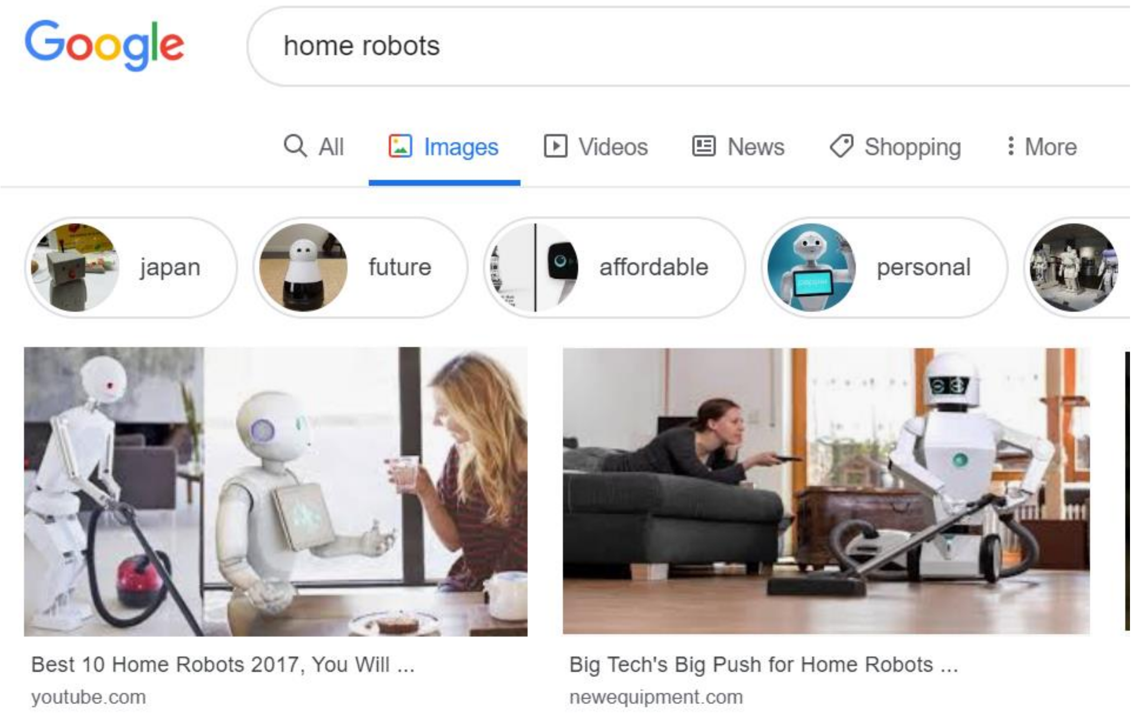
Do you want to have a home robot?

My mother does

Expectations:



Reality:

Jitendra Malik @ ECCV 2018 workshop:
“In robotics you take 25 PhD students and learn how to grasp a cup”

Embodied AI

Rebranding of “Robot”

Internet AI → Embodied AI

Physical agent
taking actions
in the world
talking to humans
in natural language

Is there smoke in any room around you?

Yes, in one room

Go there and look for people

...

Slide credit: Oleksandr Maksymets

Image Credit: Lockheed Martin; DARPA Robotics Challenge

Benchmarking Classic and Learned Navigation in Complex 3D Environments

Dmytro Mishkin, Alexey Dosovitskiy and Vladlen Koltun

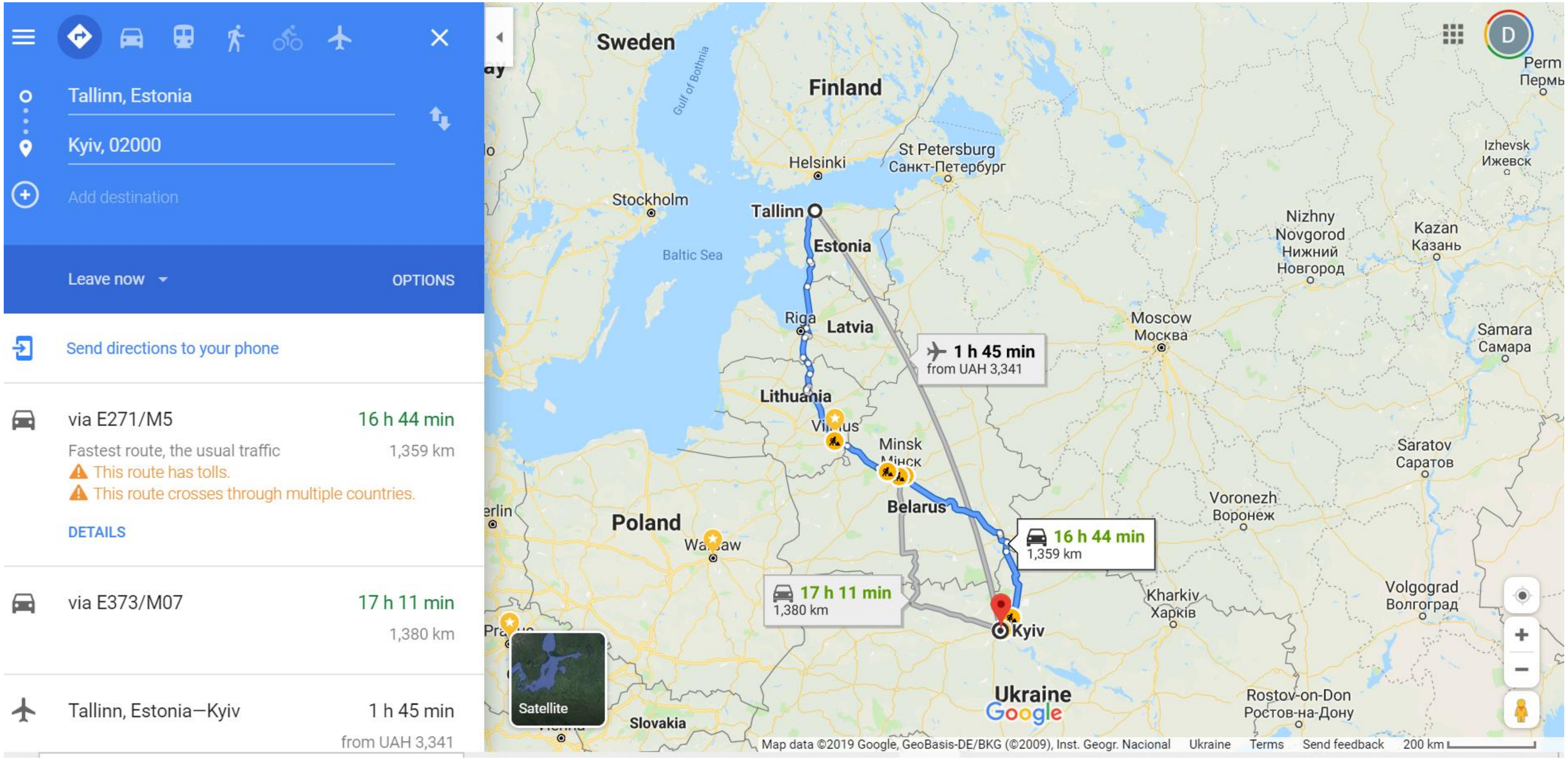
Why navigation?

Why navigation?

- Navigation is a necessary part for “embodied intelligence”:
  - Delivery
  - Home robots
  - Driving
  - basically, anything that should interact with physical world.

- It is cool, isn`t it?

Wait, but isn`t navigation solved already?



Learning to navigate in cities without a map

Impressive work by Mirowski et.al (2018):

+ Neural network navigates in the city based on image-only input + location beacons.

