Visual Navigation with Spatial Attention

Bar Mayo
Technion
mayo.bar@gmail.com

Tamir Hazan
Technion
tamir.hazan@technion.ac.il

Ayellet Tal
Technion
ayellet@ee.technion.ac.il

(a) Paths—Ours & [30]'s (b) Our agent's view (c) Our attention (d) [30]'s view

Figure 1. **Visual navigation.** (a) The agent aims at finding a TV (red rectangle) in a living room (top view), starting from a given location (black circle). Our agent's path is marked in orange and [30]'s path is in magenta. At each step, the agent is given a specific view, depending on its position. In this example, our agent starts by turning around in its starting location to gather information—a strategy it has learned. (b) shows our agent's view before the first move forward, whereas (d) shows [30]'s view before its first move forward. (c) shows our attention model, which combines semantic and spatial information of (b)'s view; it directs our agent to move forward, towards the TV. Differently, the view in (d) is part of [30]'s lengthy exploration (magenta path in (a)) after the sought-after TV.

Abstract

This work focuses on object goal visual navigation, aiming at finding the location of an object from a given class, where in each step the agent is provided with an egocentric RGB image of the scene. We propose to learn the agent's policy using a reinforcement learning algorithm. Our key contribution is a novel attention probability model for visual navigation tasks. This attention encodes semantic information about observed objects, as well as spatial information about their place. This combination of the "what" and the "where" allows the agent to navigate toward the sought-after object effectively. The attention model is shown to improve the agent's policy and to achieve state-of-the-art results on commonly-used datasets.

1. Introduction

Human and animals can navigate new environments relatively well. This adaption to new surroundings, although natural, is not trivial. It requires to find parallels between the new observations and our past experience. This is largely possible due to our ability to sort through new visual information and intelligently focus on the most relevant semantic cues. For instance, when looking for a toaster in a previously-unvisited kitchen, our intuition is to look for the refrigerator, while ignoring other "irrelevant" information, since our past experience indicates that the toaster is usually located not far from the refrigerator.

Object goal visual navigation tasks include two basic components: semantic understanding of the scene and path planning [28, 3, 14]. With the increase of data and computation power, reinforcement learning algorithms excelled in learning policies for these two components jointly in an end-to-end manner [25, 21, 13, 6]. As a result, many extensions to visual navigation were presented, including tasks specified by natural language instructions [7], by a desired goal image [37], or by a target object [30, 10]. Reinforcement learning of spatial and semantic relations is a fundamental challenge for these tasks [31].

This work focuses on object goal visual navigation, where the goal is to find an instance of the target object class (Figure 1). Like previous works, we utilize reinforcement learning. We propose to improve the agent's policy by encoding semantic information about observed objects using a convolutional net, as well as spatial information about their place, using an attention probability model. This combination of both semantic and spatial information, i.e., of "what"

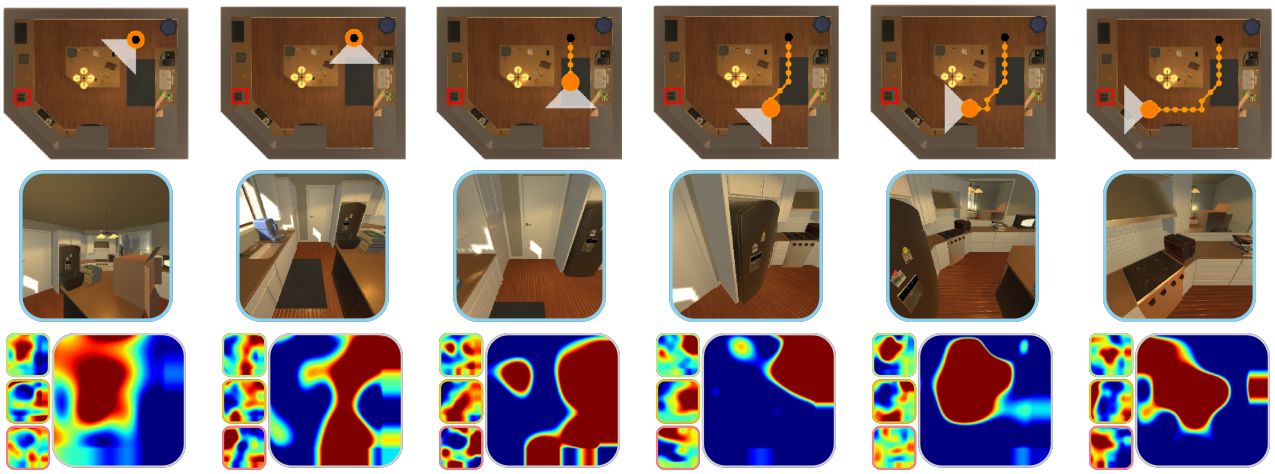


Figure 2. **Path sampling.** The first row shows a top view of the scene, with the path thus far, along with the agent's view (white triangle) at this step. The second row shows the image the agent views at this step. The third row shows the fused attention map per step, as well as the three maps that build it. The agent is looking for a toaster (red rectangle) and starts from the opposite side of the kitchen. In Step 1 the agent focuses on the refrigerator, which is an indicator to a nearby toaster; in Step 11 it moves toward the refrigerator; in Step 16 it decides to turns right and then in Step 22 the toaster becomes visible, at which point the agent's focus switches from the refrigerator to the toaster and the agent turns right; in Step 24 it starts moving forward, toward the toaster; in Step 29 it is sufficiently close and declares `Done`.

and of "where", allows the agent to navigate towards the sought objects effectively. Our novel attention mechanism consists of three types of attention probability models for navigation: target attention that considers the target information in the image; action attention that takes into account the last action of the agent; memory attention that considers the agent's previous steps in the scene. Our attention probability model results in an attended embedding, which preserves the semantic and spatial information of objects.

