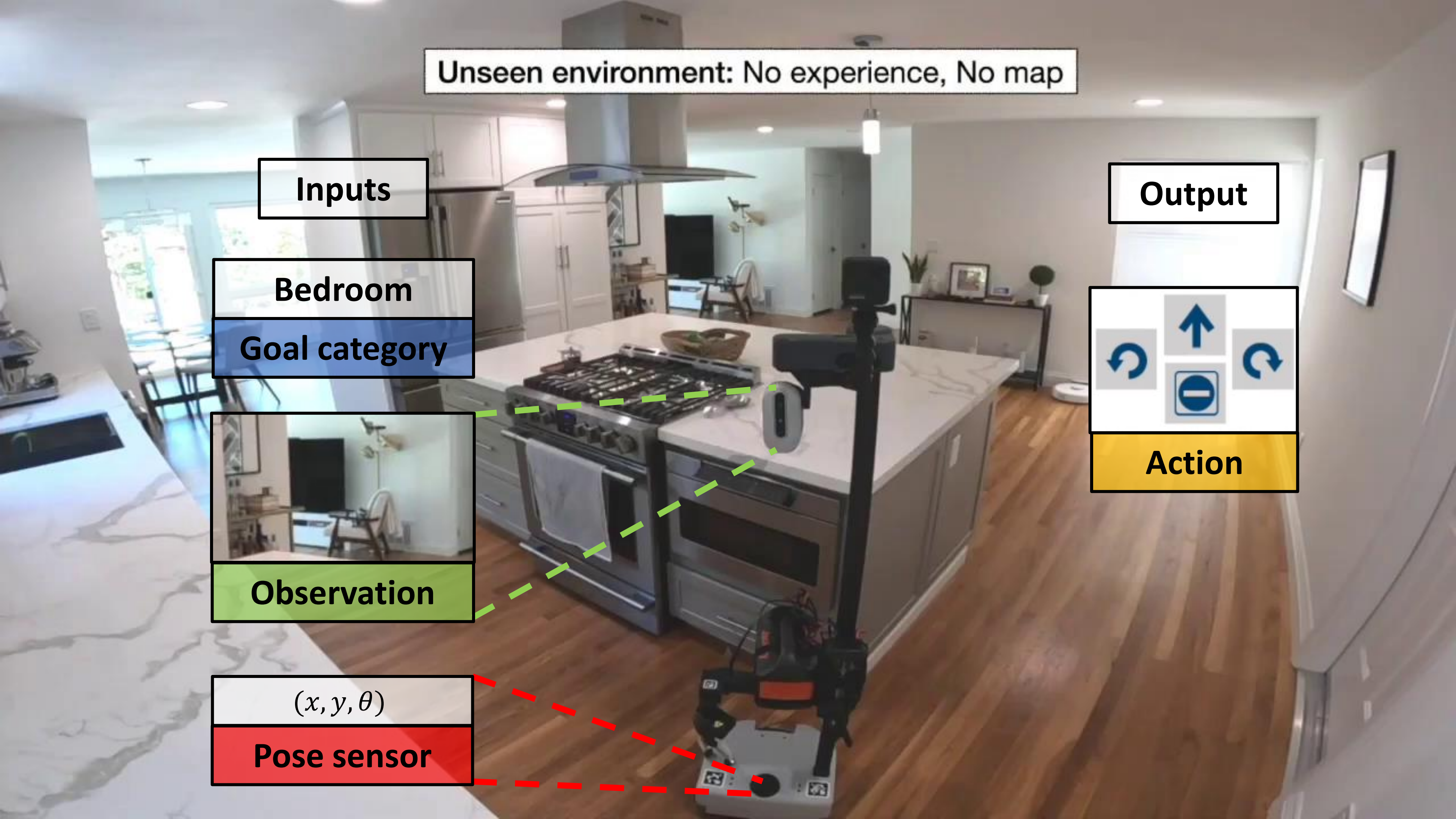
Navigating to Objects in the Real World

Theophile Gervet, Soumith Chintala, Dhruv Batra,
Jitendra Malik, Devendra Singh Chaplot

Presented by: Jemuel Stanley Premkumar


Unseen environment: No experience, No map
Inputs
Bedroom
Goal category
Observation
(x, y, θ)
Pose sensor
Output
Action


Spatial Scene Understanding
Navigable Space Detection
Semantic Scene Understanding
Object Detection
Oven
Semantic Exploration Priors
Where is a toilet more likely to be found?
Episodic Memory
Keep track of explored and unexplored areas
Unexplored
Explored

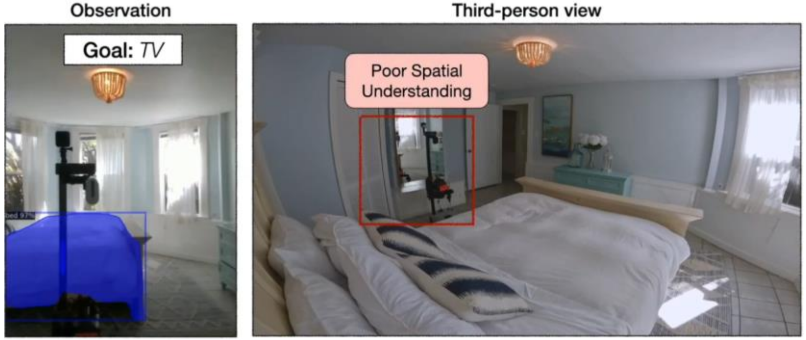
INTRODUCTION

Objective: Semantic Object Goal Navigation

- Understanding of objects
- Likely location
- Exploit prior knowledge

WHY IS NAVIGATION CHALLENGING?

