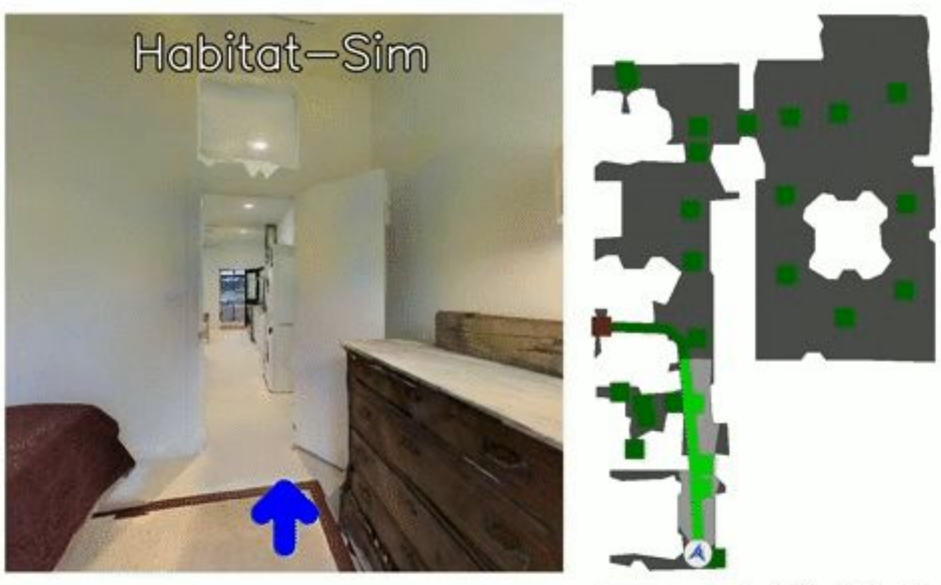
#online-vln

Sashank, Yash, Samyak, Harsh

VLN-CE setup

- Step size - 0.25m
- Turn Angle - 15°
- Goal Radius - 3m
- Inputs to agent -
  - RGB, Depth and Instruction
- Outputs -
  - One of the 4 actions (STOP, FORWARD, LEFT, RIGHT)



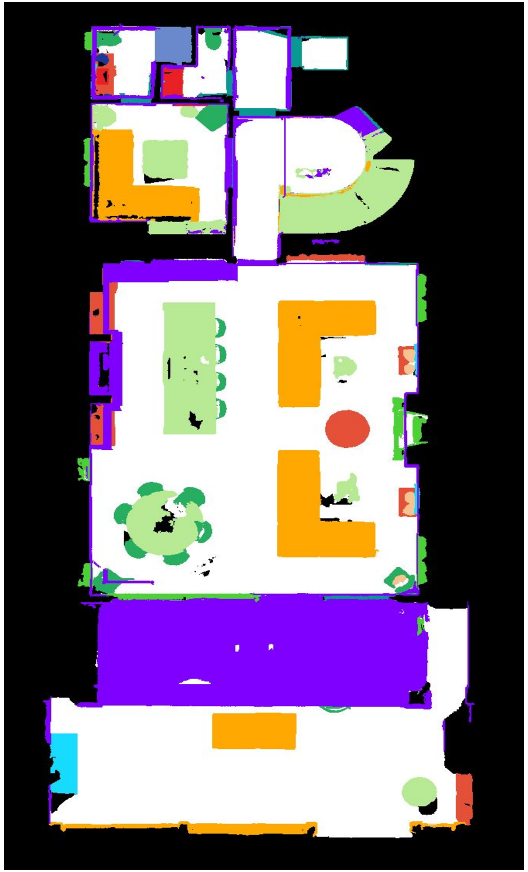
Leave the bedroom and enter the kitchen. Walk forward,
And take a left at the couch. Stop in front of the window.