We validate our approach using the AI2-THOR [37] environment. We use Wortsman et al. [30] setup with their scenes from four room categories: kitchen, living room, bedroom and bathroom, where an agent is navigating to a given object using only visual observations. In our experimental validation we show that not only we outperform the state-of-the-art, but also our attention unit carries spatial information about the objects. This is achieved using a probability distribution over areas of the observed image that are represented by the spatial locations of the topmost convolutional neurons of a standard convolutional net (e.g., ResNet18). As this attention probability distribution preserves the spatial information that is fed to the reinforcement learner, it controls the areas of the image that the agent considers when improving its policy. Hence, this attention unit also carries the promise to explain the agent's actions in visual navigation tasks.

Figure 2 illustrates this promise. For instance, in Step 1 the attention map suggests that the agent focuses on the refrigerator, which is a good indicator to the location of the toaster. Similarly, once the toaster becomes visible in Step 22, the attention map switches from focusing on the refrigerator to focusing on the sought-after toaster, and in accordance with that, the agent turns right.

Hence, this paper makes three contributions:

1. We propose a novel attention mechanism that suits navigation. It consists of three types of attentions: target, action, and memory.

2. We present an end-to-end reinforcement learning framework that realizes the attention mechanism and achieves state-of-the-art results.

3. An added benefit of the different attention maps is being able to explain the agent's actions through visualization.

2. Related Work

Navigation is one of the most fundamental problems in mobile robotics. Traditional navigation approaches decompose the problem into two separate stages: mapping the surrounding and planning a path to the goal [3, 8, 9, 14, 16, 28]. Generally, these works treat navigation as a purely geometric problem. Reinforcement learning (RL) methods were applied to learn policies for robotic tasks [17, 18, 21, 25]. While RL methods are able to learn complex tasks in an end-to-end manner, their main challenge in visual navigation tasks is to understand both the visual cues as well as the navigation plan. Recently, Shen et al. [27] fused different visual representations in navigation. In a related thread,

Gupta et al. [13] developed a cognitive mapping and planning approach whose map. Similarly, [12, 5] build semantic maps in a pre-exploration setting to capture spatial information in visual navigation. While our approach also uses a latent spatial representation, it differs in important respects: our spatial information relies on an attention probability distribution over areas in the image. This component serves as an important building block in our attended embedding, which combines both the spatial and semantic information of the image.

Target-driven visual navigation tasks have been proposed to search for an object in visual scenes. Zhu et al. [37] address target-driven navigation given a picture of the target, while Mousavian et al. [22] augment the learner with semantic segmentation and detection masks. Chaplot et al. explores visual navigation given language instructions using gated attention [7] and semantic mapping [6]. In contrast to our work, they use an attention module to represent their language modality, while we use attention probability distribution over areas of the image to better understand the spatial information. More broadly, Bayesian methods for visual navigation with that relation graphs appear in [31, 1].

We validate our visual navigation approach on AI2-THOR [37], which is an environment that consists of near photo-realistic 3D indoor scenes [19]. We augment the work on self-adaptive visual navigation (SAVN) of Wortsman et al., [30] with attended observation that serves an input to its model agnostic meta-learner (MAML) [11]. Other approaches for visual navigation were applied to AI2-THOR, e.g., learning scene priors using graph convolutional nets that are able capture the relationships between objects in the scene [34]. Recently, Du et al. augmented the AI2-THOR environment with detection information [10] albeit for different scenes.

Our work develops an embedded attention module that combines both semantic and spatial information [29, 26]. The spatial information is encoded by an attention probability distribution over areas in the image and the semantic information of these areas is encoded by a convolutional net. Attention in visual tasks has mainly been deployed for language augmented tasks [2, 15, 23, 32, 33, 35, 36, 4, 20]. Similar to our work, they construct an attention probability distribution over areas of the image. However, these attention units typically summarize the convolutional net representation by averaging with respect to the attention probability distribution. In contrast, we refrain from averaging and preserve the spatial dimension of the convolutional layer, which significantly improves its performance on navigation tasks. Similar to our work, natural language attention models, and in particular multi-head attention [29], use probability models to re-embed their preceding layer. However, they do not retain the spatial information of the image and their attended embedding ignores this information.

3. Attended Navigation

Our navigation task $\tau \in \mathcal{T}$ considers a scene S, a starting point p and a target object o. The goal of a task $\tau = (S, p, o)$ is to move an agent in a 3D indoor scene from the starting position p to an instance of the target object class o with a minimum number of steps.

The navigation is preformed by a mobile agent and is learned by a policy. The agent's policy is limited to six actions: `MoveAhead`, `RotateLeft`/`RotateRight`, `LookDown`/`LookUp`, `Done`. At each step the agent is given an egocentric RGB image $s \in S$ from the scene S and a target object class o and it can act in one of two ways: (1) choose one of the possible movement directions and move accordingly or (2) issue the `Done` action, signalling that the agent believes it has finished the task. The `Done` action ends the trial of navigation, termed an *episode*.

An episode is finished successfully if the agent issued a `Done` action and (1) The target object is sufficiently close to the agent (1 meter in practice); (2) The target object is in view; and (3) The agent did not pass the maximum number of allowed steps.

