Embodied Vision-and-Language Navigation with Dynamic Convolutional Filters

Federico Landi, Lorenzo Baraldi, Massimiliano Corsini, Rita Cucchiara

University of Modena and Reggio Emilia, Italy

30th British Machine Vision Conference - Sep 10th, 2019

Navigation is not that simple

"Go ahead and get to the end of the corridor.
Head upstairs and reach the third floor.
Wait in the room immediately on the left."

How to get to the goal?

Vision-and-Language Navigation (VLN)

VLN is a task in which an agent needs to...

- Interpret a previously unseen natural language navigation command in light of images generated by a previously unseen real environment *(Anderson et al. CVPR 2018)*
- Follow a given instruction to navigate from a starting location to a goal location *(Fried et al. NeurIPS 2018)*
- …
- …
- Reach a target location by navigating unseen environments, with a natural language instruction as only clue *(This work)*
- …
- …
- **Know where to go!** (and how to get there)

Know where to go...

360° image (surrounding environment)

Instruction can be...

a) "Take a right, going past the kitchen into the hallway"

b) "Walk into the sitting area and stop before the couch"

c) ...anything else (objects, directions, colors, ...)

Know where to go...

360° image (surrounding environment)

Dynamic convolutional filters address diversity in instructions

How to get there?

360° image (surrounding environment)

1) **Low-level action space**

(Anderson et al. CVPR 2018; Wang et al. ECCV 2018; This work)

2) **High-level action space**

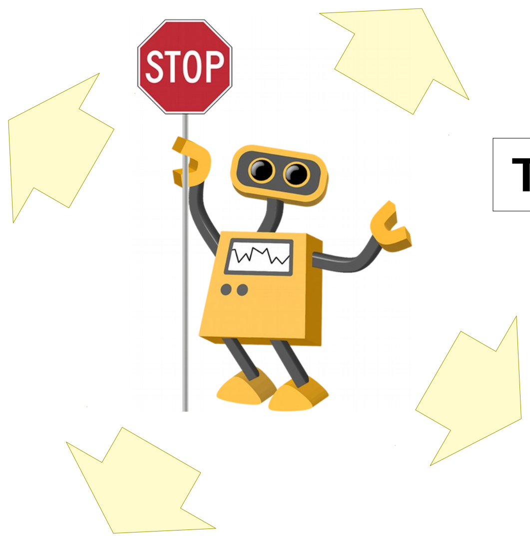
(Fried et al. NeurIPS 2018; Ma et al. ICLR 2019 & CVPR 2019; ...)

How to get there?

Low-level action space

Simulates continuous control of the agent

Move forward, turn left/right, tilt up/down, stop



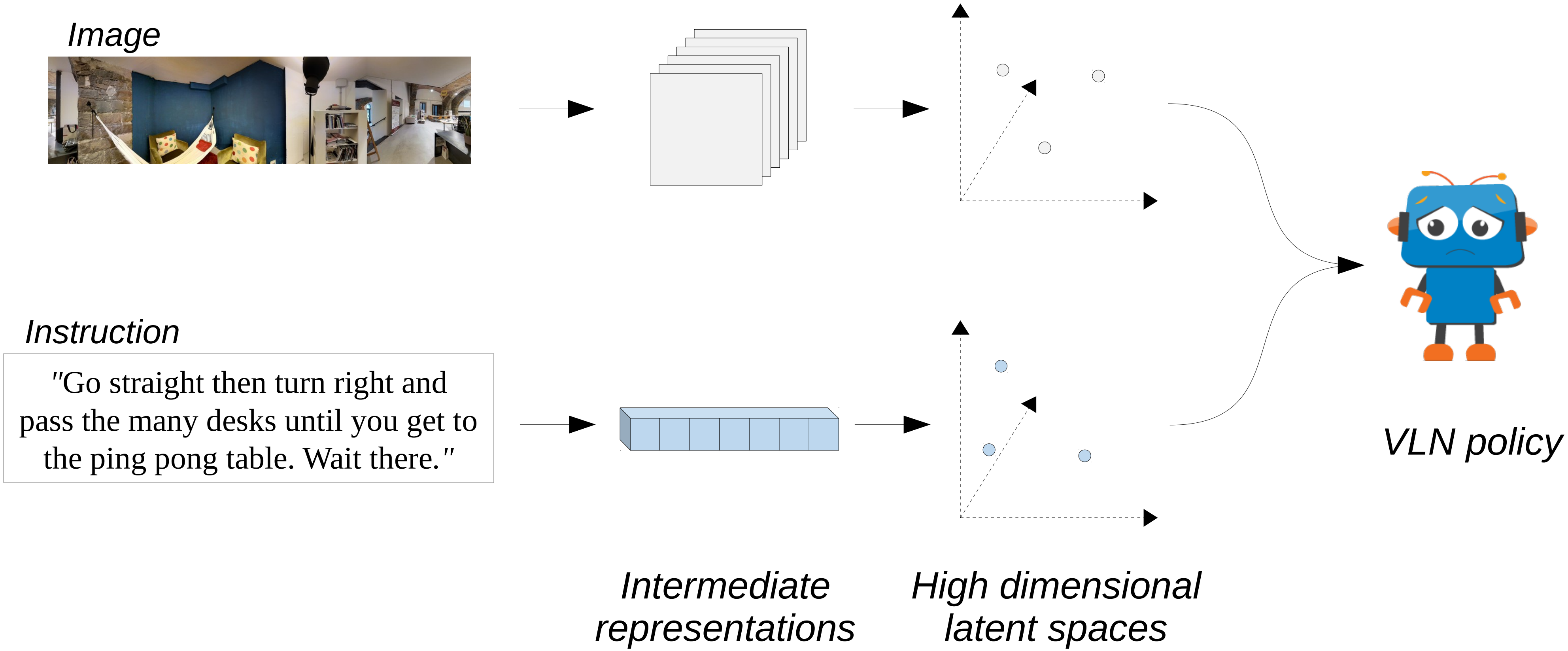
This work!

