

This is the peer reviewed version of the following article:

Multimodal Attention Networks for Low-Level Vision-and-Language Navigation / Landi, Federico; Baraldi, Lorenzo; Cornia, Marcella; Corsini, Massimiliano; Cucchiara, Rita. - In: COMPUTER VISION AND IMAGE UNDERSTANDING. - ISSN 1077-3142. - 210:(2021), pp. 1-10. [10.1016/j.cviu.2021.103255]

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

20/04/2024 07:37

(Article begins on next page)

Multimodal Attention Networks for Low-Level Vision-and-Language Navigation

Federico Landi^{a,**}, Lorenzo Baraldi^a, Marcella Cornia^a, Massimiliano Corsini^b, Rita Cucchiara^a

^aUniversity of Modena and Reggio Emilia, Italy

^bISTI - CNR, Italy

ABSTRACT

Vision-and-Language Navigation (VLN) is a challenging task in which an agent needs to follow a language-specified path to reach a target destination. The goal gets even harder as the actions available to the agent get simpler and move towards low-level, atomic interactions with the environment. This setting takes the name of low-level VLN. In this paper, we strive for the creation of an agent able to tackle three key issues: multi-modality, long-term dependencies, and adaptability towards different locomotive settings. To that end, we devise “Perceive, Transform, and Act” (PTA): a fully-attentive VLN architecture that leaves the recurrent approach behind and the first Transformer-like architecture incorporating three different modalities – natural language, images, and low-level actions for the agent control. In particular, we adopt an early fusion strategy to merge lingual and visual information efficiently in our encoder. We then propose to refine the decoding phase with a late fusion extension between the agent’s history of actions and the perceptual modalities. We experimentally validate our model on two datasets: PTA achieves promising results in low-level VLN on R2R and achieves good performance in the recently proposed R4R benchmark. Our code is publicly available at <https://github.com/aimagelab/perceive-transform-and-act>.

1. Introduction

Effective instruction-following and contextual decision-making can open the door to a new world for researchers in embodied AI. Deep neural networks have the potential to build complex reasoning rules that enable the creation of intelligent agents, and research on this subject could also help to empower the next generation of collaborative robots (Savva et al., 2019; Xia et al., 2018). In this scenario, Vision-and-Language Navigation (VLN) (Anderson et al., 2018c) plays a significant part in current research. This task requires to follow natural language instructions through unknown environments, discovering the correspondences between lingual and visual perception step by step. Additionally, the agent needs to progressively adjust navigation in light of the history of past actions and explored areas. Even a small error while planning the next move can lead to failure because perception and actions are unavoidably entangled; indeed, *we must perceive in order to move, but we must also move in order to perceive* (Gibson, 2014). For this

reason, the agent can succeed in this task only by efficiently combining the three modalities – language, vision, and actions.

Recent literature identifies two main operating settings for VLN (Landi et al., 2019), called *high-level action space* and *low-level action space* (Fig. 1a). The concept of a high-level, *panoramic* action space was first proposed by Fried et al. (2018). In this setting, navigation takes place on a graph whose connectivity is known a priori and the nodes are represented by different viewpoints (*i.e.*, the locations where the agent can step and look at the surroundings). High-level agents predict the path to the goal as a sequence of connected viewpoints, and move through the environment using a teleporting system. This aspect limits adaptability to real-world applications and prevents current research on high-level VLN from having a practical impact on embodied navigation robots. Instead, low-level methods make predictions over the agent locomotor system, hence performing actions with a one-to-one correspondence with the robot control system – *rotate X°* , *tilt up/down*, and *step forward* are examples of low-level actions. Even though low-level navigation can still be performed on a graph-like environment (with viewpoints as nodes), the agent is not aware of it and does not exploit any knowledge related to the structure of the underlying simulating platform. This setting is more

**Corresponding author:
e-mail: federico.landini@unimore.it (Federico Landi)

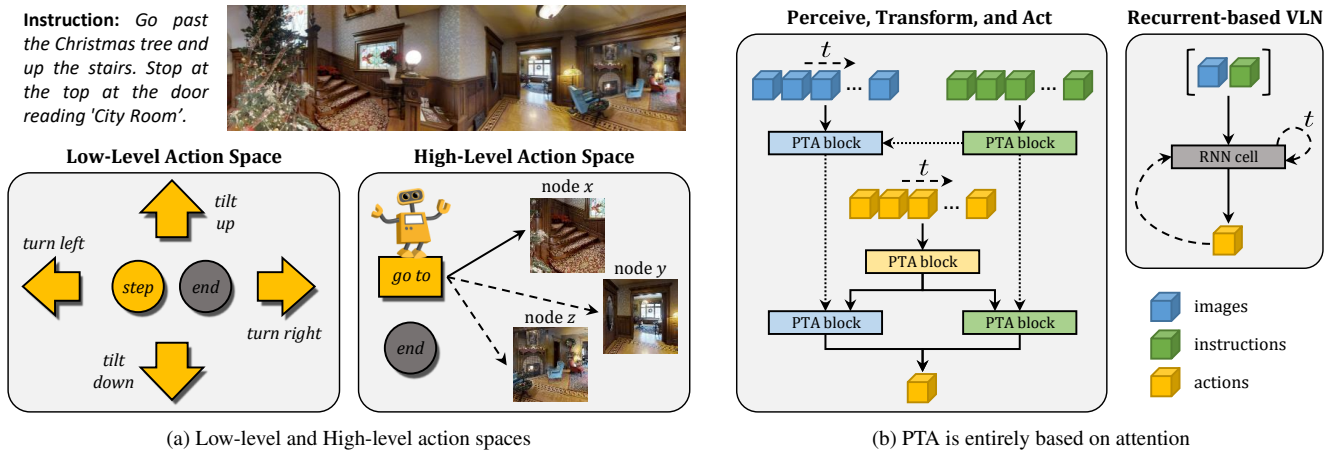


Fig. 1. Previous approaches to VLN perform high-level navigation, relaxing the assumptions on the agent action space, and build upon recurrent neural networks to model long-term dependencies among the three modalities involved – text, images, and actions. Instead, PTA implements low-level interactions with the environment and exploit only attention mechanisms

in line with recent research on embodied AI platforms (Savva et al., 2019; Xia et al., 2018), which is moving towards realistic and low-level interactions with the environment and continuous control of the agent. Since the adaptability to real-world applications represents an important challenge in this scenario, we tackle the task of low-level VLN, in which abstract reasoning (*i.e.*, teleporting from a viewpoint to the next and knowledge of the connectivity graph) is no longer available to the agent.

