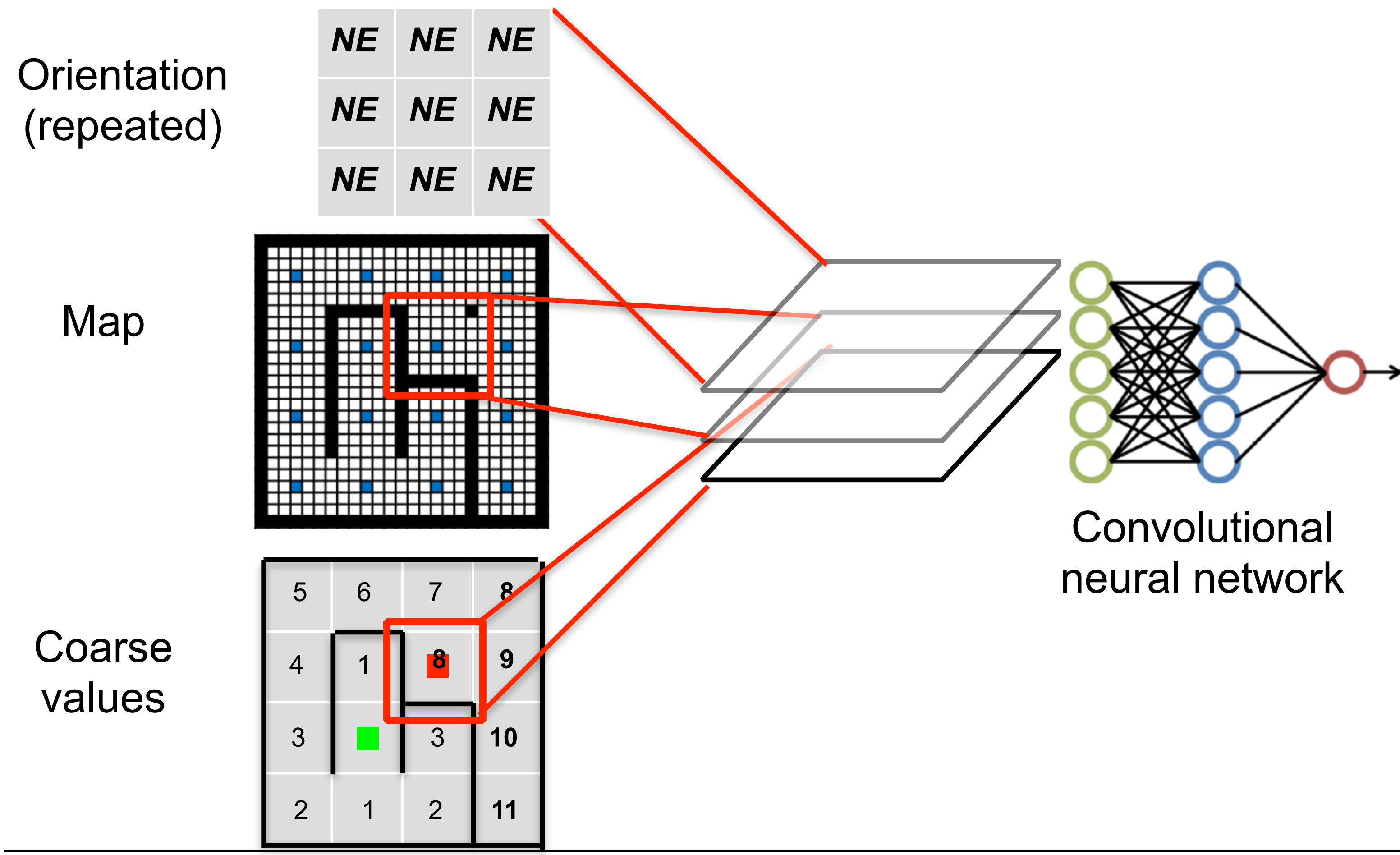


Coarse values describe global context
Can quickly compute for new environment

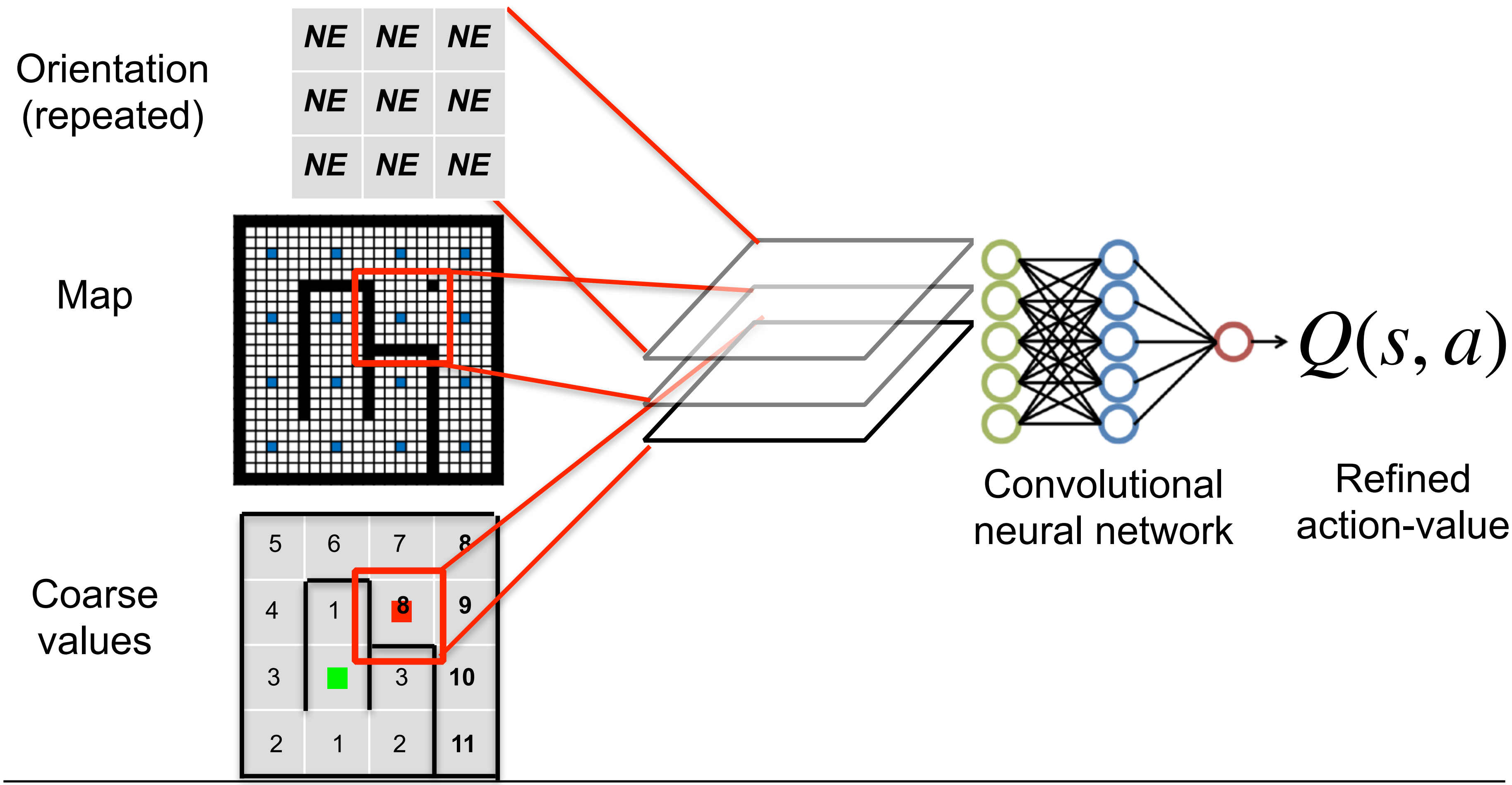
Motivating example: grid world navigation



Motivating example: grid world navigation



Motivating example: grid world navigation



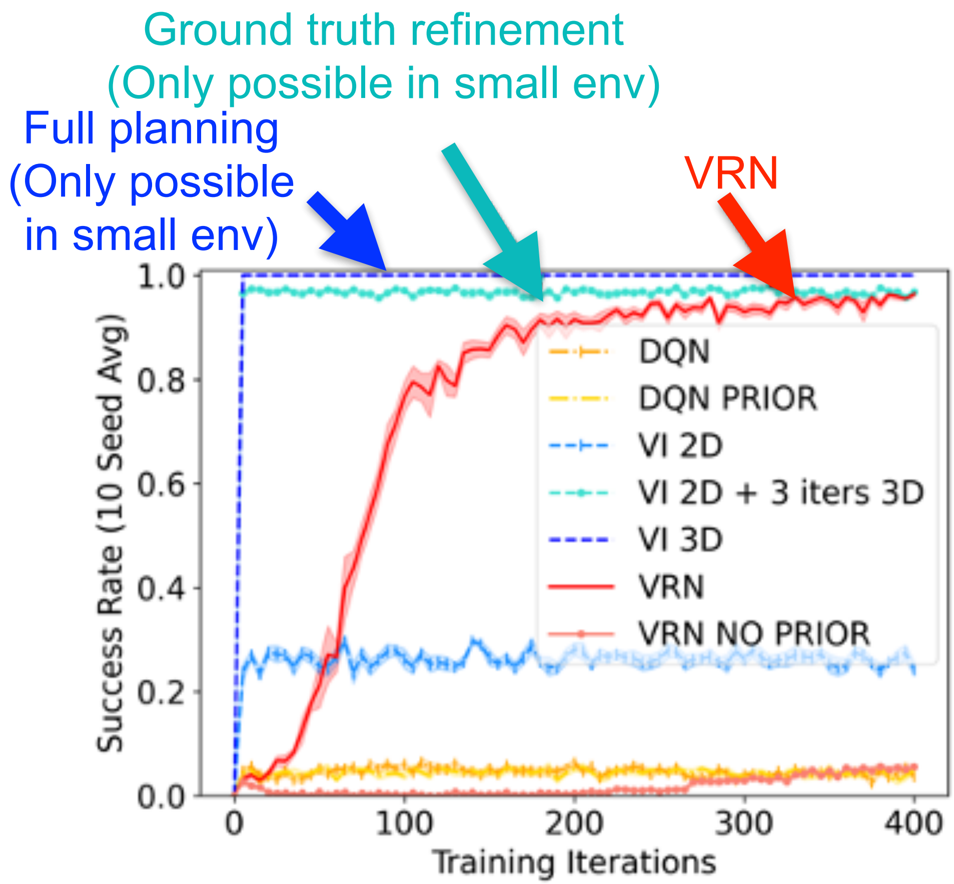
Grid world navigation result

Best performance with full planning (but most costly)

Local refinement close in performance (lower cost)

VRN can learn this refinement step

Just planning in 2D or just learning not sufficient



Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

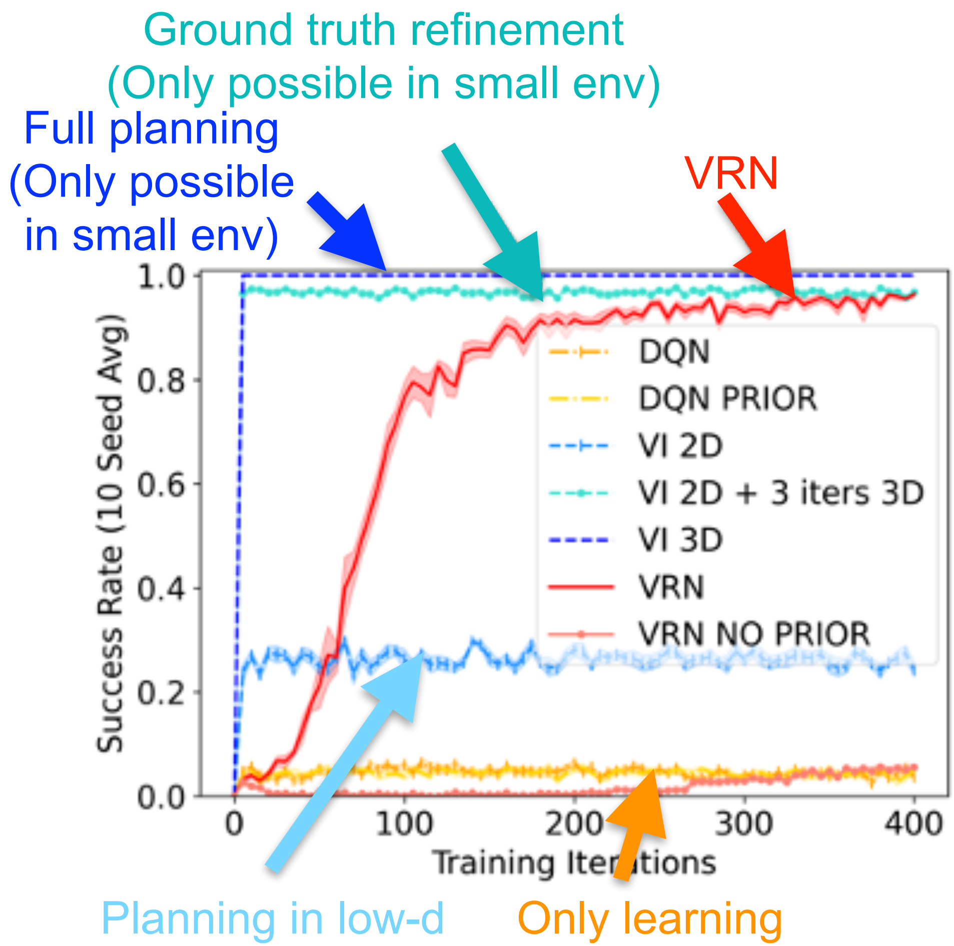
Grid world navigation result

Best performance with full planning (but most costly)