High-level action space

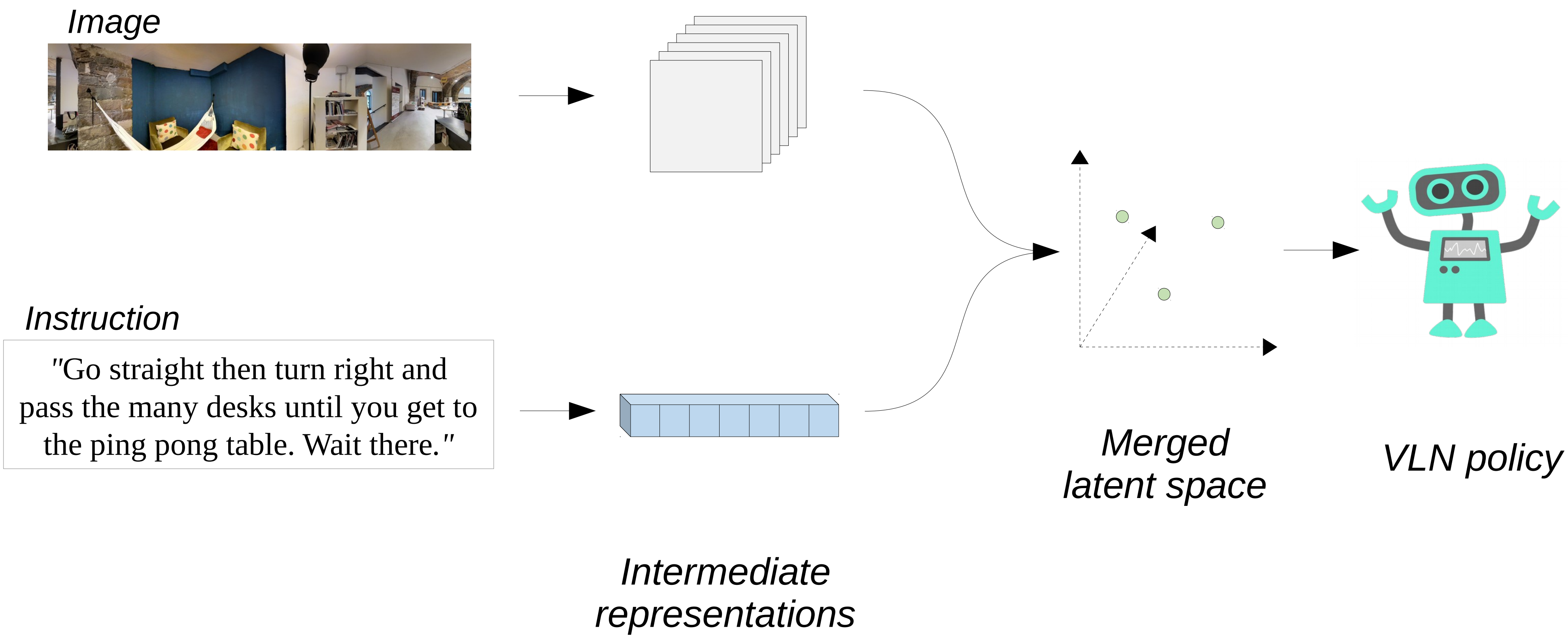
Path selection on a discrete graph

Action space is a list of adjacent nodes

Common approach to VLN



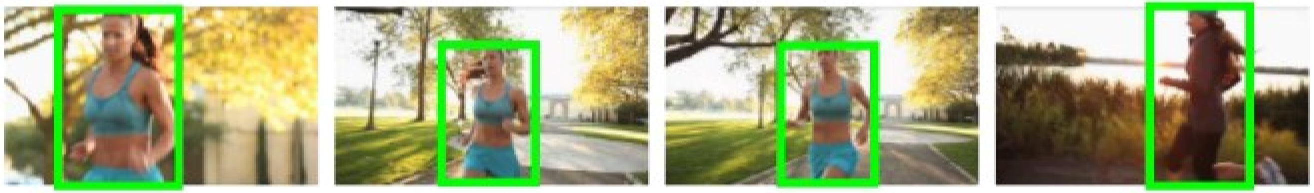
Our approach



