# Object-oriented Targets for Visual Navigation using Rich Semantic Representations

**Jean-Benoit Delbrouck**
Numediart
University of Mons, Belgium
jean-benoit.delbrouck@umons.ac.be

**Stéphane Dupont**
Numediart
University of Mons, Belgium
stephane.dupont@umons.ac.be

## Abstract

When searching for an object humans navigate through a scene using semantic information and spatial relationships. We look for an object using our knowledge of its attributes and relationships with other objects to infer the probable location. In this paper, we propose to tackle the visual navigation problem using rich semantic representations of the observed scene and object-oriented targets to train an agent. We show that both allows the agent to generalize to new targets and unseen scene in a short amount of training time.

## 1 Introduction

In this paper, we focus on the problem of navigating a space to find and reach a given object using visual input as well as rich semantic information about the observations. Given that the agent has a semantic knowledge of the world, the attributes defining the objects he sees are grounded into the environment. We would like the agent to use its capacity to understand the attributes and relationships between objects to find the quickest pathway towards a given target. We focus on two training aspects to achieve this task

Firstly, the agent is pre-trained to describe all aspects (object-wise) of a scene using natural language. This ability relies on a strong semantic understanding of a visual scene and guarantees a strong grasp of the environment. He is also able to output the localization of the described objects within the visual frame as well as the confidence about the given inference. We hope this strong ability gives the agent the semantic and spatial knowledge required for the task. Indeed, objects with similar utility or characteristics are usually close to each other (i.e. when looking for a stove, you might infer that it stands on cooking plates, as well as a keyboard lies close to a computer).

In practice, the previous statement doesn't always hold true. Some spatial relationships between objects are specific to certain space configurations. For example, in a living-room, a TV is usually in front of (and therefore close by) of sofa, even though the two objects don't share characteristics. But this spatial relation between these two objects is not especially true for, let's say, a bedroom. To tackle this second problem, our model is built in two distinct layers: one is dedicated to every scene (global knowledge about the world) and the other is scene-type specific (namely bedroom, bathroom, living-room or kitchen). No model layer is dedicated to a specific scene (an instance of a bedroom, for example) as opposed to prior work Zhu et al. [12], where a part of the model is reserved for each specific instance. In the latter contribution, the model tended to poorly transfer knowledge to unseen scene or objects.

For our experiments, we use the AI2-THOR framework Kolve et al. [7] which provides near photo-realistic 3D scene of house's room. The framework features enables the setup to follow a reinforcement setting. The state of the agent in the environment changes as the agent take actions. The location of the agent is known at each step and can be randomized for training or testing. We dispose of 20 rooms each containing 5 targets (or frame) distributed along the four scene-types: kitchen, living-room, bedroom and bathroom.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

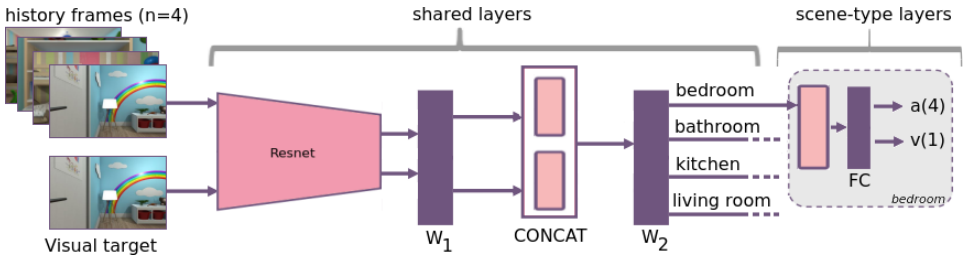



Figure 1: Baseline model. The right grey box can be seen as the policy layer, there is one for every scene-type. It means that each room instance will share its policy parameters with other instances of the same scene-type.