Poor spatial understanding



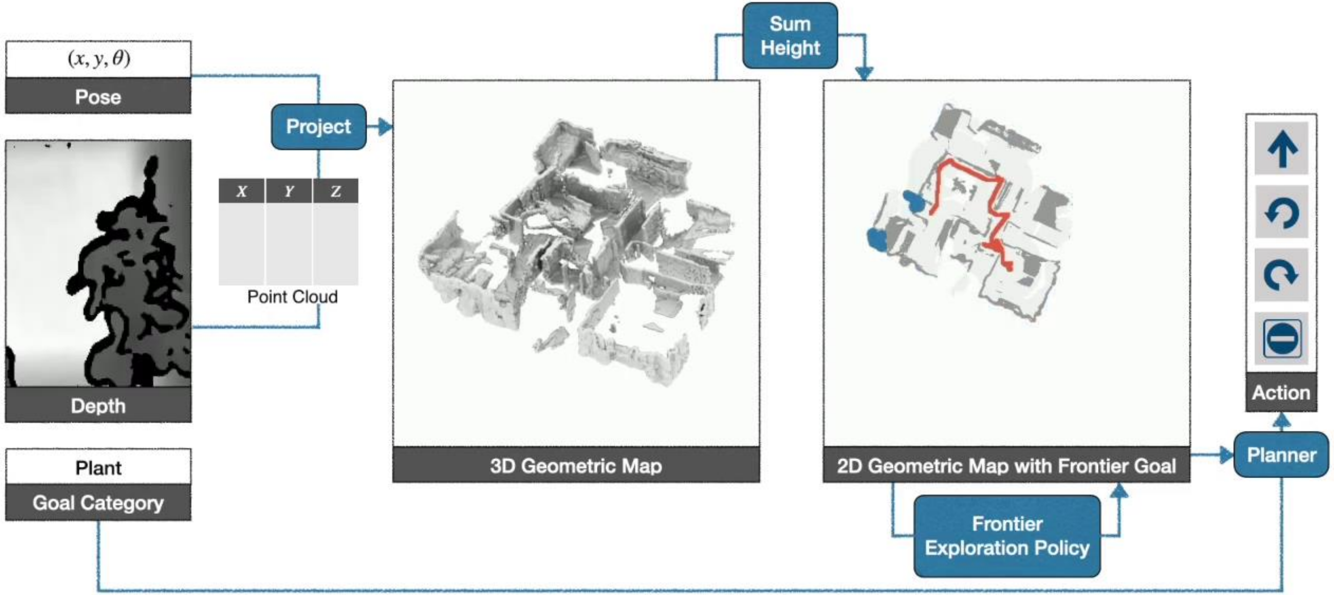
Poor semantic exploration priors

Poor semantic understanding

Poor episodic memory

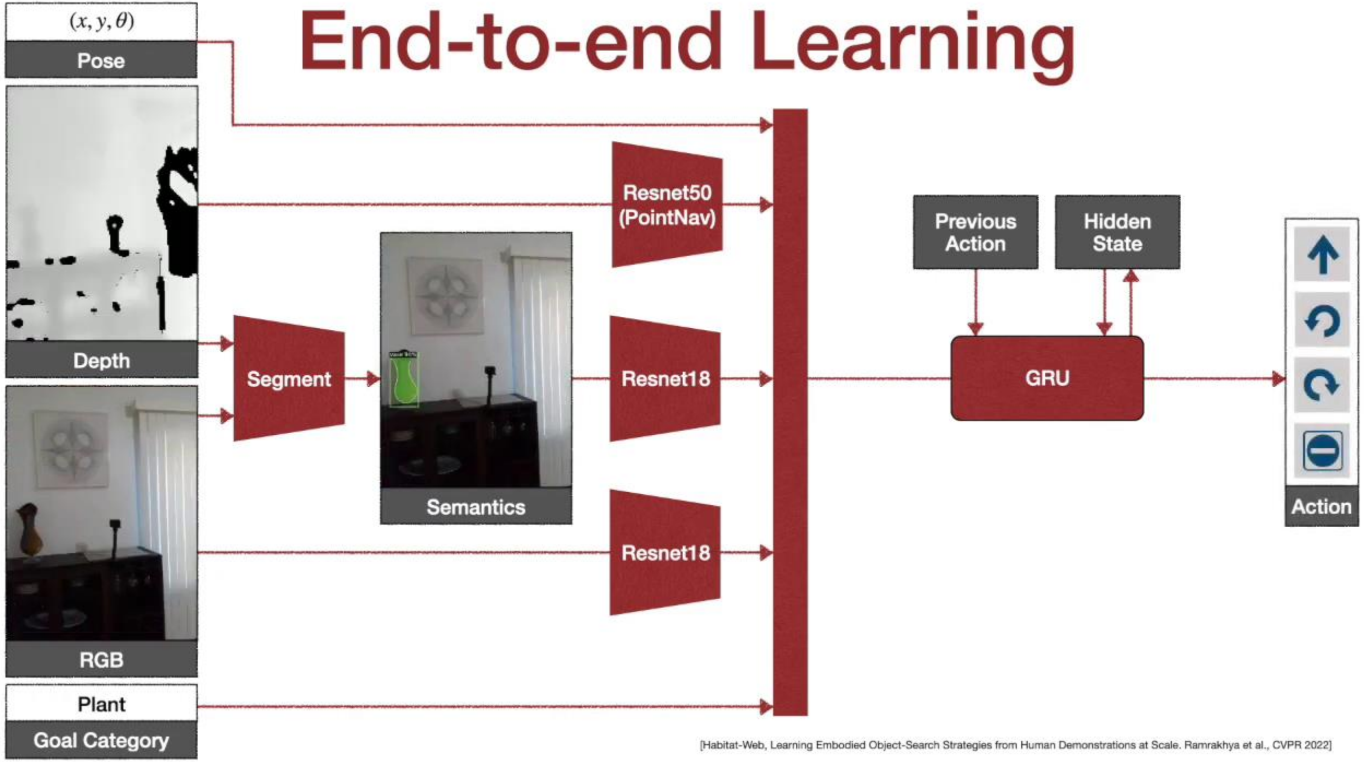
HOW IS DONE NOW?