Local refinement close in performance (lower cost)

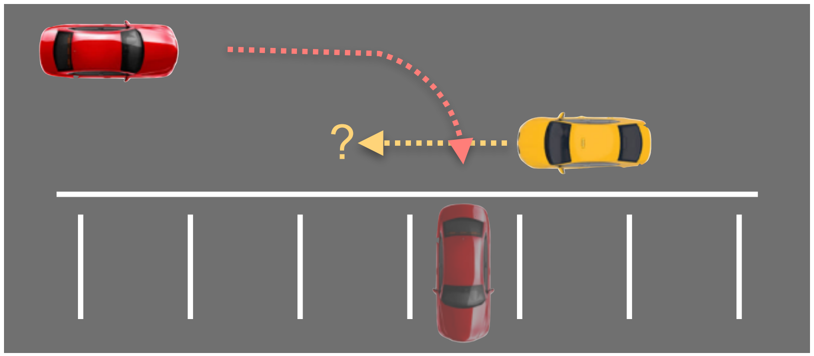
VRN can learn this refinement step

Just planning in 2D or just learning not sufficient



Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

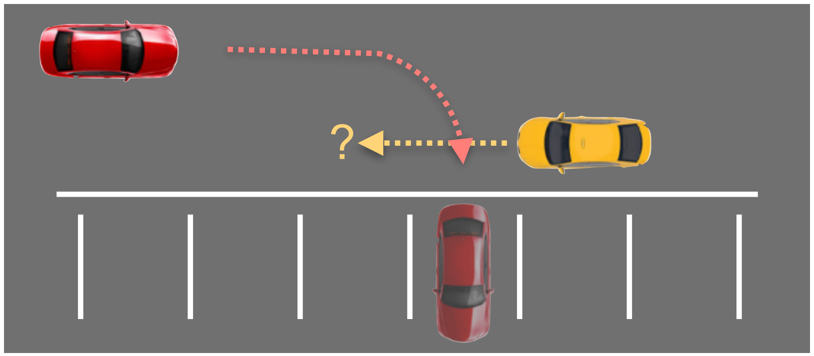
Parking task



Dynamics of 2nd vehicle unknown - exact planning not possible

Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

Parking task

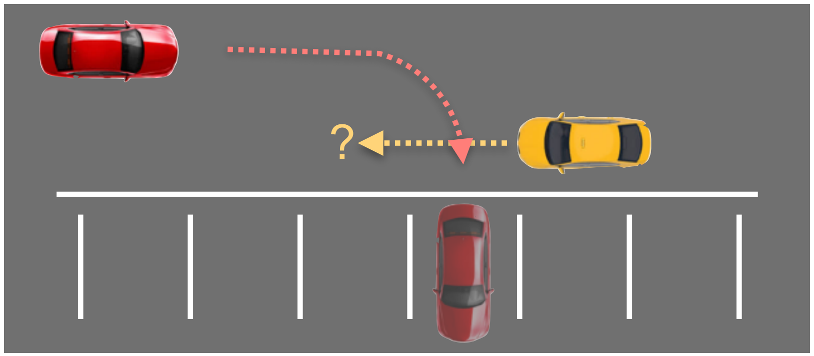


Dynamics of 2nd vehicle unknown - exact planning not possible

VRN achieves a high succes rate without costly re-planning

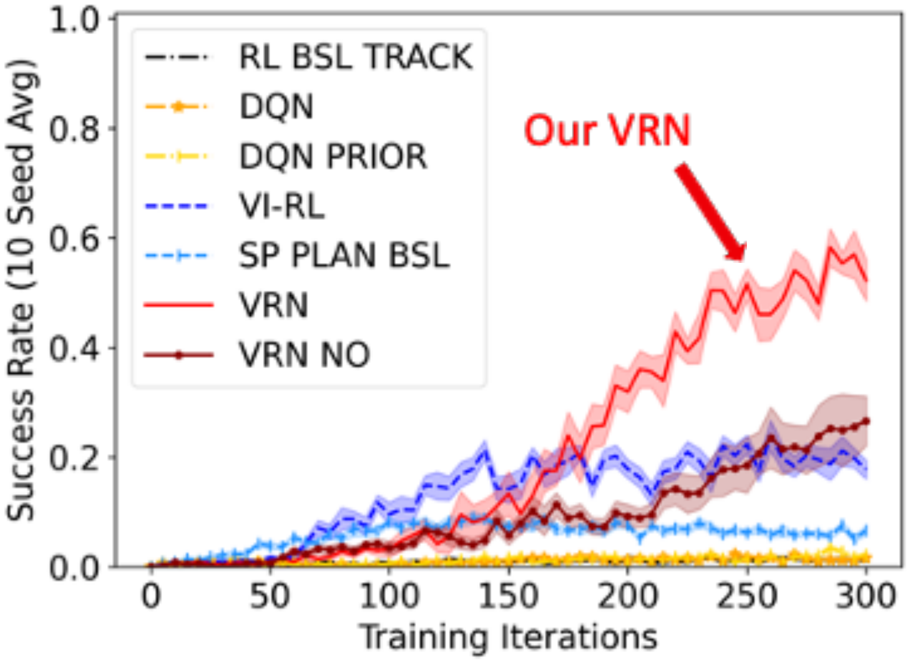
Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

Parking task



Dynamics of 2nd vehicle unknown - exact planning not possible

VRN achieves a high succes rate without costly re-planning



Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

Takeaways

Decomposing a problem into approximately independent subtasks helps find solutions efficiently

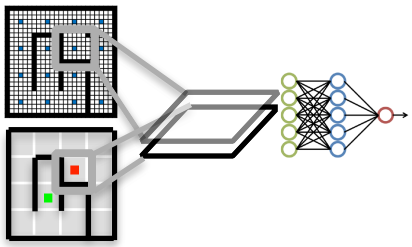
Leverages strength of learning and planning methods:

- Learning can handle high-d observations and complex dynamics
- Planning can generalize easily to different lay-outs

Wöhlke, J., Schmitt, F., & van Hoof, H. (2022). Value Refinement Network (VRN). In *IJCAI*.

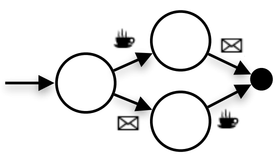
Our work on modular learning so far

Woehlke et al., 2022	✓	✓	½



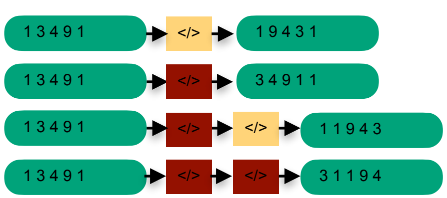
- *Learning and planning* have different advantages
- How can learning and planning modules interact?
- Does this allow tackling large and complex domains?

Kuric et al., 2024	✓	✓	✓



- How to learn for complex & never before seen *instructions*?
- Pre-learn behavior modules for different context
- Use on-the-fly *planning* to combine these modules

Macfarlane et al., 2025			✓



- Learn symbolic ‘language’ to describe mappings
- Allows *compositional generalization* in learned model
- Test-time optimization using differentiable decoder

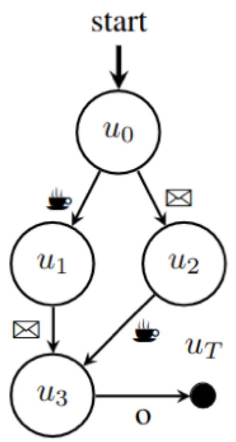
Modular instruction following

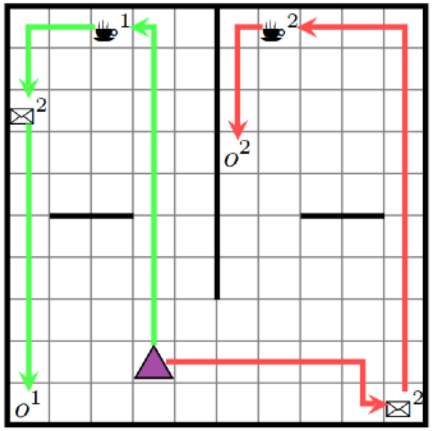
- RL agents often learn single task, possibly goal conditioned
- Capable AI assistants should be able to handle more complex instructions

Kuric, D., Infante, G., Gómez, V., Jonsson, A. & van Hoof, H. (2024). Planning with a learned policy basis to optimally solve complex tasks. In *ICAPS*.

Modular instruction following

- RL agents often learn single task, possibly goal conditioned
- Capable AI assistants should be able to handle more complex instructions
- Here: learn tasks described by Finite State Automaton

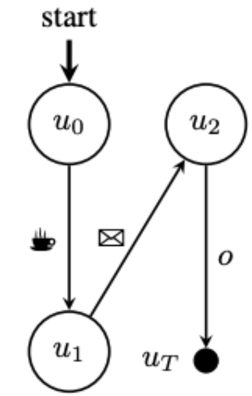


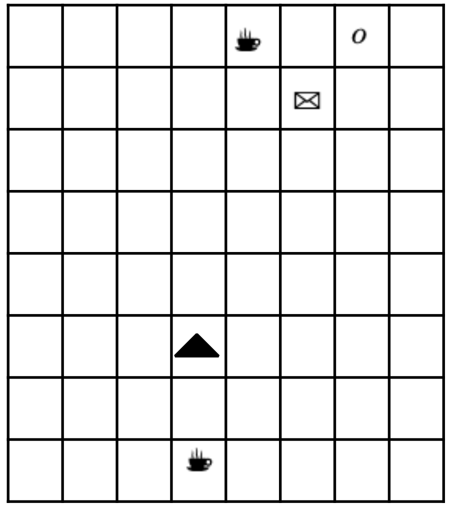


Kuric, D., Infante, G., Gómez, V., Jonsson, A. & van Hoof, H. (2024). Planning with a learned policy basis to optimally solve complex tasks. In *ICAPS*.

Modular instruction following

- Instructions described by FSA are *composed* of smaller tasks, but these tasks are *not independent*
- Assumption:
  - Layout fixed, but unknown
  - Instructions variable, given to the agent





Modular instruction following

Step 1: Learn a policy basis

Step 2: Plan with learned basis

Modular instruction following

Step 1: Learn a policy basis

- Treat long-term desirability of each labeled *exit state* ξ (☕, ✉) as a vector variable $\mathbf{w}$
- Learn *coverage set* Π containing optimal policy for any $\mathbf{w}$
- Learn the probability ψ^{π} of reaching each labeled exit state ξ for each $\pi \in \Pi$

Step 2: Plan with learned basis

Modular instruction following

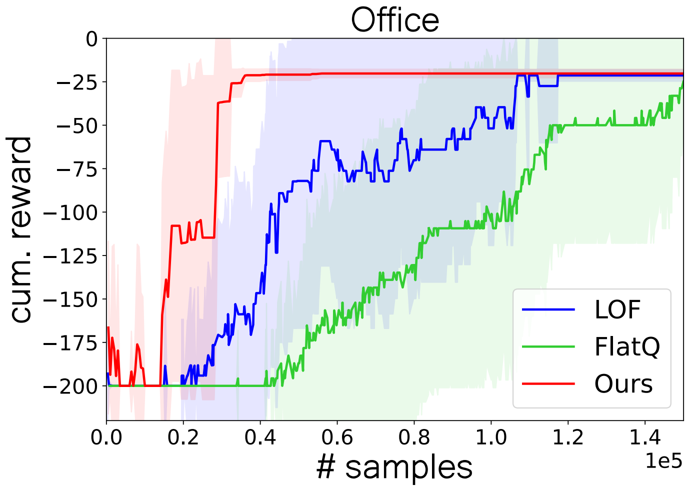
Step 1: Learn a policy basis

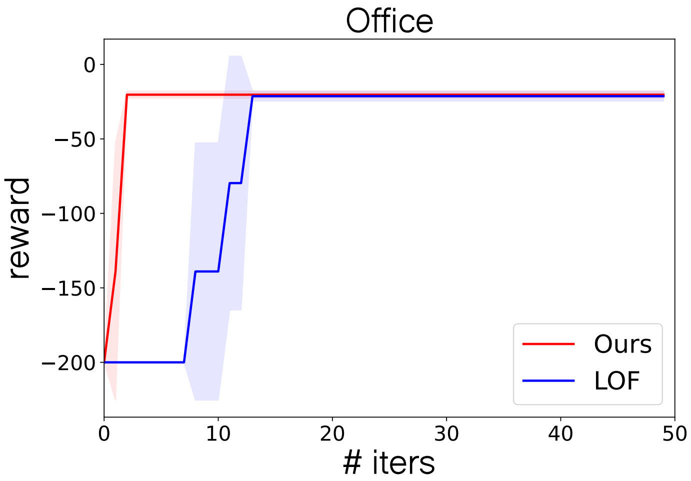
- Treat long-term desirability of each labeled *exit state* ξ (☕, ✉) as a vector variable $\mathbf{w}$
- Learn *coverage set* Π containing optimal policy for any $\mathbf{w}$
- Learn the probability ψ^{π} of reaching each labeled exit state ξ for each $\pi \in \Pi$

Step 2: Plan with learned basis

- Dynamic programming can be executed with the set of labeled exit states and the set of policies $\pi \in \Pi$
- In step 2, we need *only* access to pre-computed probabilities and the instructions. *No new interaction* with the environment is needed.

Results across different instructions





Learning is faster for modular strategies (LOF & Ours)

Planning often faster in our method as we plan at logical level

Kuric, D., Infante, G., Gómez, V., Jonsson, A. & van Hoof, H. (2024). Planning with a learned policy basis to optimally solve complex tasks. In *ICAPS*.