Dynamic convolutional filters

...or "let the sentence drive the convolution"

Query: "Woman with ponytail running"



Tracking

Li et al. CVPR 2017

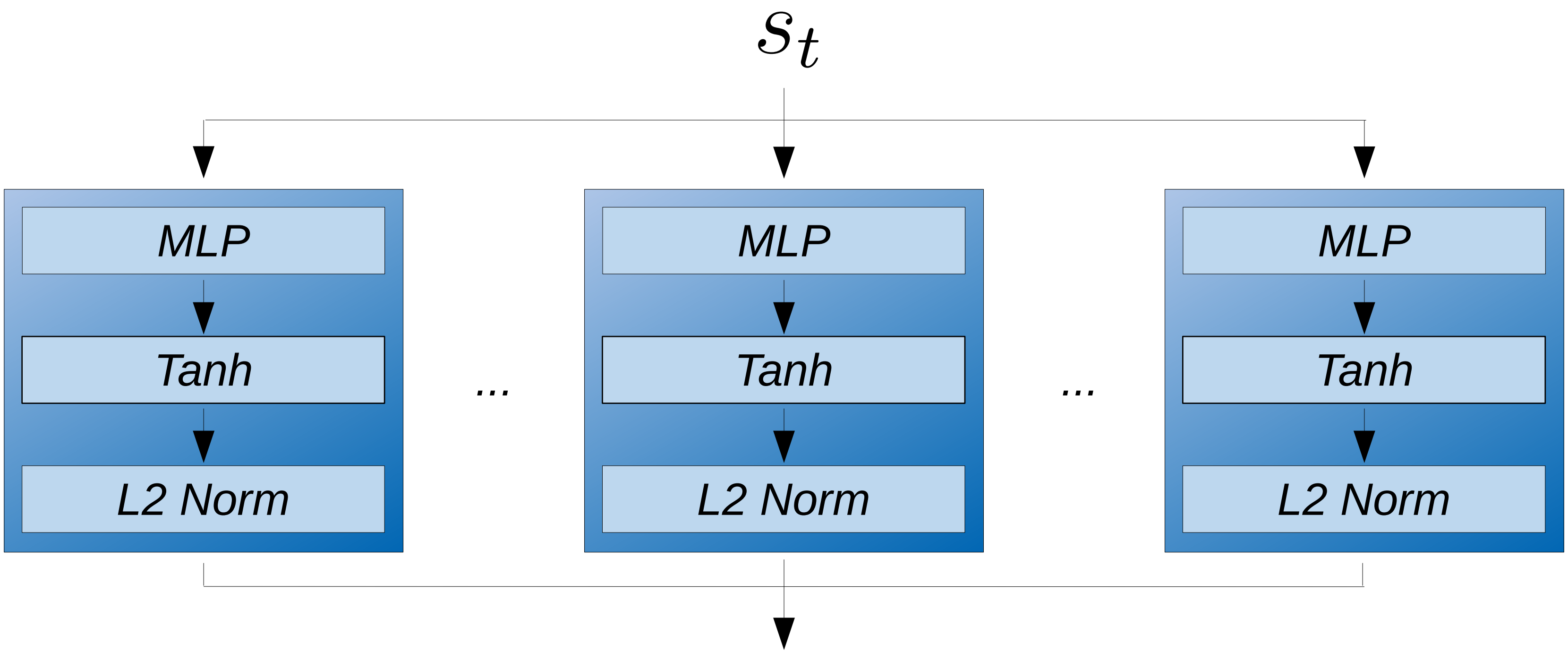
Query: "Small white fluffy puppy biting the cat"