Encouraged by the success of attention in many vision-and-language tasks (Devlin et al., 2019; Lu et al., 2019; Vaswani et al., 2017), we propose a new model for low-level VLN that exploits fully-attentive networks to merge the knowledge coming from different domains. In this work, we devise *Perceive, Transform, and Act* (PTA), in which the different modalities (text, vision, and actions) can be conditioned on the full history of previous observations. While all the previous approaches to VLN rely on a recurrent policy to track the agent’s internal status through time, we directly infer the state from the observations via attention and avoid any form of recurrence (Fig. 1b). For this reason, our agent can model the dependencies tied to navigation more efficiently and generalize to longer episodes better than other models.

At the present time, there is no study exploring the possibility for a given architecture to switch between the high-level and the low-level action spaces. In this work, we experimentally show that methods born and designed for high-level navigation experience a drop in performance when adapted for low-level VLN. Indeed, high-level reasoning and abstraction from the physical environment is too heavily exploited to let the agent walk on its own. This is not true for PTA, which is designed for low-level use but can easily adapt to high-level scenarios. We summarize our main contributions as follows:

- We propose a novel multimodal framework for low-level VLN that replaces any form of recurrence with attention mechanisms, using them to tackle both long-term dependencies and multi-modality. To the best of our knowledge, our model is the first Transformer-like architecture to merge visuo-linguistic perception with information

coming from the agent action system;

- We technically describe how it is possible to switch from a high-level output space to a low-level locomotor system and vice versa. Experimental results on this subject are the first to analyze the mutual relationships between low-level and high-level VLN, and validate the hypothesis that high-level architectures are not easily adaptable to the low-level counterpart. Such results highlight the need for more experiments in this direction for future works;
- Experimental results show that PTA achieves good performance on low-level VLN. We validate this claim on two different benchmarks of increasing instruction length and complexity. Since our setting is closer to real-world applications and requires to decode fine-grained atomic actions, we believe that low-level VLN represents the next testbed for embodied agents aiming to perform Vision-and-Language Navigation.

2. Related Work

There is a wide area of research devoted to bridge natural language processing and image understanding. Image captioning (Anderson et al., 2018b; Vinyals et al., 2015; Xu et al., 2015), visual question answering (Antol et al., 2015; Goyal et al., 2017), and visual dialog (Das et al., 2017a,b) are examples of active research areas in this field. At the same time, visual navigation (Gupta et al., 2017; Shen et al., 2019; Xia et al., 2018) and goal-oriented instruction following (Chen et al., 2019; Fu et al., 2019; Qi et al., 2020b) represent an important part of current work on embodied AI (Das et al., 2018a,b; Savva et al., 2019; Yang et al., 2019). In this context, Vision-and-Language Navigation (VLN) (Anderson et al., 2018c) constitutes a peculiar challenge, as it enriches traditional navigation with a set of visually rich environments and detailed instructions. Additionally, all the scenes are photo-realistic and unknown to the agent beforehand.

High-level Vision-and-Language Navigation. The idea of a high-level action space was first proposed by Fried et al. (2018),

and immediately allowed for an important boost in terms of performance. Following work includes visual and textual co-grounding with progress inference (Ma et al., 2019a) and backtracking with learned heuristics (Ma et al., 2019b). Other methods implement a speaker module which strengthens consistency between the chosen path and the instruction (Fried et al., 2018; Wang et al., 2019). Wang et al. (2019) propose a reinforced cross-modal matching critic, together with a new self-supervised imitation learning setting. Tan et al. (2019) devise a novel environmental dropout method to improve traditional features dropout for VLN. Ke et al. (2019) propose a FAST navigation agent which improves the performance both over greedy decoding of the next action and over beam search. Very recently, Zhu et al. (2020) exploit auxiliary reasoning tasks and the rich semantic given by the navigation in their model, while Hao et al. (2020) investigate an efficient pre-training for generic VLN agents. While pragmatic approaches with high-level reasoning allow for a boost in performance, architectures built for high-level VLN rely heavily on the information coming from the underlying simulating platform. Even when the environment is supposed to be unknown (*e.g.* during test) the agent can get a priori knowledge from the connectivity graph and exploit this information for a more efficient navigation. Recently proposed benchmarks and new evaluation metrics (Jain et al., 2019) show that traditional approaches hardly adapt to longer trajectories. Indeed, the recurrent nature of previous methods exacerbates the difficulty of learning long-term dependencies (Bengio et al., 1994) both in the instruction and in the navigation.

After the initial submission of this paper, new methods have been proposed to deal with VLN on a high-level perspective: a recent line of work designs graph operations to boost planning capabilities (Deng et al., 2020) or to model visuo-linguistic relationships in the graph nodes (Hong et al., 2020). Zhang et al. (2020) propose to employ two levels of attention-guided co-grounding, together with a new learning scheme alternating teacher-forcing and student-forcing. Qi et al. (2020a) design an architecture taking advantage from both visual tokens and action tokens in the instructions. Visual tokens are employed to identify meaningful visual features in the environment, while action tokens consider only the agent state (represented by coordinates features). In this work, we leverage the same intuition in our multi-modal decoder. In fact, we propose an additional decoding branch that does not employ visual features, but focuses on action clues provided in the sole instruction.

Low-level Vision-and-Language Navigation. In low-level VLN, the agent takes move in the environment by using actions such as *rotate*, *tilt up*, and *step ahead*. So far, only a small portion of literature has taken this direction. Anderson et al. (2018c) build on a traditional sequence-to-sequence architecture, while Wang et al. (2018) employ a mixture of model-free and model-based reinforcement learning. In these works the agent perceives only the first person view of the surrounding environment. More recently, Landi et al. (2019) propose a sequence-to-sequence model which exploits dynamic convolution to make the visual representation more compact and informative for the agent. In this last work, the agent perceives the

360° image of the surroundings. This generalization does not hurt adaptability to real-world scenarios, since it is relatively easy to enrich the agent with additional RGB cameras.

3. Perceive, Transform, and Act

Our goal is to navigate unseen environments using low-level actions with the only help of natural language instructions and egocentric visual observations. To merge multimodal knowledge coming from the environment, we devise a **two-stage encoder**. In the first stage, we focus on encoding the instruction – this step can be done once per episode as the natural language indication remains the same throughout the navigation. In the second stage, we use spatial attention to encode the visual observation and then employ the encoded instruction coming from the previous phase to enrich the agent representation of the surrounding environment. At each time step, the agent selects a move to progress towards the goal. To determine the next action, we fuse visuo-linguistic information with the history of actions via attention and build a **multimodal decoder** which merges the three modalities: actions, images, and text. We then decode a probability distribution over a low-level output space in which possible actions are atomic moves like *turn* or *step ahead*. After a first phase in which we train the agent with classical imitation learning, we implement an **extrinsic reward** function to promote coherence between ground-truth and predicted trajectories. We are the first, to the best of our knowledge, to build a VLN architecture without recurrence. Each component of our model is end-to-end trainable. Our architecture is depicted in Fig. 2 and detailed next.