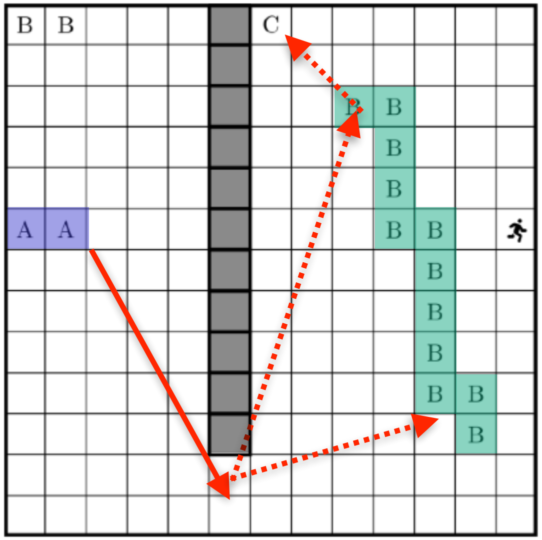
Extension to continuous state space

FSA labels are now generated by *regions* in the state space

Area within a region is represented by *basis features*

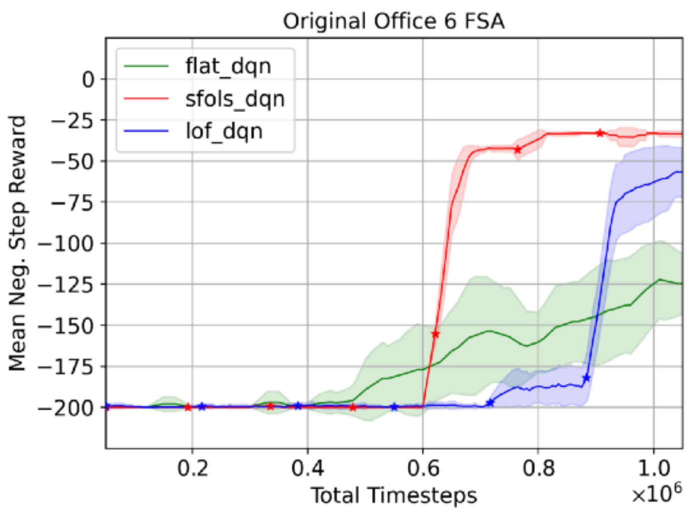
Policy basis learning unchanged

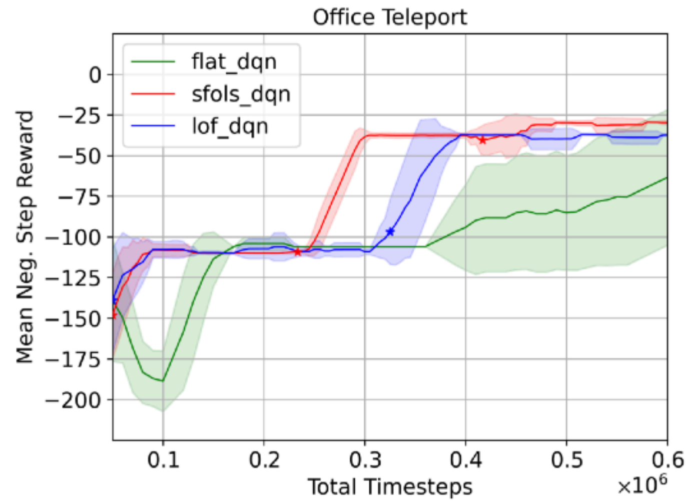
Planning now relies on a regression step to generalize cost-to-go to entire state space



Van Gelder, T. & van Hoof, H. Learning Spatially Refined Sub-Policies for Temporal Task Composition in Continuous RL. Submitted.

Results in continuous space





In several continuous-state tasks, outperform 'logical options framework' (esp. stochastic environment) and flat DQN

Van Gelder, T. & van Hoof, H. Learning Spatially Refined Sub-Policies for Temporal Task Composition in Continuous RL. Submitted.

Takeaways

We can pre-learn a policy basis that allows optimal zero-shot execution of any instruction provided as FSA

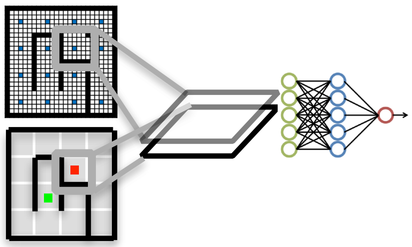
Explicit planning helps data efficiency & allows instruction following

Optimal behavior requires very large basis. Close to optimal performance is possible with a (much) smaller set.

Kuric, D., Infante, G., Gómez, V., Jonsson, A. & van Hoof, H. (2024). Planning with a learned policy basis to optimally solve complex tasks. In *ICAPS*.

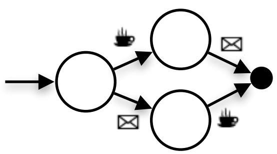
Modular learning for improving AI assistants

Woehlke et al., 2022	✓	✓	½



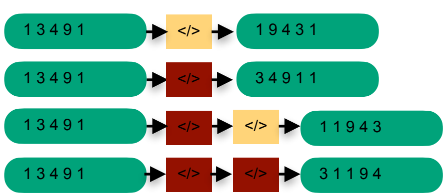
- *Learning and planning* have different advantages
- How can learning and planning modules interact?
- Does this allow tackling large and complex domains?

Kuric et al., 2024	✓	✓	✓



- How to learn for complex & never before seen *instructions*?
- Pre-learn behavior modules for different context
- Use on-the-fly *planning* to combine these modules

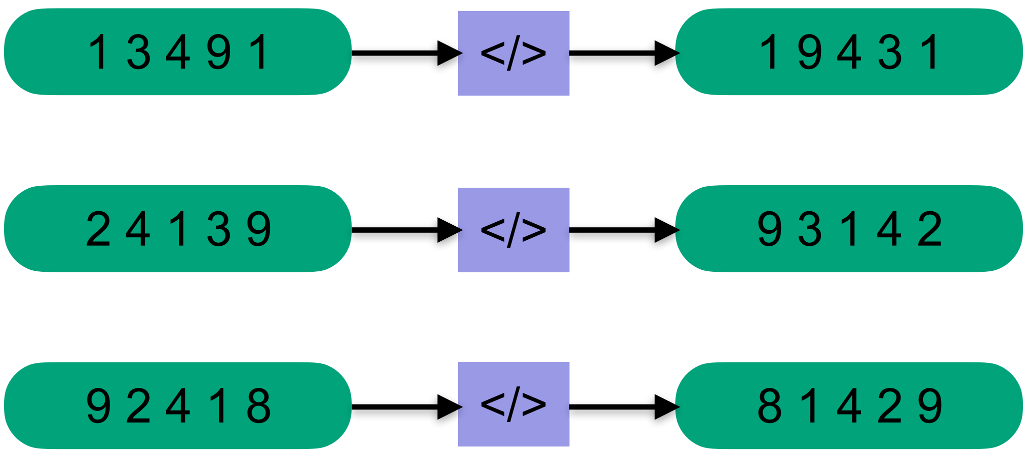
Macfarlane et al., 2025			✓



- Learn symbolic ‘language’ to describe mappings
- Allows *compositional generalization* in learned model
- Test-time optimization using differentiable decoder

Modular learning for neural program synthesis

Program synthesis: find program to explain relation between inputs and outputs



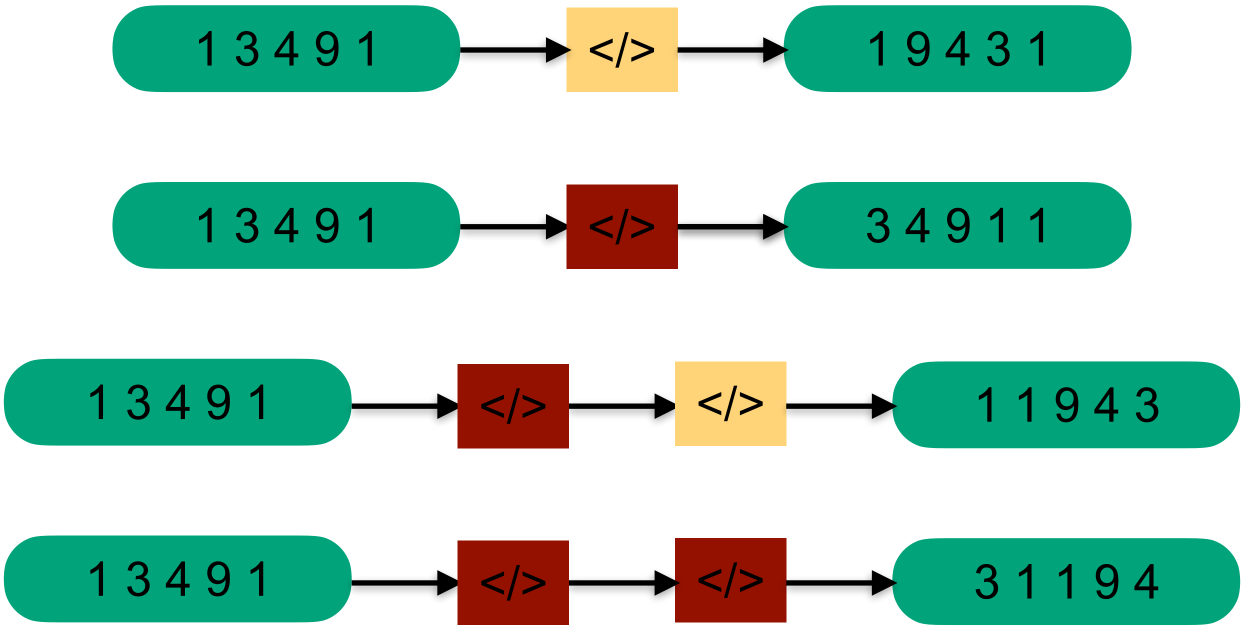
Symbolic approach: composition generalization of explicit rules

Neural approach: program is network, often monolithic

M. Macfarlane, C. Bonnet, H. van Hoof & L. Levis. Gradient-based program synthesis with Neurally Interpreted Languages. Submitted.

Search in a compositional space

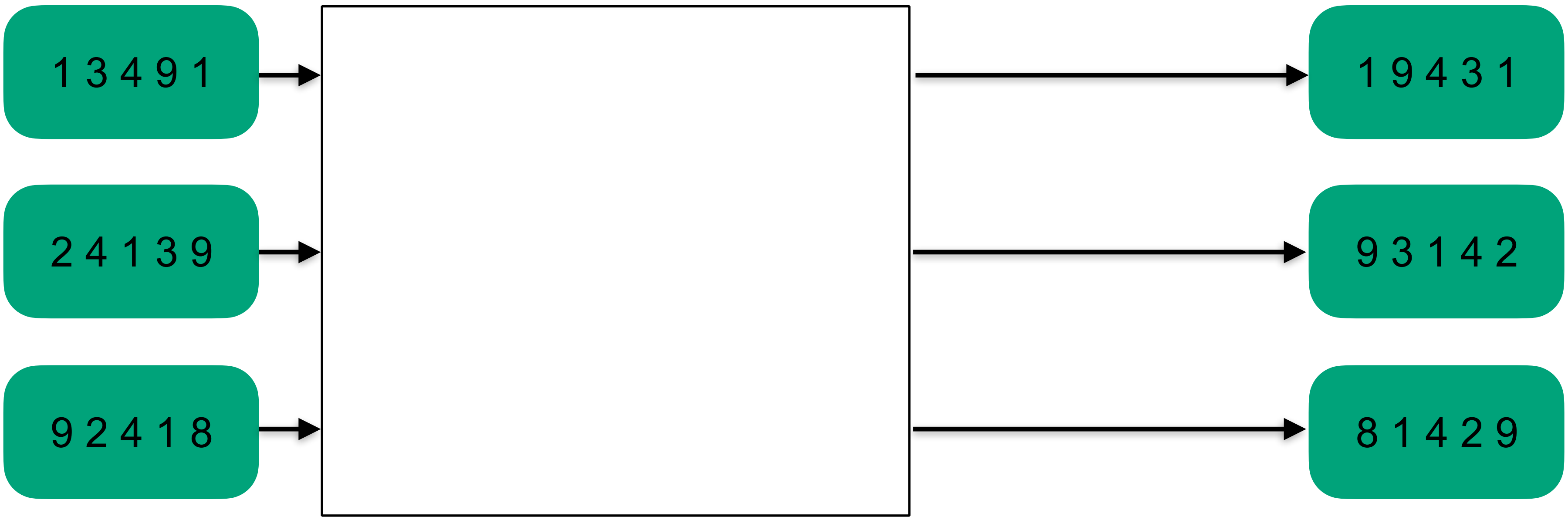
Idea 1: learned symbolic representation, allowing compositional generalization in a neural model



M. Macfarlane, C. Bonnet, H. van Hoof & L. Levis. Gradient-based program synthesis with Neurally Interpreted Languages. Submitted.