Following the *Self-Adaptive Visual Navigation (SAVN)* framework of Wortsman et al. [30], we learn a policy $\pi_\theta(\cdot|s)$, which chooses an action a given an egocentric RGB image s within a scene S. We use gradient decent (policy gradient) to improve the policy's parameters θ to navigate in each episode. These parameters are learned in order to maximize the expected reward $\mathbb{E}[\mathcal{R}^\tau]$ on a sequence of actions in a given episode. In our experimental evaluation, we use the SAVN navigation reward $\mathcal{R}^\tau_{\text{nav}}$ that subtracts 0.01 for any step except `Done` and adds 5 for a successful navigation. We also use the actor-critic family algorithms that minimize its navigation loss $\mathcal{L}^\tau_{\text{nav}}(\theta, a)$, which consists of the negative expected reward that serves the actor and a learned value function that serves the critic.

3.1. Adaptive navigation

SAVN [30] relies on adaptive navigation, hence its policy benefits from adapting to the relevant navigation subtask, e.g., entering a hallway, approaching a refrigerator and so on. To deal with such a complex task, SAVN applies model agnostic meta-learning (MAML) that shifts the parameters of the policy as the agent interacts with the scene. This shift of parameters allows the agent to adapt to the scene while interacting with it. SAVN achieves this behavior by using an interaction loss $\mathcal{L}^\tau_{\text{int}}(\theta, \alpha)$ that is being applied on a $\hat{k}$-prefix α of actions a, i.e., $\alpha = (a_1, ..., a_{\hat{k}})$. Thus the loss function for learning a policy $\pi_\theta(\cdot|s)$ for a task τ for a sequence of actions a and their prefix α is

$$\min_\theta \sum_{\tau \in \mathcal{T}_{\text{train}}} \mathbb{E}_{a \sim \pi_\theta} \Big[\mathcal{L}^\tau_{\text{nav}}\big(\theta - \nabla_\theta \mathcal{L}^\tau_{\text{int}}(\theta, \alpha), a\big) \Big]. \qquad (1)$$

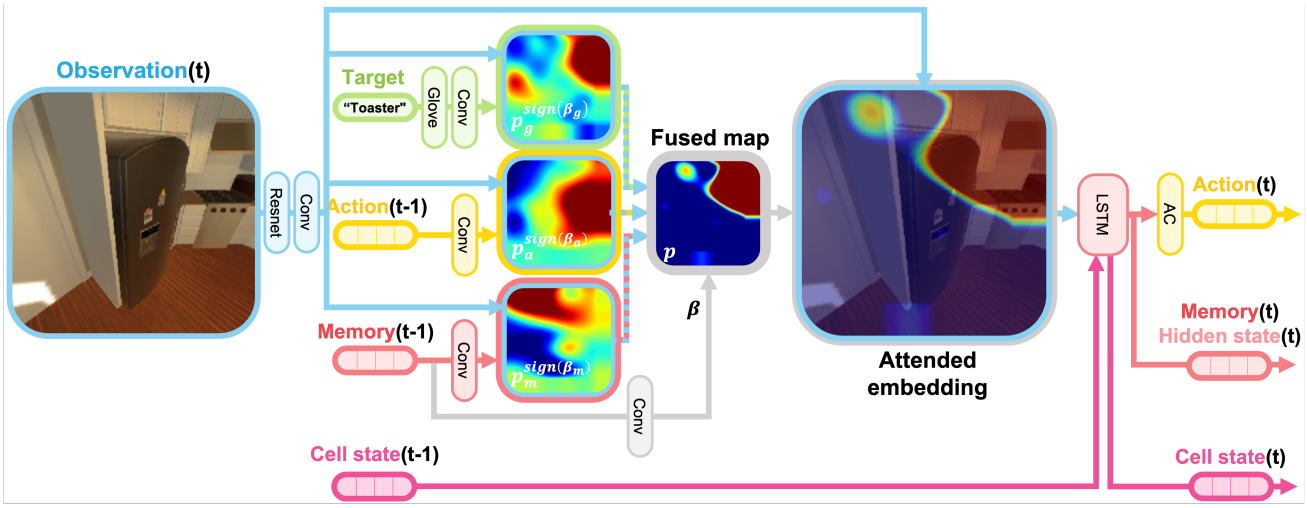


Figure 3. **Architecture overview.** The adaptive navigation unit, which is described in Section 3.1 follows Wortsman et al. [30]. The attended embedding, described in Equation 2, encodes semantic information about observed objects using a convolutional net, as well as spatial information about their place, using the fused attention probability distribution. The fused attention, described in Section 3.2.4 balances the target/action/memory attention distributions. The target attention, described in Section 3.2.1, combines the target word GloVe embedding with the image information. In this example, the target word is "toaster" and $t = 22$. One can verify that the inferred probability distribution overlaps the area in the image that contains part of the toaster (the black rectangle to the right). The action attention, described in Section 3.2.2, combines the last action of the actor with the image information. In this example, the action's attention probability distribution is focused on the area on the right of the image, and the agent is about to turn right to locate the toaster. The memory attention, described in Section 3.2.3, summarizes the agent's experience and aims to focus on sections of the image based on the information already gathered in the episode. In this example, the memory attention probability distribution takes into account the refrigerator, as it was learned to be a correlated to the toaster class in a kitchen.

We also learn parameters ϕ of the interaction loss, although we omit this dependence for readability. This loss essentially minimizes the navigation loss while encouraging the gradient $\nabla_\theta \mathcal{L}^\tau_{\text{int}}(\theta, \alpha)$ to be similar to the gradient $\nabla_\theta \mathcal{L}^\tau_{\text{nav}}(\theta, a)$. This allows to adjust the policy parameters in test time, to reduce the navigation loss of a single trajectory, i.e., to better adapt the policy to navigation subtasks.