3.1. Two-stage Encoder

At the beginning of each navigation episode, the agent receives a natural language instruction $\{w_0, w_1, \dots, w_{n-1}\}$ of variable length n . The agent also perceives a panoramic 360° image of the surroundings I_t at each timestep t . Our encoder consists of a single branch for each modality: text and images, and then employs attention to create a fused representation which specifically models the relevance of the source instruction into the visual observation.

Instruction Encoding. To encode the textual instruction, we employ an attention mechanism with multiple heads, followed by a feed-forward network. As a first step, we filter stop words and apply GloVe embeddings (Pennington et al., 2014) to obtain a meaningful representation for each word. We then apply the following transformation:

$$\tilde{X} = \text{LayerNorm}(\max(0, XW_x + \mathbf{b}_x)), \quad (1)$$

where X is the GloVe embedding for the natural language instruction, $W_x \in \mathbb{R}^{d_{\text{GloVe}} \times d_{\text{model}}}$ and $\mathbf{b}_x \in \mathbb{R}^{d_{\text{model}}}$ are learnable parameters, and $\text{LayerNorm}(\cdot)$ stands for layer normalization. Since the instruction encoder has no recurrence, we must inject information about the relative position of the words in the sentence. Such information is added in the form of positional encoding to the input embeddings. The positional encodings have the same dimension as the embeddings, so that the two

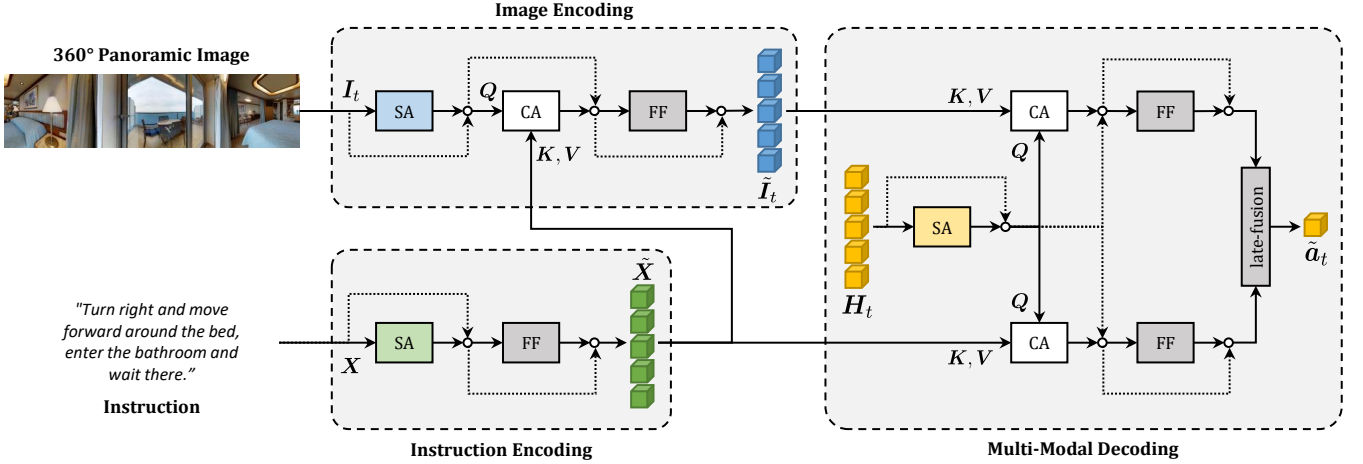


Fig. 2. Overview of our approach. Our attention-based architecture for VLN builds upon three main blocks: an instruction encoder, an image encoder, and a multimodal decoder. SA, CA, and FF stand for self-attention, cross-attention, and feed-forward networks respectively. Dotted lines stand for residual connections between the results of the attention blocks and their inputs. For sake of clarity, we omit layer normalization after each block

can be summed. We employ sine and cosine functions of different frequencies, in line with (Vaswani et al., 2017):

$$\begin{aligned} PE_{(pos,2j)} &= \sin(pos/10000^{2j/d_{model}}) \\ PE_{(pos,2j+1)} &= \cos(pos/10000^{2j/d_{model}}) \end{aligned} \quad (2)$$

where pos is the position in the sequence and j is the channel index. At this point we use multi-head attention to create a representation that models temporal dependencies inside the instruction. Multi-head attention is defined as:

$$\begin{aligned} \text{MH}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \text{Concat}(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_h) \mathbf{W}^O \\ \text{with } \mathbf{h}_i &= \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), \end{aligned} \quad (3)$$

where $\mathbf{W}_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{model}}$ denote learnable weight matrices, and the index i stands for the i^{th} head in the multi-head attention module. As also stated in our implementation details, $d_k = d_v = d_{model}/h$. In each head, we employ the scaled dot-product attention defined by Vaswani et al. (2017):

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V}. \quad (4)$$

The attention mechanism described by Eq. 4 computes a weighted sum of the values (\mathbf{V}) basing on the similarity between the keys and the queries (\mathbf{K} and \mathbf{Q}). In the self-attention, the same source sequence ($\tilde{\mathbf{X}}$ in this case) is employed to model the $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ triplet of Eq. 3. Following the attention layer, we place a feed-forward multilayer perceptron:

$$\text{FF}(\tilde{\mathbf{X}}) = \max(0, \tilde{\mathbf{X}}\mathbf{W}_1 + \mathbf{b}_1) \mathbf{W}_2 + \mathbf{b}_2, \quad (5)$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_{model} \times d_{ff}}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times d_{model}}$, $\mathbf{b}_1 \in \mathbb{R}^{d_{ff}}$, $\mathbf{b}_2 \in \mathbb{R}^{d_{model}}$. At the end of this step, we obtain the attended representation for the current instruction $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_{n-1}\}$, that we use both during image encoding and in our multimodal decoder.

Image Encoding. As a first step, we discretize the 360° panoramic image of the surroundings \mathbf{I}_t in 36 squared locations

and we extract the corresponding visual features with a ResNet-152 (He et al., 2016) trained on ImageNet (Deng et al., 2009). Each viewpoint covers 30° in the equirectangular image representing the agent surroundings, hence the image representation takes the form of a 3×12 grid. We then project visual features with a transformation similar to Eq. 1, but instead of using sinusoidal positional encodings, we append a coordinate vector given by:

$$\text{coord}_t = (\sin \phi_t, \cos \phi_t, \sin \theta_t), \quad (6)$$

where $\phi_t \in (-\pi, \pi)$ and $\theta_t \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ are the heading and elevation angles for each viewpoint in the 3×12 grid relative to the agent position at timestep t . We then apply multi-head self-attention according to Eq. 3 to help modeling concepts such as relative positions between objects. In this layer, the input sequence modeling $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ is composed by the features extracted from the 36 squared regions of \mathbf{I}_t .