Search in a compositional space

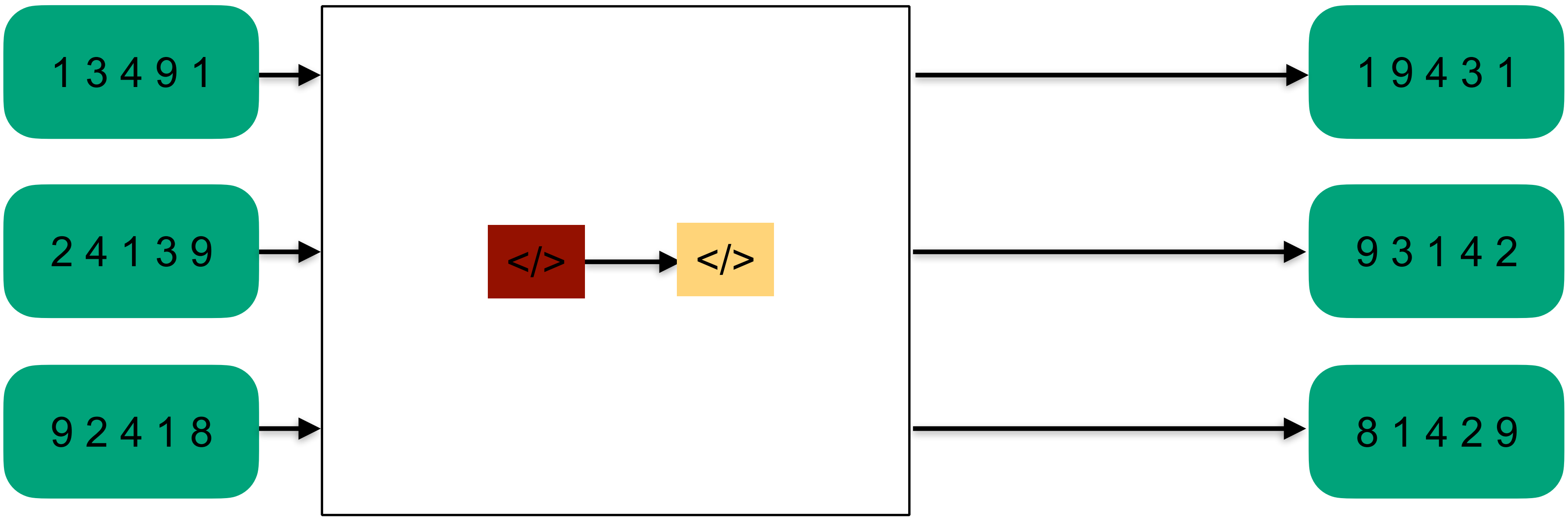
Idea 2: Test-time gradient-based search for best program in learned language



M. Macfarlane, C. Bonnet, H. van Hoof & L. Levis. Gradient-based program synthesis with Neurally Interpreted Languages. Submitted.

Search in a compositional space

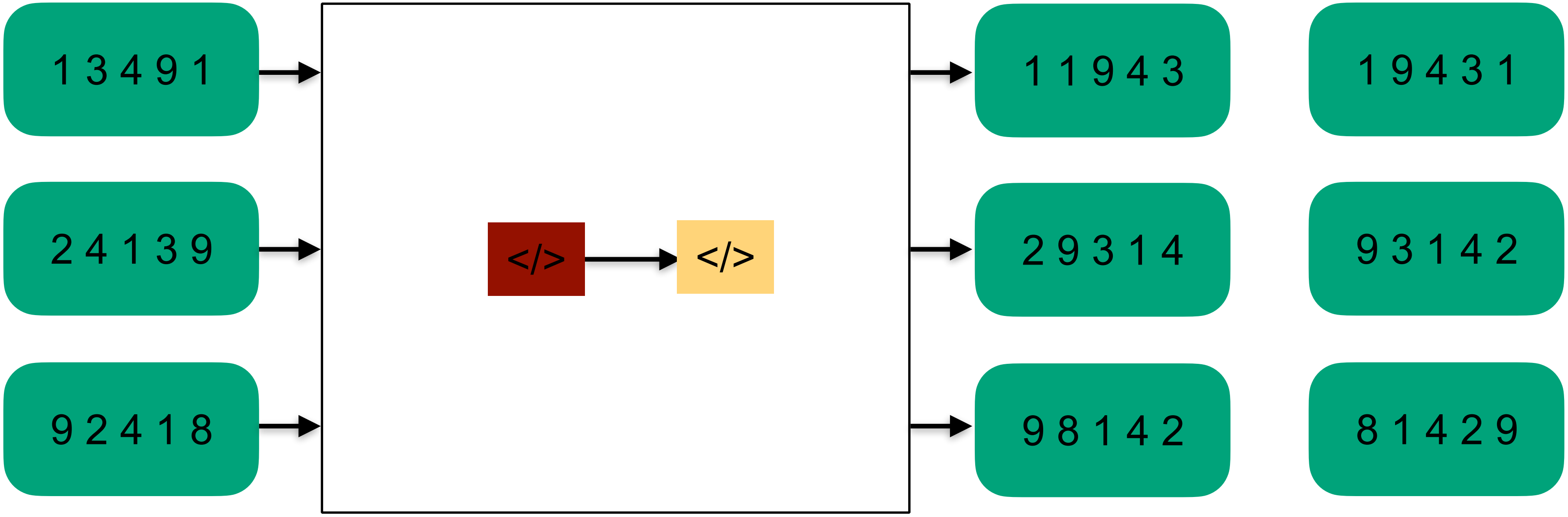
Idea 2: Test-time gradient-based search for best program in learned language



M. Macfarlane, C. Bonnet, H. van Hoof & L. Levis. Gradient-based program synthesis with Neurally Interpreted Languages. Submitted.

Search in a compositional space

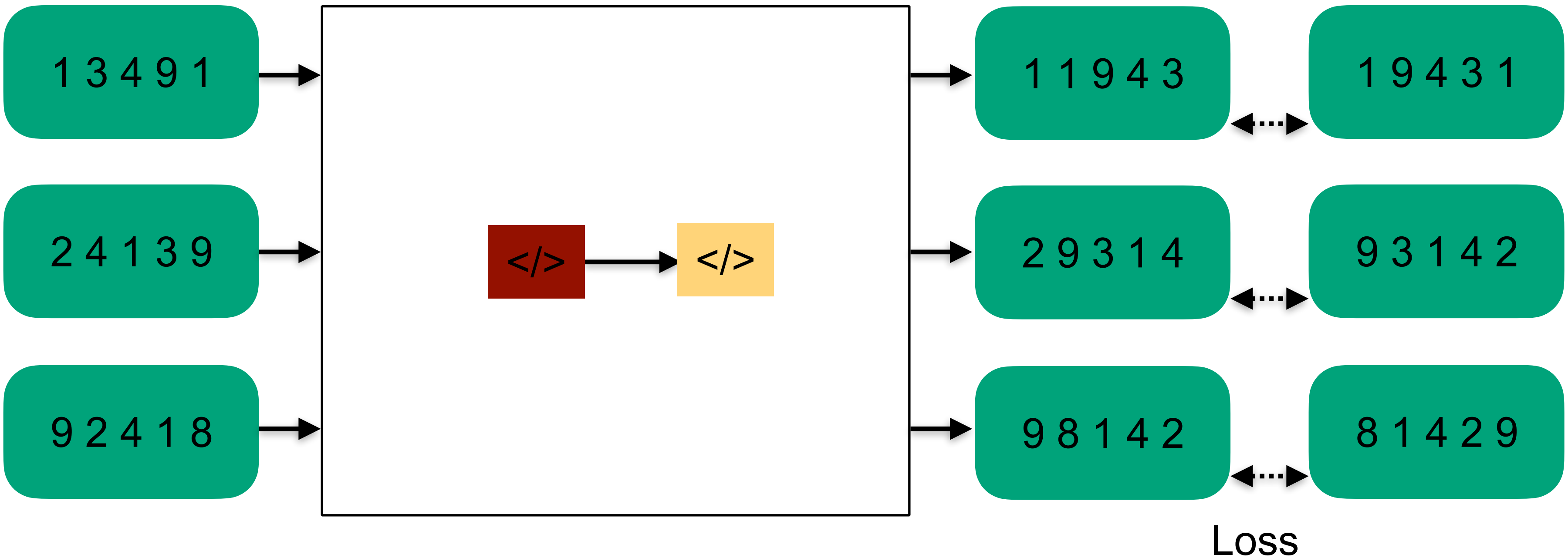
Idea 2: Test-time gradient-based search for best program in learned language



M. Macfarlane, C. Bonnet, H. van Hoof & L. Levis. Gradient-based program synthesis with Neurally Interpreted Languages. Submitted.

Search in a compositional space

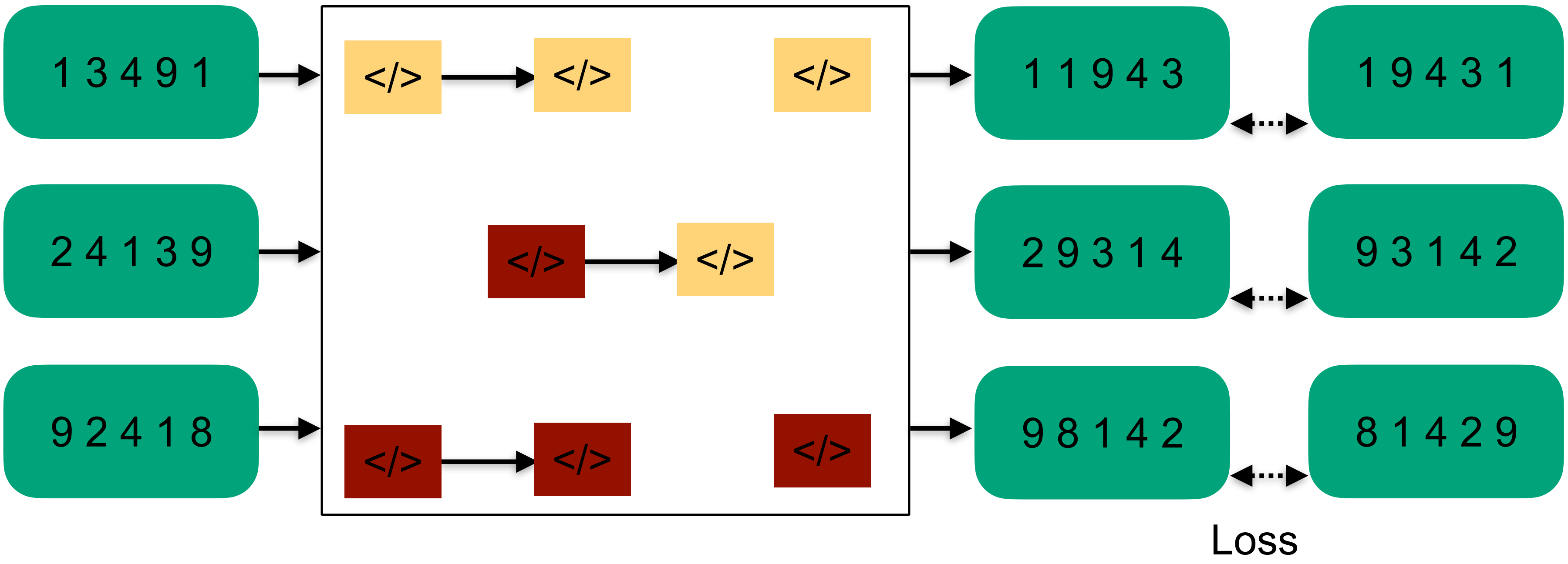
Idea 2: Test-time gradient-based search for best program in learned language



M. Macfarlane, C. Bonnet, H. van Hoof & L. Levis. Gradient-based program synthesis with Neurally Interpreted Languages. Submitted.

Search in a compositional space

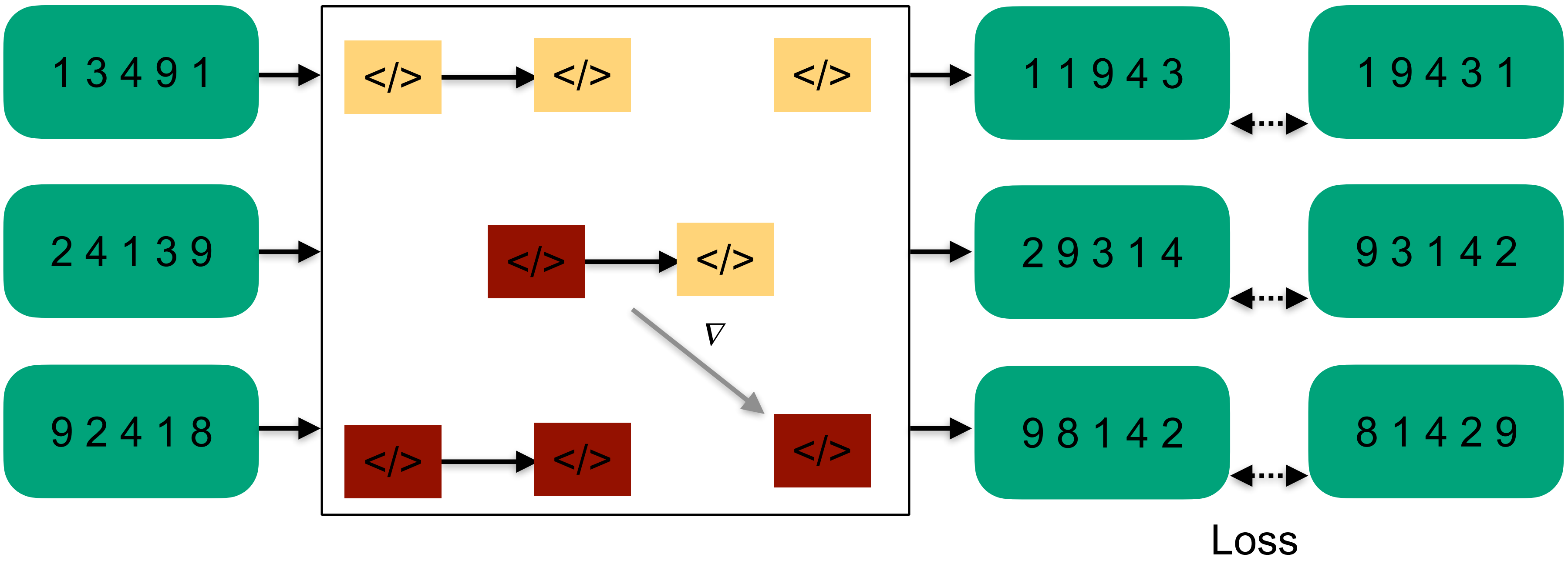
Idea 2: Test-time gradient-based search for best program in learned language



M. Macfarlane, C. Bonnet, H. van Hoof & L. Levis. Gradient-based program synthesis with Neurally Interpreted Languages. Submitted.