Classical



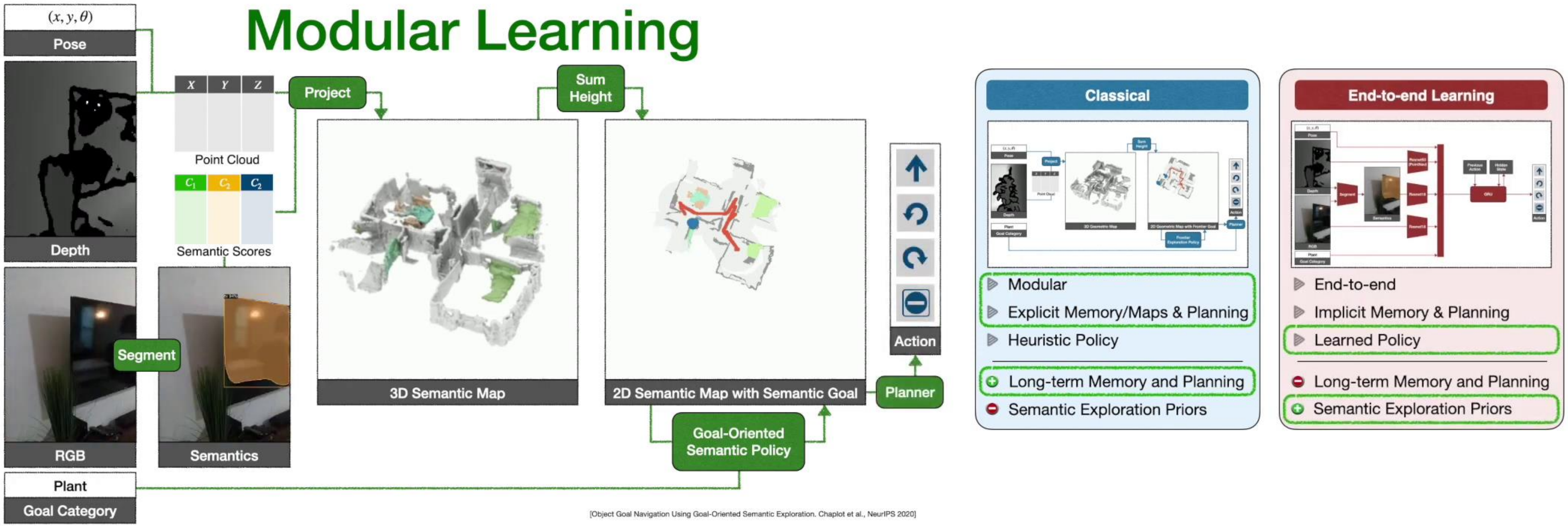
[A Frontier-based Approach for Autonomous Exploration. Yamauchi, *CIRA 1997*]

HOW IS DONE NOW?



[Habitat-Web, Learning Embodied Object-Search Strategies from Human Demonstrations at Scale. Ramrakhya et al., CVPR 2022]

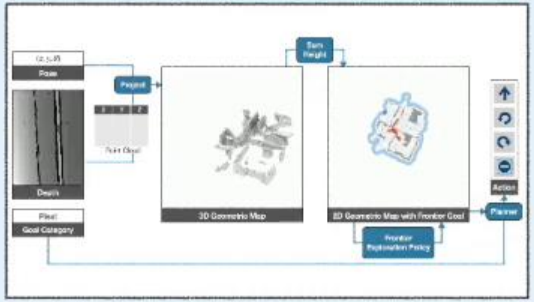
HOW IS DONE NOW?



[Object Goal Navigation Using Goal-Oriented Semantic Exploration. Chaplot et al., NeurIPS 2020]

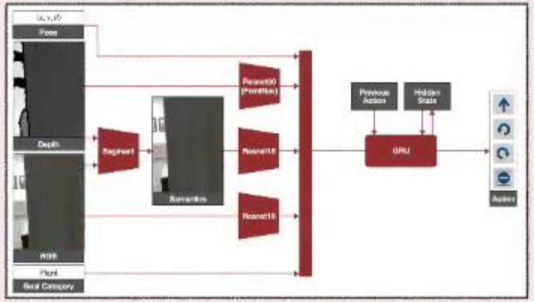
How well do they perform?