### 1.1 Related work

Since the emergence of virtual environments, a few works are related to ours. Zhu et al. [12] proposed a deep reinforcement learning framework for target-driven visual navigation. As input to the model, only the target frame is given without any other priors about the environment. To transfer knowledge to unseen scene or object, a small amount of fine-tuning was required. We use their siamese network as baseline for our experiments. To adapt to unseen data, Gupta et al. [4] proposed a mapper and planner with semantic spatial memory to output navigation actions. The use of spatial memory for visual question answering in such interactive environments has been explored by Gordon et al. [3]. Visually grounded navigation instruction have also been addressed by a plurality of works (Anderson et al. [1], Yu et al. [11], Hermann et al. [5]). The closest work related to ours is probably from Yang et al. [10], they also use semantic priors for the virtual navigation task. Nevertheless, we differ in how the semantic representation is constructed. Their approach uses a knowledge graph built on Visual Genome (Krishna et al. [8]) where the input of each node is based on the the current state (or current visual frame) and the word vector embedding of the target. Our idea is rather based on natural language as well as object localization (more detailed in section 3). The task goal is also defined differently: their end state is reached when the target object is in the field of view and within a threshold of distance whilst we require the agent to go to a specific location.

## 2 Baseline model

This section describes the model without any semantic knowledge about the environment. We also explain how we redefine target to be object-oriented.

### 2.1 Network Architecture

First, both history frames and target frames (tiled 4 times to match history frames input) are passed through a ResNet-50 pre-trained on ImageNet Deng et al. [2] that produces 2048-d features for each frame. We freeze the ResNet parameters during training. The resulting history and target 8196-d output vectors is linearly projected into the 512-d embedding space by the same matrix $\boldsymbol{W}_1$. The CONCAT layer takes a 1024-d concatenated embedding and generates a 512-d joint representation with matrix $\boldsymbol{W}_2$ that serves as input for each scene-type layer. This vector is subsequently passed through scene-type specific fully-connected layers $f_s, s \in \{bathroom, bedroom, kitchen, livingroom\}$ , producing 4 policy outputs (actions) and a single value output. Matrices sizes of the models are $\boldsymbol{W}_1 \in \mathbb{R}^{8196\times512}$, $\boldsymbol{W}_2 \in \mathbb{R}^{1024\times512}$ and each $f_s$ contains two matrices $\boldsymbol{W}_{s_1} \in \mathbb{R}^{512\times512}$ and $\boldsymbol{W}_{s_2} \in \mathbb{R}^{512\times5}$. Each matrix has its subsequent bias and uses ReLu as activation function.

As the shared layers parameters are used across different tasks, it can benefit from learning with multiple goals simultaneously. We then use A3C reinforcement learning model (Mnih et al. [9]) that learns by running multiple copies of training threads in parallel and updates the shared set of model parameters in an asynchronous manner. The reward upon completion is set to 10. Each action taken decreases the reward by 0.01 hence favoring shorter paths to the target.

### 2.2 Object-oriented targets

We use the same data-set as used in Zhu et al. [12] for comparison. It's composed of 20 rooms distributed along the four scene-types and for each room, five targets are chosen randomly. Our main assumption is that the use of semantic knowledge about objects improves the navigation task. As a first test-bed, we redefine the targets manually so that each target frame clearly contains an object. We make sure that an object picked appears on multiple other targets. To test this change, we make use of the model described in section 2.1.


Figure 2: Handpicked differences between randomly sampled target (top) and object-oriented target (bottom) across 4 scenes.

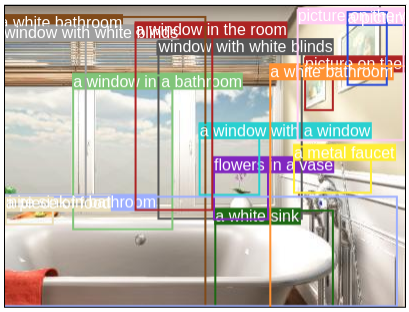
Results show that, after a certain amount of training, the baseline model is able to learn to navigate to random targets but the training convergence is much quicker when using object-oriented targets, which confirms our intuition. Providing object-oriented target also improves the overall training and enables the model to better generalize to other random frames (more details in section 4).

## 3 Semantic model

In this section, we explain how we collected data for our semantic knowledge database. We also show how we plug the semantics into the baseline model presented in section 2.1.