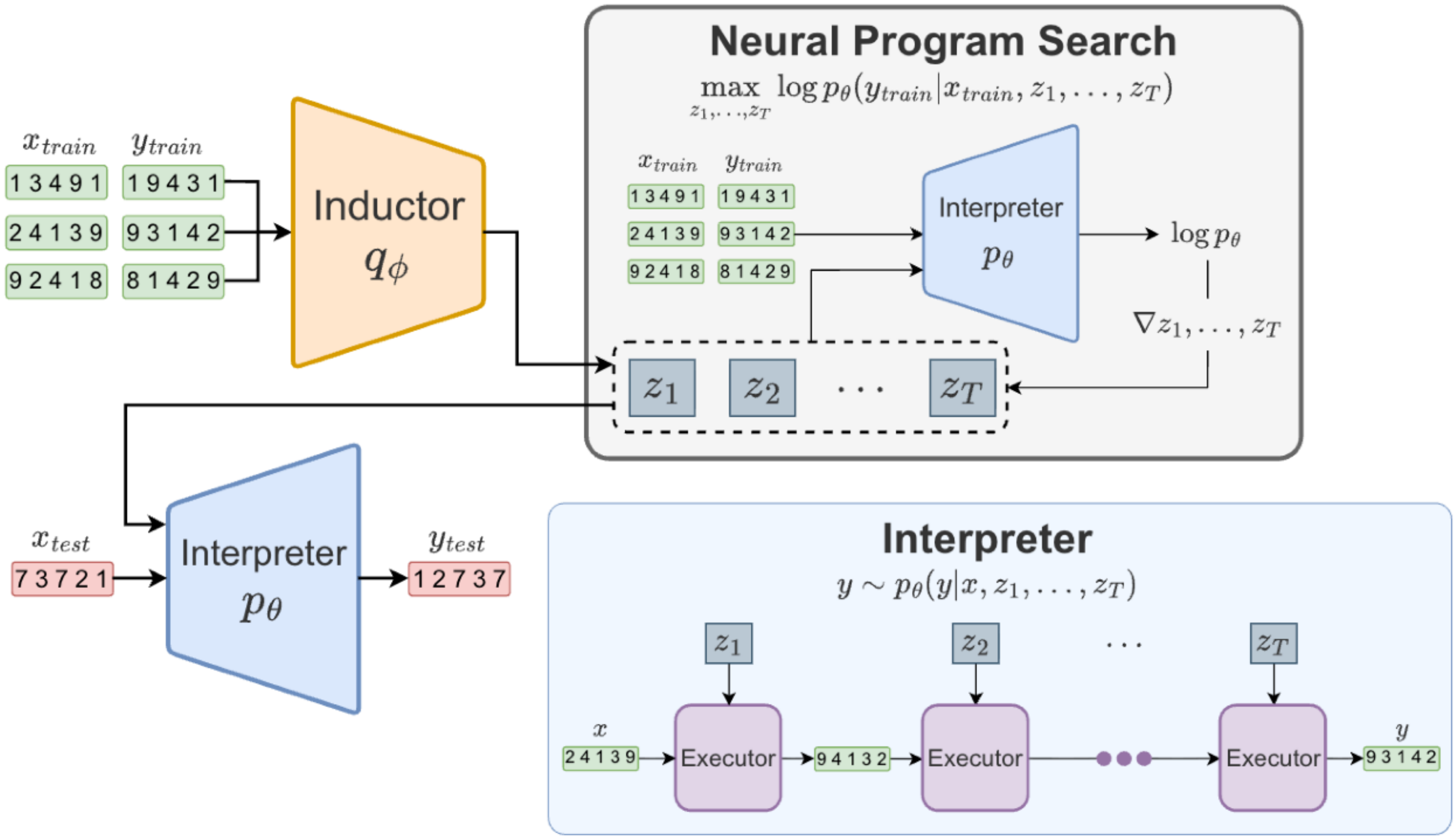
Search in a compositional space

Idea 2: Test-time gradient-based search for best program in learned language



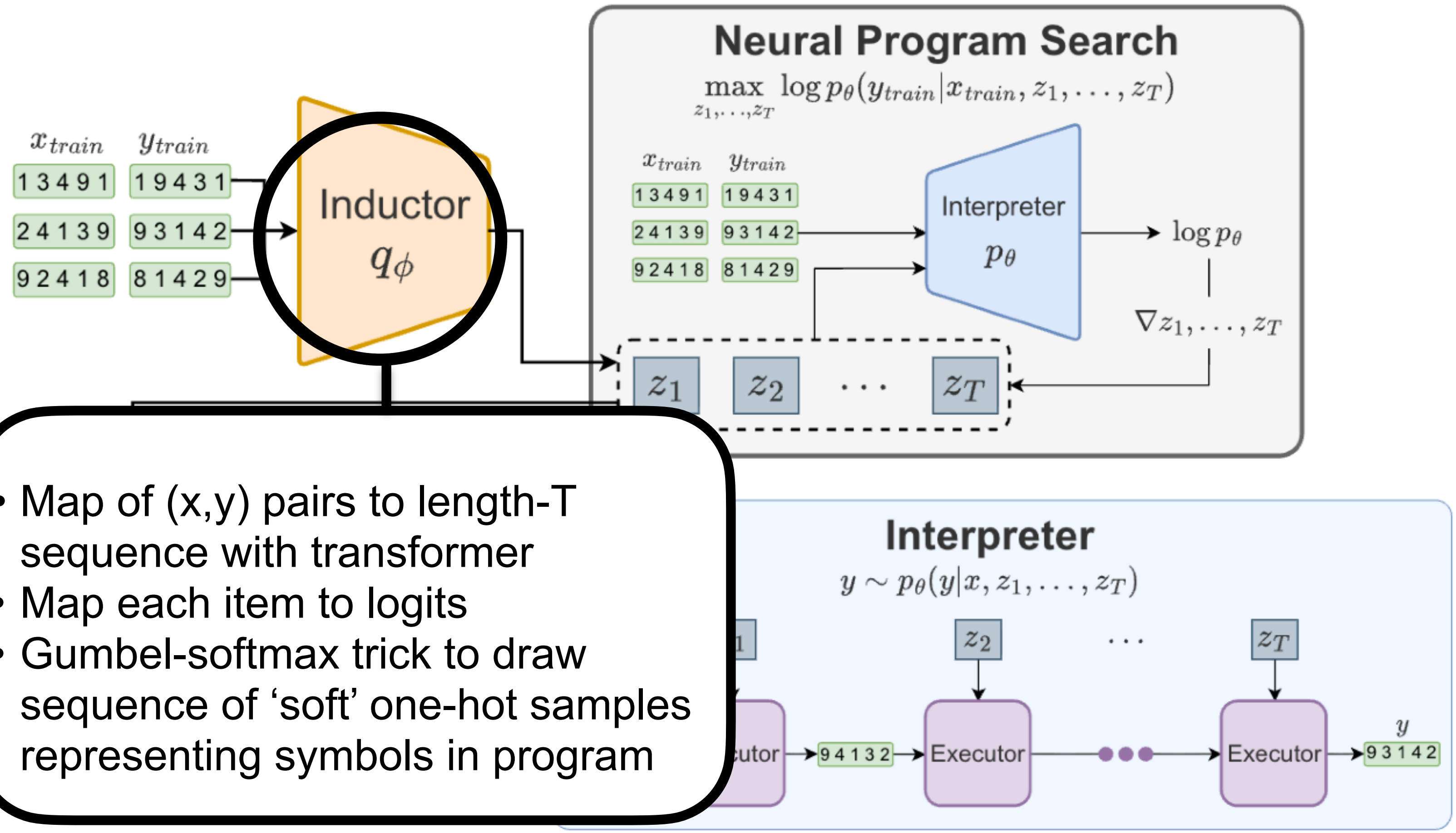
M. Macfarlane, C. Bonnet, H. van Hoof & L. Levis. Gradient-based program synthesis with Neurally Interpreted Languages. Submitted.

Search in a compositional space



M. Macfarlane, C. Bonnet, H. van Hoof & L. Levis. Gradient-based program synthesis with Neurally Interpreted Languages. Submitted.

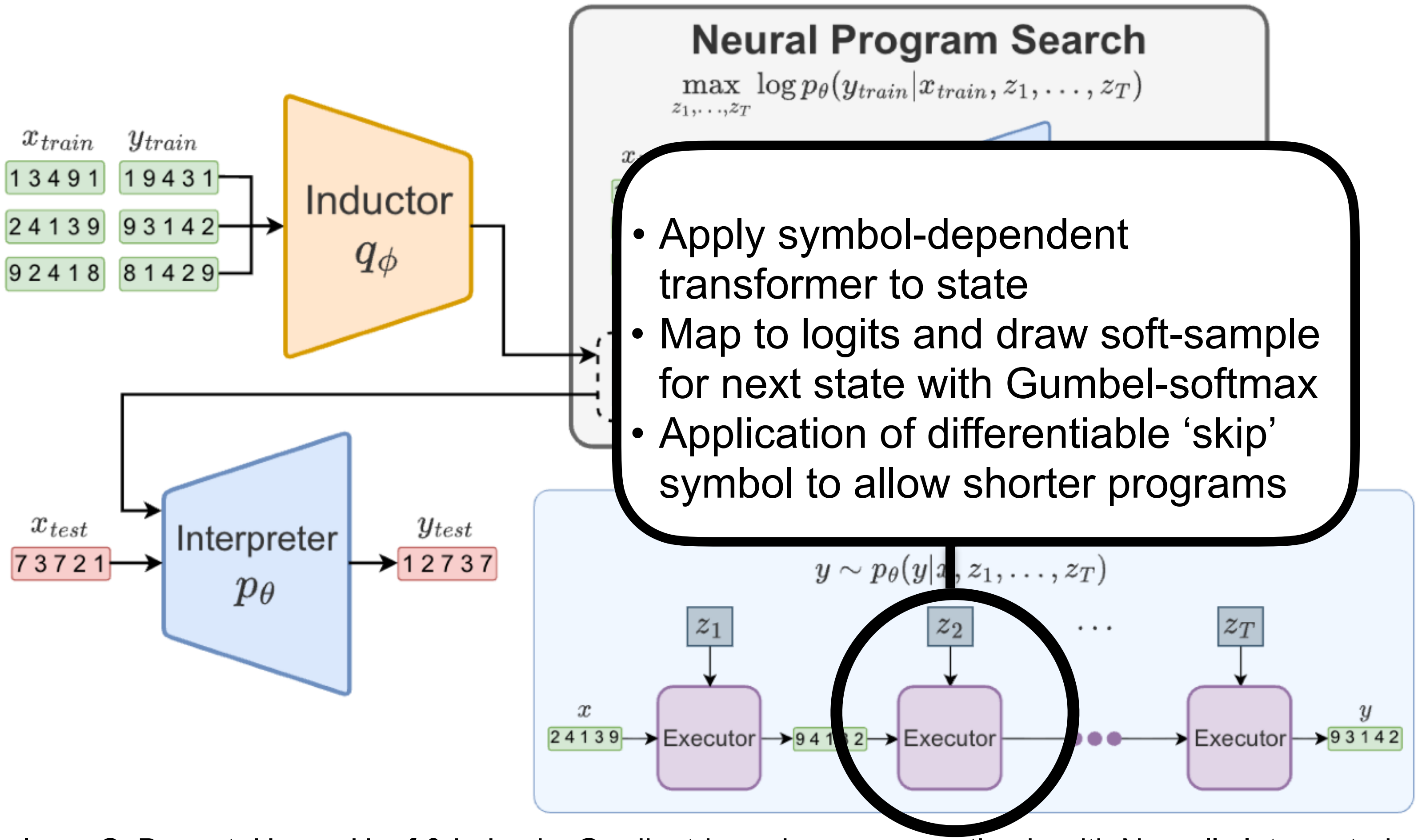
Search in a compositional space



- Map of (x,y) pairs to length-T sequence with transformer
- Map each item to logits
- Gumbel-softmax trick to draw sequence of 'soft' one-hot samples representing symbols in program

M. Macfarlane, C. Bonnet, H. van Hoof & L. Levis. Gradient-based program synthesis with Neurally Interpreted Languages. Submitted.