+ Large scale, efficient

- Environment (city) is known
- Environment (city) is fixed

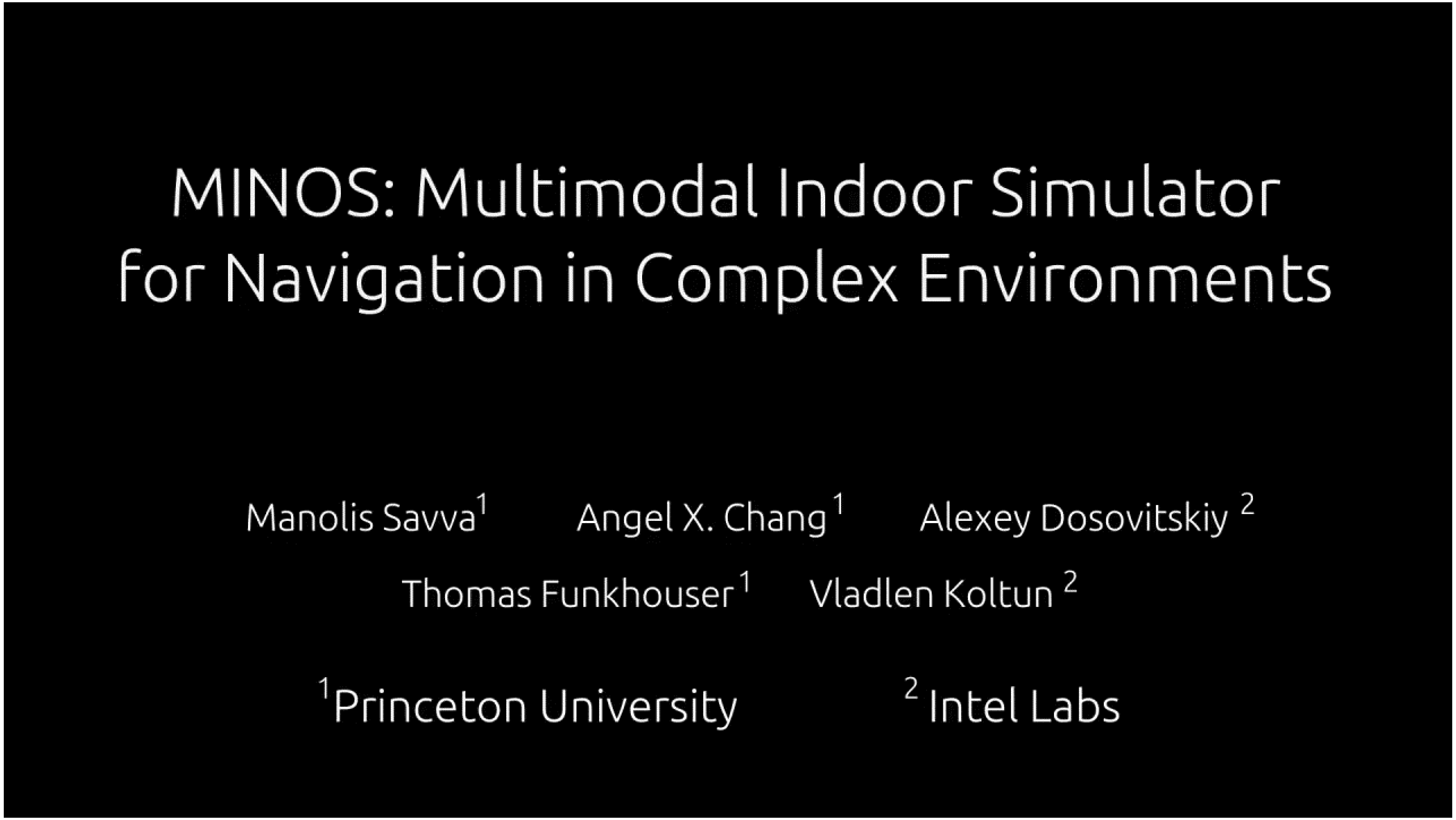
https://deepmind.com/blog/article/learning-to-navigate-cities-without-a-map

No, navigation is far from being “solved”

- Real-world solutions (google maps, driving assistants, etc) typically use a lots of sensors, GPS. Nobody sane uses just RGB or RGBD in production.
- Even more important – current solutions need infrastructure, e.g. beacons, pre-built maps.
- Or may be robotics research HAVE solved it…but never open-sources code or evaluation

Navigation tasks

Examples of basic navigation tasks



Let’s start with the basic

Point goal task

Point Goal

- You wake up in unknown house
- The invisible “goal” is sending you its relative coordinates
- You have a limited amount of time (actions) to find it

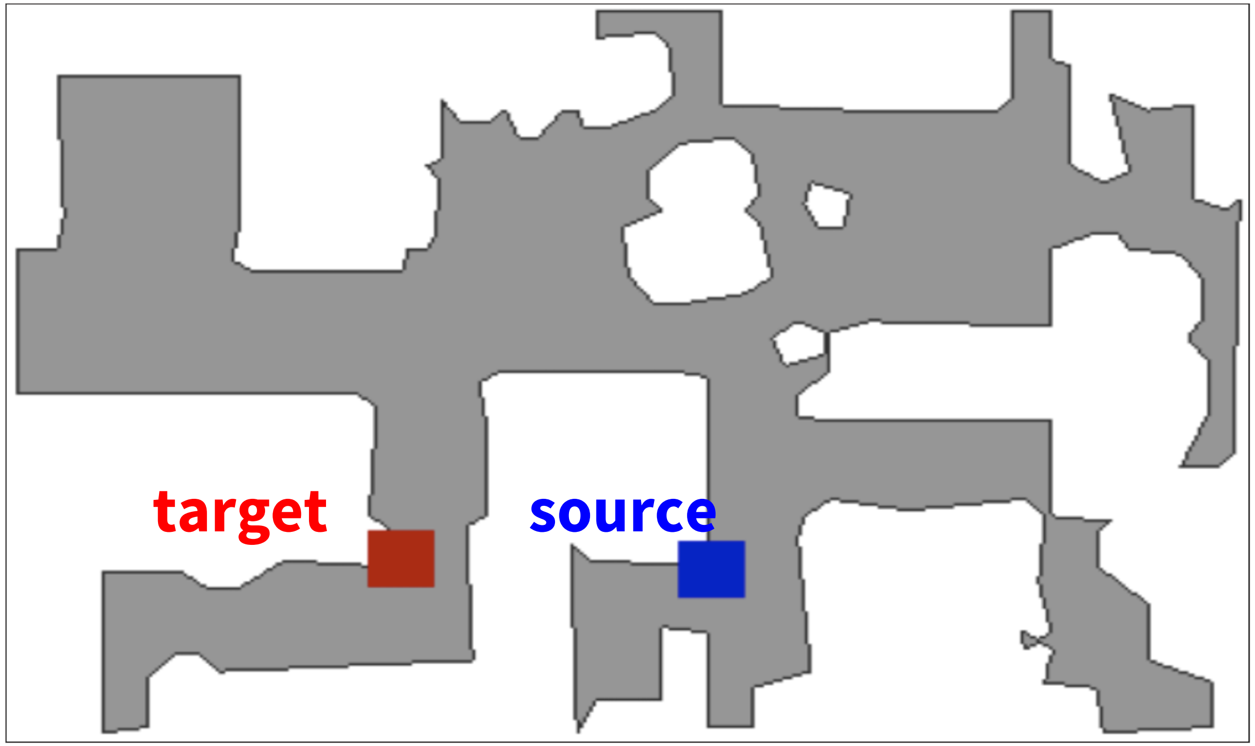
Point Goal

- You wake up in unknown house
- The invisible “goal” is sending you its relative coordinates
- You have a limited amount of time (actions) to find it

I want to play a navigation game

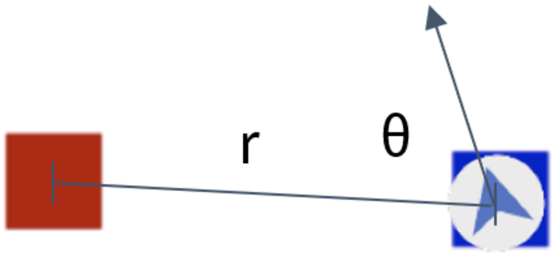
Point Goal

- You wake up in unknown house
- The invisible "goal" is sending you its relative coordinates
- You have a limited amount of time (actions) to find it