### 3.1 Semantic knowledge

To build the semantics data of the observations, we use DenseCap (Johnson et al. [6]), a vision system to both localize and describe salient regions in images in natural language. DenseCap is trained on Visual Genome Krishna et al. [8], an image data-set that contains dense annotations of objects, attributes, and relationships within each image. Every frame available in the 20 scenes are passed through the trained DenseCap model to generate our semantic knowledge database. An output for a frame is as follows:



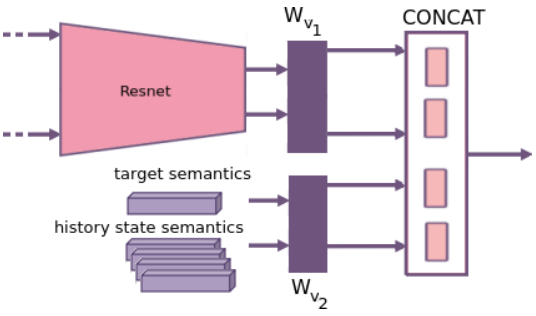

picture on the wall. a picture on the wall. window with white blinds. a white sink.
bathroom. a window with a window. a white bathroom. a metal faucet. a piece of food.
a white bathroom. flowers in a vase. window with white blinds. a window in a bathroom.
picture on the wall. a window in the room. white sink in

picture on the wall. 4.9513 [62.49,7.55,539.14,503.60]
a picture on the wall. 3.6825 [275.91,65.66,260.71,298.42]
window with white blinds. 2.7136 [673.27,67.40,45.13,103.78]
a white sink. 2.48694 [59.49,108.68,249.14,249.70]
a white bathtub. 2.4288 [594.52,116.99,54.78,97.48]

Figure 3: Output of DenseCap. We select the top-5 anchor boxes according to their confidence score.

For each anchor box, the model predicts a confidence score and four scalars ($[x\_min, y\_min, x\_max, y\_max]$) which represent the predicted box coordinates. DenseCap also includes a language model to generate a rich sentence for each box. Regions with high confidence scores are more likely to correspond to ground-truth regions of interest, hence we keep 5 entries per frame having the top-5 confidence score. Even tough DenseCap already has a language model, we train an auto-encoder [1] with all the selected sentences amongst all scenes as training set. It allows us to perform an in-domain training (4915 sentences with a 477 words vocabulary) and to define ourselves the feature-size for one sentence. Once the training has converged, we extract a 64-d vector per sentence. A semantic knowledge of a single frame is thus a concatenation of five 69-d vectors (64-d for the sentence, 4-d for the anchor coordinates and one scalar for the confidence score).

### 3.2 Semantic architecture




For our semantic model (left), there is two new inputs. First, the history semantic vectors from past frames (n=4) and secondly the semantic vector about the target frame. We now use two $\boldsymbol{W}_1$ matrices. $\boldsymbol{W}_{v_1}$ is still of size $8192 \times 512$ and $\boldsymbol{W}_{v_2}$ is of size $345 \times 512$. The two visual inputs (history frames and visual targets) stays as explained in section 2.1. When testing this architecture, we don't handpick object-oriented target as previously explained in section 2.2. We automatically define new top-semantic targets by taking, per scene, the top-5 targets that have overall the highest score confidences from DenseCap. A target to reach is now a frame the agent has the best semantic information about.

## 4 Results

We now detail our results in two parts: first we illustrate the convergence of the reward (Figure 4) from the baseline model being trained on random targets or the hand-picked object-oriented targets from section 2.2. Finally, we describe the results of the semantic model using the top semantic targets.

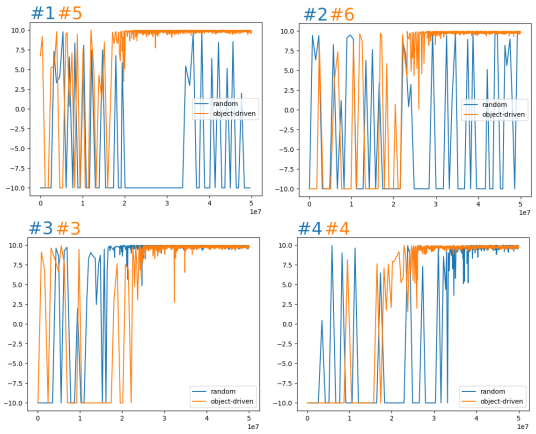



Figure 4: Evolution of the reward for the targets of Figure 2 ranging from 0 to 50 millions training frames.