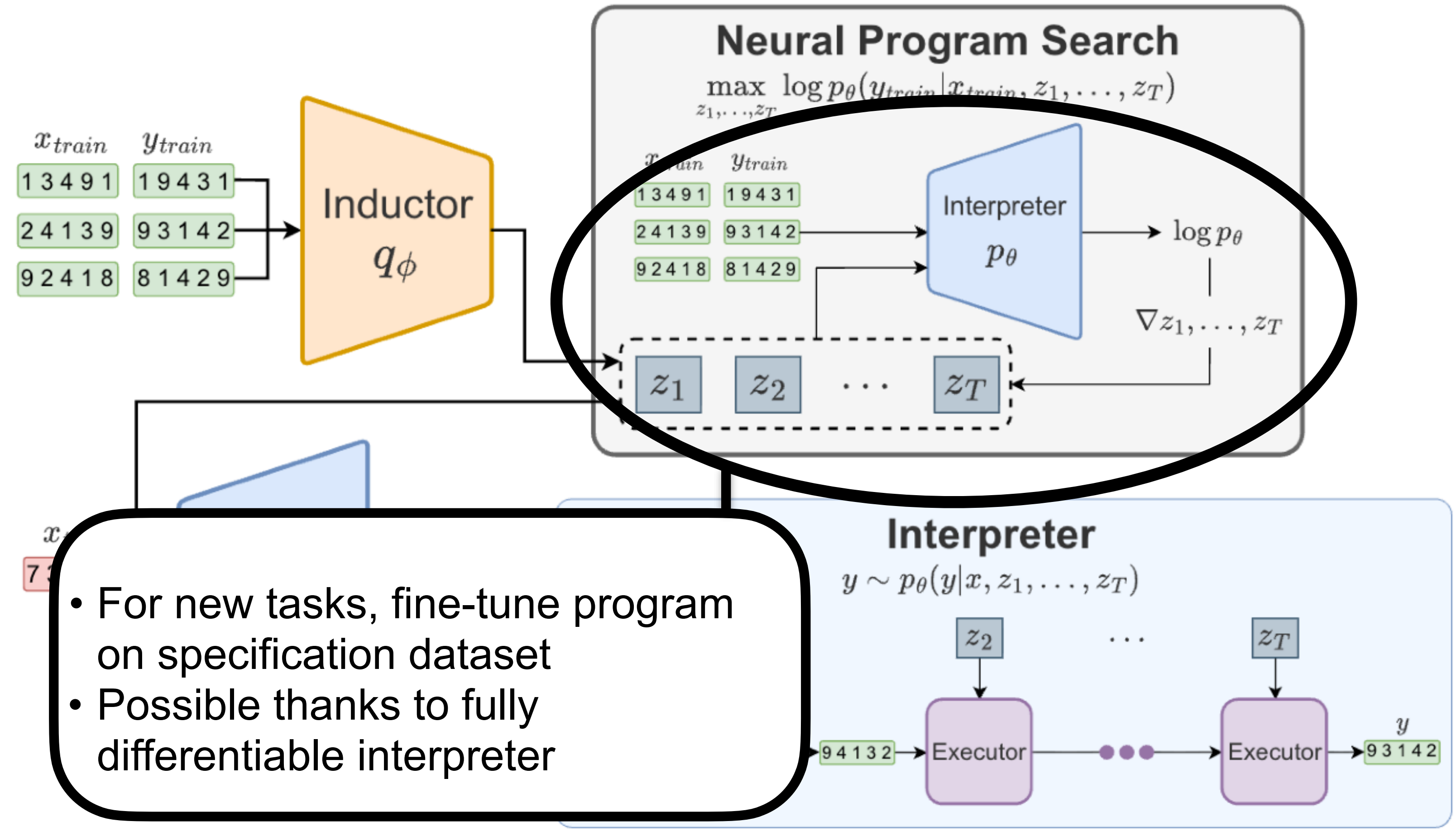
Search in a compositional space



- Apply symbol-dependent transformer to state
- Map to logits and draw soft-sample for next state with Gumbel-softmax
- Application of differentiable 'skip' symbol to allow shorter programs

M. Macfarlane, C. Bonnet, H. van Hoof & L. Levis. Gradient-based program synthesis with Neurally Interpreted Languages. Submitted.

Search in a compositional space



- For new tasks, fine-tune program on specification dataset
- Possible thanks to fully differentiable interpreter

M. Macfarlane, C. Bonnet, H. van Hoof & L. Levis. Gradient-based program synthesis with Neurally Interpreted Languages. Submitted.

Results on generalization benchmark

	Shift-L	
Method	ID	OOD
In-Context	1.00	0.00
TTT	1.00	0.00
LPN	1.00	0.00
LPN Gradient Search	1.00	0.03
D-LPN	1.00	0.02
D-LPN Gradient Search	1.00	0.01
NLI	1.00	0.00
NLI Prior Search	1.00	0.10
NLI Gradient Search	1.00	**0.99**

Train on left shift by 1 - 5 places
OOD on shifts of 6 - 10 places

Results on generalization benchmark

Method	Shift-L ID	Shift-L OOD
In-Context	1.00	0.00
TTT	1.00	0.00
LPN	1.00	0.00
LPN Gradient Search	1.00	0.03
D-LPN	1.00	0.02
D-LPN Gradient Search	1.00	0.01
NLI	1.00	0.00
NLI Prior Search	1.00	0.10
NLI Gradient Search	1.00	**0.99**

Train on left shift by 1 - 5 places
OOD on shifts of 6 - 10 places

Compared to neural baselines, only proposed "neural language interpreter" + gradient search does well out-of-distribution

Results on generalization benchmark

Method	Shift-L		Shift-P		Comp-I	
	ID	OOD	ID	OOD	ID	OOD
In-Context	1.00	0.00	1.00	0.00	1.00	0.13
TTT	1.00	0.00	1.00	0.00	0.95	0.14
LPN	1.00	0.00	1.00	0.00	1.00	0.18
LPN Gradient Search	1.00	0.03	1.00	0.00	1.00	0.29
D-LPN	1.00	0.02	1.00	0.00	0.99	0.15
D-LPN Gradient Search	1.00	0.01	1.00	0.00	0.99	0.20
NLI	1.00	0.00	1.00	0.00	1.00	0.17
NLI Prior Search	1.00	0.10	1.00	0.00	1.00	0.23
NLI Gradient Search	1.00	**0.99**	1.00	**1.00**	1.00	**0.91**

Compared to neural baselines, only proposed "neural language interpreter" + gradient search does well out-of-distribution

Results on generalization benchmark

Learned Program Representations for Shift-L

Ground Truth Program	NLI Program Representation
shift_left(1)	231
shift_left(2)	231 231
shift_left(3)	231 231 231
shift_left(4)	231 476 231
shift_left(5)	231 476 476
...	
shift_left(8) (OOD)	231 231 231 231 476 476

Compared to neural baselines, only proposed “neural language interpreter” + gradient search does well out-of-distribution

Results on generalization benchmark

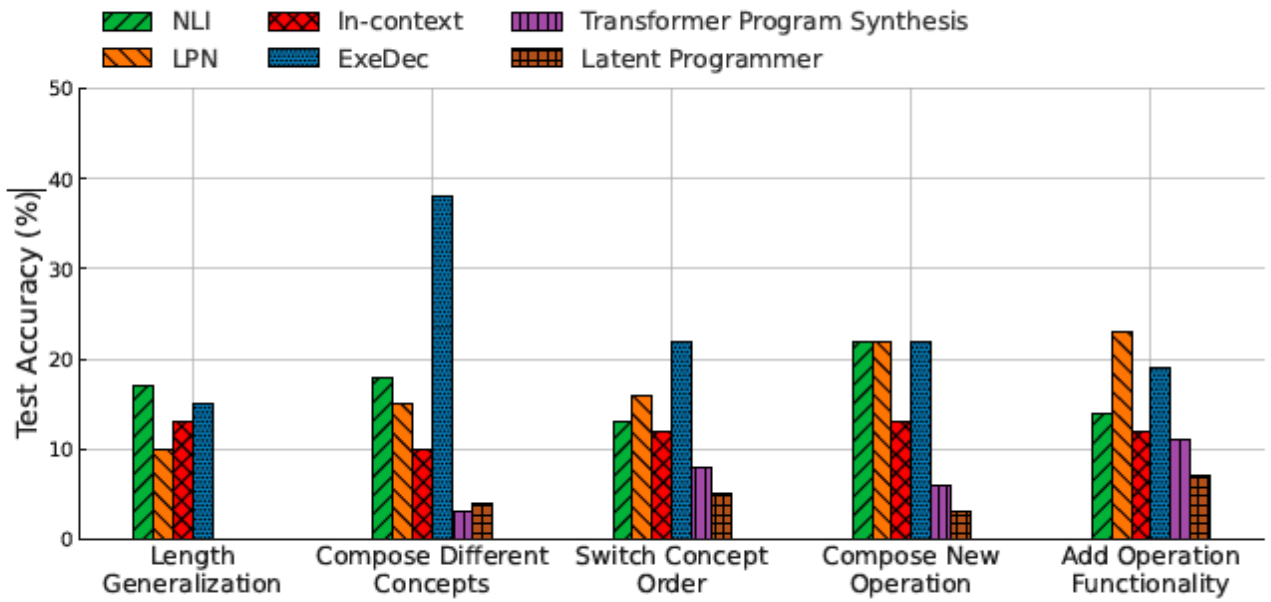
Learned Program Representations for Shift-L

Ground Truth Program	NLI Program Representation
shift_left(1)	231
shift_left(2)	231 231
shift_left(3)	231 231 231
shift_left(4)	231 476 231
shift_left(5)	231 476 476
...	
shift_left(8) (OOD)	231 231 231 231 476 476

Compared to neural baselines, only proposed “neural language interpreter” + gradient search does well out-of-distribution

NLI symbols can aid interpretation

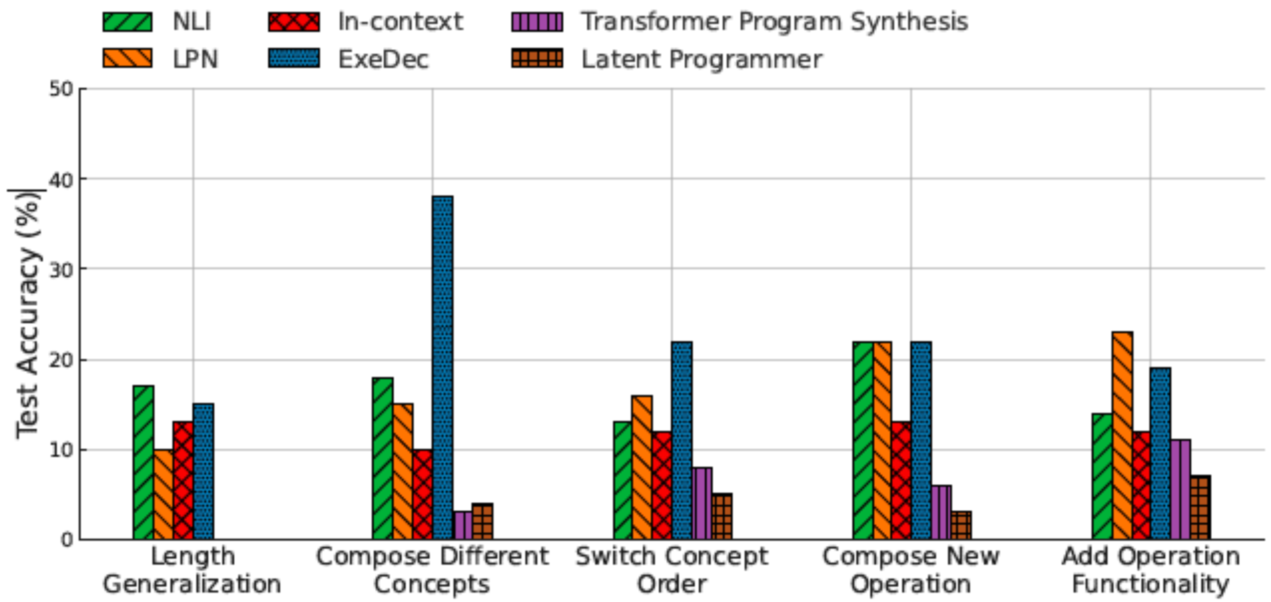
Results on DeepCoder benchmark



Compared to neural baselines, only proposed “neural language interpreter” + gradient search does well out-of-distribution

NLI symbols can aid interpretation

Results on DeepCoder benchmark



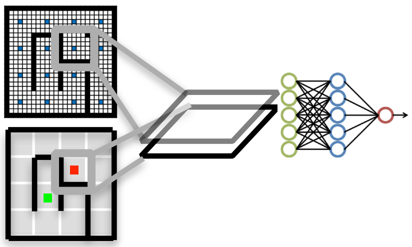
Compared to neural baselines, only proposed “neural language interpreter” + gradient search does well out-of-distribution

NLI symbols can aid interpretation

Competitive performance with neuro-symbolic methods, that require ground-truth programs during training

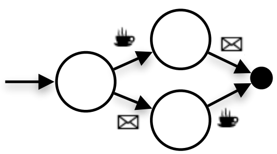
Modular learning for improving AI assistants

Woehlke et al., 2022	✓	✓	½



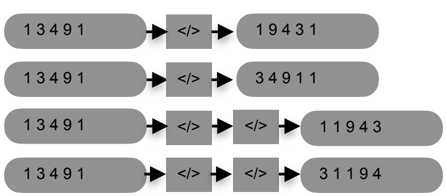
- *Learning and planning* have different advantages
- How can learning and planning modules interact?
- Does this allow tackling large and complex domains?