We introduce spatial attention into the SAVN framework. Intuitively, attention may improve navigation by orienting the agent to the correct direction. We show that this is indeed the case and that we outperform SAVN using a spatial attention mechanism that takes into account the target, the agent's actions, and the memory of images seen so far. Hereafter, we present our novel attention mechanism, designed particularly for efficient visual navigation in 3D, and explain how we incorporated it into the architecture.

3.2. Spatial embedding using attention

Visual navigation requires not only semantic reasoning, but also spatial reasoning. This is due to the fact that we control an agent that interacts with a 3D environment. In our work we learn a policy for navigating in the 3D space given an egocentric RGB image. In the following we present an approach that is able to encode semantic information about observed objects using convolutional net, as well as spatial information about their place, using an attention probability model. Our approach is illustrated in Figure 3.

The navigation is preformed by a mobile agent and is learned by a policy $\pi_\theta(\cdot|s_0)$, which chooses an action given an egocentric RGB image s_0 at the beginning of the episode, within a scene S. The policy samples its actions iteratively. At time t the agent is given the egocentric image s_t and chooses the action a_t.

We use convolutional nets to extract semantic information about a given image in the scene, as they were proven to be very effective in encoding mid-level and high-level semantic information in the image. We encode the t^{th} image by the spatial locations of the topmost convolutional layer, whose dimension is $n_v \times n_v \times d_v$, of a standard convolutional net (ResNet18) that is pretrained on Imagenet. The spatial location of each topmost convolutional neuron is indexed by $i, j = 1, ..., n_v$ and its $(i, j)^{th}$ location corresponds to an area in the observed image and is described by the vector $v^t_{i,j} \in \mathbb{R}^{d_v}$. In the following we refer to the area that is represented by such $(i, j)^{th}$ neuron, namely $v^t_{i,j}$, as the $(i, j)^{th}$ sub-window in the image. We then emphasize the spatial information of the objects in the relevant sub-windows using an attention probability distribution.

At each time step of the agent, we construct an attention probability distribution over the $n_v \times n_v$ sub-windows of the input image. Intuitively, this probability distribution as-

signs high probability to sub-windows that have relevant information in the image and assigns low-probability to sub-windows that do not. By doing so, the attention probability distribution introduces spatial information to the process. Our attention probability distribution is composed of three attention units: (i) target attention unit, which incorporates the target information in the image; (ii) action attention unit, which takes into account the agent's last action; (iii) memory attention unit, which "remembers" relevant information from previously-seen images in the scene. These three distributions over the $n_v \times n_v$ sub-windows are then fused into a single attention probability distribution over the image sub-windows. We denote by $p^t(i,j)$ the fused probability distribution at time t over the $n_v \times n_v$ sub-windows $i,j = 1,...,n_v$.

The spatially attended embedding, $\hat{v}^t_{i,j}$, of the t^{th} image combines both the semantic information in the image as well as the spatial information about the location of the different objects. The semantic information is represented by the vectors $v^t_{i,j} \in \mathbb{R}^{d_v}$, while the spatial information is represented by the attention probability distribution $p^t(i,j)$. We combine these two components using the pointwise multiplication: $\hat{v}^t = p^t \odot v^t$, which is defined by

$$\hat{v}^t_{i,j} = p^t(i,j) \cdot v^t_{i,j}. \tag{2}$$

The dimension of the attended embedding is the same dimension as the image embedding $v^t_{i,j}$. Intuitively, the attention probability distribution $p^t(i,j)$ has high values for relevant $(i,j)^{th}$ sub-windows of the image, i.e., sub-windows that contain semantic information for the visual navigation task. Equivalently $p^t(i,j) \approx 0$ for irrelevant sub-windows. Hence, the attended embedding in Equation 2 consists of the vector $\hat{v}^t_{i,j} \approx 0$ whenever $p^t(i,j) \approx 0$, i.e., for sub-windows that are irrelevant for navigation in the t^{th} step. Equivalently, for semantically meaningful sub-windows the attended embedding is similar to the original image embedding, i.e., $\hat{v}^t_{i,j} \approx v^t_{i,j}$. This embedding highlights the spatial locations of the semantically meaningful sub-windows and populates them with the respective semantic information of the image. This embedding allows the agent to choose its next step according to both the semantic and the spatial information of the image, as it is fed as the input to the navigation policy; see Figure 3.

3.2.1 Target attention unit

This unit learns a probability distribution function over the image. It gets as input the image at the t^{th} step and the target (given by a word) and aims to focus on target-relevant information in the image, including the target and visual clues for the target's location. For example, if the target is a soap bottle, which is invisible, the agent should focus on the bathtub or the sink, since soap bottles are usually found next to them. In other words, we want to learn the interaction of each sub-window in the image with the target.

The target word is encoded by a vector of length d_g; in our system we used the GloVe embedding [24]. We denote by $u_g \in \mathbb{R}^{d_g}$ the GloVe embedding and by $v^t_{i,j} \in \mathbb{R}^{d_v}$ the $n_v \times n_v$ image vectors at the t^{th} time step, for $i,j = 1,...,n_v$. The interaction of the word vector u_g with an image sub-window embedding $v^t_{i,j}$ relies on the inner product of these vectors, after embedding both representations in a d-dimensional space.