Classical



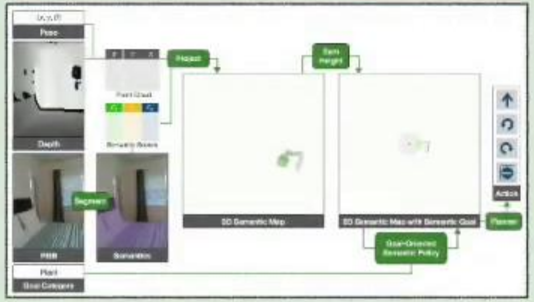
- Geometric Map
- Heuristic Exploration
  - No Training

End-to-end Learning



- End-to-end
- Large-scale IL + RL fine-tuning
  - 77,000 human trajectories
  - 200M frames of RL

Modular Learning

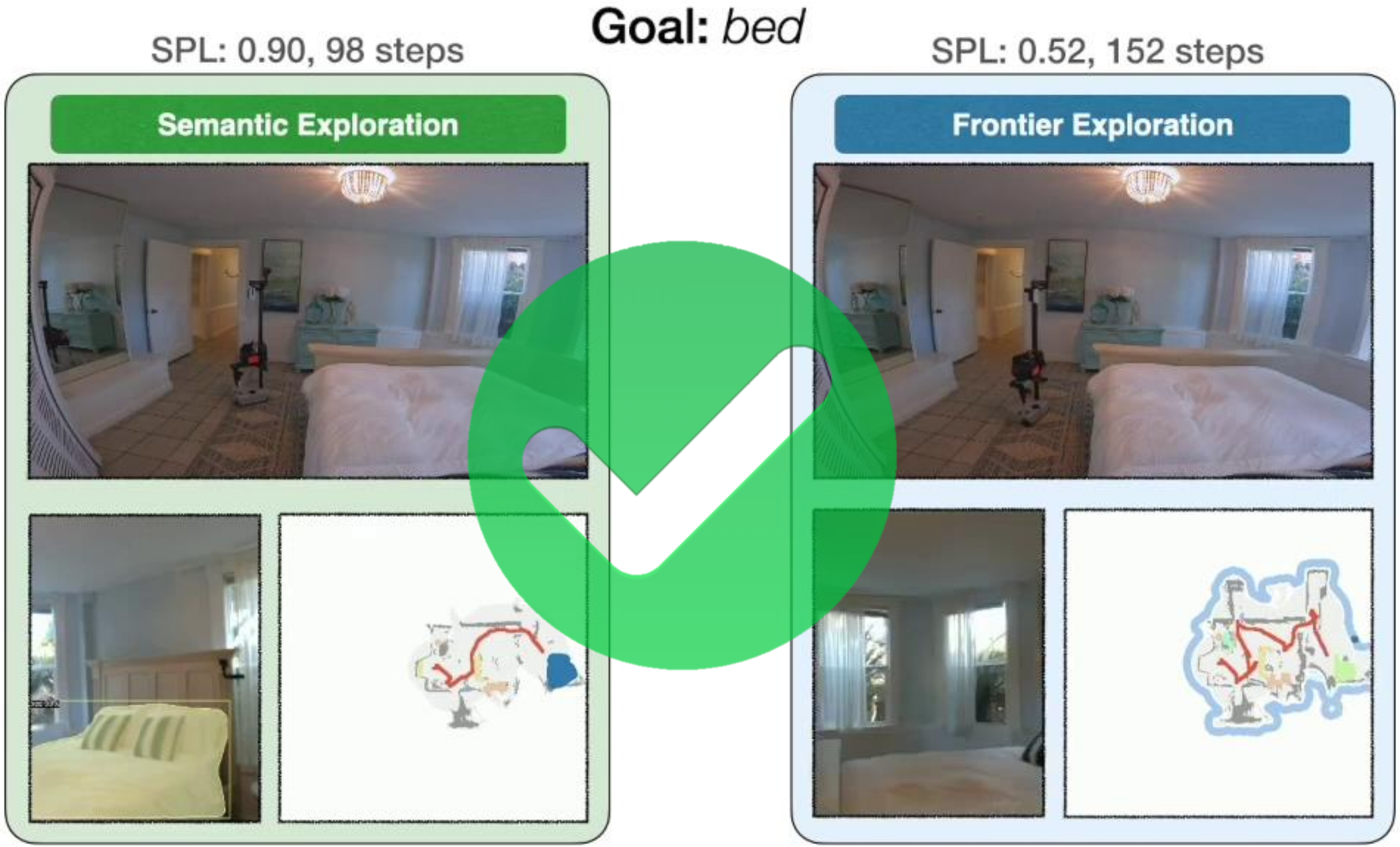


- Semantic Map
- Goal-Oriented Exploration
  - 10M frames of RL


Empirical Evaluation
3 Approaches
6 unseen homes
6 Global Object categories


Goal: couch
SPL: 0.74, 78 steps
SPL: 0.0, 121 steps
SPL: 0.33, 181 steps
Modular
End-to-End
Classical
Third-person view
Third-person view
Third-person view
Observation
Predicted Semantic Map
Observation
Predicted Semantic Map