Kuric et al., 2024	✓	✓	✓



- How to learn for complex & never before seen *instructions*?
- Pre-learn behavior modules for different context
- Use on-the-fly *planning* to combine these modules

Macfarlane et al., 2025			✓



- Learn symbolic ‘language’ to describe mappings
- Allows *compositional generalization* in learned model
- Test-time optimization using differentiable decoder

Wrap-up

Capable AI assistants require at least

- Data-efficient learning and generalization to handle niche domains
- Explicit reasoning or planning to provide transparency & predictability
- Instructability: provide a channel for specifying the user’s wishes

Wrap-up

Capable AI assistants require at least

- Data-efficient learning and generalization to handle niche domains
- Explicit reasoning or planning to provide transparency & predictability
- Instructability: provide a channel for specifying the user's wishes

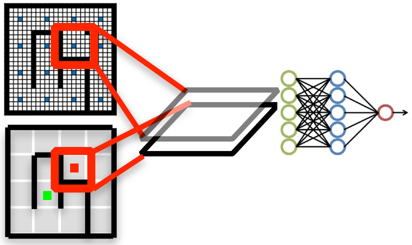
Modular learning helps to make progress on all these dimensions

Wrap-up

Detailed look at three research projects

Wrap-up

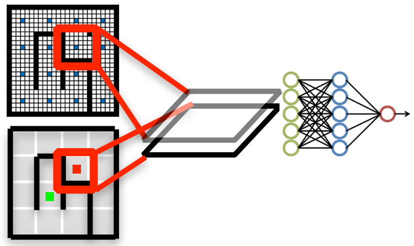
Detailed look at three research projects



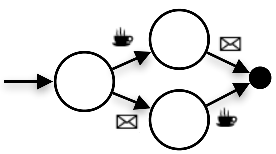
Composition of global planning with local learning module to strike a balance between flexible generalization and reactivity

Wrap-up

Detailed look at three research projects



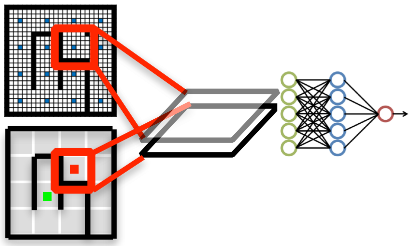
Composition of global planning with local learning module to strike a balance between flexible generalization and reactivity



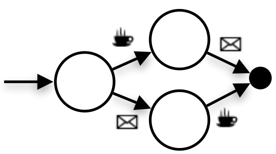
Composition of policy modules from a pre-learned basis to allow optimal zero-shot generalization to never-before seen instructions

Wrap-up

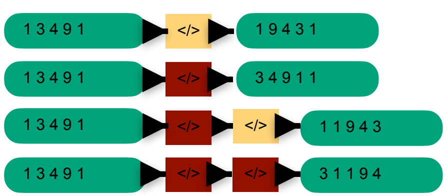
Detailed look at three research projects



Composition of global planning with local learning module to strike a balance between flexible generalization and reactivity



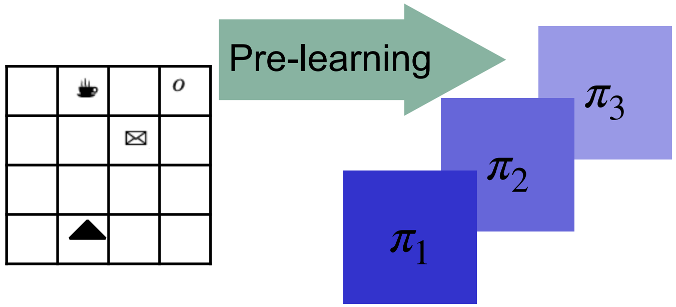
Composition of policy modules from a pre-learned basis to allow optimal zero-shot generalization to never-before seen instructions



Neural learning of symbolic 'language' allows compositional generalization and test-time optimization

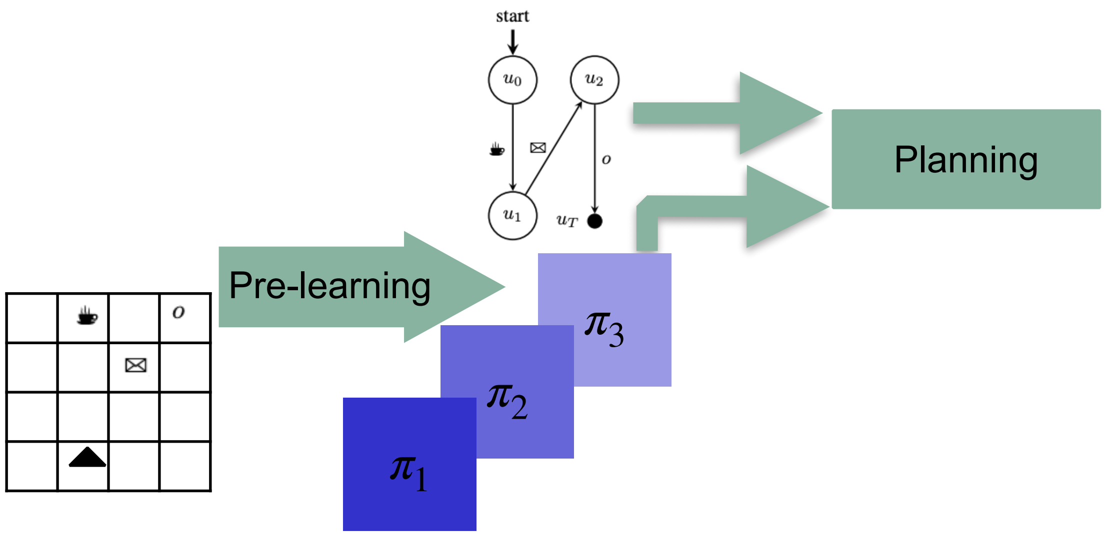
Modular learning for improving AI assistants

A possible architecture



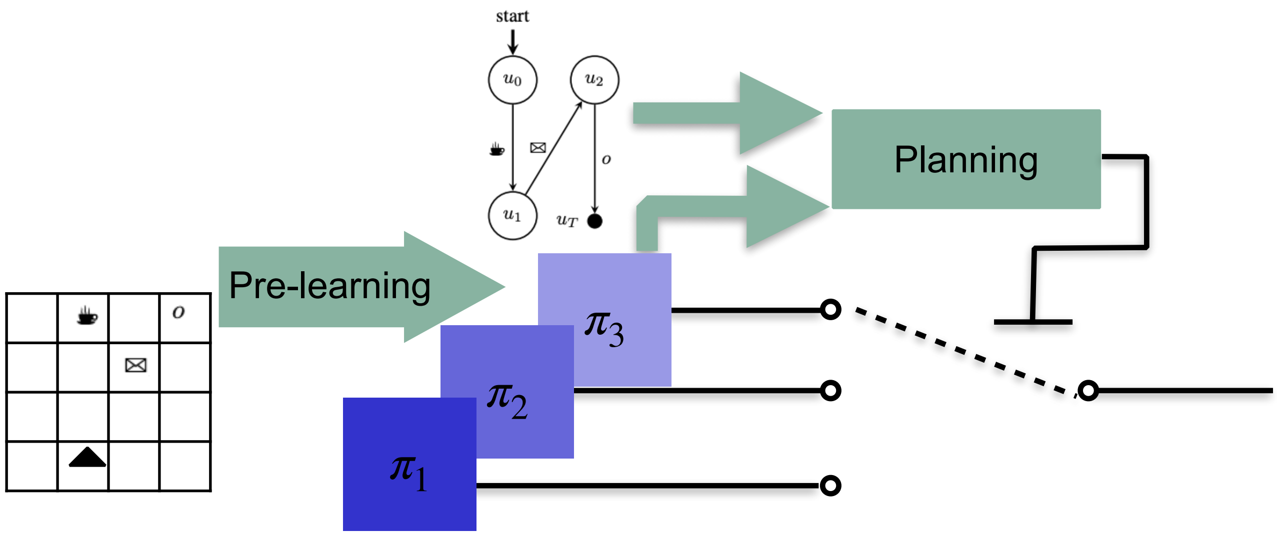
Modular learning for improving AI assistants

A possible architecture



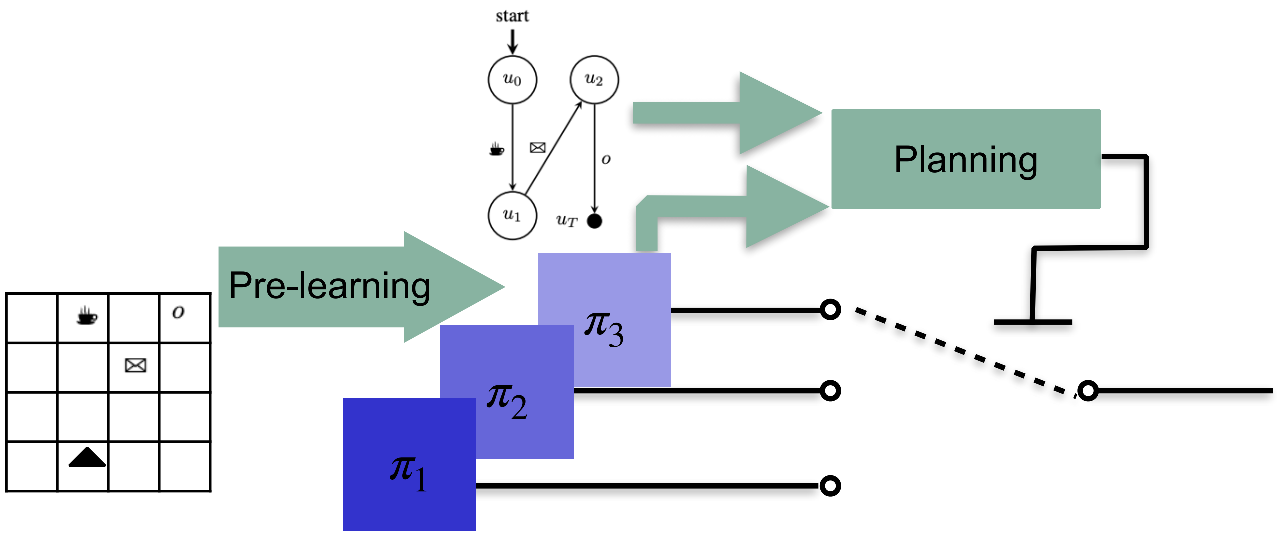
Modular learning for improving AI assistants

A possible architecture



Modular learning for improving AI assistants

A possible architecture

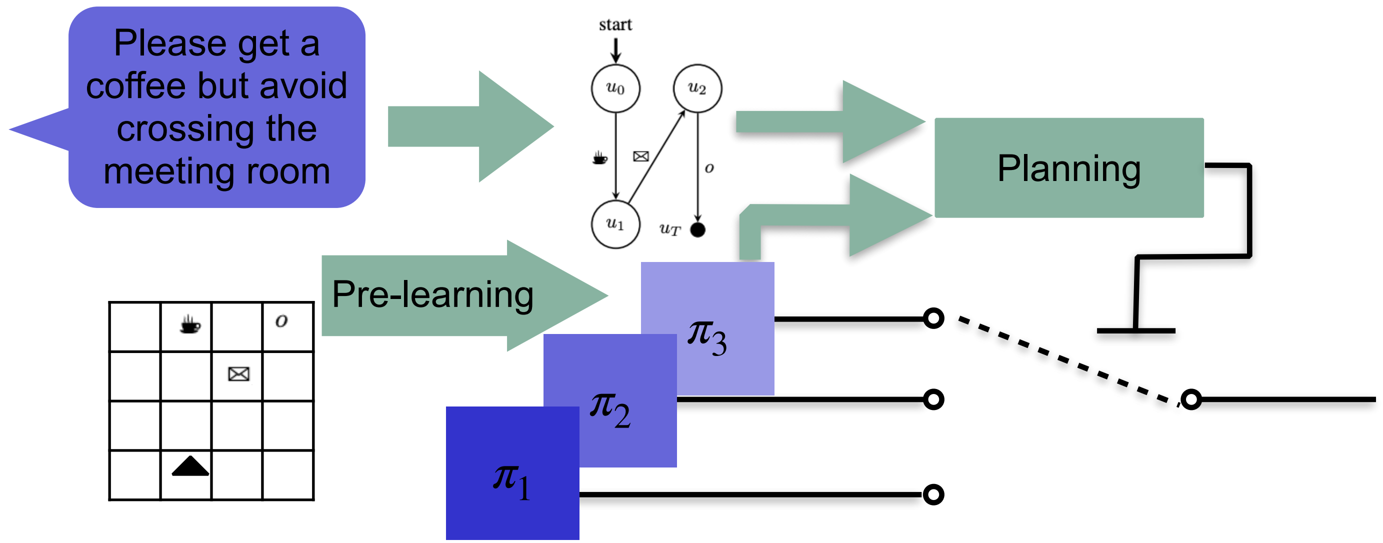


What is still required

- Intuitive instructability (e.g., language) while maintaining predictability
- Compositional algorithms with bounded sub-optimality at scale
- Predictable behavior with real-world sensors
- Study of interpretability and useability of modular systems