Actor and Action Segmentation

Gavrilyuk et al. CVPR 2018

Dynamic convolutional filters

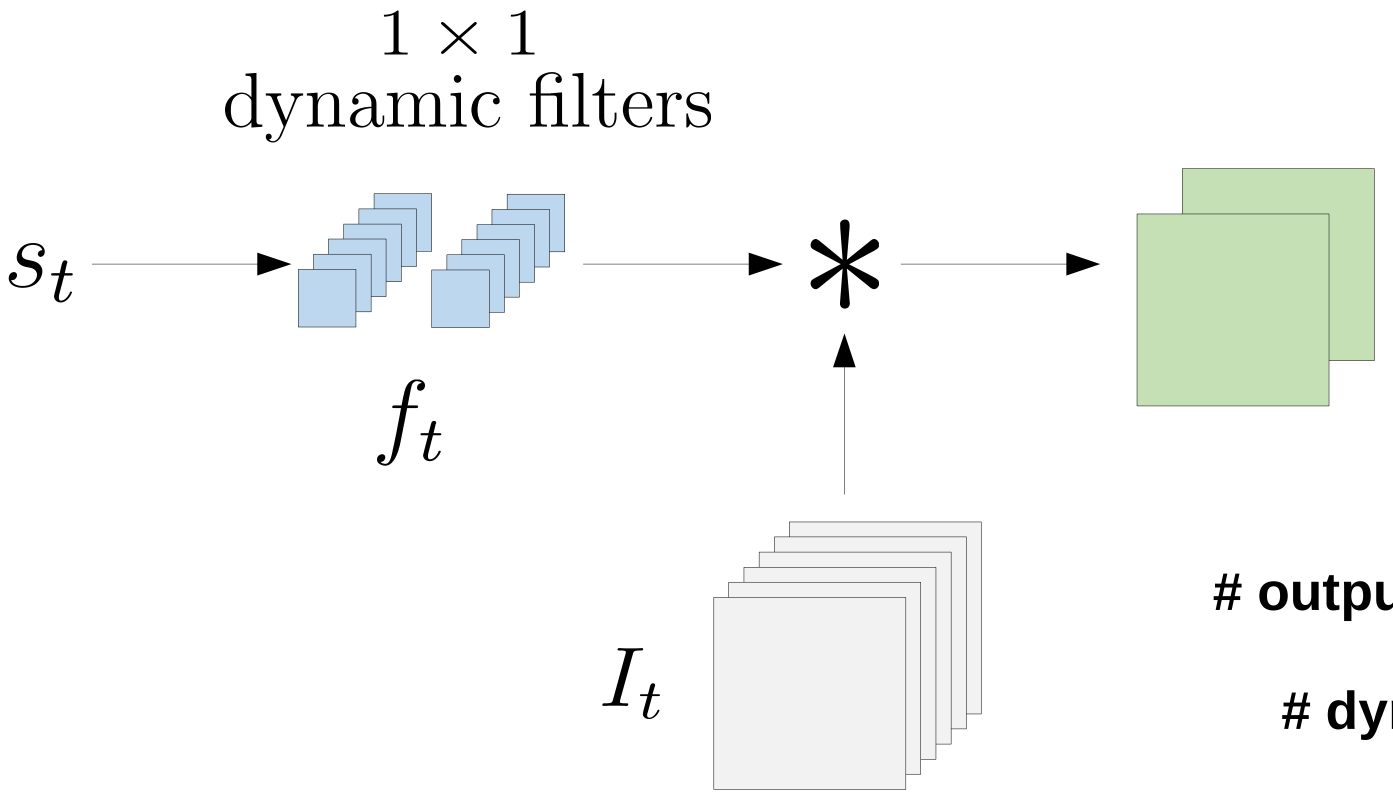
...or "let the sentence drive the convolution"



$$f_t^i = \ell_2[\tanh(W_f^i s_t + b_f^i)]$$

Dynamic convolutional filters

...or “let the sentence drive the convolution”

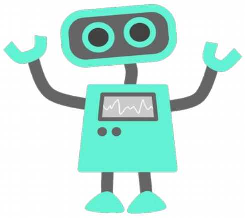


$$D_t = f_t * I_t$$

 # output feature maps
=
 # dynamic filters

Architecture

"Go straight then turn right and pass the many desks until you get to the ping pong table. Wait there."