**Baseline** Each graph of Figure 4 contains two training curves (or reward evolution) of a target. The orange curve depicts a baseline model trained on the hand-picked object-driven targets and, in blue, a baseline model trained on random targets. The target trained is given by the number above each graph and corresponds to a target presented in Figure 2. As we see, the blue model didn't converge for the random target #1 and #2. Indeed, both target are object-less: an empty corridor and a wall whilst orange model for target #5 and #6 has converged. To compare performances, both baseline models have frames #3 and #4 (that are random) within their training pool of 100 targets. We see that being trained on object-oriented target benefits the overall training: orange converges quicker for frames #3 and #4.

**Semantic** We evaluate the model's ability of 1) generalization across targets (the scene is already seen, but not the targets) and 2) the generalization across scene (the scene has never been seen). We compare the semantic model with our baseline model and [12]. After training, each target is evaluated

[1] `https://github.com/erickrf/autoencoder`

for 100 episodes and then averaged. We consider an episode successful if the target is reached in less than 1000 actions. For task 1), we train the models for 5 millions frames shared along all scenes and then evaluate on one seen scene picked randomly. For task 2), the models are trained for 10 millions frames shared along all scenes except one per scene-type. The latter are used for evaluation. All models are trained on the top-semantic targets as described in 3.2 unless stated otherwise.

| **1)** | **Bedroom** | | **Bathroom** | | **Kitchen** | | **Living** | | **2)** | **Bedroom** | | **Bathroom** | | **Living** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | E.L. | % | E.L. | % | E.L. | % | E.L. | % | | E.L. | % | E.L. | % | E.L. | % |
| R | 963 | 6.6 | 941 | 9.4 | 988 | 6.4 | 968 | 4.8 | | 941 | 8.6 | 928 | 10.7 | 991 | 2.8 |
| [12] | 903 | 19.9 | 691 | 71.4 | 885 | 23.9 | 921 | 16.2 | | - | - | - | - | - | - |
| B | 893 | 22 | 611 | 78.2 | 890 | 22.1 | 901 | 18.1 | | 921 | 9.6 | 861 | 12.4 | 966 | 4.1 |
| S | 619 | 66.2 | 361 | 94.1 | 824 | 32.7 | 882 | 22.1 | | 861 | 14.5 | 798 | 19.1 | 954 | 4.7 |
| S_H | 699 | 61.9 | 358 | 92.4 | 870 | 31.5 | 906 | 18.1 | | - | - | - | - | - | - |

Table 1: Score table. R = Random, B = baseline model, S = semantic model, E.L. = mean episode length (rounded), % = episode success rate. S_H is the semantic model trained on hand-picked target from section 2.2 rather than top-semantic targets. Task 2 is not performed on kitchen as we have only one instance.

## 5 Conclusion

For task 1, our semantic model (S) is able to generalize to new targets within trained scene with a low amount of training frames. We also see that the baseline model (B) using scene-type policy perform better overall than a scene-specific policy ([12]). Importantly, training the semantic model on the top-semantic target (S) instead of hand-picked target (S_H) offers a better generalization to new targets. Task 2 is harder but nonetheless, the semantic model still comes ahead. It it possible that more training frames are required to transfer knowledge to new scenes.

## 6 Acknowledgements

This work was partly supported by the Chist-Era project IGLU with contribution from the Belgian Fonds de la Recherche Scientique (FNRS), contract no. R.50.11.15.F, and by the FSO project VCYCLE with contribution from the Belgian Waloon Region, contract no. 1510501.

## References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[3] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[4] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*, 2017.

[6] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[7] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

[8] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, 123(1):32–73, May 2017. ISSN 0920-5691. doi: 10.1007/s11263-016-0981-7. URL `https://doi.org/10.1007/s11263-016-0981-7`.

[9] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `http://proceedings.mlr.press/v48/mniha16.html`.

[10] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018.

[11] Haonan Yu, Haichao Zhang, and Wei Xu. Interactive grounded language acquisition and generalization in a 2d world. In *ICLR*, June 2018.

[12] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3357–3364. IEEE, 2017.