After this step, we aim to create an image representation enriched with the textual concepts expressed by the attended instruction $\tilde{\mathbf{X}}$. We use cross-attention to achieve this goal, and employ $\tilde{\mathbf{X}}$ as keys and values for multi-head attention (Eq. 3), while the queries come from the output of the previous self-attention layer. Using cross-attention, we enrich visual information with a weighted sum of the instruction tokens. From the resulting representation it is possible to draw concepts such as the *tableness* or the *redness* of an image region, given an instruction that refers to concepts such as *table* or *red*. Finally, a feed-forward network as in Eq. 5 is applied to obtain the attended visual observation $\tilde{\mathbf{I}}_t$.

3.2. Multimodal Decoder

Our decoder predicts the next action to perform among the following instructions: *turn right/left 30°*, *tilt up/down*, *step forward*, and *end episode* – to signal that it has reached the goal.

Contextual History for Action Decoding. The first part of our decoder takes into account the history of past actions. While previous methods employ a recurrent neural network to keep track of previous steps (see for instance Anderson et al. (2018c);

Ma et al. (2019a); Wang et al. (2019)), we explicitly model $\mathbf{H}_t = \{a_0, a_1, \dots, a_{t-1}\}$ as the set of actions performed before the current timestep t . Note that a_0 coincides with the `<start>` token. We add sinusoidal positional encoding (Eq. 2) to provide temporal information and apply multi-head self-attention to obtain an attended history representation $\tilde{\mathbf{H}}_t = \{\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_{t-1}\}$.

Late Fusion of Perception and Action. At this point, $\tilde{\mathbf{H}}_t$ contains the relevant information regarding the action history of the navigation episode. However, this information must be enriched with the perception coming from the environment. We merge textual and visual information with $\tilde{\mathbf{H}}_t$ via attention, allowing mutual influence between perception and motion. We build two branches of multi-head cross-attention accepting respectively $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{I}}_t$ as key/value pairs and using $\tilde{\mathbf{H}}_t$ as query. The image-action cross-attention is motivated by the fact that the agent needs to *look around* before decoding the next action. Since $\tilde{\mathbf{I}}_t$ already contains information coming from the instruction, this cross-attention layer is sufficient to achieve decent results on the VLN task (as demonstrated by our ablation studies). However, we find out that adding a separate text-action cross-attention layer helps generalization in unseen environments. After this step, we concatenate the two representations and apply a FC layer to obtain the output sequence whose last element corresponds to \tilde{a}_t . With this last layer, we perform a late fusion of visuo-linguistic information with the agent internal state (given by its previous history). It is worth noting that PTA also comprises an early fusion mechanism: the cross-attention between $\tilde{\mathbf{X}}_t$ and the attended visual input introduced in the Image Encoder. In our ablation study, we discuss the positive effects given by the early fusion and the late fusion mechanisms.

Action Selection. To select the next low-level action, we project the final representation \tilde{a}_t in a six-dimensional space corresponding with the agent locomotor space containing the following actions: *turn right/left 30°*, *tilt up/down*, *step forward*, and *end episode*. The output probability distribution over the action space can therefore be written as:

$$\mathbf{p}_t = \text{softmax}(\tilde{a}_t \mathbf{W}_p + \mathbf{b}_p), \quad (7)$$

where $\mathbf{W}_p \in \mathbb{R}^{d_{\text{model}} \times n_{\text{actions}}}$ and $\mathbf{b}_p \in \mathbb{R}^{n_{\text{actions}}}$ are learned parameters ($n_{\text{actions}} = 6$). During training, we sample the next action to perform a_t from \mathbf{p}_t , while we select $a_t = \text{argmax}(\mathbf{p}_t)$ during evaluation and test.

3.3. Training

Our training setup includes two distinct objective functions. The first estimates the policy by imitation learning, while the second enforces similarity between the ground-truth and predicted trajectories via reinforcement learning.

Imitation Learning. To approximate a good policy, we first train our agent using strong supervision. At each timestep t , the simulator outputs the ground-truth action y_t . In the low-level setup, the ground-truth action is the one that allows getting to the next target viewpoint in the minimum amount of steps. In this phase, we aim to minimize the cross-entropy loss of the predicted distribution \mathbf{p}_t w.r.t. the ground-truth action y_t .

Extrinsic Reward. After a first training phase with supervised learning, we finetune our agent using an extrinsic reward function. Recently, Magalhaes et al. (2019) propose to employ Dynamic Time Warping (DTW) (Berndt and Clifford, 1994) to evaluate the trajectories performed by navigation agents. In particular, they define the *normalized Dynamic Time Warping* (nDTW) as:

$$\text{nDTW}(R, Q) = \exp\left(-\frac{\text{DTW}(R, Q)}{|R| \cdot d_{th}}\right), \quad (8)$$

where R and Q are respectively the reference and the query paths, $|R|$ is the length of the reference path, and d_{th} is the success threshold distance. At each navigation step t , the agent receives a reward equal to the gain in terms of nDTW:

$$R_t = \text{nDTW}(q_{0,\dots,t}, R) - \text{nDTW}(q_{0,\dots,t-1}, R). \quad (9)$$

Additionally, we give an episode-level reward to the agent if it terminates the navigation within a success threshold distance d_{th} from the goal, given by $R_s = \max(0, 1 - d_{goal}/d_{th})$, where d_{goal} is the final distance between the agent and the target. We can write our final reinforcement learning objective function as:

$$L_{rl} = -\mathbb{E}_{a_t \sim \pi_\theta} [A_t]. \quad (10)$$

where the advantage function $A_t = R_t + R_s$. Based on REINFORCE algorithm (Williams, 1992), we derive the gradient of our reward-based objective as:

$$\nabla_\theta L_{rl} = -A_t \nabla \log \pi_\theta(a_t | s_t). \quad (11)$$

4. Low-level and High-Level Navigation

Section 3 describes our approach to *low-level* VLN. Here, we discuss the main technical differences with the high-level counterpart and explain how PTA can switch from one setting to the other. Differently from the low-level architectures, a high-level method aims to predict the next node to traverse in the navigation graph, as physical navigation takes place with a teleport mechanism. The choice at time step t is done with a similarity measure between the agent internal state s_t and the appearance vector for the navigable locations \mathbf{v}_t . This similarity function is normally mapped into a bilinear dot-product:

$$\mathbf{p}_t = \text{softmax}(f(s_t)^\top g(\mathbf{v}_t)) \quad (12)$$

where $f(\cdot)$ and $g(\cdot)$ are generic transformations.

