Actional Atomic-Concept Learning for Demystifying Vision-Language Navigation

Bingqian Lin[1*], **Yi Zhu**[2], **Xiaodan Liang**[1,3†], **Liang Lin**[4], **Jianzhuang Liu**[2]
[1]Shenzhen Campus of Sun Yat-sen University, Shenzhen
[2]Huawei Noah's Ark Lab
[3]PengCheng Laboratory
[4]Sun Yat-sen University
linbq6@mail2.sysu.edu.cn, zhuyi36@huawei.com, liangxd9@mail.sysu.edu.cn,
linliang@ieee.org, liu.jianzhuang@huawei.com

Abstract

Vision-Language Navigation (VLN) is a challenging task which requires an agent to align complex visual observations to language instructions to reach the goal position. Most existing VLN agents directly learn to align the raw directional features and visual features trained using one-hot labels to linguistic instruction features. However, the big semantic gap among these multi-modal inputs makes the alignment difficult and therefore limits the navigation performance. In this paper, we propose Actional Atomic-Concept Learning (AACL), which maps visual observations to actional atomic concepts for facilitating the alignment. Specifically, an actional atomic concept is a natural language phrase containing an atomic action and an object, e.g., "go up stairs". These actional atomic concepts, which serve as the bridge between observations and instructions, can effectively mitigate the semantic gap and simplify the alignment. AACL contains three core components: 1) a concept mapping module to map the observations to the actional atomic concept representations through the VLN environment and the recently proposed Contrastive Language-Image Pretraining (CLIP) model, 2) a concept refining adapter to encourage more instruction-oriented object concept extraction by re-ranking the predicted object concepts by CLIP, and 3) an observation co-embedding module which utilizes concept representations to regularize the observation representations. Our AACL establishes new state-of-the-art results on both fine-grained (R2R) and high-level (REVERIE and R2R-Last) VLN benchmarks. Moreover, the visualization shows that AACL significantly improves the interpretability in action decision. Code will be available at https://gitee.com/mindspore/models/tree/master/research/cv/VLN-AACL.

Introduction

Vision-Language Navigation (VLN) (Anderson et al. 2018; Ku et al. 2020; Chen et al. 2019; Nguyen and Daumé 2019) has attracted increasing interests in robotic applications since an instruction-following navigation agent is practical and flexible in real-world scenarios. For accomplishing successful navigation, a VLN agent needs to align complicated visual observations to language instructions to reach the required target point. For example, when asking to "turn left to the bathroom", the agent should choose the right observation which not only contains the mentioned object "bathroom" but also indicates the direction "turn left".

Most of early VLN approaches adopt the LSTM-based encoder-decoder framework (Fried et al. 2018; Tan, Yu, and Bansal 2019; Wang et al. 2019; Ma et al. 2019; Zhu et al. 2020), which encodes both the visual observations and language instructions and then generates the action sequence. With the development of large-scale cross-modal pretraining models in vision-language tasks (Li et al. 2020a,b; Chen et al. 2020; Lu et al. 2019), emerging works attempt to introduce them into VLN tasks (Hao et al. 2020; Hong et al. 2021; Chen et al. 2021; Moudgil et al. 2021). However, both the non-pretraining-based or pretraining-based approaches represent the visual observations by raw directional features and visual features trained using one-hot labels, which are difficult to be aligned to the linguistic instruction features due to the large semantic gap among them. This direct alignment process also leads to poor interpretability of action decision and therefore makes the VLN agents unreliable to be deployed to real environments.

In this work, we aim to *mitigate the semantic gap and simplify the alignment* in VLN by proposing a new framework, called Actional Atomic-Concept Learning (AACL). Since the instructions usually consist of atomic action concepts, e.g., "turn right", and object concepts[1], e.g., "kitchen", in AACL, the visual observations are mapped to actional atomic concepts, which are natural language phrases each containing an action and an object. The actions are extracted from a predefined atomic action set. These actional atomic concepts, which can be viewed as the bridge between observations and instructions, effectively facilitate the alignment as well as provide good interpretability for action decision.

AACL consists of three main components. Firstly, a **concept mapping module** is constructed to map each single view observation to the actional atomic concept. For deriving the object concept, we resort to the recently

[*] Part of this work was done during an internship in Huawei Noah's Ark Lab.

[†] Corresponding author.

[1] In this work, we also treat the scene concept, e.g., "bathroom", as the object concept.