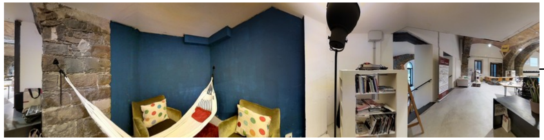
Instruction Encoder

Instruction Embedding

GloVe + LSTM

s_t

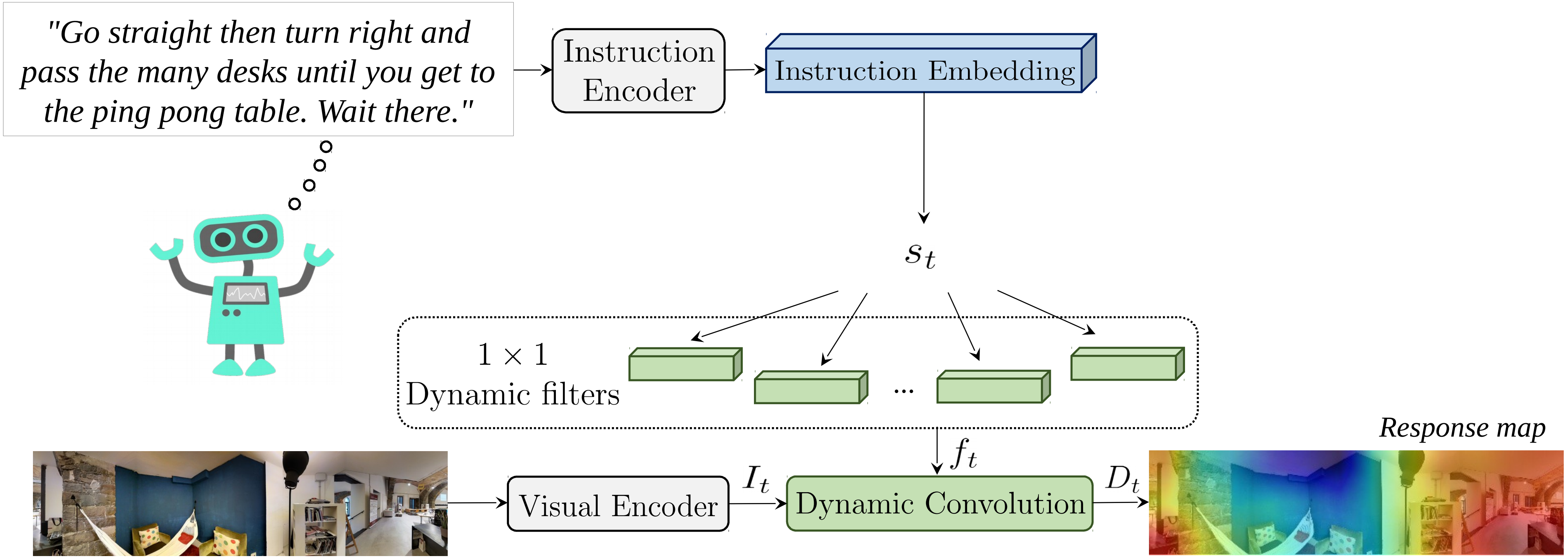
ResNet-152



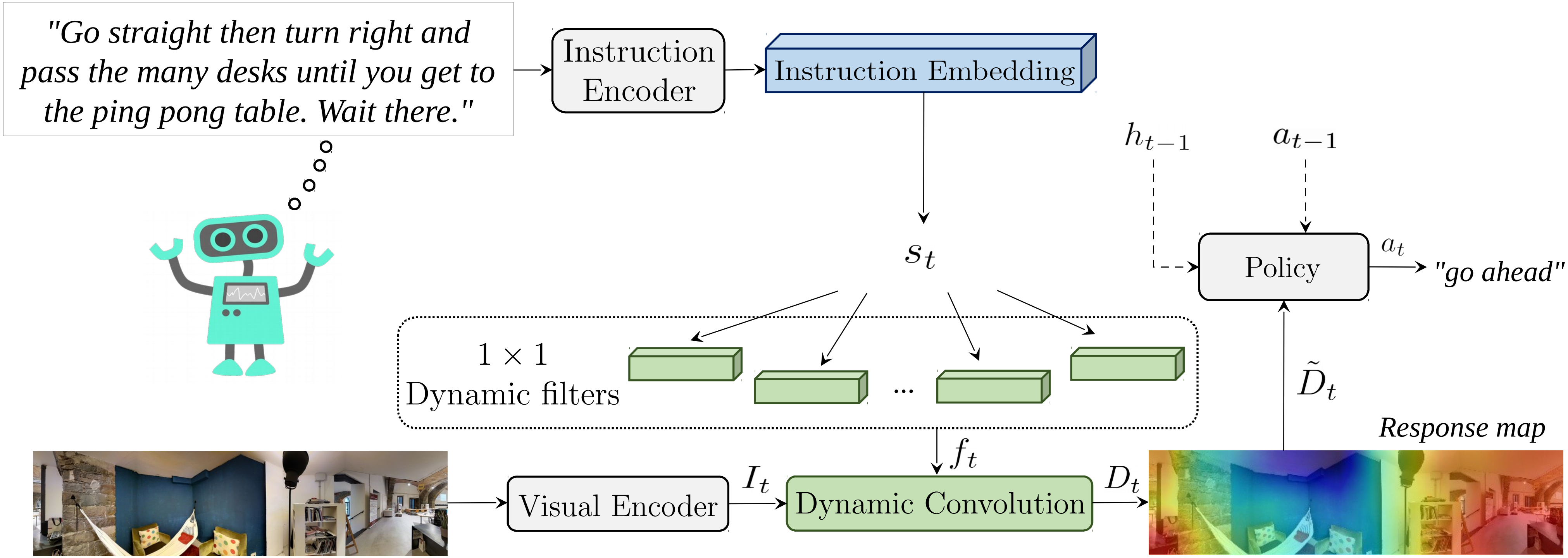
Visual Encoder

I_t

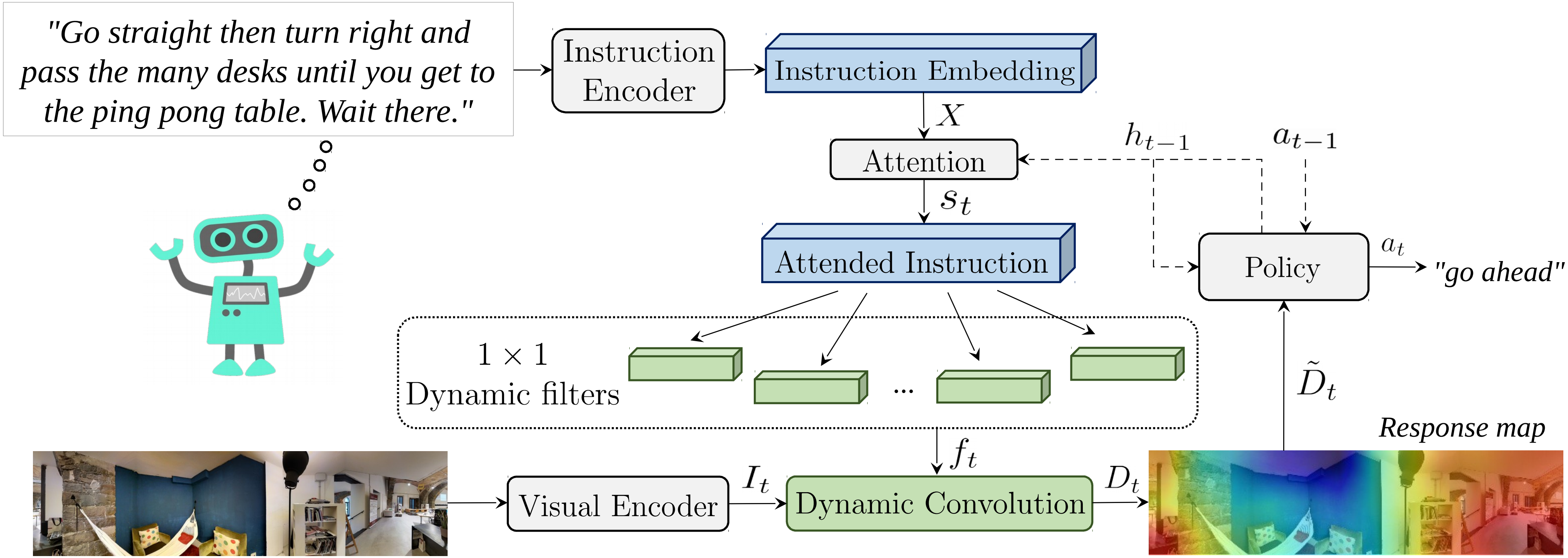
Architecture



Architecture

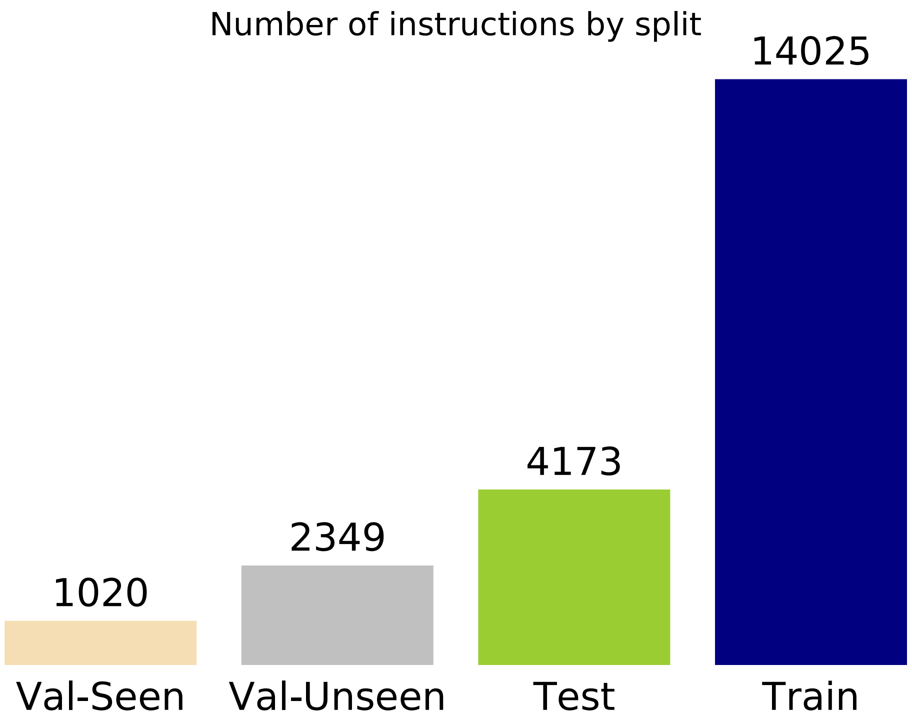


Architecture



Room-to-Room dataset (R2R)