Idea

Combining two fairly natural ideas to model a VLN-CE agent

1. Hierarchical Models
2. Using a top-down semantic map

Semantic map of scene
(40 categories)



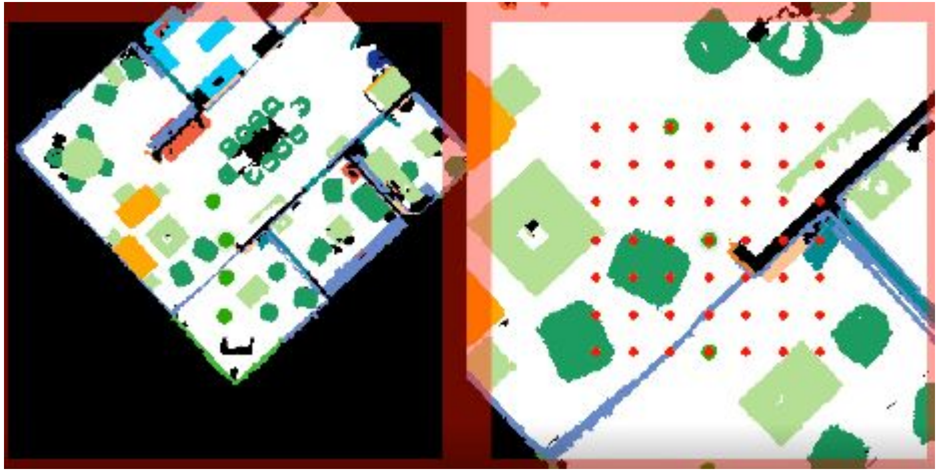
Hierarchical Model

Planner - predicts an intermediate waypoint. It is trained to predict a waypoint *w* forward steps away in the shortest path to the goal

Controller - takes a waypoint as input and navigates towards it for a max of *w* forward steps. Currently using a ‘teleporting’ model

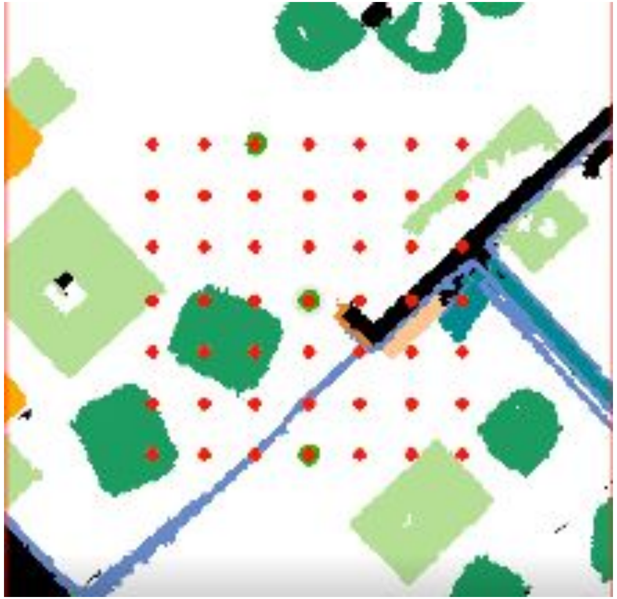
Planner - Input

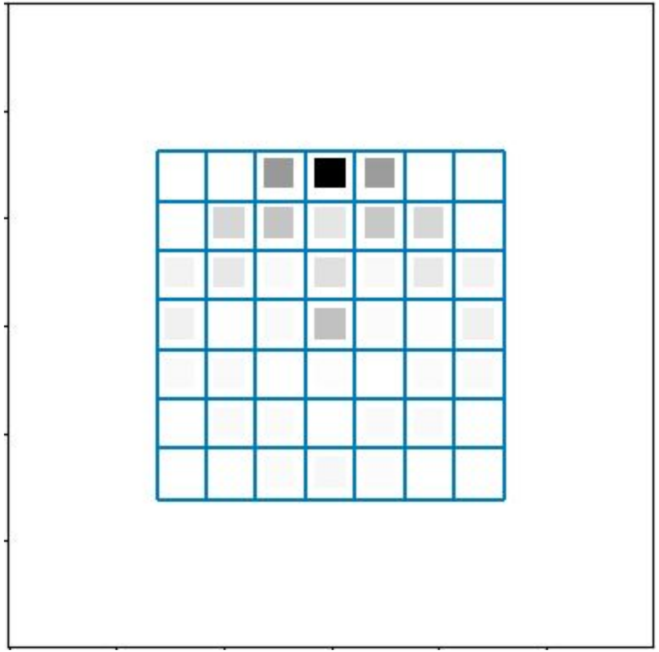
1. RGB
2. Depth
3. Instruction
4. Egocentric Semantic Map
    - Two crops of different resolutions to capture immediate and overall scene
    - Agent at the center and facing upwards
    - Fine crop - Corresponds to a region of 12 forward steps from center till the egde.
    - Coarse crop - Corresponds to a region of 36 forward steps from center till the egde.