In principle, it is possible to substitute the final softmax classifier of a low-level architecture (Eq. 7) with Eq. 12 and change the corresponding action space. According to this observation, we can swap the action space of a model to test its adaptability to different navigation settings. While traditional approaches start from the hidden state of the recurrent policy to estimate the agent’s internal state s_t , we can derive it directly from \tilde{a}_t :

$$s_t = \tilde{a}_t \mathbf{W}_s + \mathbf{b}_s, \quad (13)$$

where \mathbf{W}_s and \mathbf{b}_s are learned parameters. As \mathbf{v}_t , we select the unattended visual features augmented with the coordinate vector described by Eq. 6, and apply the following transformation:

$$g(\mathbf{v}_t) = \max(0, \mathbf{v}_t \mathbf{W}_v + \mathbf{b}_v), \quad (14)$$

where \mathbf{W}_v and \mathbf{b}_v are learned parameters.

In our architecture, $\tilde{\mathbf{a}}_t$ can fit to represent any kind of information about the current navigation. This is because it can draw knowledge from the perceptual modalities and the history of past actions directly and without the bottleneck represented by a recurrent network. Our experiments on this subject (Sec. 5.3) show that our model stands out from the literature in terms of adaptability. In other words, PTA can adapt to a different action space because it does not make any assumptions on the underlying simulating platform. Instead, our architecture relies on efficient visuo-linguistic fusion mechanisms designed to be agnostic towards the final action space. We will see that methods making stronger assumptions on the action space experience a larger drop in performance than PTA.

5. Experiments and Discussion

5.1. Experimental Setup

Datasets. In our experiments, we primarily test our architecture on the R2R dataset for VLN (Anderson et al., 2018c). This dataset builds on the Matterport3D dataset of spaces (Chang et al., 2017), which contains complete scans of 90 different buildings. The visual data is enriched with more than 7 000 navigation paths and 21 000 natural language instructions. The episodes are divided into a training set, two validation splits (*validation-seen*, with environments that the agent has already seen during training, and *validation-unseen*, containing only unexplored buildings), and a test set. The testing phase takes place in previously unseen environments and is accessible via a test-server with a public leaderboard. While the instructions in R2R are quite long and complex (about 29 words on average), navigation episodes usually involve a limited number of steps – max 6 steps for high-level action space and max 23 steps for the low-level setup. In the R4R dataset, Jain et al. (2019), merge the paths in R2R to create a more complex and challenging setup. Episodes become considerably longer, pushing the traditional approaches to their limits and testing their generalizability to arbitrary long instructions and more complex trajectories.

Evaluation Metrics. In line with previous literature, we mainly focus on four metrics. NE (Navigation Error) measures the mean distance from the goal and the stop point. SR (Success Rate) is the fraction of episodes concluded within a threshold distance from the target – 3 meters for all of the previous papers on the subject. OSR (Oracle SR) represents the SR that the agent would achieve if it received an oracle stop signal when passing within the threshold distance from the goal, while SPL (SR weighted by inverse Path Length) penalizes navigation episodes that deviate from the shortest path to the goal. SPL is accredited to be the most reliable metric on the R2R dataset (Anderson et al., 2018a), as it strongly penalizes exhaustive exploration and search methods like beam search. Recently, Jain et al. (2019) propose to use Coverage weighted by Length Score (CLS) to replace SR for generic navigation trajectories, as this metric is also sensitive to intermediate nodes in the reference path. Additionally, Magalhaes

et al. (2019) propose Dynamic Time Warping (DTW) and derived metrics (Normalized DTW and Success weighted by normalized DTW) to measure the similarity between reference and predicted paths. These three last metrics are more meaningful on the R4R dataset than SR and SPL (Jain et al., 2019).

Implementation Details. In the instruction encoder, $d_{\text{GloVe}} = 300$. In each component of our model, we project the input features into a d_{model} -dimensional space, with $d_{\text{model}} = 512$. For multi-head attention, we employ $h = 8$ heads, thus $d_k = d_v = d_{\text{model}}/h = 64$. The internal representation of feed-forward networks has size $d_{\text{ff}} = 2048$. After each sub-module, we add a residual connection followed by layer normalization. We also apply dropout (Srivastava et al., 2014) with drop probability $p = 0.1$ after each linear layer. During training, we use Adam optimizer (Kingma and Ba, 2015) with learning rate 10^{-4} , we set the batch size to 32 and reduce the learning rate by a factor 10 if the SPL on the validation unseen split does not improve for 5 consecutive epochs. We stop the training after 30 epochs without improvement on the same metric. When finetuning using REINFORCE, we set the initial learning rate to 10^{-7} .

5.2. Ablation Study

In our ablation study, we experimentally validate the importance of each module in our architecture. First, we ablate multimodality in our decoder and we do not apply late fusion before decoding the next action. In a second experiment, we remove cross-attention between visual and lingual information in the encoder. Finally, we show the impact of synthetic data augmentation (Fried et al., 2018) and the role of REINFORCE. Results are shown in Table 1 and discussed below.

Multimodal Decoder. In our first ablation study, we use only one of the two decoder branches at the time, and we do not perform late fusion between lingual and visually-grounded information. When removing the textual branch (Table 1, line 3), our agent performs worse on unseen environments, hence losing potential in terms of generalization. When removing the visual modality, our PTA agent is blinded and can only count on the natural language instruction. This setup leads to success only when the instruction does not involve references to objects or visual properties of the environment – a nearly empty subset of the dataset. Indeed, the metrics for our *blind* agent are extremely low, and they do not vary between seen and unseen environments (Table 1, line 4). This result is meaningful in light of recent studies proving that some single-modality agents perform better than their multimodal version by removing the visual perception and overfitting on dataset biases (Thomason et al., 2018).

Early Fusion of Textual and Visual Perception. As a second experiment, we remove the early fusion mechanism, namely the cross-attention layer between the textual and visual branches of our encoder, to check its contribution. If this fusion layer is redundant, we expect that the late fusion stage will compensate for the loss. Instead, we experience a drop in performance: -12% in SPL in unseen environments (Table 1, line 5). We thus prove the importance of early textual and visual fusion in our architecture for VLN.