Input to the agent:

Distance and direction to the goal



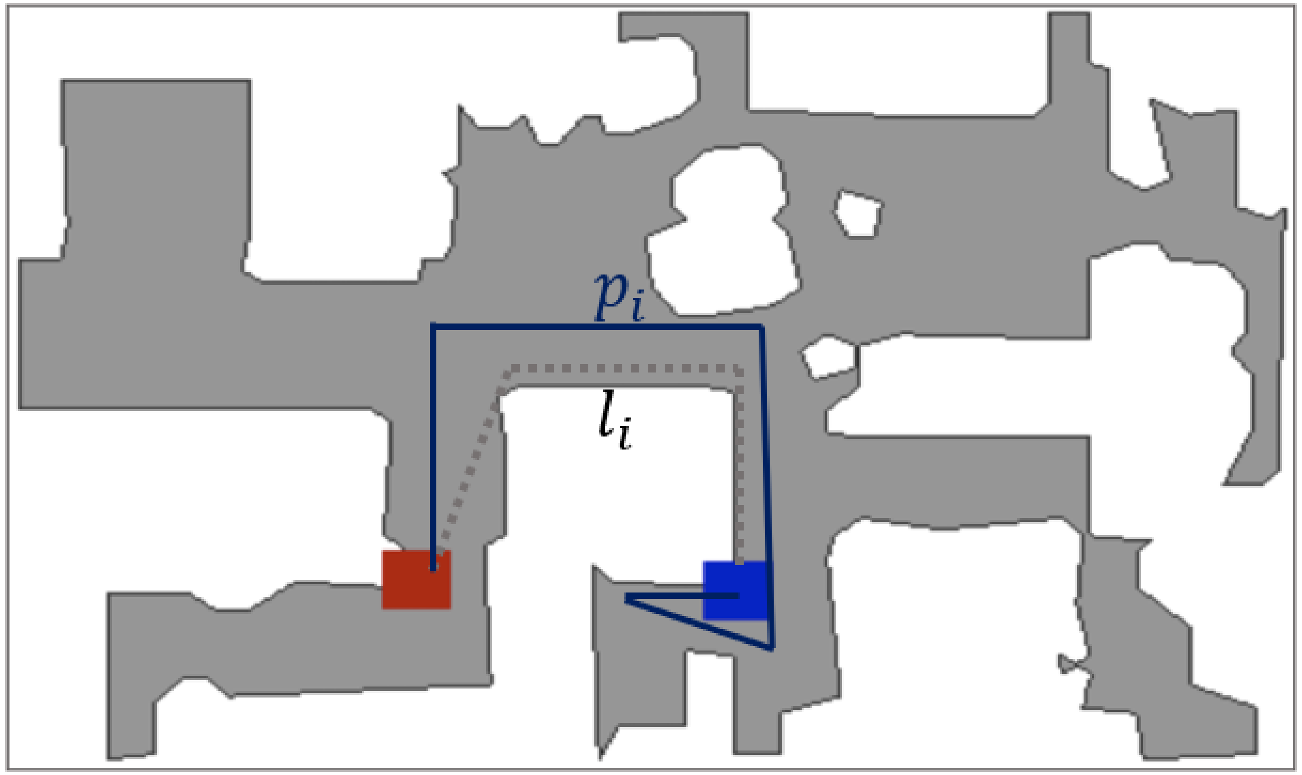
Slide credit: Abhishek Kadian

PointGoal Task: Action Space

1. Turn-left (impulse of acceleration)
2. Turn-Right (impulse of acceleration)
3. Move-Forward (impulse of acceleration)
4. Stop

Slide credit: Abhishek Kadian

Success weighted by Path Length (SPL)



$$SPL = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i}{\max(p_i, l_i)}$$

l_i : length of shortest path between goal and target for episode

p_i: length of path taken by agent in episode

S_i: binary indicator of success in episode

Slide credit: Abhishek Kadian

Current approaches to navigation

Classic: handcrafted SLAM + planner + something. Works? Nobody reproducibly tested.

Naively learned via RL: does not work in novel complex environments

Hybrid. Reformulate classical components into NNs and learn them: most of recent publications

VIN, GPPN, CogMap, MapNet

A*, ORB-SLAM

A3C, UNREAL, DFP

Current trend

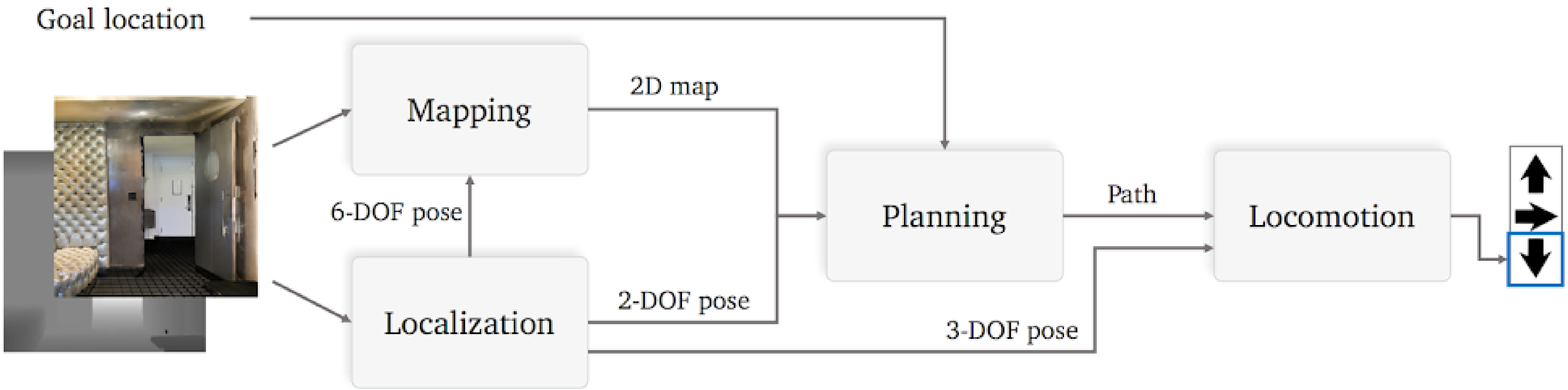
Classic

Learned

Problems with handcrafted solutions

- There is no open-source ready solution to just run in simulation.
- Moreover, most of the “classical” solutions need lots of parameter tuning, which are not stated in papers
- Each sensor setup is unique, which makes fair comparison hard

Our classical agent: ORBSLAM2 + D*-lite



- ORB-SLAM2 RGB-D localization
- Depth-based mapper
- D*-lite path planner
- Rule-based locomotion module

Mishkin et.al, [Benchmarking Classic and Learned Navigation in Complex 3D Environments](#) arXiv 2019