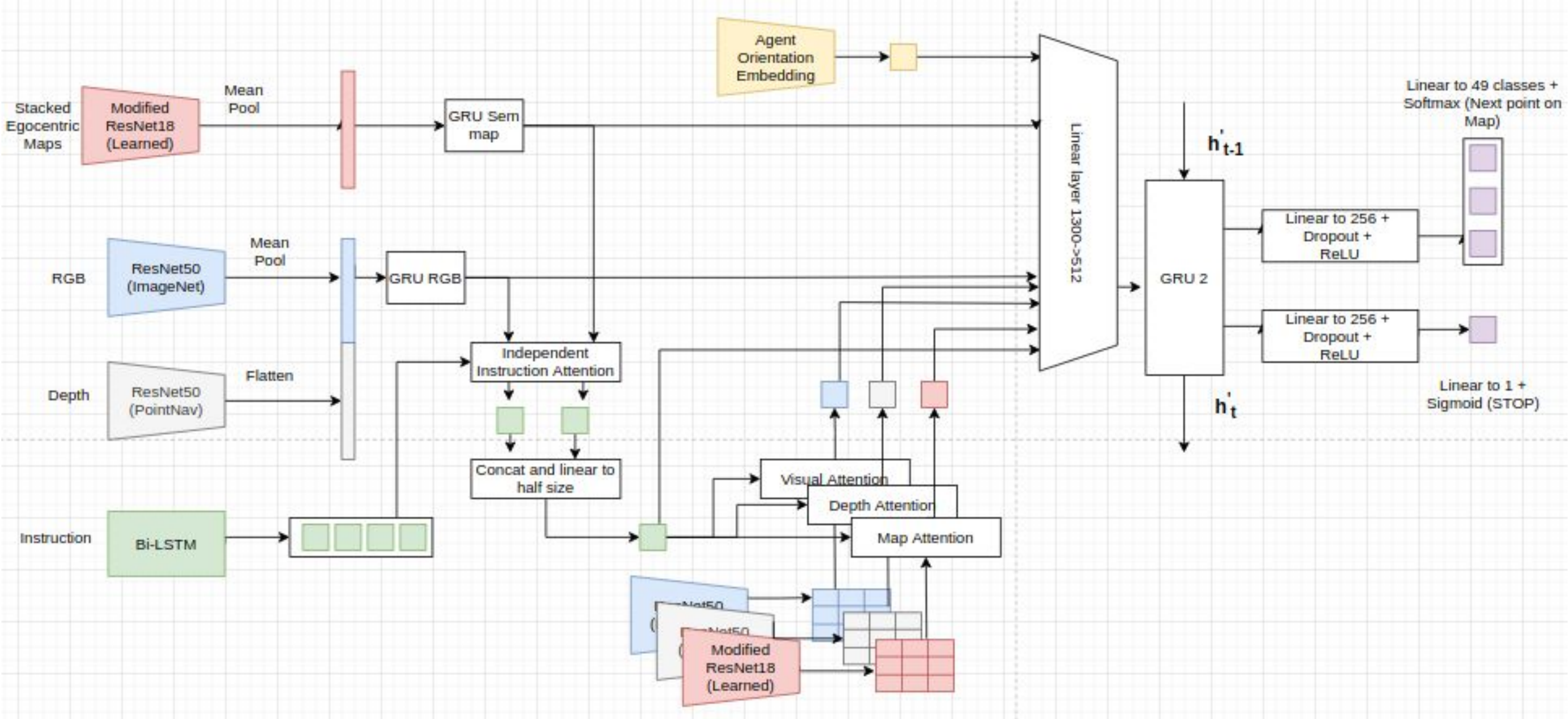
Planner - Output

1. Next waypoint on fine crop (classification across 7*7 grid)
2. Probability of STOP

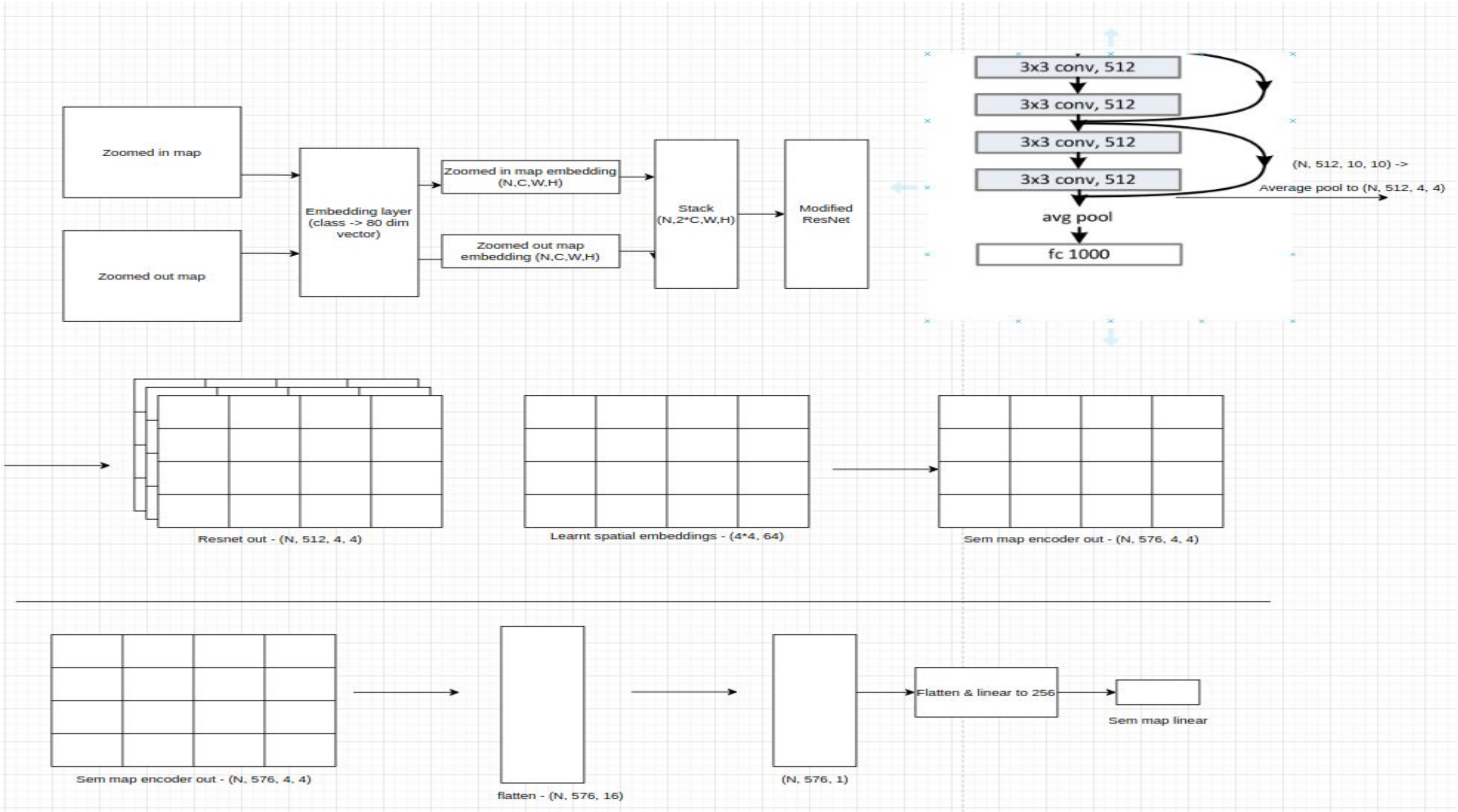




Planner model


Agent Orientation Embedding
Stacked Egocentric Maps
Modified ResNet18 (Learned)
Mean Pool
GRU Sem map
Linear to 49 classes + Softmax (Next point on Map)
h'_{t-1}
Linear layer 1300->512
GRU 2
Linear to 256 + Dropout + ReLU
RGB
ResNet50 (ImageNet)
Mean Pool
GRU RGB
Linear to 256 + Dropout + ReLU
Depth
ResNet50 (PointNav)
Flatten
Independent Instruction Attention
h'_t
Linear to 1 + Sigmoid (STOP)
Concat and linear to half size
Visual Attention
Depth Attention