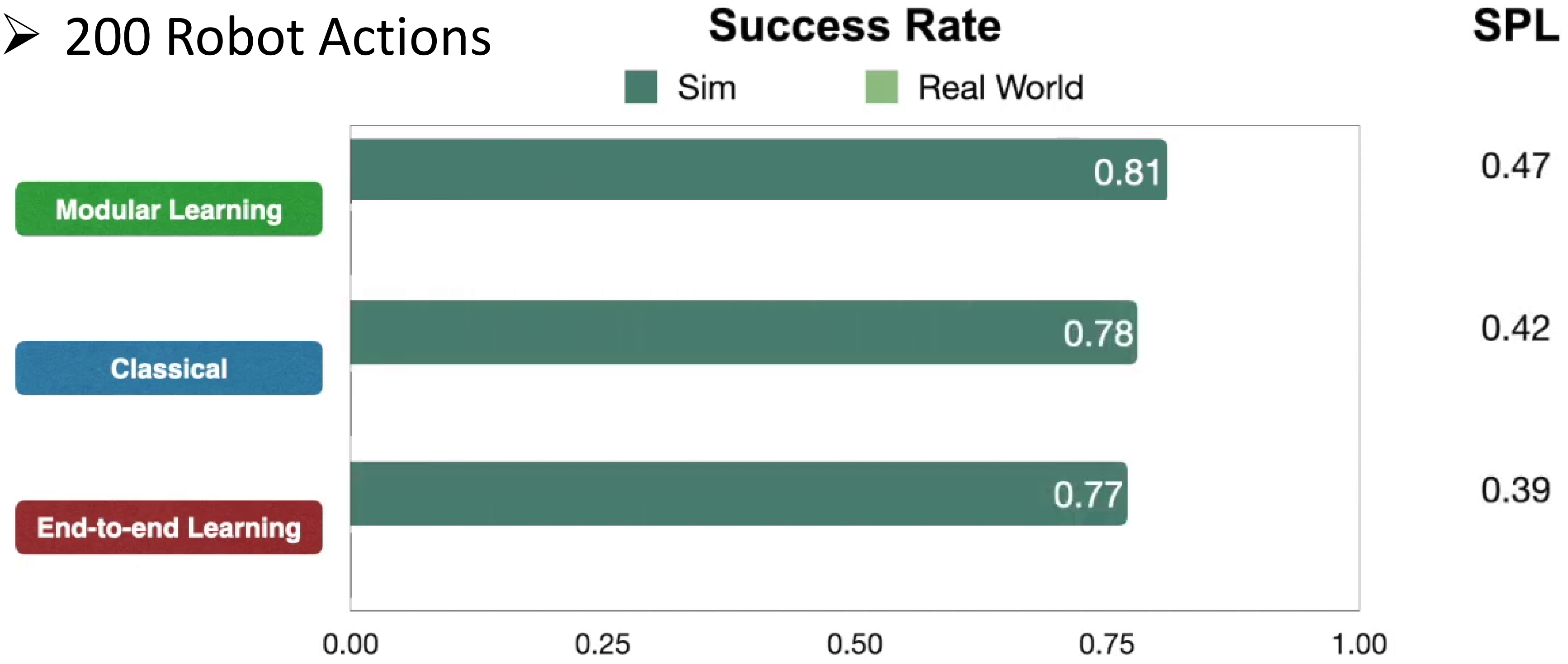
Modular vs Classical



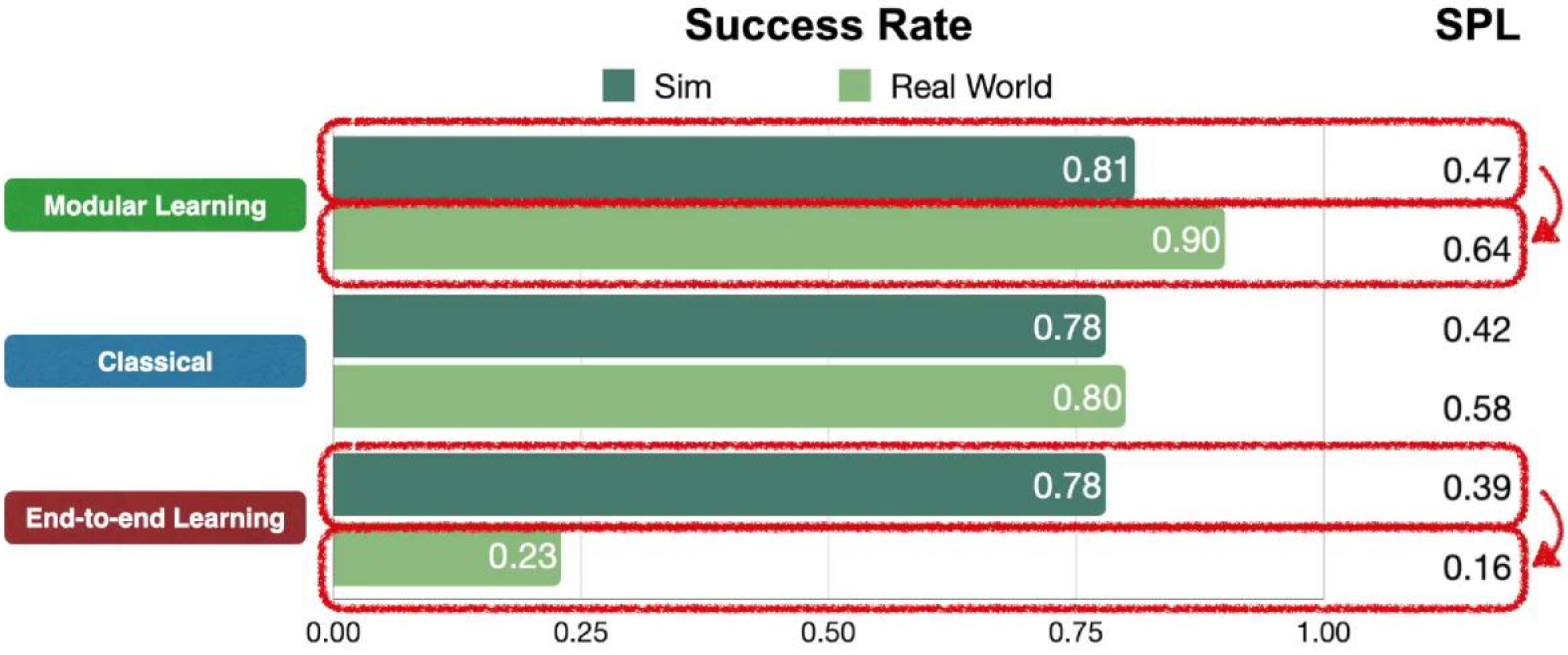
End-to-end Learning Failure

RESULTS

- 200 Robot Actions



RESULTS



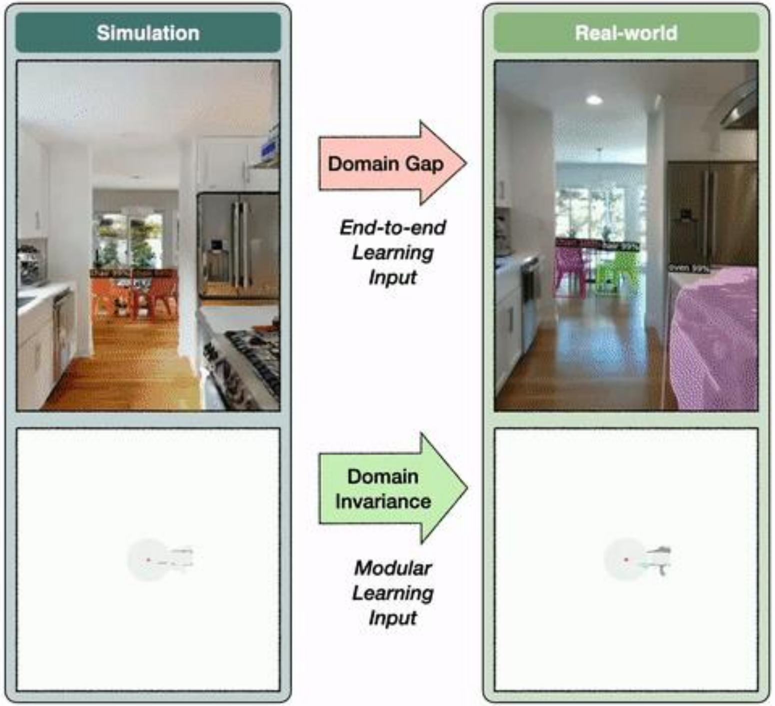
Success Rate
SPL
Sim
Real World
Modular Learning
Classical
End-to-end Learning
0.81
0.90
0.78
0.80
0.78
0.23
0.47
0.64
0.42
0.58
0.39
0.16
0.00
0.25
0.50
0.75
1.00

How is Modular Learning better than End-to-end Learning?

Reconstructed one real-world home in simulation and conducted experiments with identical episodes in sim and reality

How is Modular Learning better than End-to-end Learning?