- Builds upon Matterport3D dataset of spaces *(Chang et al. 3DV 2017)*
- 90 different buildings
- ~7k navigation paths
- 3 different descriptions / path
- ~29 words / instruction on average
- 2 different validation splits
- Test server with public leaderboard



R2R - Evaluation metrics

- **NE** (Navigation Error)
  distance between the agent final position and the goal

- **SR** (Success Rate)
  fraction of episodes terminated within 3 meters from the goal

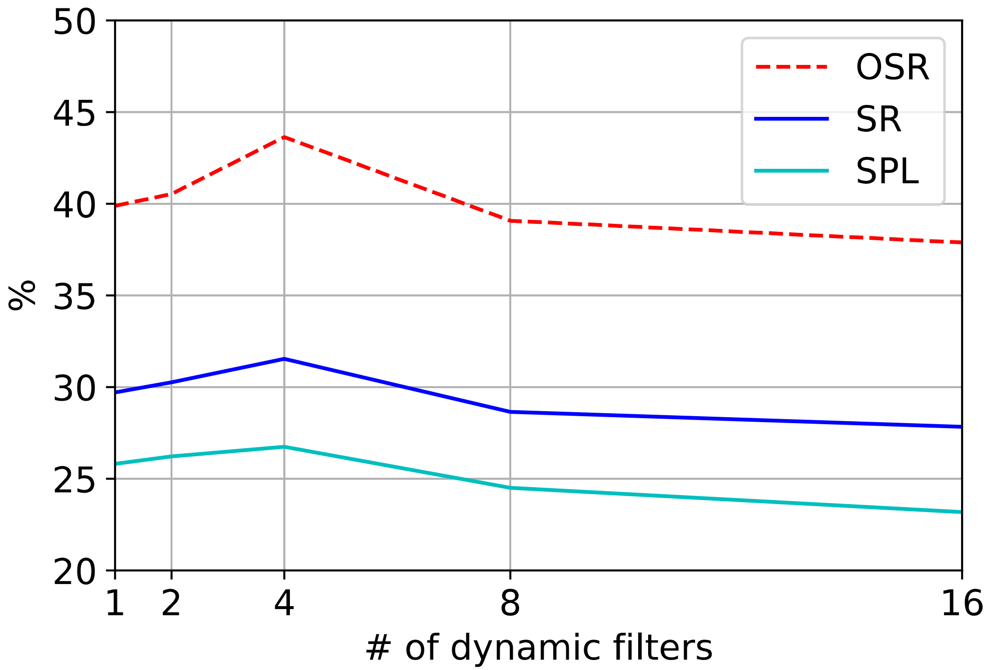
- **OSR** (Oracle SR)
  SR that the agent would have achieved if it received an oracle stop signal

- **SPL** (SR weighted by Path Length)
  SR weighted by normalized inverse path length (penalizes overlong navigations)

Number of dynamic filters

- How many dynamic filters do we need to encode meaningful information?
- The more the better?