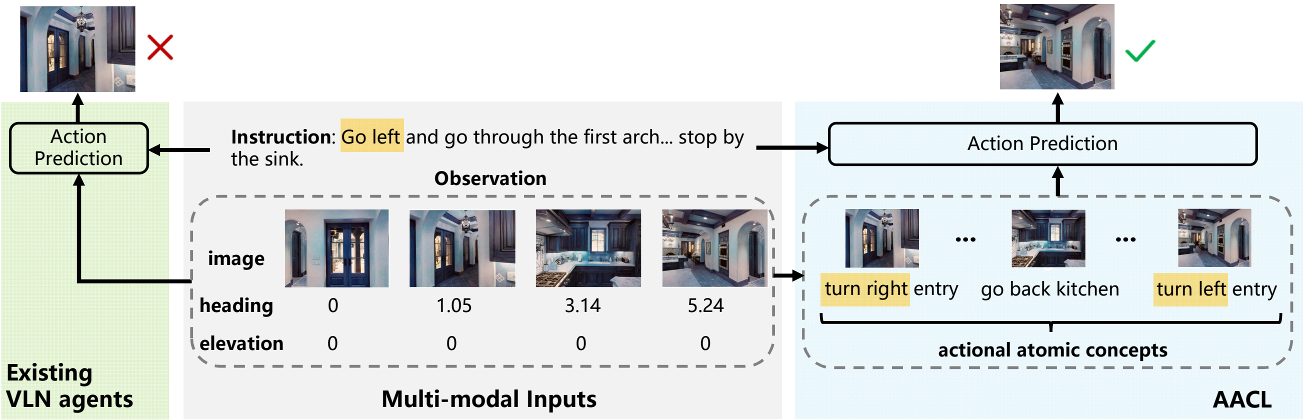


Figure 1: Comparison between existing VLN agents and the proposed AACL. Through mapping the visual observations to actional atomic concepts, AACL can simplify the multi-modal alignment and distinguish different observation candidates easily to make accurate action decision.

proposed Contrastive Language-Image Pretraining (CLIP) model (Radford et al. 2021) rather than image classification or object detection models pretrained on a fixed category set. Benefiting from the powerful open-world object recognition ability of CLIP, AACL can better generalize to diverse navigation scenarios. And we map the sequential direction information in VLN environments during navigation to the action concept. Secondly, to encourage more instruction-oriented object concept extraction for facilitating the multi-modal alignment, a **concept refining adapter** is further introduced to re-rank the predicted object concepts of CLIP according to the instruction. Lastly, an **observation co-embedding module** embeds each observation and its paired actional atomic concept, and then uses concept representations to regularize the observation representations through an observation contrast strategy. Figure 1 presents an action selection comparison between existing VLN agents and our AACL. Through mapping visual observations to actional atomic concepts formed by language, AACL can simplify the modality alignment and distinguish different action candidates easier to make correct actions.

We conduct experiments on several popular VLN benchmarks, including one with fine-grained instructions (R2R (Anderson et al. 2018)) and two with high-level instructions (REVERIE (Qi et al. 2020) and R2R-Last (Chen et al. 2021)). Experimental results show that our AACL outperforms the state-of-the-art approaches on all benchmarks. Moreover, benefiting from these actional atomic concepts, AACL shows excellent interpretability in making action decision, which is a step closer towards developing reliable VLN agents in real-world applications.

Related Work

Vision-Language Navigation. Developing navigation agents which can follow natural language instructions has attracted increasingly research interests in recent years (Anderson et al. 2018; Ku et al. 2020; Chen et al. 2019; Nguyen and Daumé 2019; Qi et al. 2020). Most of early Vision-Language Navigation (VLN) approaches employ the LSTM-based encoder-decoder framework (Fried et al. 2018; Tan, Yu, and Bansal 2019; Wang et al. 2019; Ma et al. 2019; Zhu et al. 2020; Wang, Wu, and Shen 2020; Qi et al. 2020; Fu et al. 2020). Due to the success of Transformer-based cross-modal pretraining (Li et al. 2020a,b; Chen et al. 2020; Lu et al. 2019; Li et al. 2019), recent works have explored transformer architectures into VLN tasks (Hao et al. 2020; Hong et al. 2021; Chen et al. 2021; Moudgil et al. 2021; Qi et al. 2021; Lin et al. 2022). HAMT (Chen et al. 2021) develops a history aware multimodal transformer to better encode the long-horizon navigation history. DUET (Chen et al. 2022) constructs a dual-scale graph transformer for joint long-term action planning and fine-grained cross-modal understanding. However, these pretraining-based methods still learn to align the raw directional features and visual features trained using one-hot labels to the linguistic instruction features, leading to limited performance due to the large semantic gap among these multi-modal inputs.