Depth mapper

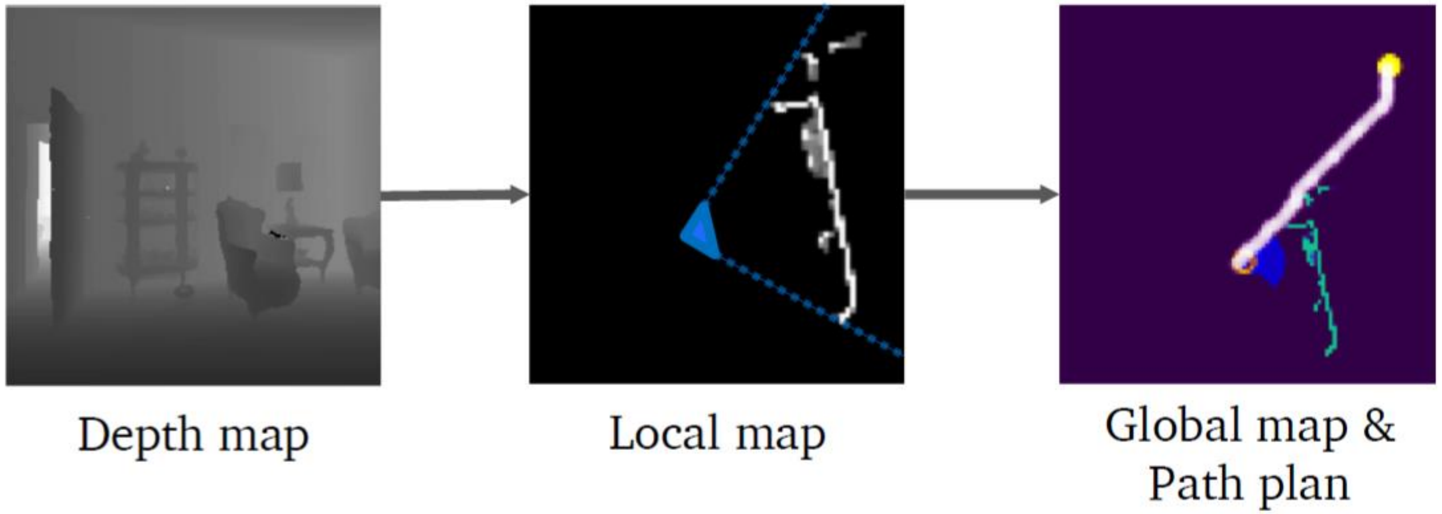
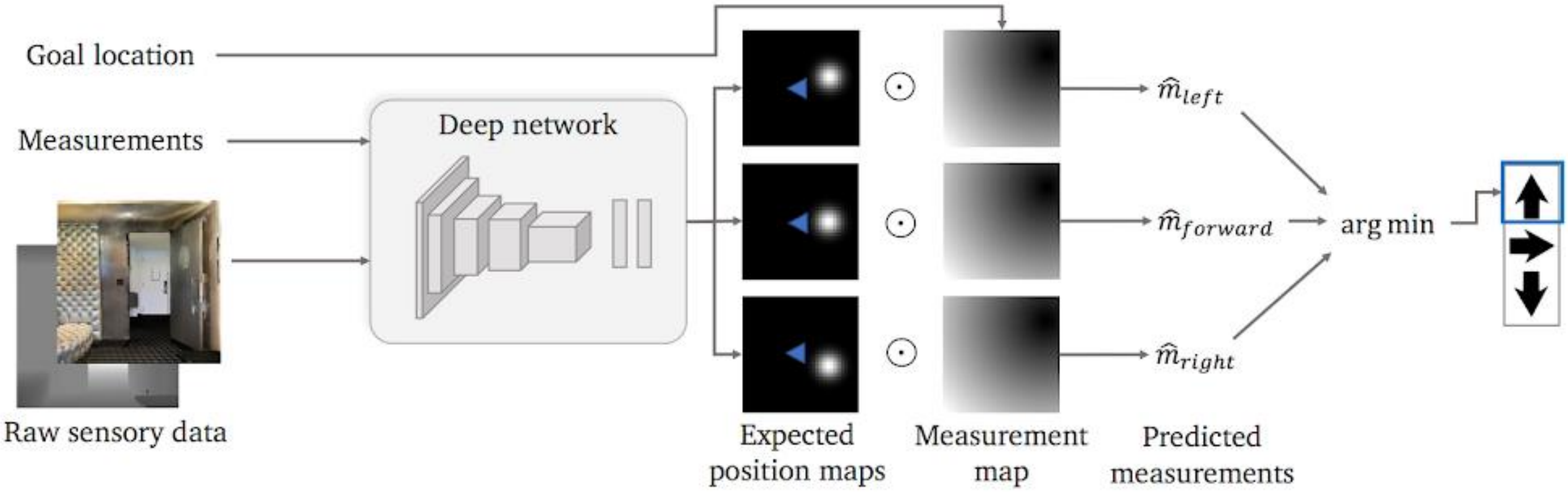


Figure 3. Illustration of the mapping and planning modules of the classic pipeline. A local map is generated by reprojecting the depth map to the top view and binning. Using the estimated agent's pose, the local map is integrated into the global map. The global map is used to plan a path towards the goal.

Mishkin et.al, [Benchmarking Classic and Learned Navigation in Complex 3D Environments](#) arXiv 2019

Our RL-agent: Belief - Direct Future Prediction



Mishkin et.al, [Benchmarking Classic and Learned Navigation in Complex 3D Environments](#) arXiv 2019

Naive learned agents: fail in realistic scenarios

Environment				Agent				
Task	Dataset	Clutter	Size	Random	A3C-FF	A3C-LSTM	DFP	UNREAL
PointGoal	SUNCG	Empty	Small	23.8	10.1	69.1	**80.3**	72.9
PointGoal	SUNCG	Empty	Medium	8.6	7.2	57.4	**64.1**	63.2
PointGoal	SUNCG	Furnished	Small	9.5	16.1	60.9	**64.5**	64.1
PointGoal	SUNCG	Furnished	Medium	6.3	7.9	41.3	43.6	**45.3**
PointGoal	Matterport3D	Furnished	Small	0.0	2.0	32.0	27.3	**38.0**
PointGoal	Matterport3D	Furnished	Medium	0.0	2.0	0.0	18.2	**20.0**
RoomGoal	SUNCG	Furnished	Small	10.0	25.7	30.0	22.5	**58.6**
RoomGoal	SUNCG	Furnished	Medium	3.1	6.9	7.2	4.0	**32.0**
RoomGoal	Matterport3D	Furnished	Small	5.0	12.0	**14.0**	13.6	**14.0**

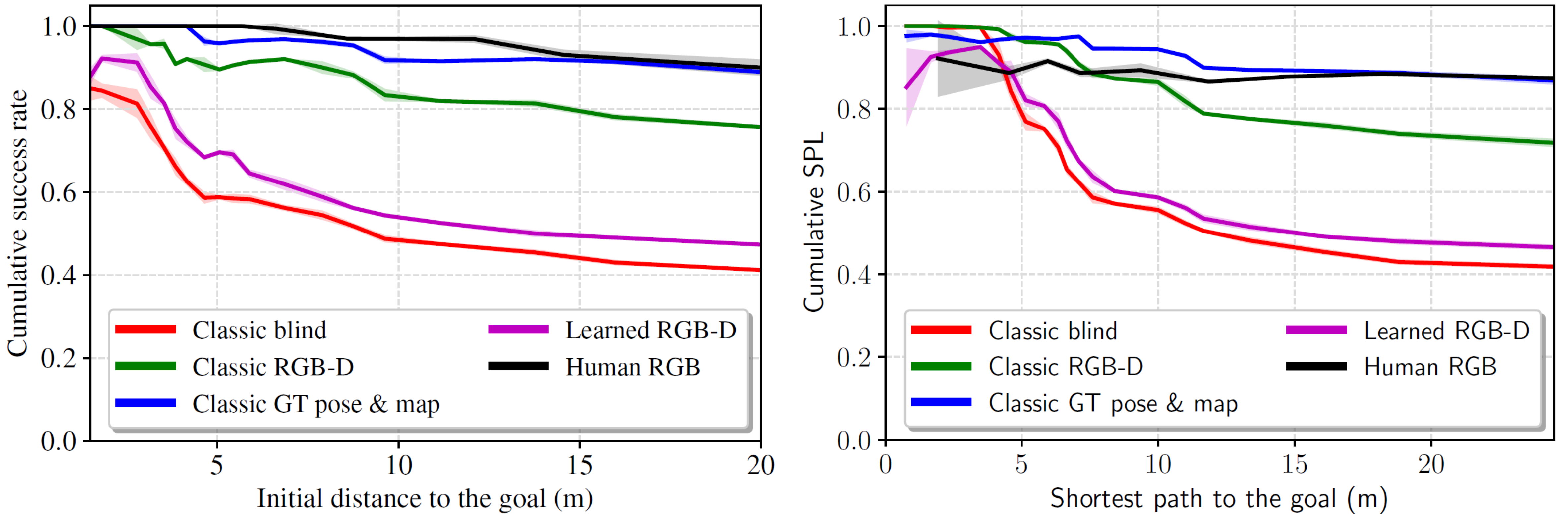
Table 2: Average episode success rate for agents trained on PointGoal and RoomGoal tasks, tested on novel environments of varying complexity. SUNCG 'Small' refers to two-room houses while SUNCG 'Medium' contains three-to-five-room houses. Matterport3D 'Small' contains environments with up to 10 rooms, and Matterport3D 'Medium' refers to environments with up to 24 rooms. Note that all agents exhibit significant performance degradation as the size and complexity of the environment increases.