#	Method	Validation-Seen							Validation-Unseen						
		NE ↓	SR ↑	OSR ↑	SPL ↑	CLS ↑	nDTW ↑	SDTW ↑	NE ↓	SR ↑	OSR ↑	SPL ↑	CLS ↑	nDTW ↑	SDTW ↑
1	Anderson et al. (2018c)	6.01	0.39	0.53	-	-	-	-	7.81	0.22	0.28	-	-	-	-
2	PTA (pure IL, no extrinsic reward)	4.14	0.58	0.70	0.50	0.63	0.48	0.39	6.44	0.39	0.49	0.32	0.48	0.32	0.24
3	- multi-modal decoder (only visual)	3.90	0.61	0.72	0.54	0.65	0.52	0.44	6.56	0.36	0.46	0.29	0.47	0.32	0.22
4	- multi-modal decoder (only textual)	9.64	0.03	0.04	0.03	0.28	0.19	0.02	9.13	0.04	0.04	0.04	0.28	0.21	0.02
5	- early fusion (cross attention)	6.41	0.34	0.44	0.30	0.54	0.28	0.18	7.70	0.23	0.29	0.20	0.43	0.20	0.12
6	- action history (only last action)	5.40	0.42	0.54	0.36	0.55	0.39	0.27	7.19	0.22	0.31	0.18	0.41	0.26	0.12
7	+ data augmentation	3.47	0.66	0.76	0.58	0.67	0.54	0.47	5.91	0.40	0.48	0.34	0.50	0.36	0.25
8	+ extrinsic reward	3.58	0.65	0.74	0.59	0.69	0.60	0.50	6.00	0.40	0.47	0.36	0.52	0.41	0.28

Table 1. Ablation study proving the effectiveness of our main modules. We also show that our model can be initialized using synthetic data augmentation and then finetuned with a limited set of refined data. Adding an extrinsic reward function further improves the performance in the final model.

Contextual History for Action Decoding. H_t stores past actions as a series of one hot vectors, and it is extremely helpful to model navigation history. It acts as a sort of memory for the agent, so that it knows what actions have already been made. A similar trick in LSTM-based VLN consists in adding the last action as input to the policy RNN at each step. In our model, removing H_t and using only the last action (losing all the history) causes a drop in performance: -14% and -17% on SPL and SR respectively for the Val-Unseen split (Table 1, line 6).

Data Augmentation. In line with previous literature, we find the use of additional synthetic instructions useful to initialize our agent. The synthetic training set was provided by Fried et al. (2018) using a *Speaker* module. After a first training with the full set of instructions (synthetic and human-generated), we finetune using *only* the original R2R train set. Results are reported in Table 1, line 7.

Extrinsic Reward. While imitation learning allows approximating a good policy, there is still room for improvement via reinforcement learning. Wang et al. (2019) were the first to use REINFORCE in the context of VLN to refine their navigation policy based on cross-modal matching. In line with them, we find REINFORCE beneficial for our model: our final agent sticks more closely to the reference trajectory and penalizes overlong navigations (Table 1, line 8).

5.3. Results on R2R

In our experiments on the R2R dataset (Anderson et al., 2018c), we test the ability of our agent to navigate unseen environments in light of previously unseen natural language instructions. The main test-bed for this experiment is represented by the R2R evaluation leaderboard, which is publicly available online.

Comparison with SOTA. In Table 2, we report our results on the R2R test set, together with the results achieved by other state-of-the-art architectures on VLN. Other methods that operate in the *low-level action space* are the sequence-to-sequence architecture proposed by Anderson et al. (2018c), the RPA model using a mixture of model-free and model-based reinforcement learning (Wang et al., 2018), and the recurrent architecture with dynamic convolutional filters proposed by Landi et al. (2019). Our method overcomes the state-of-the-art on low-level VLN by a large margin (5% in terms of SPL and SR).

Low-level Methods	Test (Unseen)			
	NE ↓	SR ↑	OSR ↑	SPL ↑
Random	9.77	0.13	0.18	0.12
Anderson et al. (2018c)	7.85	0.20	0.27	0.18
Wang et al. (2018)	7.53	0.25	0.33	0.23
Landi et al. (2019)	6.55	0.35	0.45	0.31
PTA	6.17	0.40	0.47	0.36

High-level Methods	Test (Unseen)			
	NE ↓	SR ↑	OSR ↑	SPL ↑
Fried et al. (2018)	6.62	0.35	0.44	0.28
Ma et al. (2019a)	5.67	0.48	0.59	0.35
Wang et al. (2019)	6.01	0.43	0.51	0.35
Ma et al. (2019b)	5.69	0.48	0.56	0.40
Ke et al. (2019)	5.14	0.54	0.64	0.41
Tan et al. (2019)	5.23	0.51	0.59	0.47
Li et al. (2019)	4.53	0.57	0.63	0.53

Table 2. Results on the R2R test server for low-level (top) and high-level (bottom) methods. We chose the best version of each model basing on SPL.

Although a direct comparison between the two settings is not feasible, we notice that PTA performs better than some high-level architectures in terms of SPL. Notably, we achieve this result without making any assumption on the underlying simulating platform and decoding a longer sequence of atomic moves, instead of target viewpoints. Moreover, high-level architectures can often count on efficient graph-search methods (impractical when dealing with continuous controls) to decode the final trajectory, and on additional modules that are not present in our method. While these are effective for high-level VLN, their generalizability to a low-level setup, closer to real-world application, is yet to be tested.

Switching from Low-level to High-level. Our second experiment on R2R aims to test the effects of retraining existing models after switching their final action spaces (from high-level to low-level and vice-versa). To that end, we change the final classifier of PTA as described in Section 4. In this new setting, the output of the action decoder becomes a probability distribution over the adjacent nodes of the navigation graph. Once the agent decides where to go, the displacements are made automatically and there is no need to decode lower-level actions such as rotations. We train PTA from scratch in this setup, without any further hyperparameter tuning. In Table 3 we detail the full set of metrics obtained using PTA with the high-level classifier,

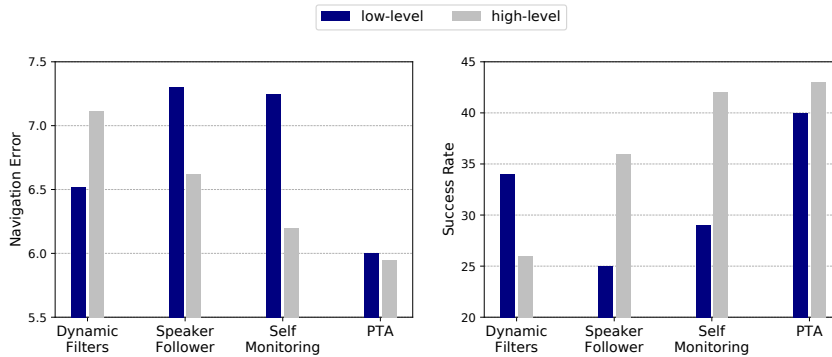


Fig. 3. Visualization of the navigation error (left) and success rate (right) on the R2R val-unseen split. A larger difference between the blue and gray bars denotes a lower degree of adaptability. The metric gap is reduced when using PTA

and compare with the model incorporating the low-level control system. The small gap between the metrics in the two setups suggests that PTA does not take any particular advantage from the underlying action space. Of course, metrics that directly evaluate the final trajectory (like DTW-based metrics) benefits from using high-level actions with automatic oracle displacements.