Modular learning for improving AI assistants

A possible architecture

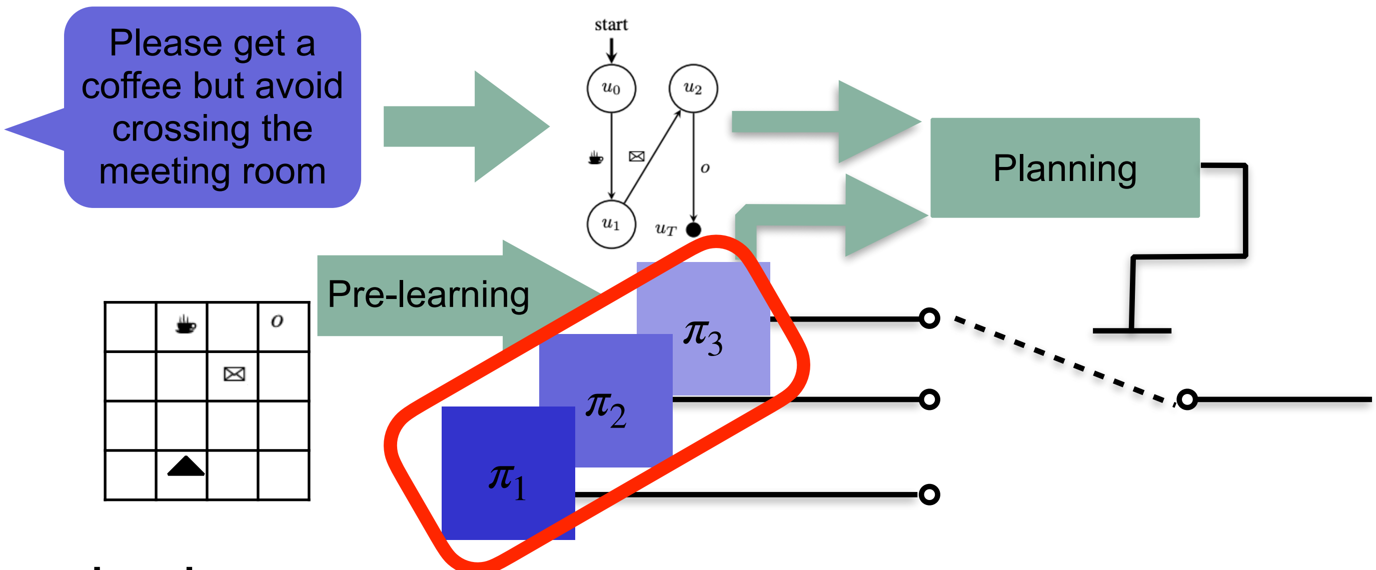


What is still required

- Intuitive instructability (e.g., language) while maintaining predictability
- Compositional algorithms with bounded sub-optimality at scale
- Predictable behavior with real-world sensors
- Study of interpretability and useability of modular systems

Modular learning for improving AI assistants

A possible architecture

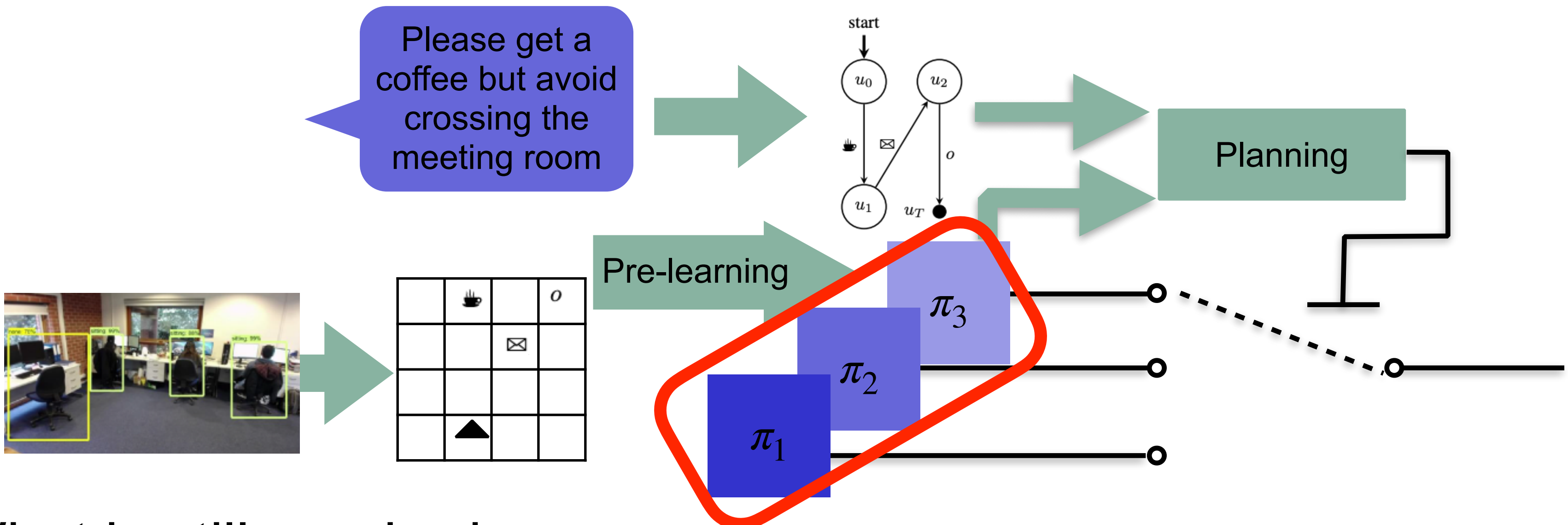


What is still required

- Intuitive instructability (e.g., language) while maintaining predictability
- Compositional algorithms with bounded sub-optimality at scale
- Predictable behavior with real-world sensors
- Study of interpretability and useability of modular systems

Modular learning for improving AI assistants

A possible architecture

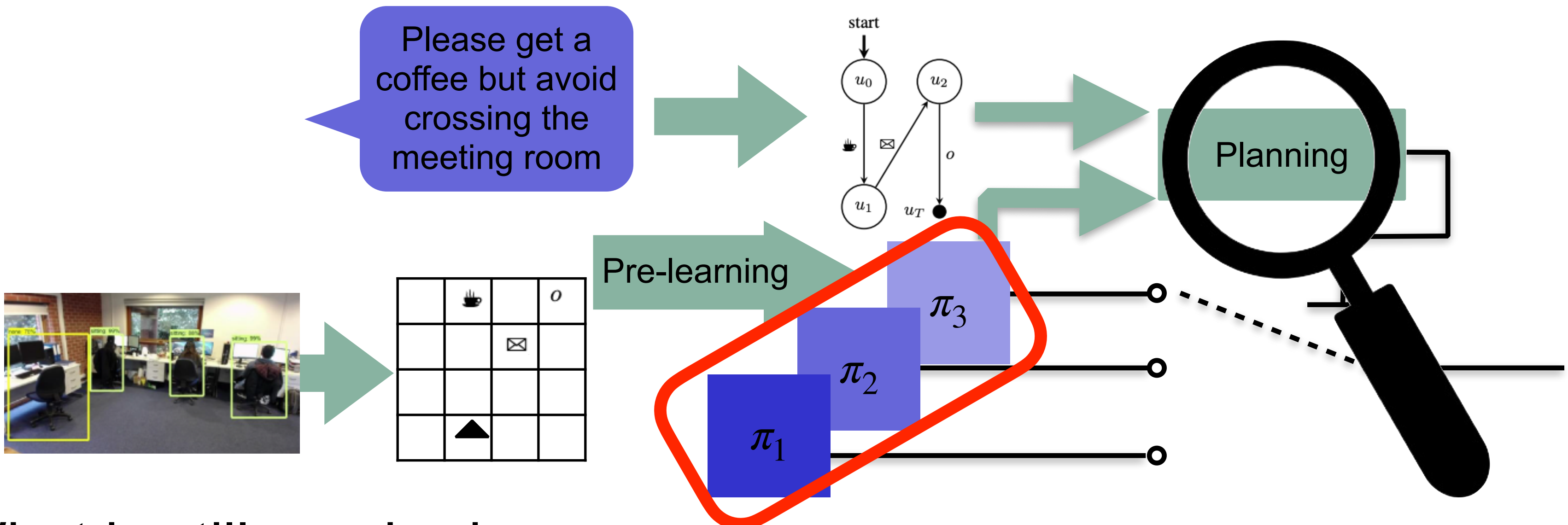


What is still required

- Intuitive instructability (e.g., language) while maintaining predictability
- Compositional algorithms with bounded sub-optimality at scale
- Predictable behavior with real-world sensors
- Study of interpretability and useability of modular systems

Modular learning for improving AI assistants

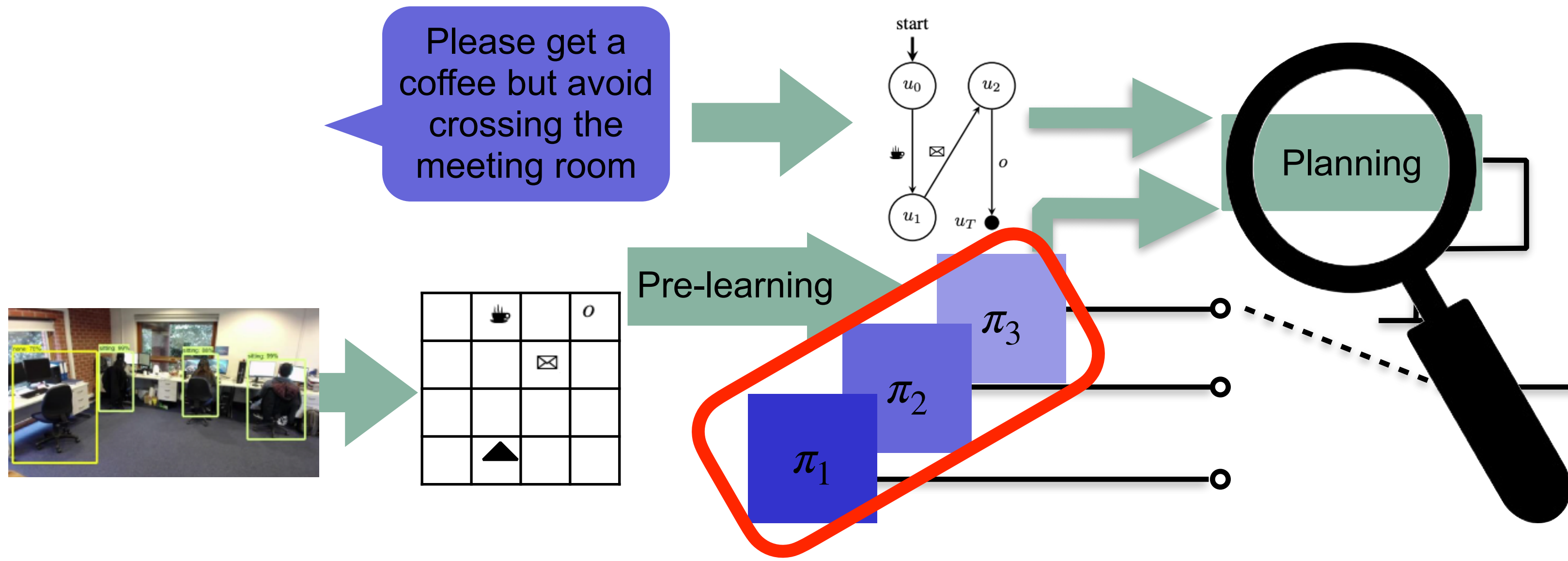
A possible architecture



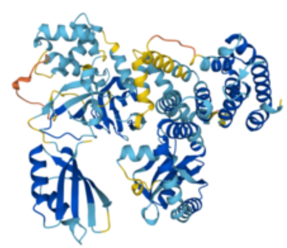
What is still required

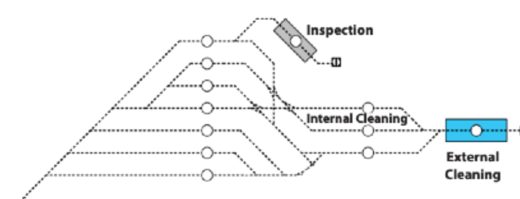
- Intuitive instructability (e.g., language) while maintaining predictability
- Compositional algorithms with bounded sub-optimality at scale
- Predictable behavior with real-world sensors
- Study of interpretability and useability of modular systems

Modular learning for improving AI assistants



www.switchmybusiness.com


Wikipedia/DeepMind



[Peer et al., 2018]

Acknowledgement

This work received funding from the European Union's Horizon Europe Research and Innovation Programme under grant agreement no. 101119527 (AI4REALNET) and is supported by the action CNS2022-136178 financed by MCIN/AEI/10.13039/501100011033 and by the European Union Next Generation EU/PRTR. This work has been co-funded by MCIN/AEI/10.13039/501100011033 under the Maria de Maeztu Units of Excellence Programme (CEX2021-001195-M). Anders Jonsson is partially supported by the EU ICT-48 2020 project TAILOR (No. 952215), AGAUR SGR, and the Spanish grant PID2019-108141GB-I00. David Kuric acknowledges travel support from the ELISE Network funded by European Union's Horizon 2020 research and innovation programme (GA No 951847).

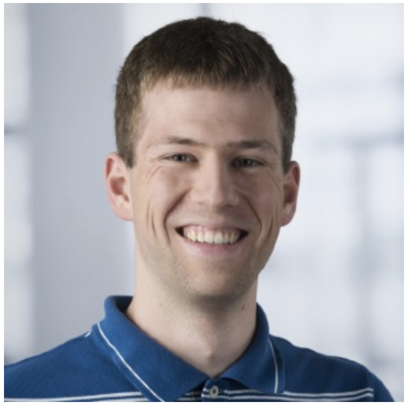
Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

This research was partly funded by the funded by LIFT-project 019.011 which is partly financed by the Dutch Research Council (NWO).

Levi Lelis was supported by the Canada CIFAR AI Chair and NSERC.

Thanks for your attention!

Collaborators



Jan Wöhlke
Bosch Center for AI

David Kuric
U. of Amsterdam

Guillermo
Infante Molina
Universitat Pompeu Fabra

Matthew Macfarlane
U. Of Amsterdam

Felix Schmitt (BCAI)

Vicenç Gómez (UPF)
Anders Jonsson (UPF)

Clément Bonnet (Ndea)
Levi Lelis (U. Alberta)

Questions?