Savva et al. MINOS: Multimodal Indoor Simulator for Navigation in Complex Environments, arXiv 2017

Overview

Mishkin et.al, [Benchmarking Classic and Learned Navigation in Complex 3D Environments](#) arXiv 2019

Handcrafted (w RGB-D) >> Vanilla RL



Mishkin et.al, [Benchmarking Classic and Learned Navigation in Complex 3D Environments](#) arXiv 2019

Handcrafted (w RGB) << Vanilla RL

Table 2. Comparison of learned navigation approaches in MINOS. We report average success rate. Our DFP implementation generally performs on par with previously reported results. Belief DFP slightly outperforms DFP in most environments. *The exact set of environments and start-goal pairs used for evaluation on Matterport3D is not identical, which may account for part of the performance difference.

	SunCG Empty		SunCG Furnished		Matterport3D		Mean	
	Classic	Learned	Classic	Learned	Classic	Learned	Classic	Learned
Blind	52.3	**52.9**	**44.4**	13.8	**40.9**	27.7	**45.9**	31.4
RGB	9.4	**49.9**	4.4	**44.8**	4.7	**37.4**	6.1	**44.0**
RGB-D	**65.7**	54.6	**67.7**	47.2	**70.2**	45.5	**67.9**	49.1
RGB-D + MINOS pose	82.5	–	84.5	–	78.8	–	82.0	–
RGB-D + MINOS pose & map	81.8	–	81.5	–	85.1	–	82.8	–

Mishkin et.al, Benchmarking Classic and Learned Navigation in Complex 3D Environments arXiv 2019

RGB + (predicted) depth = PROFIT

SLAM input	SunCG-E	SunCG-F	M3D
RGB	9.4	4.4	4.7
RGB + MonoDepth [18]	43.8	47.3	37.5
RGB + StereoDepth [6]	30.2	34.7	47.3
RGB + GT Depth	65.7	67.7	70.2

The main problem of the estimated depth: error accumulation.
Might be handled by Bayesian setup or learned mapper.

Mishkin et.al, Benchmarking Classic and Learned Navigation in Complex 3D Environments arXiv 2019

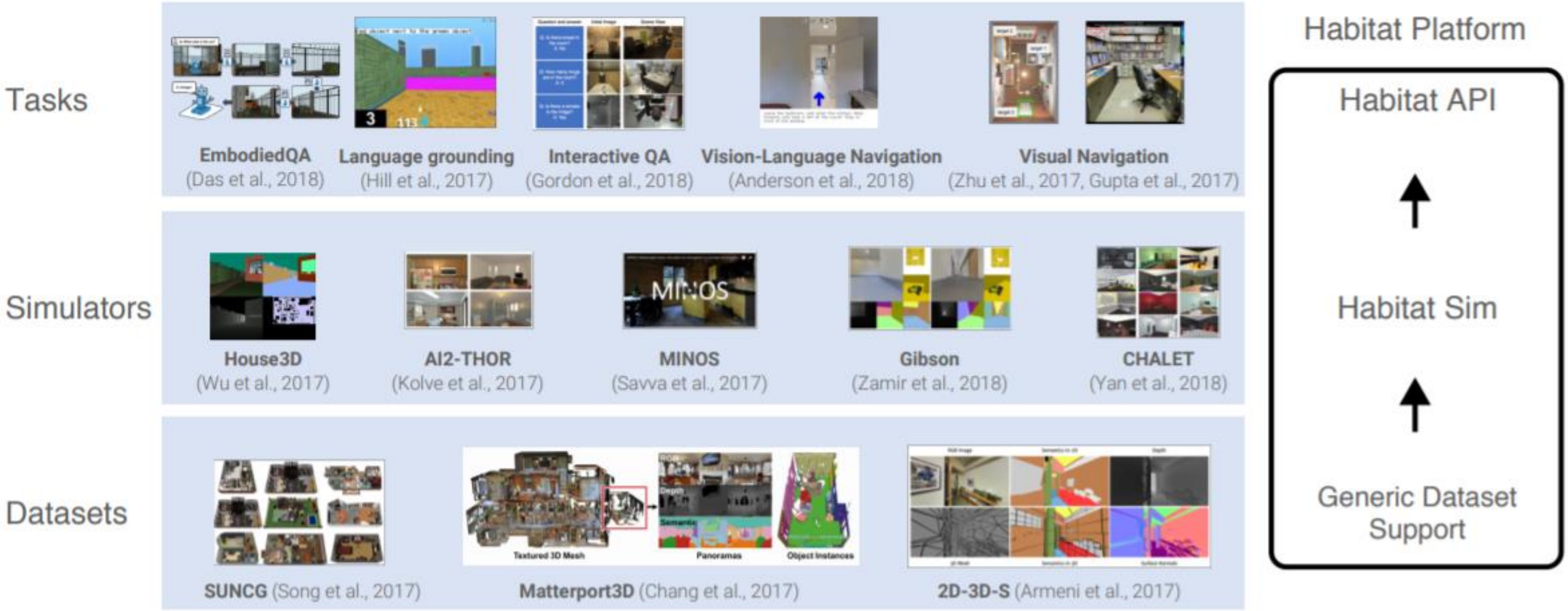
What about humans? It depends

Success rate, test split

Human	SunCG Furnished	Matterport 3D
Paper co-author	100 %	95 %
Gamer	100 %	84 %
Non-gamer	88 %	48 %
RGBD-SLAM	67%	70%

- Fine control is not easy
- Hard to interpret (angle + distance) w/o visible goal