Instruction
Bi-LSTM
Map Attention
Modified ResNet18 (Learned)

Semantic map encoder



Zoomed in map

Zoomed out map

Embedding layer (class -> 80 dim vector)

Zoomed in map embedding (N,C,W,H)

Zoomed out map embedding (N,C,W,H)

Stack (N,2*C,W,H)

Modified ResNet

3x3 conv, 512

3x3 conv, 512

3x3 conv, 512

3x3 conv, 512

avg pool

fc 1000

(N, 512, 10, 10) ->

Average pool to (N, 512, 4, 4)

Resnet out - (N, 512, 4, 4)

Learnt spatial embeddings - (4*4, 64)

Sem map encoder out - (N, 576, 4, 4)

Sem map encoder out - (N, 576, 4, 4)

flatten - (N, 576, 16)

(N, 576, 1)

Flatten & linear to 256

Sem map linear

Demo

Experiments

Model	RGB-D	Semantic Map	Floor Map	SR	SR - Base	Remarks
Full Model	Yes	Yes	-	24.60	0	
Full Model ablated RGBD	No	Yes	-	21.80	-2.8	
Full Model ablated Sem map	Yes	No	-	20.85	-3.75	Sem map is more important than RGB-D
Full Model with floor map	Yes	No	Yes	23.44	-1.16	Just using floor annotations is pretty good too!

Analysis of RGBD model vs Semantic Map model per number of semantic objects in instruction

As no. of objects in the instruction increase, we find a general drop in Success Rate of all models.

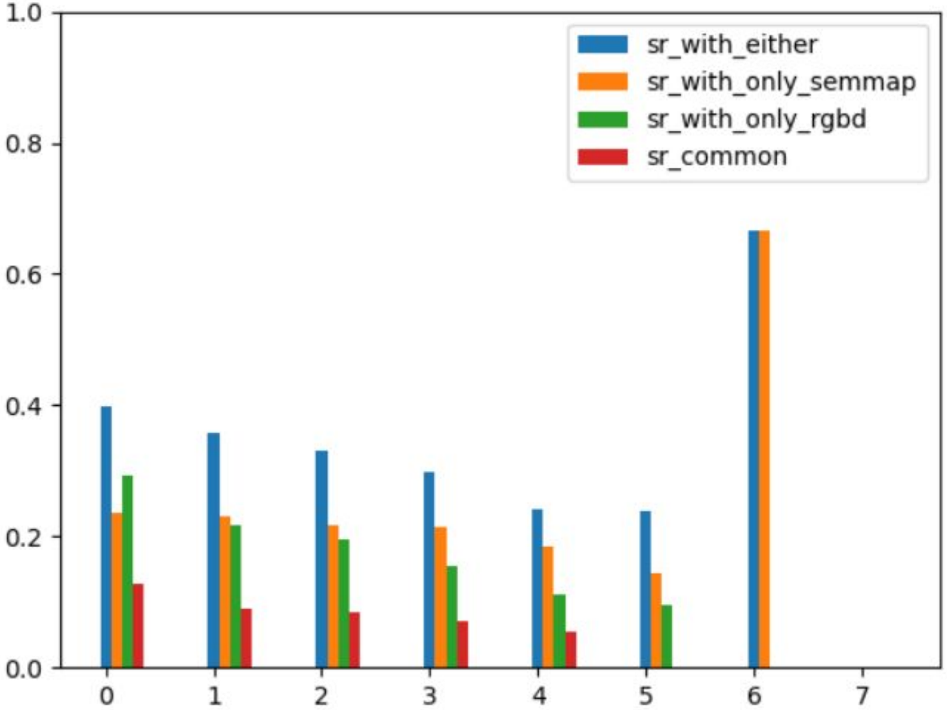
But the drop is much less in the model using only Semantic Map compared to the one using only RGBD. This is an encouraging trend.