Let $W_v \in \mathbb{R}^{d \times d_v}$ be trainable parameters that embed a sub-window embedding $v^t_{i,j}$ in the d-dimensional space, and let $W_g \in \mathbb{R}^{d \times d_g}$ be trainable parameters that embed the target embedding u_g in the same space. For every sub-window index $i,j \in \{1,...,n_v\}$, the visual-target attention potential $\phi^t_g(\cdot)$ at time t takes the form:

$$\phi^t_g(i,j) = \left\langle \frac{W_v v^t_{i,j}}{\|W_v v^t_{i,j}\|}, \frac{W_g u_g}{\|W_g u_g\|} \right\rangle. \tag{3}$$

We apply ℓ_2-normalization before the multiplication, i.e., we use the *cosine* similarity to compute the potential interaction between the target u_g and the image sub-window $v^t_{i,j}$. The corresponding attention probability distribution is attained by applying the softmax operation:

$$p^t_g(i,j) = \frac{e^{\phi^t_g(i,j)}}{\sum_{s,t=1}^{n_v} e^{\phi^t_g(s,t)}}. \tag{4}$$

See Figure 3 for an example of the target attention probability distribution, $p^t_g(\cdot)$, for a target word "toaster". One can verify that the inferred probability distribution is focused on the area in the image that contains the toaster.

3.2.2 Action attention unit

This unit gets as input the image and the last step's action distribution. In practice, the action probability distribution correlates with the agent's movement. The actor's actions are sampled from the policy $\pi_\theta(\cdot|s)$ that chooses an actions, given an egocentric RGB image $s \in S$. At time $t = 1,...,k$, the agent samples an action a_t from one of the six actions $a_t \in \{$`MoveAhead`, `RotateLeft`,`RotateRight`, `LookDown`, `LookUp`, `Done`$\}$. At step t, the policy $\pi_\theta(\cdot|s)$ utilizes the actions distribution at time $t-1$, which we denote by $u_a^{(t-1)} \in \mathbb{R}^{d_a}$.

Similarly to the target attention unit, each of the $n_v \times n_v$ sub-windows of the image that is seen by the agent at time t is encoded by a vector of length d, using the matrix W_v. We also embed $u_a^{(t-1)}$ to the d dimensional space by the learned matrix $W_a \in \mathbb{R}^{d \times d_a}$. For every sub-window index $i,j \in \{1,...,n_v\}$, the visual-action attention potential $\phi^t_a(\cdot)$

at time t takes the form:

$$\phi_a^t(i,j) = \left\langle \frac{W_v v_{i,j}^t}{\|W_v v_{i,j}^t\|}, \frac{W_a u_a^{(t-1)}}{\|W_a u_a^{(t-1)}\|} \right\rangle. \quad (5)$$

Importantly, the action potential function considers the observed image at time t and the preceding action (at time $t-1$). The corresponding attention probability distribution is attained by applying the softmax operation:

$$p_a^t(i,j) = \frac{e^{\phi_a^t(i,j)}}{\sum_{s,t=1}^{n_v} e^{\phi_a^t(s,t)}}. \quad (6)$$

An example of the action attention probability distribution $p_a^t(\cdot)$ appears in Figure 3.

3.2.3 Memory attention unit

The memory attention unit summarizes the agent's experience and aims to focus on sections of the image based on the information already gathered in the episode. For example, the agent should avoid focusing its attention on irrelevant areas that were previously explored. This unit gets as input the image and the agent's gathered experience in the episode up to the t^{th} step, i.e., the actions, the observed images, and the internal state representations that appear in the $(t-1)^{th}$ prefix of the agent's trajectory $(a_1, ..., a_{t-1})$. This experience is represented by the hidden state of an LSTM-cell, whose input is the spatially attended observed image; its output is fed into the actor-critic module.

As before, this unit learns a probability distribution function over the observed image at time t. The $n_v \times n_v$ sub-windows are encoded by vectors of length d using the matrix W_v. The memory is extracted from the hidden state of an LSTM-cell at time $t-1$. We denote this state by $u_m^{(t-1)} \in \mathbb{R}^{d_m}$ and embed it in the d^{th} dimensional space by the learned matrix $W_m \in \mathbb{R}^{d \times d_m}$. For every sub-window index $i, j \in \{1, ..., n_v\}$, the visual-memory potential $\phi_m^t(\cdot)$ at time t takes the form:

$$\phi_m^t(i,j) = \left\langle \frac{W_v v_{i,j}^t}{\|W_v v_{i,j}^t\|}, \frac{W_m u_m^{(t-1)}}{\|W_m u_m^{(t-1)}\|} \right\rangle. \quad (7)$$

The corresponding attention probability distribution is:

$$p_m^t(i,j) = \frac{e^{\phi_m^t(i,j)}}{\sum_{s,t=1}^{n_v} e^{\phi_m^t(s,t)}}. \quad (8)$$

The role of the memory attention probability distribution is demonstrated in Figure 3.

3.2.4 Fused attention unit

The different attention units are constructed to capture different behaviors of the agent. However, these attention maps should be fused to summarize the target, the action and the memory attentions, which are represented by the respective probability distributions over the image.