Habitat: new stack for indoor navigation



https://github.com/facebookresearch/habitat-sim

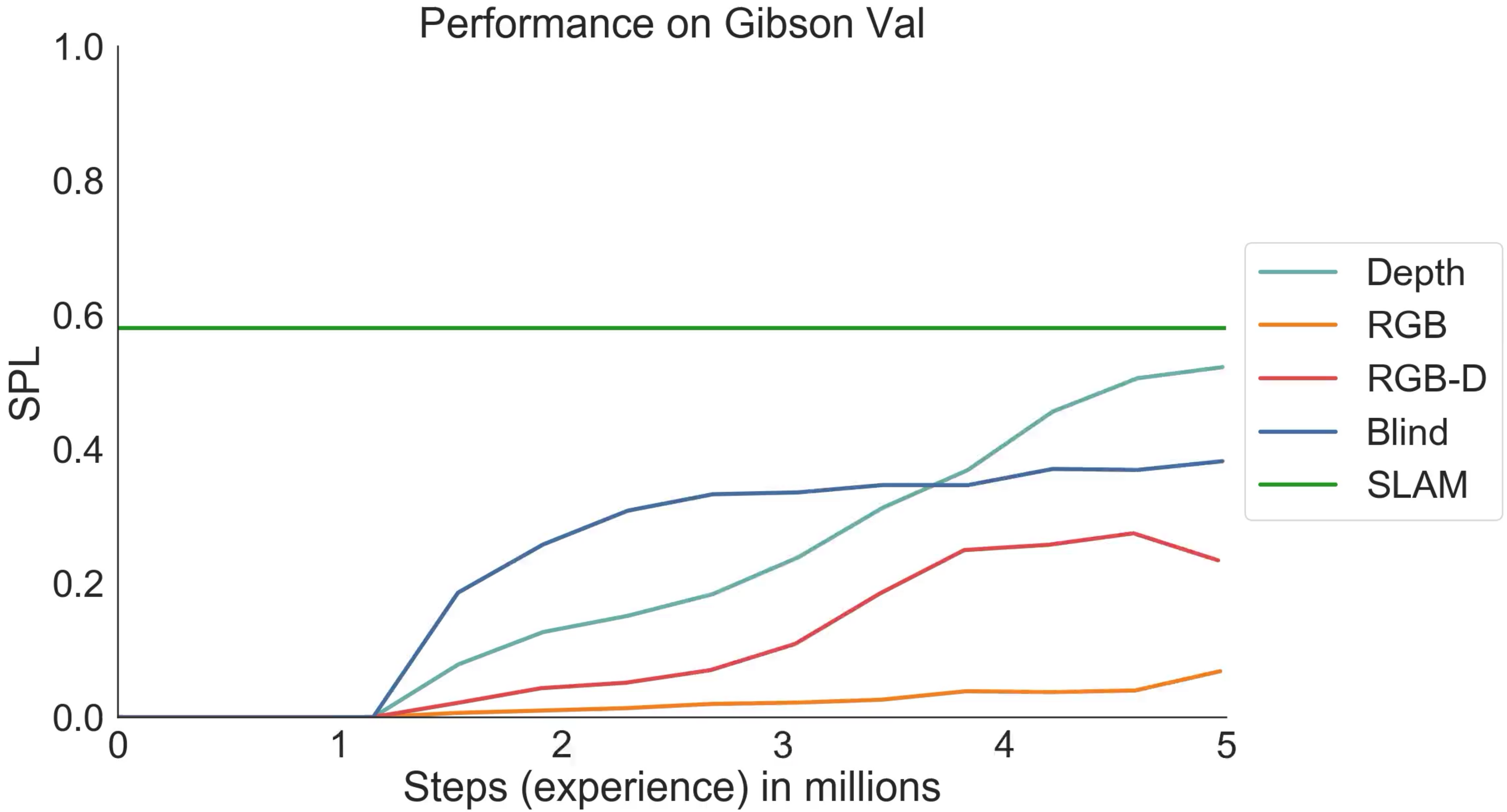
Habitat-sim: fast and scalable training

	1 proc			5 procs		
Sensors / Resolution	128	256	512	128	256	512
RGB	4,093	1,987	848	10,592	3,574	2,629
RGB + depth	2,050	1,042	423	5,223	1,774	1,348

Table 1: Performance of `Habitat-Sim` in frames per second for an example Matterport3D scene (id 17DRP5sb8fy) on an Intel Xeon E5-2690 v4 CPU and Nvidia Titan Xp GPU, measured at different frame resolutions and with a varying number of concurrent simulator processes sharing the GPU.

Renders faster than loading images from SSD!

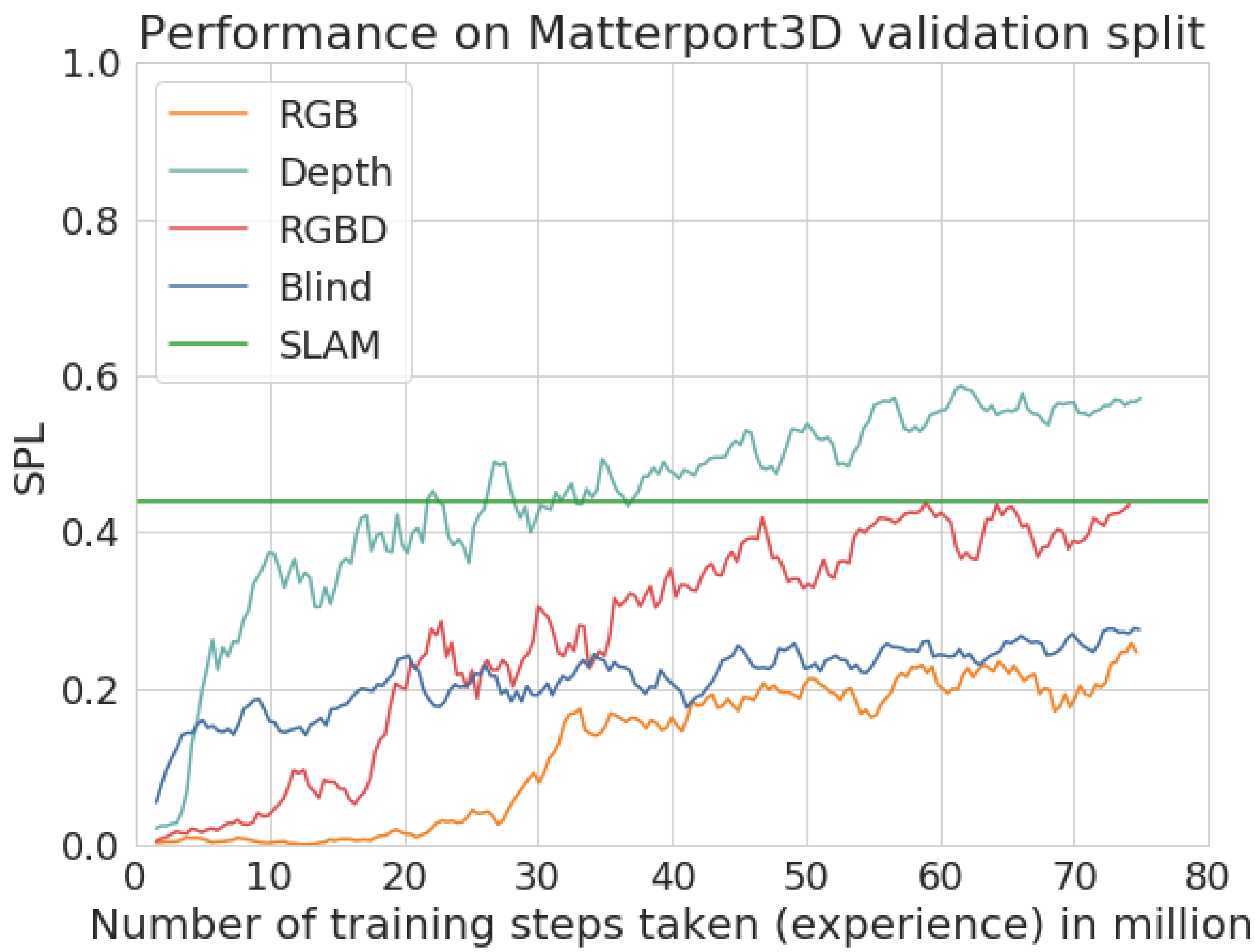
Savva et al. [Habitat: A Platform for Embodied AI Research](), ICCV 2019
https://github.com/facebookresearch/habitat-sim