In principle, every model should exhibit a decent level of elasticity towards different locomotor settings. In practice, we find out that architectural choices that strongly help high-level VLN often end up hindering the other setup. This is especially true when the agent exploits high-level reasoning and makes strong assumptions on the nature of the underlying simulator. As a result, current high-level methods experience a drop in performance when adopting a simple, atomic action space (see Figure 3). PTA, instead, does not rely on such assumptions and builds on more efficient modules to merge multi-modal information entailed in the VLN task. The plots in Figure 3 show that our model exhibits far greater flexibility to the final action space than other architectures. The considerably narrow step between the blue and the gray bars (representing the low-level and the high-level actions spaces respectively) denotes that a change in the final action space does not prevent PTA from reaching its goal. We compare with the Speaker-Follower (Fried et al., 2018) and the Self-Monitoring agent (Ma et al., 2019a) from the high-level setup, which experience a sizeable loss in performance. In fact, results drop of 11% and 13% respectively in terms of SR when adapted for low-level use. We also compare PTA with a recurrent architecture exploiting dynamic convolution (Landi et al., 2019) from the low-level category. The lower degree of adaptability shown by this competitor is motivated by the fact that it operates a strong compression on the visual input basing on the current instruction. In this step, much information that could ease high-level action selection is lost.

To conduct this experiment we adjust the codes from Landi et al. (2019) and Ma et al. (2019a), which are publicly available online, and report the results in the paper for Fried et al. (2018). We choose the Speaker-Follower and the Self-Monitoring agents because they are flexible frameworks by design, and for this reason they are the most suitable models among their high-level peers for this comparison. We believe

Method	NE ↓	SR ↑	OSR ↑	SPL ↑	CLS ↑	nDTW ↑	SDTW ↑
PTA <i>low-level</i>	6.00	0.40	0.47	0.36	0.52	0.41	0.28
PTA <i>high-level</i>	5.95	0.43	0.49	0.39	0.53	0.53	0.35

Table 3. Comparison between the low-level and the high-level version of PTA. On all the metrics, a small gap denotes high adaptability. DTW-based metrics highly benefits from the use of a high-level action space

that the findings and insights provided in this experiment will motivate further experiments in this direction, and help to unravel the main reasons of improvements in new architectures for VLN.

Qualitative Results. In Fig. 4, we report a qualitative result from the R2R val-unseen set. Remarkably, PTA is able to ground concepts such as “*the second doorway on your left*” and terminates the navigation episode successfully. Since our agent operates in a low-level setup, it needs to orientate towards the next viewpoint before stepping ahead, making the decoding phase more challenging.

5.4. Results on R4R

R4R (Jain et al., 2019) builds upon R2R and aims to provide an even more challenging setting for embodied navigation agents. While navigation in R2R is usually direct and takes the shortest path between the starting position and the goal viewpoint, trajectories in R4R may bend and return on the agent’s previous steps. This change calls for adaptation in evaluation metrics: SPL and SR are now less indicative because the agent might stop close the goal in the first half of the navigation and still fail to complete the second part. In this sense, an important role is played by recently proposed metrics: CLS (Jain et al., 2019) and nDTW (Magalhaes et al., 2019) take into account the agent’s steps and are sensitive to intermediate errors in the navigation path. For this reason, these last metrics are more meaningful when evaluating navigation agents on R4R.

Comparison with SOTA. In this experiment, we compare PTA with other state-of-the-art architectures for VLN and report the results in Table 4. In the low-level setup, we compare to the recurrent architecture with dynamic convolution proposed by Landi et al. (2019). Results show that our approach performs better on all of the main metrics. In particular, a lower NE and a



Instruction: Exit the bathroom and walk down the hall to the second doorway on your left. Turn left and enter the room through that doorway.

Fig. 4. Navigation episode from the R2R unseen validation split. For each step, we report the agent first-person point of view and the next predicted action (from left to right, top to bottom)

Method	R4R Validation-Seen							R4R Validation-Unseen						
	PL ↓	NE ↓	SR ↑	SPL ↑	CLS ↑	nDTW ↑	SDTW ↑	PL ↓	NE ↓	SR ↑	SPL ↑	CLS ↑	nDTW ↑	SDTW ↑
Landi et al. (2019)	11.9	5.74	0.51	0.39	0.50	0.38	0.24	9.98	9.03	0.20	0.11	0.33	0.19	0.06
PTA low-level	11.9	5.11	0.57	0.45	0.52	0.42	0.29	10.2	8.19	0.27	0.15	0.35	0.20	0.08
Fried et al. (2018)	15.4	5.35	0.52	0.37	0.46	-	-	19.9	8.47	0.24	0.12	0.30	-	-
RCM <i>goal oriented</i> (Jain et al., 2019)	24.5	5.11	0.56	0.32	0.40	-	-	32.5	8.45	0.29	0.10	0.20	-	-
RCM <i>fidelity oriented</i> (Jain et al., 2019)	18.8	5.37	0.53	0.31	0.55	-	-	28.5	8.08	0.26	0.08	0.35	-	-
PTA high-level	16.5	4.54	0.58	0.39	0.60	0.58	0.41	17.7	8.25	0.24	0.10	0.37	0.32	0.10

Table 4. Results on the R4R validation splits. Our model is the new state-of-the-art on the two splits in both of its versions – *low-level* and *high-level*. Note that, since the trajectories can bind and return on the agent previous steps, CLS and nDTW are the more indicative metrics. Metrics with ‘-’ were not reported in the original papers.

higher CLS indicate that our agent tends to get closer to the goal while sticking to the natural language instruction better than the competitor. We also report the results obtained by our model incorporating the high-level decision space. We compare with Speaker-Follower (Fried et al., 2018) and RCM (Wang et al., 2019), as implemented in (Jain et al., 2019). PTA performs better than its high-level competitors on the majority of the metrics. In particular, the higher CLS score shows that PTA can generally select a path that follows the instruction better than the competitors. When considering the reference metrics proposed for R4R (Jain et al., 2019), our architecture achieves the best results on both the setups.

6. Conclusion

In this paper, we have presented *Perceive, Transform, and Act* (PTA), the first fully-attentive model for VLN. In particular, we tackle the challenging task of low-level VLN, in which high-level information about the environment is no longer accessible to the agent. We show that previous work on high-level VLN suffers from low flexibility and experiences a drop in performance when adapted for low-level use, while our agent naturally adapts to the other action space. These results suggest that boosts in performance observed in high-level VLN may be due to the use of a simpler action space, and encourage further research in this direction. Our architectural choices allow for a significant boost in performance: PTA achieves good results on low-level VLN, and when testing on the recently proposed R4R dataset, PTA achieves promising results in both the setups.

Acknowledgement

This work has been supported by ‘‘Fondazione di Modena’’ and by the national project ‘‘IDEHA: Innovation for Data Elaboration in Heritage Areas’’ (PON ARS01_00421), cofunded by the Italian Ministry of University and Research.