Sem map (i.e,. no RGB-D) - 22.40

RGB-D (i.e., no Sem map) - 21.26

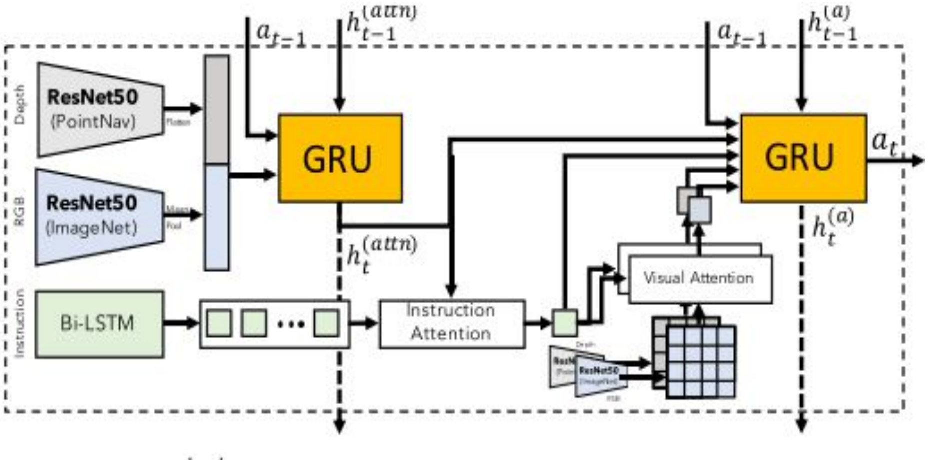
Union of both - 34.63

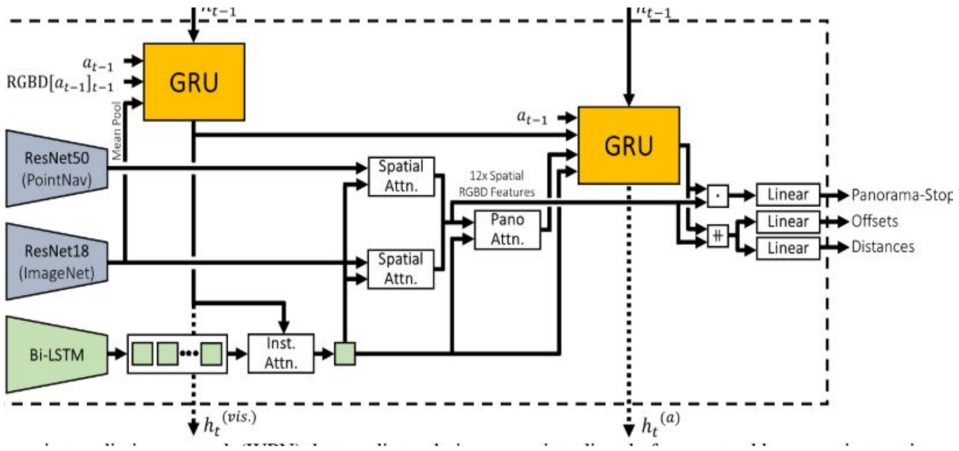
Intersection of both ~9



Motivation for floor map - Jacob’s work

Paper	Input	Model	Algorithm	Hierarchy	Overall SR
1	FPV	CMA V1	TF/dAgger	No	27
2	Panos	CMA V2	RL	Yes	33