# of filters	Validation-Unseen			
	NE ↓	SR ↑	OSR ↑	SPL ↑
1	6.79	29.7	39.9	25.8
2	6.77	30.3	40.5	26.2
4	**6.65**	**31.6**	**43.6**	**26.8**
8	7.19	28.7	39.1	24.5
16	7.03	27.8	37.9	23.2



Best results with four filters

One is enough to make things work well

Qualitative results

Instruction:

a) Take a right, going past the kitchen into the hallway

b) Walk into the sitting area and stop before the couch

Qualitative results

Instruction:

a) Take a right, going past the kitchen into the hallway

b) Walk into the sitting area and stop before the couch

Qualitative results

Instruction:

a) Take a right, going past the kitchen into the hallway

b) Walk into the sitting area and stop before the couch

Qualitative results

Instruction:

a) Take a right, going past the kitchen into the hallway

b) Walk into the sitting area and stop before the couch

Ablation study

Method	Validation-Seen NE ↓	SR ↑	OSR ↑	SPL ↑	Validation-Unseen NE ↓	SR ↑	OSR ↑	SPL ↑
Random agent	9.45	15.9	21.4	-	9.23	16.3	22.0	-
Baseline w/ traditional convolution	6.01	38.6	52.9	-	7.81	21.8	28.4	-
Ours w/o encoder-decoder attention	5.86	41.3	51.2	36.3	7.72	22.0	29.3	19.3
Ours w/o pre-trained embedding	5.62	42.0	54.0	36.3	7.32	25.8	33.3	22.1
Ours w/ dynamic filters	**4.68**	**53.1**	**66.1**	**46.0**	**6.65**	**31.6**	**43.6**	**26.8**

Every component contributes to the overall performance

→ **Dynamic convolution is the most valuable module**

Comparison with the State of the Art

Low-level Actions Methods	Validation-Seen				Validation-Unseen				Test (Unseen)			
	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑
Random	9.45	0.16	0.21	-	9.23	0.16	0.22	-	9.77	0.13	0.18	0.12
Student-forcing [1]	6.01	0.39	0.53	-	7.81	0.22	0.28	-	7.85	0.20	0.27	0.18
RPA [2]	5.56	0.43	0.53	-	7.65	0.25	0.32	-	7.53	0.25	0.33	0.23
Ours	4.68	0.53	0.66	0.46	6.65	0.32	0.44	0.27	7.14	0.31	0.42	0.27
Ours w/ data augmentation	**3.96**	**0.58**	**0.73**	**0.51**	**6.52**	**0.34**	**0.43**	**0.29**	**6.55**	**0.35**	**0.45**	**0.31**

State of the Art for low-level actions methods

[1] Anderson et al, CVPR 2018
[2] Wang et al, ECCV 2018

Comparison with the State of the Art

Low-level Actions Methods	Validation-Seen				Validation-Unseen				Test (Unseen)			
	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑
Ours	4.68	0.53	0.66	0.46	6.65	0.32	0.44	0.27	7.14	0.31	0.42	0.27
Ours w/ data augmentation	**3.96**	**0.58**	**0.73**	**0.51**	**6.52**	**0.34**	**0.43**	**0.29**	**6.55**	**0.35**	**0.45**	**0.31**

High-level Actions Methods	Validation-Seen				Validation-Unseen				Test (Unseen)			
	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑
Speaker-Follower [3]	3.36	0.66	0.74	-	6.62	0.36	0.45	-	6.62	0.35	0.44	0.28
Self-Monitoring [4]	**3.22**	0.67	**0.78**	0.58	5.52	0.45	0.56	0.32	5.99	0.43	0.55	0.32
RCM [5]	3.37	0.67	0.77	-	5.88	0.43	0.52	-	6.01	0.43	0.51	0.35
Regretful [6]	3.23	**0.69**	0.77	**0.63**	**5.32**	**0.50**	**0.59**	**0.41**	**5.69**	**0.48**	**0.56**	**0.40**

Competitive with high-level actions methods

→ **But direct comparison is not feasible**

[3] Fried et al, NeurIPS 2018
[4] Ma et al, ICLR 2019
[5] Wang et al, CVPR 2019
[6] Ma et al, CVPR 2019

Conclusion

- VLN is not simple. Do not add further complexity in the model
- Dynamic convolutional filters act as specialized and flexible feature extractors
- Different action spaces dramatically influence the results on R2R

→ be aware of that when making comparisons

Thank you!
federico.landi@unimore.it