Savva et al. Habitat: A Platform for Embodied AI Research, ICCV 2019

Slide credit: Oleksandr Maksymets

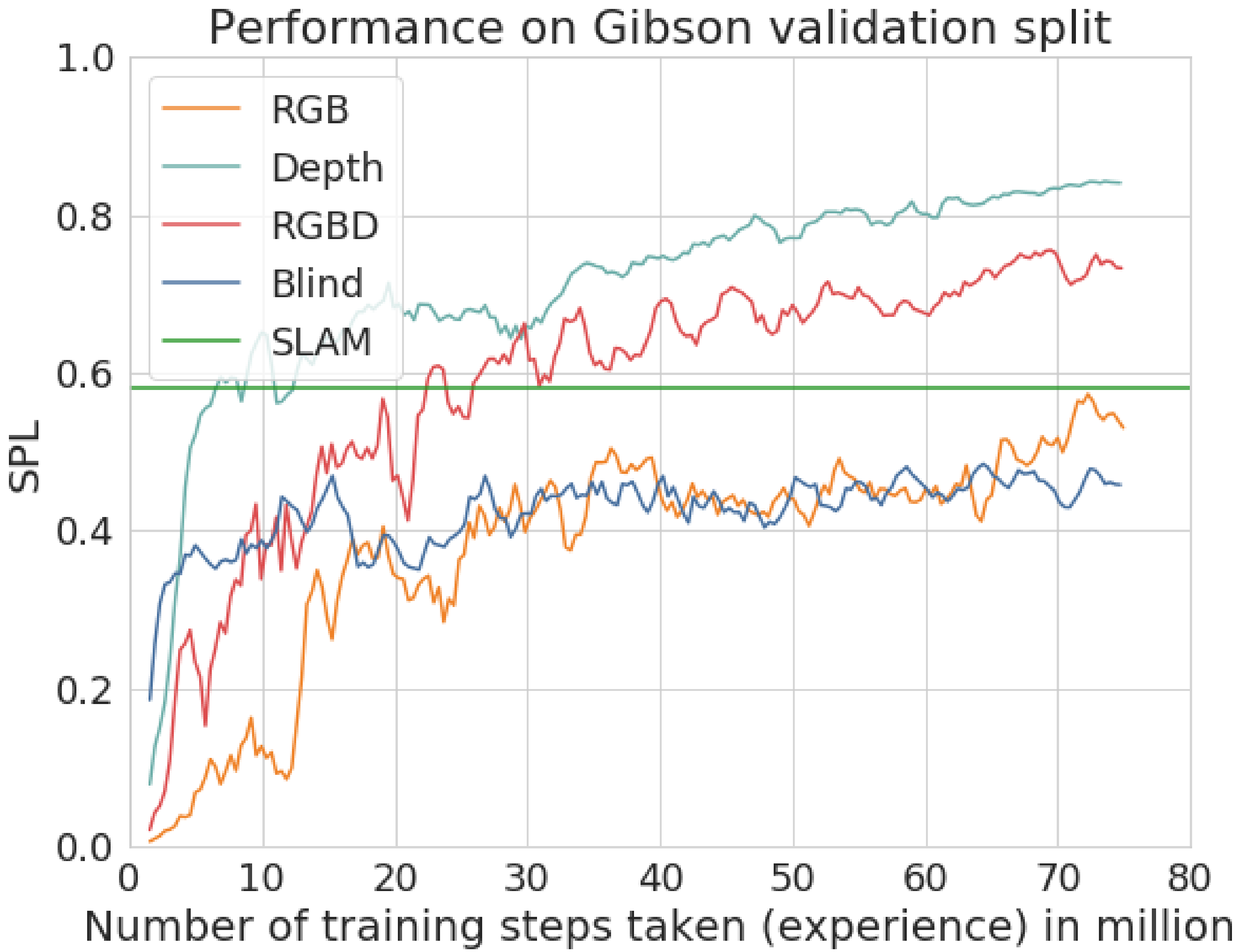
Habitat: PPO RL trained longer and faster



- Depth-only is much better than RGB-D
- RL is good, when you have huge training budget: > 50M steps

