

# cognitive\_planning

## Visual Representation for Semantic Target Driven Navigation

Arsalan Mousavian, Alexander Toshev, Marek Fiser, Jana Kosecka, James Davidson

This is the implementation of semantic target driven navigation training and evaluation on Active Vision dataset.

ECCV Workshop on Visual Learning and Embodied Agents in Simulation Environments 2018.



Target: Fridge



Target: Television



Target: Microwave



Target: Couch

Paper: <https://arxiv.org/abs/1805.06066>

## 1. Installation

### Requirements

### Python Packages

```
networkx  
gin-config
```

## Download cognitive\_planning

```
git clone --depth 1 https://github.com/tensorflow/models.git
```

## 2. Datasets

### Download ActiveVision Dataset

We used Active Vision Dataset (AVD) which can be downloaded from [here](#). To make our code faster and reduce memory footprint, we created the AVD Minimal dataset. AVD Minimal consists of low resolution images from the original AVD dataset. In addition, we added annotations for target views, predicted object detections from pre-trained object detector on MS-COCO dataset, and predicted semantic segmentation from pre-trained model on NYU-v2 dataset. AVD minimal can be downloaded from [here](#). Set `$AVD_DIR` as the path to the downloaded AVD Minimal.

### TODO: SUNCG Dataset

Current version of the code does not support SUNCG dataset. It can be added by implementing necessary functions of `envs/task_env.py` using the public released code of SUNCG environment such as [House3d](#) and [MINOS](#).

### ActiveVisionDataset Demo

If you wish to navigate the environment, to see how the AVD looks like you can use the following command:

```
python viz_active_vision_dataset_main -- \
  --mode=human \
  --gin_config=envs/configs/active_vision_config.gin \
  --gin_params='ActiveVisionDatasetEnv.dataset_root=$AVD_DIR'
```

## 3. Training

Right now, the released version only supports training and inference using the real data from Active Vision Dataset.

When RGB image modality is used, the Resnet embeddings are initialized. To start the training download pre-trained Resnet50 check point in the working directory `./resnet_v2_50_checkpoint/resnet_v2_50.ckpt`

```
wget http://download.tensorflow.org/models/resnet_v2_50_2017_04_14.tar.gz
```

### Run training

Use the following command for training:

```
# Train
python train_supervised_active_vision.py \
  --mode='train' \
  --logdir=$CHECKPOINT_DIR \
  --modality_types='det' \
  --batch_size=8 \
  --train_iters=200000 \
  --lstm_cell_size=2048 \
```

```
--policy_fc_size=2048 \  
--sequence_length=20 \  
--max_eval_episode_length=100 \  
--test_iters=194 \  
--gin_config=envs/configs/active_vision_config.gin \  
--gin_params='ActiveVisionDatasetEnv.dataset_root=$AVD_DIR' \  
--logtostderr
```

The training can be run for different modalities and modality combinations, including semantic segmentation, object detectors, RGB images, depth images. Low resolution images and outputs of detectors pretrained on COCO dataset and semantic segmentation pre trained on NYU dataset are provided as a part of this distribution and can be found in Meta directory of AVD\_Minimal. Additional details are described in the comments of the code and in the paper.

## Run Evaluation

Use the following command for unrolling the policy on the eval environments. The inference code periodically check the checkpoint folder for new checkpoints to use it for unrolling the policy on the eval environments. After each evaluation, it will create a folder in the \$CHECKPOINT\_DIR/evals/\$ITER where \$ITER is the iteration number at which the checkpoint is stored.

```
# Eval  
python train_supervised_active_vision.py \  
--mode='eval' \  
--logdir=$CHECKPOINT_DIR \  
--modality_types='det' \  
--batch_size=8 \  
--train_iters=200000 \  
--lstm_cell_size=2048 \  
--policy_fc_size=2048 \  
--sequence_length=20 \  
--max_eval_episode_length=100 \  
--test_iters=194 \  
--gin_config=envs/configs/active_vision_config.gin \  
--gin_params='ActiveVisionDatasetEnv.dataset_root=$AVD_DIR' \  
--logtostderr
```

At any point, you can run the following command to compute statistics such as success rate over all the evaluations so far. It also generates gif images for unrolling of the best policy.

```
# Visualize and Compute Stats  
python viz_active_vision_dataset_main.py \  
--mode=eval \  
--eval_folder=$CHECKPOINT_DIR/evals/ \  
--output_folder=$OUTPUT_GIFS_FOLDER \  
--gin_config=envs/configs/active_vision_config.gin \  
--gin_params='ActiveVisionDatasetEnv.dataset_root=$AVD_DIR'
```

## Contact

To ask questions or report issues please open an issue on the tensorflow/models [issues tracker](#). Please assign issues to @arsalan-mousavian.

## Reference

The details of the training and experiments can be found in the following paper. If you find our work useful in your research please consider citing our paper:

```
@inproceedings{MousavianECCVW18,  
  author = {A. Mousavian and A. Toshev and M. Fiser and J. Kosecka and J. Davidson},  
  title = {Visual Representations for Semantic Target Driven Navigation},  
  booktitle = {ECCV Workshop on Visual Learning and Embodied Agents in Simulation  
Environments},  
  year = {2018},  
}
```