Motivation for floor map

However, we find that both hierarchy and panos aren’t the components that are improving SR

All improvements are via RL
Infact, RL with direction prediction (36) is better than waypoint prediction (33)

	CMA V1	CMA V2
Teacher Forcing	23	24
dAgger	27	27
RL	-	33

Motivation for floor map

Probable reasons

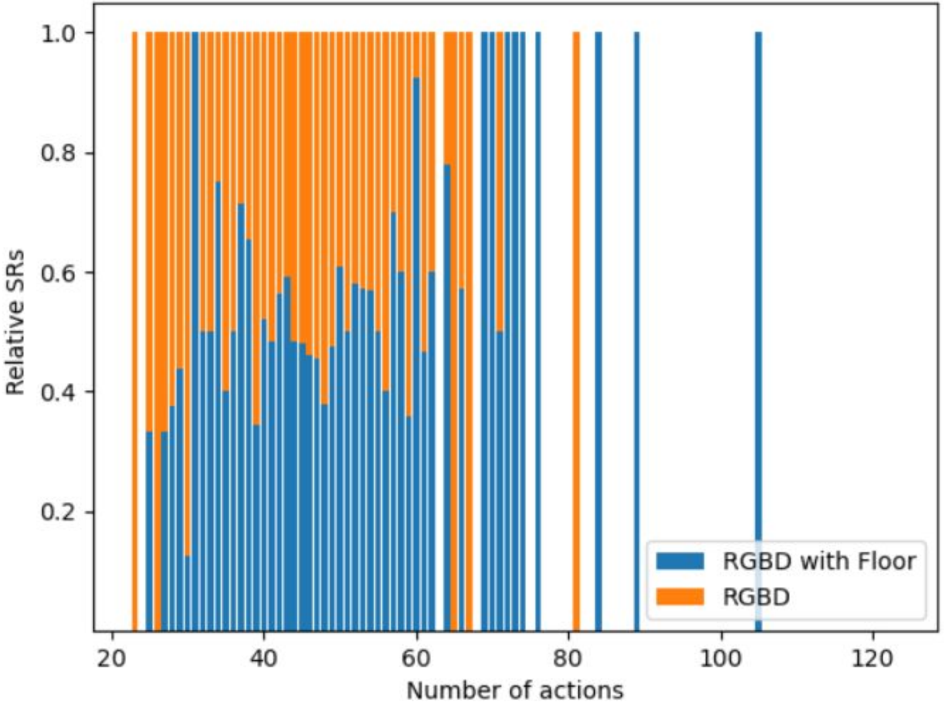
- Planner is missing observations (unlikely as they have panos)
- Projecting a point into space from first person view is difficult - May be just floor map would help?

Model	SR	Remarks
RGBD	20.85	
RGBD + Semantic map	24.6	
RGBD + floor map	23.44	69% of improvement

Analysis of RGBD model vs RGBD + floor map per number of actions needed to finish episode

As length of episode increases, floor map seems to help more (although noisy indicator due to number of instructions at each value)

SR with RGBD + Floor - 23.81
SR with RGBD - 21.26
Union of both - 35.29
Intersection ~9 (Similar to RGBD case)



Strange observation - 1

Model A	Model B	SR A	SR B	SR A & B
RGB-D	RGB-D + Floor	20.85	23.44	9.78
RGB-D	RGB-D + SemMap	20.85	24.6	10.05
RGB-D + Floor	RGB-D + SemMap	23.44	24.6	12.01

Strange observation - 2

Found out that models trained on the task **without instruction input** work very well

Model	SR with Instruction	SR without Instruction
Full Model	24.6	21.21
Model with floor map instead of semantic map	23.05	22.10
Model with no inputs	-	9

Next steps

- Explore mechanisms to better fuse all the modalities
- Start with Jacob’s codebase and check how removing instructions perform there

Thank you.