Sim vs Real



Sim: Operates directly on RGB-D frames

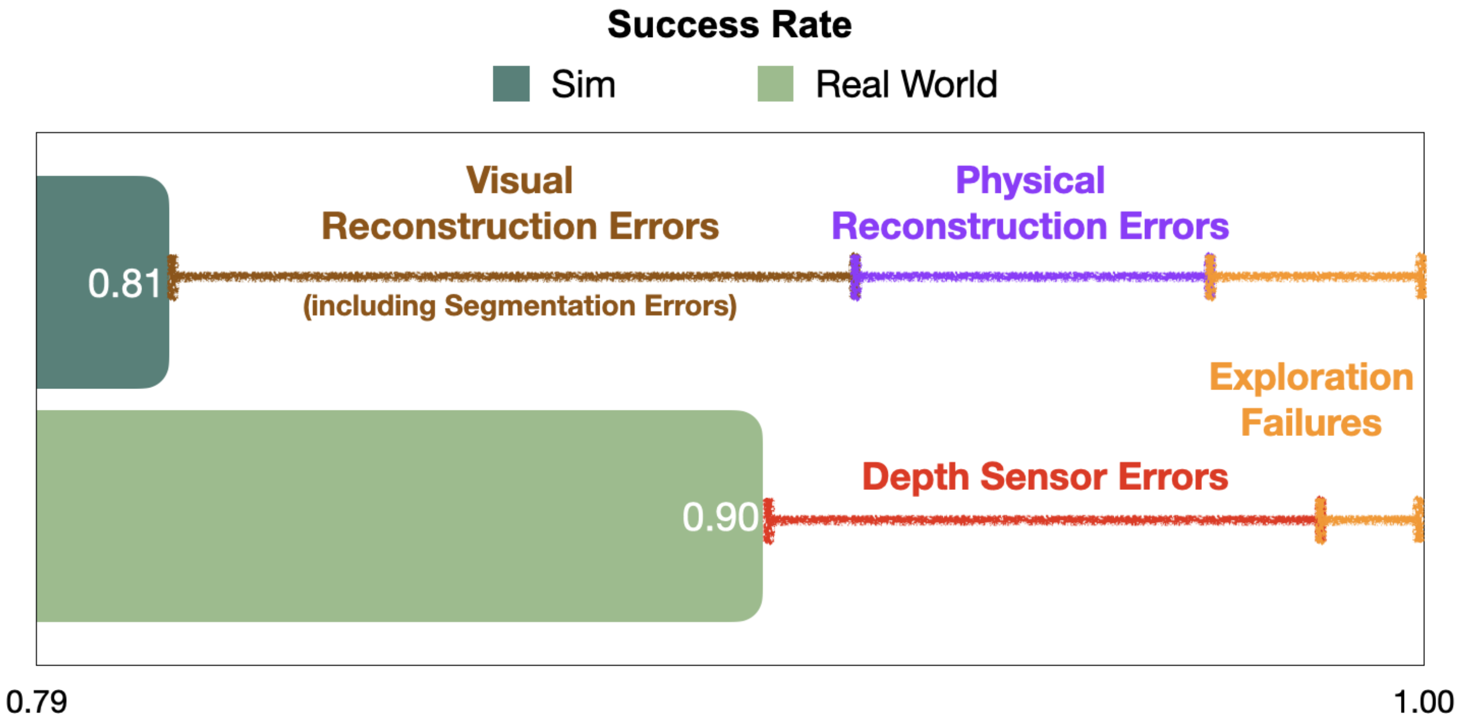
✓ Domain gap

Real: Semantic map

✓ Invariant between sim and reality

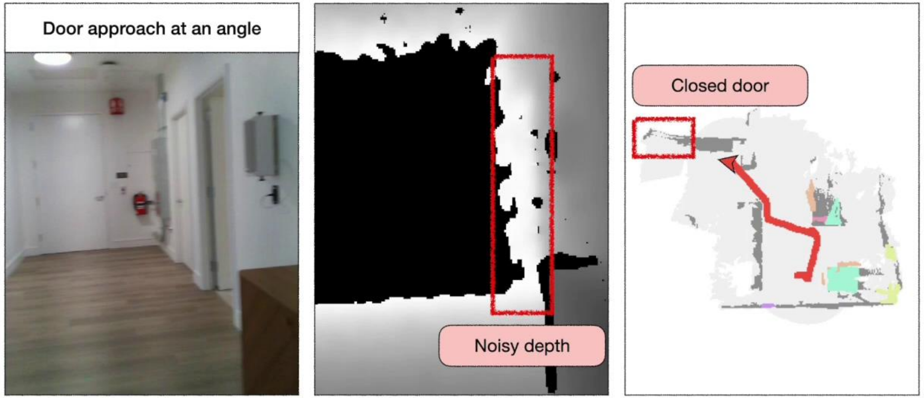
How is Modular Learning better than End-to-end Learning?

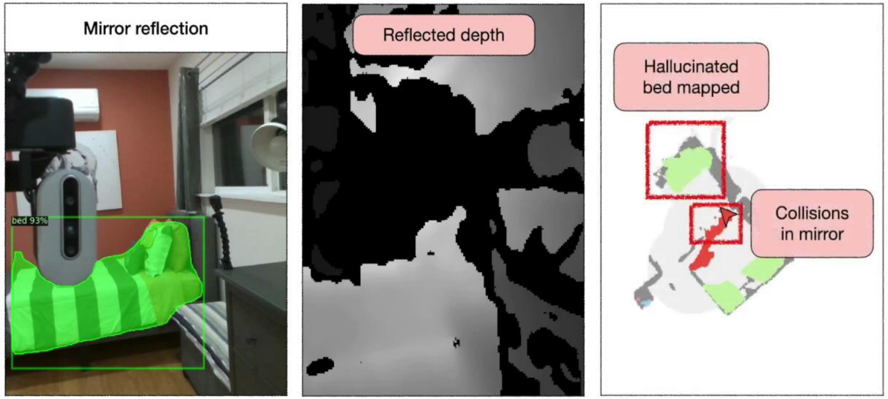
Error Modes



How is Modular Learning better than End-to-end Learning?

Error Modes





Reconstructed TV

Key Takeaways

Practitioners

✓ Modular learning more reliable (90% success)

Researchers

Issues:

➢ Sim-to-real gap:

- Leverage modularity and abstraction in policies

➢ Existence of disconnection between sim and real error modes

- Evaluate semantic navigation on real robots

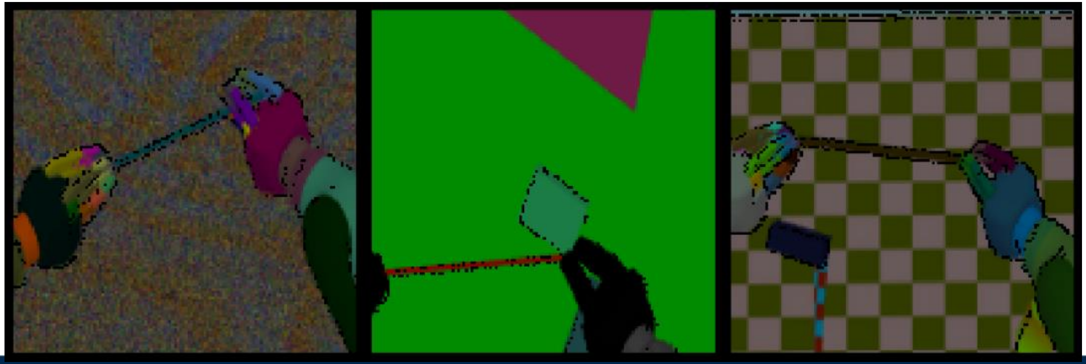
Discussion

Priya Thanneermalai 38 minutes ago
It makes sense that there are errors in Sim due to visual and physical reconstruction errors that are not applicable to the real world. Real world errors mostly stem from Reflections in TV and mirrors causing depth sense errors and downstream navigation failures. There can also be noise in depth that can block it in the map which can be solved by map denoising mechanisms. I wonder how can we make Sim better so as to do away with reconstruction errors, this may help to get easier empirical studies and make Sim a better representation of the real world.
helpful! | 0