Naively, we can fuse these three probability distributions by normalizing their product $p_g^t(i,j) \cdot p_a^t(i,j) \cdot p_m^t(i,j)$. This allows to fuse the attention while accounting for each probability in a symmetric manner. However, this does not allow to learn the importance of each probability at time step t. Instead, we learn the importance of each probability at time t by considering the hidden state of the LSTM-cell at time $t-1$, denoted by $u_m^{(t-1)}$. Specifically, we learn the real-valued weight functions $\beta_g(u_m^{(t-1)}), \beta_a(u_m^{(t-1)}), \beta_m(u_m^{(t-1)})$ to fuse the different attention probability distributions at time t. We use the short hand notation $\beta_g, \beta_a, \beta_m$ for these functions and attain the fused attention probability distribution:

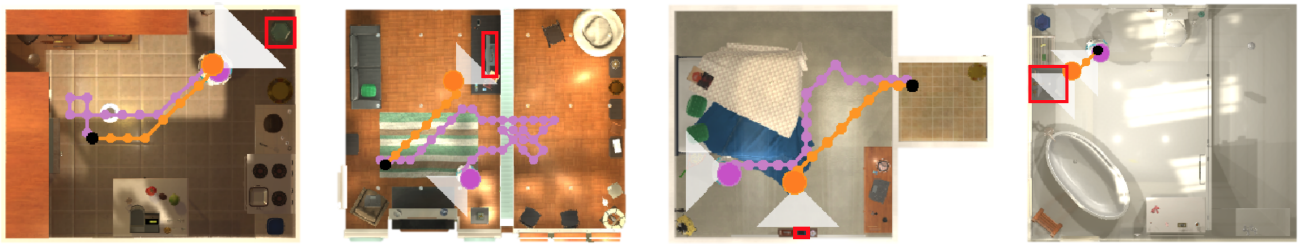
$$p^t(i,j) \propto \left(p_g^t(i,j)^{\beta_g} p_a^t(i,j)^{\beta_a} p_m^t(i,j)^{\beta_m} \right). \quad (9)$$

The fused attention is able to combine all attention probability distribution to a coherent distribution, see Figure 3.

4. Experimental Validation

We follow Wortsman et al. [30] and train and evaluate our models using the AI2-THOR [37] environment with their scenes from the four room categories: kitchen, living room, bedroom and bathroom. For each room type, we use the same $20/5/5$ split of train/validation/test for a total of 120 scenes. The objects in their scenes, per room type are: 1) Living room: pillow, laptop, television, garbage can, box, and bowl. 2) Kitchen: toaster, microwave, refrigerator, coffee maker, garbage can, box, and bowl. 3) Bedroom: plant, lamp, book, and alarm clock. 4) Bathroom: sink, toilet paper, soap bottle, and light switch. We also use the reward function of [30], with reward of 5 for finding the object and -0.01 for taking a step. We learn a policy for this reward using actor-critic reinforcement learner with an advantage function and 12 synchronous agents (A2C). The scene, initial state of the agent and the target object were chosen by [30] and for each training run we select the model that performs best on the validation set in terms of success.

The agent moves with the `MoveAhead` action. The `RotateLeft` and `RotateRight` actions occur in increments of $45°$, while the `LookDown` and `LookUp` actions tilt the camera by $30°$. During training, the maximal trajectory consists of 30 actions and during validation and testing the maximal trajectory is limited to 200 actions to living rooms and 100 actions to other room types. The agent successfully completes a navigation task if it performs the `Done` action when an instance from the target object class is within 1 meter from the agent's camera and within the agent's field of view.



(a) Both reached the goal; our path is shorter

(b) Reaching the goal thanks to semantic external clues

(c) Reaching the goal despite target's irregular position

(d) Reaching the goal while SAVN is too far from target

Figure 4. **Qualitative results.** This figure compares our agent's trajectories to those of (SAVN) [30]. The starting position of the agent is drawn as a black circle and the target as a red box; the path of our agent is in orange and SAVN's path is in magenta; accordingly the large orange/magenta points show the final locations of the respective agents. (a) While both agents reach the target (a garbage can), our path is shorter, since our attention model enables our agent to gather information on the scene early on. This is also expressed in the SPL evaluation in Table 1. (b) Our agent found the TV, whereas SAVN misses it, probably due to ignoring the spatial cues in the living room, such as the carpet or the TV stand; (c) The alarm clock is situated in irregular position (near the dresser) and is very small. While our agent is able to focus on a small region in the observation and locate the alarm clock, SAVN's agent continues its search near the bed and misses it. The lack of attention unit in SAVN results in more emphasis on object locations than on visual characteristics. (d) Our agent found a sink, whereas SAVNs agent stopped too far, and yet declared it found the target; i.e. the distance estimation is wrong.

Architecture	SPL	Success	SPL $L \geq 5$	Success $L \geq 5$
Scene Prior [34]	15.47	35.13	11.37	22.25
SAVN [30]	16.15	40.86	13.91	28.70
Ours (A3C)	16.99	43.20	15.51	31.71
Ours (A2C)	**17.88**	**46.20**	**15.94**	**32.63**

Table 1. **Quantitative results.** Our best results are attained for synchronous actor-critic learner (A2C). However, our asynchronous learner (A3C) outperforms the asynchronous learner of [30] as well.

The methods are evaluated using both Success Rate and Success weighted by Path Length (SPL). Success is defined as $\frac{1}{N}\sum_{i=1}^{N}\mathcal{S}_i$ where $N = 1000$ is the number of episodes (250 episodes for each scene type in the test set) and $\mathcal{S}_i$ is a binary indicator of success in episode i. The SPL is defined as $\frac{1}{N}\sum_{i=1}^{N}\mathcal{S}_i\frac{L_i}{\max(P_i,L_i)}$ and it measures the quality of the agent's path when it succeeds in finding the object in episode i, where P_i denotes path length and L_i is the length of the optimal trajectory to any instance of the target object class in that scene. As the behavior of the agent's policy is different for short and long paths. we also refer to trajectories where the optimal path length is at least 5 and denote this by $L \geq 5$ (L refers to optimal trajectory length).