Savva et al. Habitat: A Platform for Embodied AI Research, ICCV 2019

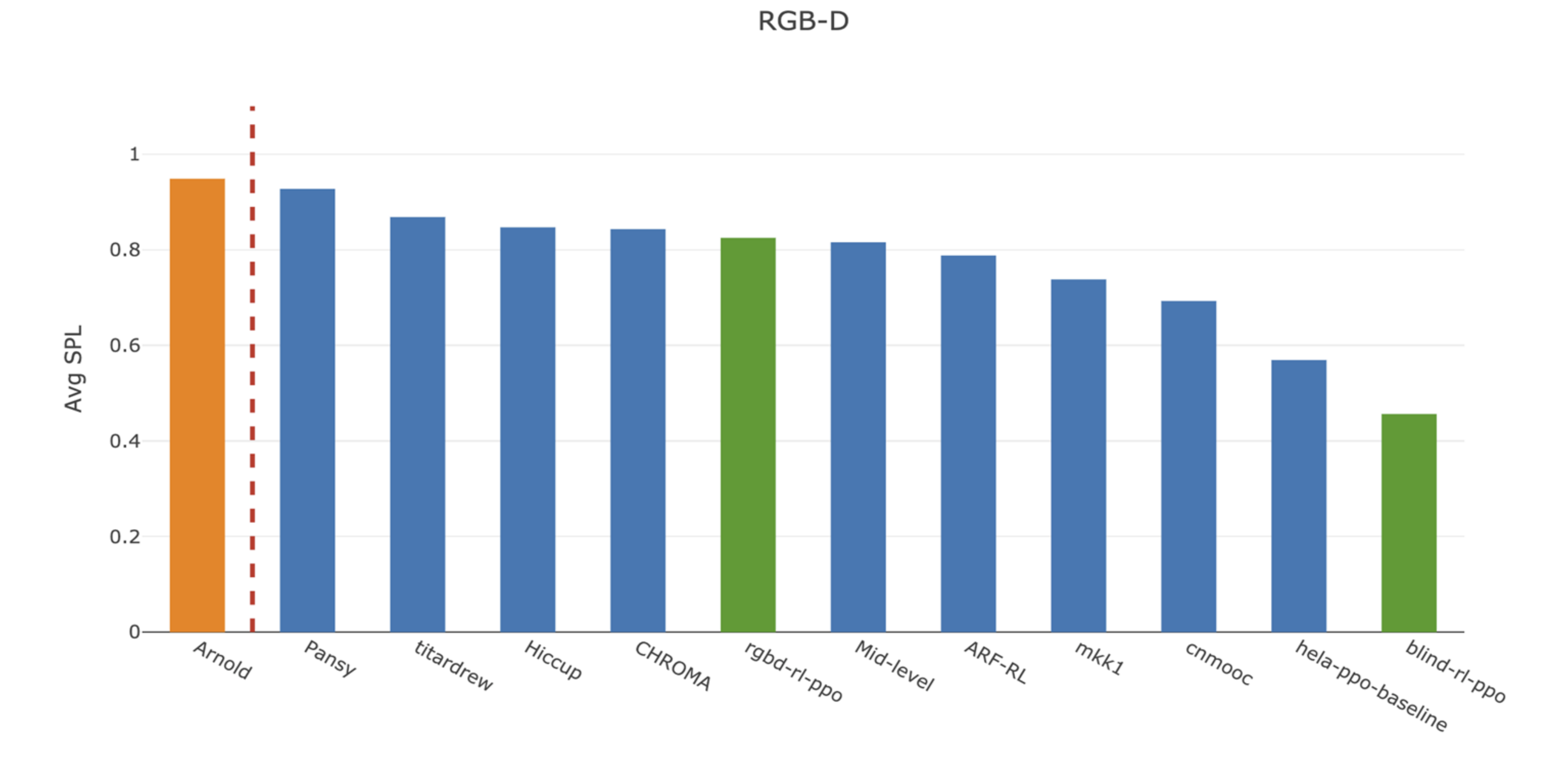
Habitat: RL trained longer and faster



- Less obstacles ⟶ better for RL
- Depth-only still better than RGB-D

Savva et al. [Habitat: A Platform for Embodied AI Research](), ICCV 2019

Habitat Challenge (CVPR 2019) results



Winner: Arnold,
Learned components in structured framework

https://evalai.cloudcv.org/web/challenges/challenge-page/254

Arnold: combination of classic and learned

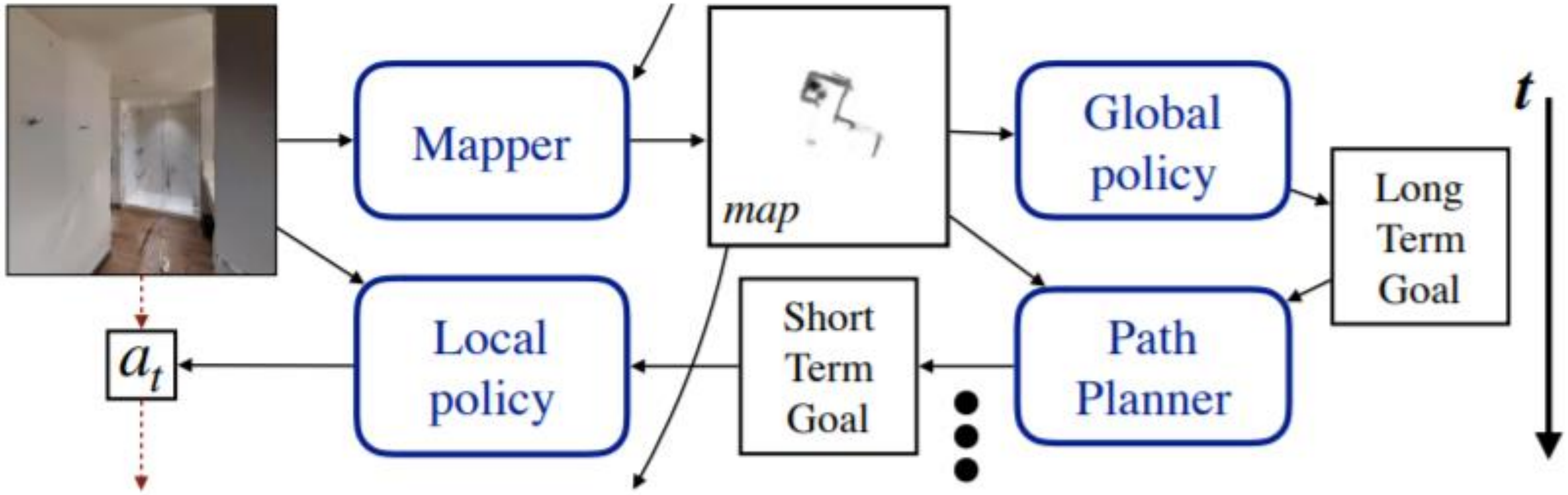


Figure 1: Overview of our approach. We use a neural network based mapper that predicts a map from incoming RGB observations. This map is used by a task-specific global policy to output a long term goal. An analytic path planner computes a plan to achieve the long term goal. This plan is used to generate a short term goal. This short-term goal is used in conjunction with the current image to output actions that the agent should execute.

Chaplot et.al [Modular Visual Navigation using Active Neural Mapping](#) CVPRW2019

Arnold: combination of classic and learned

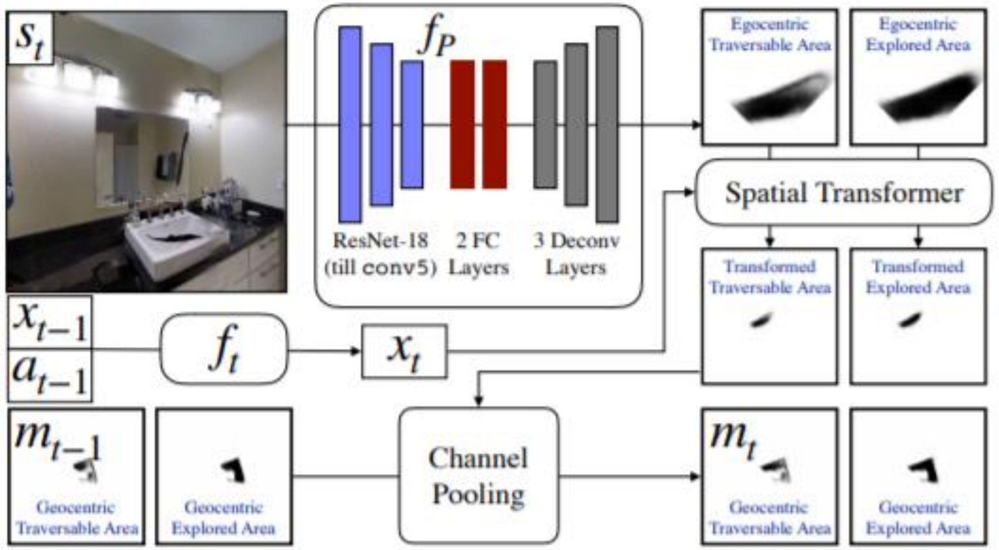


Figure 2: Architecture of the mapper: An encoder decoder architecture predicts an egocentric map from first person RGB images. This is transformed using a spatial transformer to update the geocentric map m_t.

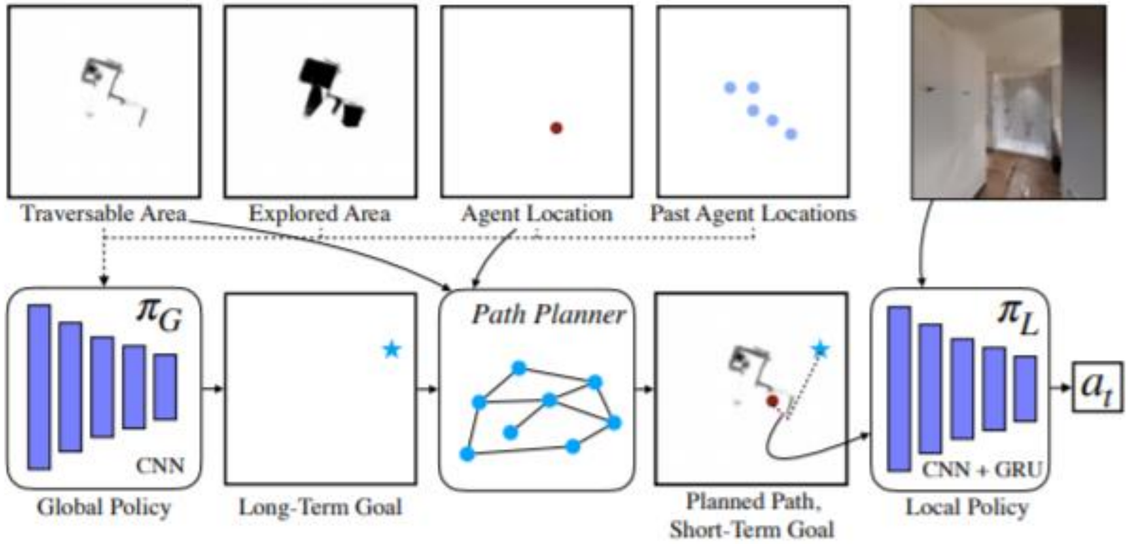


Figure 3: Information flow between global policy, path planner and local policy. Global policy takes in the maps to output a long-term goal. An analytic path planner computes a path to the long-term goal, to output a short term goal. This is used by the Local policy to output actions.

Chaplot et.al [Modular Visual Navigation using Active Neural Mapping](#) CVPRW2019

Conclusions

- Handcrafted structure + learned components is a way to go
- Fast simulator is a key for RL success
- If you need RL – use PPO (proximal policy optimization), not others.
  PPO is also the base for OpenAI Dota bots - Five
- It is much better to estimate depth and use it as an input, than hope that CNN will figure it out itself
- Humans are not perfect: one should estimate human baseline
- https://sites.google.com/view/classic-vs-learned-navigation

Thank you and let’s meet

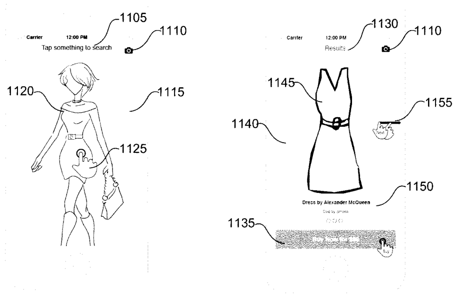
Dmytro Mishkin

ducha.aiki@gmail.com

Industry

- CTO of Clear Research (2014-2017)
  AI-based shopping app

Consulting: 2018 - now

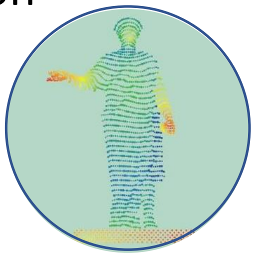


Research

- CTU in Prague. Marrying computer vision and deep learning

Community building

- Co-founder of Eastern Europe Computer Vision Conference EECVC (2018 – now)
- Co-founder of Ukrainian Research Group “Szkocka” (2016 – now)



Competitive ML

- Kaggle Master