Nathaniel Chong (nychong) 33 minutes ago
The authors stated that the reconstruction errors may be rectified by better 3D meshes, but did not elaborate much beyond that. I think that the most practical thing to do is to implement a realistic noise model into the simulation, so the learned policy can become invariant to realistic perturbations and transfer better to the real world.
helpful! | 0

Ruohua Li 6 hours ago
I think with the continuous development of the rendering abilities of game engines, pushing simulations to be more similar to the real world for perception model is something that could be done.
helpful! | 0

- ➢ How to make Sim better?
- Data augmentation techniques: Adding noise, varying light conditions
- Photo realism?



Discussion

- Develop real-world error modes for simulators

- Limits usefulness of sim to diagnose bottlenecks

- Modeling occlusion, sensor noise

Discussion

- ➢ Design policies that can be transferred from sim to real
- \- Prioritize real-world transfer
- \- Replace policy architectures that directly operate on RGB-D with ones leveraging abstractions as common practices in other domains
- \- Avoid training a segmentation model on sim data if policy architecture does not allow easy swapping

Discussion

Oliver A Wang (oliveraw) 8 hours ago
I thought it was interesting that the modular approach performed better in the real world than in simulation. It seemed that some aspects of the simulations themselves were faulty. For example, segmentation failing because objects don't look like they're supposed to, or simulated hallways that are too narrow to pass through. It makes me curious if simulated datasets are checked for quality or if it is solely up to the researchers who create them.
helpful! | 0

Alan Van Omen 3 hours ago
I was also surprised that the modular approach actually had significantly better performance in the real-world. I imagine if they somehow fixed some of the issues they talked about in simulation, such as the reflection of the tv or mirror, then there would be better performance in the simulation.

But it seems to be a common theme in papers we have read, that it is very useful to extract a useful lower-dimensional representation from raw pixel data rather than try to learn direclty in an end-to-end manner, as the latent representations will not be as susceptible to the large amount of noise and distractor data present in raw sensory inputs.
helpful! | 0

Nathaniel Chong (nychong) 1 hour ago
I agree with Alan in that latent representations and abstractions of real sensory input are more useful for model generalization, even to the real world.
The classical approach also performed better in the real world, indicating that the construction of a map and the use of semantic exploration are a reason for this success.
helpful! | 0

- ➢ Abstractions

Rajiv Govindjee 9 hours ago
It seems to me that the issue is not inherently with end-to-end training, but rather that we don't currently have good structures and model architectures that allow for these intermediate signals to be learned (stably). If we can learn these intermediate signals, it should theoretically be possible to achieve much better performance than introducing artificial bottlenecks that may be inefficient.

In an ideal architecture, a model can learn different submodules and pass information between them as needed; that information might contain some version of a semantic map (for example) without the engineer ever having to specify this. Presumably, the human brain is not genetically preprogrammed to pass around semantic maps. The human brain is, however, genetically programmed to have certain physical structures and to produce neurons with physical/chemical properties in different areas. That structure is what we seem to be lacking for many problems in embodied AI.

Giant transformer models appear to suffice for language and vision problems alone, but integrating them together for a task like visual navigation remains difficult.
helpful! | 0

Sawan Patel (sawanpa) 9 hours ago
I agree with Rajiv, it's interesting how these characteristic features are very common and reproducible in the human brain but are not precisely replicated in embodied AI. The lack of a stable representation for intermediate signaling is certainly a limiting factor.
helpful! | 0

Discussion

Haoyuan Ma 24 minutes ago
Is it also possible that we learned a transformer model that learns the difference between simulated environment and the real world so we could transform the abstraction learned from either world to the other?
helpful! | 0

Harikrishnan Seetharaman 8 minutes ago
This is an exciting idea that you mentioned about the transformer model. However, more or less the task you said resembles domain adaptation which is commonly done, where the model is trained on one domain and then adapted to perform well on a different domain. In the context of robotics, this could involve training a model on simulated data and then adapting it to perform well on real-world data. with careful design and training, I think it is possible to train a transformer model that can learn to perform well in both simulated and real-world environments by leveraging domain adaptation techniques.
helpful! | 0

Jemuel Stanley Premkumar 4 minutes ago
Actions
One approach is to use unsupervised domain adaptation which involves training the transformer on data from the simulated environment, and then fine-tuning it on a small amount of labeled data from the real world. This approach can be effective if the distribution of the data in the simulated and real environments is similar.
Another approach is to use adversarial domain adaptation which involves training the transformer to generate outputs that are indistinguishable between the simulated and real environments, while also preserving the semantic meaning of the input. This approach can be more effective if there is a large distribution shift between the simulated and real environments. However, both call for careful selection and preprocess of the data used for training and fine-tuning, to ensure that it accurately reflects the challenges and variations present in the real world environment.
helpful! | 0

- Domain adaptation