Table 1 compares our results to the state-of-the-art and shows improvement over previous works in terms of both success rate and path length (SPL), for short paths as well as for long paths. During our experimental validation we noticed that synchronization is important to get stable results over different platforms and GPUs. Empirically, our best results are attained for synchronous actor-critic learner (A2C) rather than asynchronous actor-critic learner (A3C). Nevertheless, our asynchronous learner outperforms asynchronous learner of SAVN [30].

Figure 4 shows different scenarios and compares the behavior of our agent to that of SAVN. It demonstrates how the added spatial attention information allows our agent to better navigate in the $3D$ Euclidean space. In particular, in all these scenarios our agent reaches the goal efficiently whereas SAVN's agent (a) takes longer to find the target object; (b) misses the target, as its non-optimal trajectory sets it on a path for which the object is not in view; (c) misses the target due to being small and situated in a irregular position; (d) stops too far. Our attention model enhances the agent's ability to notice and use visual clues. This is turn, manifests in better object mapping and path planing.

Recall that our spatial attention is based on three attention probability distributions that are based on the target, the agent's previous action and the agent's memory, which is represented in its LSTM-cell. These three distributions are fused to a single attention probability model, which weighs the three factors according to the agent's LSTM-cell. Figure 5 provides a quantitative assessment of the weights, $\beta_g(u_m^t), \beta_a(u_m^t), \beta_m(u_m^t)$, which control the fused distribution in Equation 9. We also compared the aforementioned fused module with a more expressive baseline that concatenates the three attended feature maps and use it as the hidden feature for the actor-critic module. While our system requires less parameters (7.5M vs. 20.4M) it performs better than the expressive baseline: our SPL is 31% better, Success is 22% better, SPL $L \geq 5$ is 41% better, and Success $L \geq 5$ is 30% better than the baseline.

We also tested our attention module in the densely annotated setting of [10]. This setting use the simulator information to extract detected objects which allows us to integrate their information to the attention module. In this setting

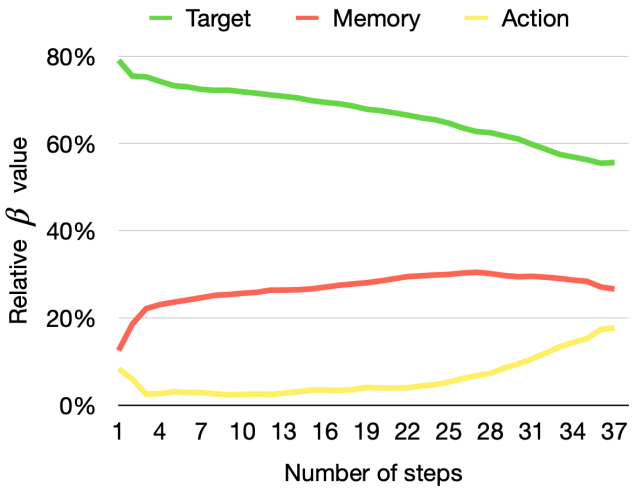


Figure 5. **The β weights.** This graph shows how $\beta_g(u_m^t), \beta_a(u_m^t), \beta_m(u_m^t)$ of the target/action/memory attention units change along the navigation. The y-axis is proportion of the respective unit, e.g., $|\beta_g(u_m^t)| / \sum_{i \in g,a,m} |\beta_i(u_m^t)|$ and the t^{th} tick of the x-axis is the average of the respective proportion for all test episodes in their t^{th} step. We capped t by 37 as there are negligible number of trajectories that have more than 37 steps.

we our attention module, built over the architecture of [10], achieve an improvement of 9.2% in Success rate, 9.1% in SPL, 13.7% in Success $L \geq 5$, 13.2% in SPL $L \geq 5$, when measured over the 4000 test scenes. Also, since our attention adds spatial information to our agent, we note that the number of test images in which we detected the target object is 15.8% higher in our case than when using [10].

Ablation study. The aim of the ablation study is to verify the validity of our attention unit and the importance of its different components. Table 2 compares our attended embedding with the state-of-the-art multi-head embedding of the transformer [29] (MHA in the table)). The difference in performance is related to the different embedding strategies of the two methods. The transformer embeds the data using learned probability distributions (of key and query) and learns representation (value). While this embedding is very effective in language processing, its embedding ignores spatial information in visual tasks.

Table 2 also shows how our fused attention model performs when we take out its target/action/memory attention components. One can verify that each of the components is vital to gain good performance. The target attention unit is the most important module, as we are focused on target-driven visual navigation, while the contribution of the action & memory units is comparable. We also see the importance of changing the balance β of the various attention units, as when we use a fixed $\beta = 1$ during learning and testing, the results deteriorate.

Architecture	SPL	Success	SPL $L \geq 5$	Success $L \geq 5$
SAVN [30]	16.15	40.86	13.91	28.7
Ours (MHA)	9.80	30.70	8.29	20.36
Ours w/o p_g	9.41	29.60	7.93	19.01
Ours w/o p_a	14.41	41.00	13.07	29.64
Ours w/o p_m	15.39	45.6	14.07	**33.08**
Ours $\beta = 1$	13.88	38.80	11.30	26.35
Ours (A2C)	**17.88**	**46.20**	**15.94**	32.63

Table 2. **Ablation study.** The table presents the significance of our attention module and its separate units. Replacing our attention unit with multi-head attention of [29] (MHA) decreases our success rate by 8% for all paths and 7% for long paths ($L \geq 5$). Omitting the target attention unit decreases our success rate by about half, while the contributions of the action and memory units are comparable. Fixing the balance β of the various attention units during a test episode decreases our success rate by more than 20%.

(a) erronous classification (b) hidden object

Figure 6. **Limitations.** (a) The bathtub is classified as a sink, due to shape similarity. (b) The agent failed to find the partially-hidden box under the table.