References

- Anderson, P., Chang, A., Chaplot, D.S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., et al., 2018a. On evaluation of embodied navigation agents. arXiv preprint arXiv:1807.06757.
- Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L., 2018b. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., van den Hengel, A., 2018c. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D., 2015. VQA: Visual Question Answering, in: Proceedings of the International Conference on Computer Vision.
- Bengio, Y., Simard, P., Frasconi, P., et al., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks 5, 157–166.
- Berndt, D.J., Clifford, J., 1994. Using dynamic time warping to find patterns in time series, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining.
- Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y., 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments, in: Proceedings of the International Conference on 3D Vision.

- Chen, H., Suhr, A., Misra, D., Snavely, N., Artzi, Y., 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., Batra, D., 2018a. Embodied Question Answering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Das, A., Gkioxari, G., Lee, S., Parikh, D., Batra, D., 2018b. Neural modular control for embodied question answering, in: Proceedings of the Conference on Robot Learning.
- Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J.M., Parikh, D., Batra, D., 2017a. Visual Dialog, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Das, A., Kottur, S., Moura, J.M., Lee, S., Batra, D., 2017b. Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning, in: Proceedings of the International Conference on Computer Vision.
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Deng, Z., Narasimhan, K., Russakovsky, O., 2020. Evolving graphical planner: Contextual global planning for vision-and-language navigation, in: Advances in Neural Information Processing Systems.
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics.
- Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., Darrell, T., 2018. Speaker-follower models for vision-and-language navigation, in: Advances in Neural Information Processing Systems.
- Fu, J., Korattikara, A., Levine, S., Guadarrama, S., 2019. From Language to Goals: Inverse Reinforcement Learning for Vision-Based Instruction Following, in: Proceedings of the International Conference on Learning Representations.
- Gibson, J.J., 2014. The Ecological Approach to Visual Perception: Classic Edition. Psychology Press.
- Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D., 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Gupta, S., Davidson, J., Levine, S., Sukthankar, R., Malik, J., 2017. Cognitive mapping and planning for visual navigation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Hao, W., Li, C., Li, X., Carin, L., Gao, J., 2020. Towards Learning a Generic Agent for Vision-and-Language Navigation via Pre-training, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Hong, Y., Rodriguez-Opazo, C., Qi, Y., Wu, Q., Gould, S., 2020. Language and visual entity relationship graph for agent navigation, in: Advances in Neural Information Processing Systems.
- Jain, V., Magalhaes, G., Ku, A., Vaswani, A., Ie, E., Baldrige, J., 2019. Stay on the Path: Instruction Fidelity in Vision-and-Language Navigation, in: Proceedings of Annual Meeting of the Association for Computational Linguistics.
- Ke, L., Li, X., Bisk, Y., Holtzman, A., Gan, Z., Liu, J., Gao, J., Choi, Y., Srinivasa, S., 2019. Tactical Rewind: Self-Correction via Backtracking in Vision-and-Language Navigation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Kingma, D., Ba, J., 2015. Adam: a method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations.
- Landi, F., Baraldi, L., Corsini, M., Cucchiara, R., 2019. Embodied Vision-and-Language Navigation with Dynamic Convolutional Filters, in: Proceedings of the British Machine Vision Conference.
- Li, X., Li, C., Xia, Q., Bisk, Y., Celikyilmaz, A., Gao, J., Smith, N., Choi, Y., 2019. Robust Navigation with Language Pretraining and Stochastic Sampling, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing.
- Lu, J., Batra, D., Parikh, D., Lee, S., 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, in: Advances in Neural Information Processing Systems.
- Ma, C.Y., Lu, J., Wu, Z., AlRegib, G., Kira, Z., Socher, R., Xiong, C., 2019a. Self-monitoring navigation agent via auxiliary progress estimation, in: Proceedings of the International Conference on Learning Representations.
- Ma, C.Y., Wu, Z., AlRegib, G., Xiong, C., Kira, Z., 2019b. The Regretful Agent: Heuristic-Aided Navigation through Progress Estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Magalhaes, G., Jain, V., Ku, A., Ie, E., Baldrige, J., 2019. Effective and general evaluation for instruction conditioned navigation using dynamic time warping. arXiv preprint arXiv:1907.05446 .
- Pennington, J., Socher, R., Manning, C.D., 2014. GloVe: Global Vectors for Word Representation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing.
- Qi, Y., Pan, Z., Zhang, S., van den Hengel, A., Wu, Q., 2020a. Object-and-action aware model for visual language navigation, in: Proceedings of the European Conference on Computer Vision.
- Qi, Y., Wu, Q., Anderson, P., Wang, X., Wang, W.Y., Shen, C., Hengel, A.v.d., 2020b. REVERIE: Remote Embodied Visual Referring Expression in Real Indoor Environments, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., Batra, D., 2019. Habitat: A Platform for Embodied AI Research, in: Proceedings of the International Conference on Computer Vision.
- Shen, W.B., Xu, D., Zhu, Y., Guibas, L.J., Fei-Fei, L., Savarese, S., 2019. Situational Fusion of Visual Representation for Visual Navigation, in: Proceedings of the International Conference on Computer Vision.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1929–1958.
- Tan, H., Yu, L., Bansal, M., 2019. Learning to Navigate Unseen Environments: Back Translation with Environmental Dropout, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics.
- Thomason, J., Gordon, D., Bisk, Y., 2018. Shifting the Baseline: Single Modality Performance on Visual Navigation & QA, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: Advances in Neural Information Processing Systems.
- Vinyals, O., Toshev, A., Bengio, S., Erhan, D., 2015. Show and tell: A neural image caption generator, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.F., Wang, W.Y., Zhang, L., 2019. Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Wang, X., Xiong, W., Wang, H., Yang Wang, W., 2018. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation, in: Proceedings of the European Conference on Computer Vision.
- Williams, R.J., 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8, 229–256.
- Xia, F., Zamir, A.R., He, Z., Sax, A., Malik, J., Savarese, S., 2018. Gibson env: Real-world perception for embodied agents, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y., 2015. Show, attend and tell: Neural image caption generation with visual attention, in: Proceedings of the International Conference on Machine Learning.
- Yang, J., Ren, Z., Xu, M., Chen, X., Crandall, D., Parikh, D., Batra, D., 2019. Embodied Amodal Recognition: Learning to Move to Perceive Objects, in: Proceedings of the International Conference on Computer Vision.
- Zhang, W., Ma, C., Wu, Q., Yang, X., 2020. Language-guided Navigation via Cross-Modal Grounding and Alternate Adversarial Learning. *IEEE Transactions on Circuits and Systems for Video Technology* .
- Zhu, F., Zhu, Y., Chang, X., Liang, X., 2020. Vision-language navigation with self-supervised auxiliary reasoning tasks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.