Limitations. Figure 6 shows two cases where our system fails. In (a), the agent is looking for a sink and thinks it found it despite of the fact that it actually found a bathtub. We note that this specific bathtub is similar in shape to some sinks in the training data. In (b), the agent is looking for a box, which is partially hidden. The visible part of the box is insufficient for our agent to conclude it found the object.

5. Conclusion

This work presented an end-to-end reinforcement learning for visual navigation. Our framework is based on a novel attention probability model that suits visual navigation, as it encodes both semantic information about the observed objects and spatial information about their place. Specifically, the attention model consists of three components: target, action and memory. The framework is shown to achieve SOTA results on commonly-used scenarios.

Our results are achieved using only RGB images. In the future, RGBD images can be utilized, as suggested by [6]. This has the potential to shorten the path, as the distance requirement can be achieved more precisely and obstacles may be more easily avoided.

References

[1] Peter Anderson, Ayush Shrivastava, Devi Parikh, Dhruv Batra, and Stefan Lee. Chasing ghosts: Instruction following as bayesian state tracking. *arXiv preprint arXiv:1907.02022*, 2019. 3

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 3

[3] Michael Blösch, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Vision based mav navigation in unknown and unstructured environments. In *2010 IEEE International Conference on Robotics and Automation*, pages 21–28. IEEE, 2010. 1, 2

[4] Valts Blukis, Dipendra Misra, Ross A Knepper, and Yoav Artzi. Mapping navigation instructions to continuous control actions with position-visitation prediction. In *Conference on Robot Learning*, pages 505–518. PMLR, 2018. 3

[5] Vincent Cartillier, Zhile Ren, Neha Jain, Stefan Lee, Irfan Essa, and Dhruv Batra. Semantic mapnet: Building allocentric semanticmaps and representations from egocentric views. *arXiv preprint arXiv:2010.01191*, 2020. 3

[6] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *arXiv preprint arXiv:2007.00643*, 2020. 1, 3, 8

[7] Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. Gated-attention architectures for task-oriented language grounding. *arXiv preprint arXiv:1706.07230*, 2017. 1, 3

[8] Mark Cummins and Paul Newman. Probabilistic appearance based navigation and loop closing. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2042–2048. IEEE, 2007. 2

[9] MWM Gamini Dissanayake, Paul Newman, Steve Clark, Hugh F Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation*, 17(3):229–241, 2001. 2

[10] Heming Du, Xin Yu, and Liang Zheng. Learning object relation graph and tentative policy for visual navigation. *arXiv preprint arXiv:2007.11018*, 2020. 1, 3, 7, 8

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017. 3

[12] Georgios Georgakis, Yimeng Li, and Jana Kosecka. Simultaneous mapping and target driven navigation. *arXiv preprint arXiv:1911.07980*, 2019. 3

[13] 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*, pages 2616–2625, 2017. 1, 3

[14] Raia Hadsell, Pierre Sermanet, Jan Ben, Ayse Erkan, Marco Scoffier, Koray Kavukcuoglu, Urs Muller, and Yann LeCun. Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, 26(2):120–144, 2009. 1, 2

[15] Saumya Jetley, Nicholas A Lord, Namhoon Lee, and Philip HS Torr. Learn to pay attention. *arXiv preprint arXiv:1804.02391*, 2018. 3

[16] Kiyosumi Kidono, Jun Miura, and Yoshiaki Shirai. Autonomous visual navigation of a mobile robot using a human-guided experience. *Robotics and Autonomous Systems*, 40(2-3):121–130, 2002. 2

[17] H Jin Kim, Michael I Jordan, Shankar Sastry, and Andrew Y Ng. Autonomous helicopter flight via reinforcement learning. In *Advances in neural information processing systems*, pages 799–806, 2004. 2

[18] Nate Kohl and Peter Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 3, pages 2619–2624. IEEE, 2004. 2

[19] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 3

[20] Keuntaek Lee, Gabriel Nakajima An, Viacheslav Zakharov, and Evangelos A Theodorou. Perceptual attention-based predictive control. In *Conference on Robot Learning*, pages 220–232. PMLR, 2020. 3

[21] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015. 1, 2

[22] Arsalan Mousavian, Alexander Toshev, Marek Fišer, Jana Košecká, Ayzaan Wahid, and James Davidson. Visual representations for semantic target driven navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8846–8852. IEEE, 2019. 3

[23] Jonghwan Mun, Minsu Cho, and Bohyung Han. Text-guided attention model for image captioning. *arXiv preprint arXiv:1612.03557*, 2016. 3

[24] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 5

[25] Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697, 2008. 1, 2

[26] Idan Schwartz, Seunghak Yu, Tamir Hazan, and Alexander G Schwing. Factor graph attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2039–2048, 2019. 3

[27] William B Shen, Danfei Xu, Yuke Zhu, Leonidas J Guibas, Li Fei-Fei, and Silvio Savarese. Situational fusion of visual representation for visual navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2881–2890, 2019. 2

[28] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998. 1, 2

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3, 8

[30] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 3, 4, 6, 7, 8

[31] Yi Wu, Yuxin Wu, Aviv Tamar, Stuart Russell, Georgia Gkioxari, and Yuandong Tian. Bayesian relational memory for semantic visual navigation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2769–2779, 2019. 1, 3

[32] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer, 2016. 3

[33] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 3

[34] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018. 3, 7

[35] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016. 3

[36] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016. 3

[37] 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 *IEEE International Conference on Robotics and Automation*, 2017. 1, 2, 3, 6