In contrast to the above mentioned VLN approaches, in this work, we build a bridge among multi-modal inputs for facilitating the alignment by introducing actional atomic concepts formed by language. Through these actional atomic concepts, the alignment can be significantly simplified and good interpretability can be provided.

Contrastive Language-Image Pretraining (CLIP). CLIP (Radford et al. 2021) is a cross-modal pretrained model using 400 million image and text pairs collected from the web. Through natural language supervision (Jia et al. 2021; Sariyildiz, Perez, and Larlus 2020; Desai and Johnson 2021) rather than one-hot labels of fixed size of object categories, CLIP has shown great potential in open-world object recognition. Recently, many works have attempted to introduce CLIP into various computer vision (CV) or vision-language (V&L) tasks to improve the generalization ability of the downstream models (Song et al. 2022; Subramanian et al. 2022; Khandelwal et al. 2022; Shen et al. 2022; Rao et al. 2022; Dai et al. 2022; Liang et al. 2022). Dense-

CLIP (Rao et al. 2022) introduces CLIP into dense prediction tasks, e.g., semantic segmentation, through converting the original image-text matching in CLIP to the pixel-text matching. (Dai et al. 2022) distills the vision-language knowledge learned in CLIP to enhance the multimodal generation models. EmbCLIP (Khandelwal et al. 2022) investigates the ability of CLIP's visual representations in improving embodied AI tasks. Some works have also tried to apply CLIP into VLN tasks (Liang et al. 2022; Shen et al. 2022). ProbES (Liang et al. 2022) utilizes the knowledge learned from CLIP to build an in-domain dataset by self-exploration for pretraining. (Shen et al. 2022) replaces the ResNet visual encoder pretrained on ImageNet in the conventional VLN models with the pretrained CLIP visual encoder.

In this paper, we resort to the powerful object recognition ability of CLIP to provide object concepts for each single-view observation. To encourage instruction-oriented object concept extraction for better alignment, a concept refining adapter is further introduced beyond CLIP to re-rank its predicted object concepts according to the instruction.

Preliminaries

VLN Problem Setup

Given a natural language instruction $I = \{w_1, ..., w_l\}$ with l words, a VLN agent is asked to navigate from a start viewpoint S to the goal viewpoint G. At each timestep t, the agent receives a panoramic observation, containing N_o image views $O_t = \{O_{t,n}\}_{n=1}^{N_o}$. Each $O_{t,n}$ contains the image $B_{t,n}$ and the attached direction information $A_{t,n}$. The visual feature $\mathbf{v}_{t,n}$ for $B_{t,n}$ is obtained by a pretrained ResNet (He et al. 2016) or ViT (Dosovitskiy et al. 2021). $A_{t,n}$ is usually composed of the heading $\psi_{t,n}$ and the elevation $\theta_{t,n}$. Each panoramic observation contains d navigable viewpoints $C_t = \{C_{t,i}\}_{i=1}^{d}$ as the action candidates. At timestep t, the agent predicts an action $\mathbf{a}_t$ from C_t based on the instruction I and current visual observations O_t.

Baseline Agents

Our AACL can be applied to many previous VLN models. In this work, two strong baseline agents HAMT (Chen et al. 2021) and DUET (Chen et al. 2022) are selected. In this section, we briefly describe one baseline HAMT. In HAMT, the agent receives the instruction I, the panoramic observation O_t, and the navigation history H_t at each timestep t. H_t is a sequence of historical visual observations. A standard BERT (Devlin et al. 2019) is used to obtain the instruction feature $\mathbf{f}_I$ for I. For each view n with the angle information $< \psi_{t,n}, \theta_{t,n} >$ in O_t, the direction feature is defined by $\mathbf{e}_{A_{t,n}} = (\sin\psi_{t,n}, \cos\psi_{t,n}, \sin\theta_{t,n}, \cos\theta_{t,n})$. With the visual feature $\mathbf{v}_{t,n}$ and the direction feature $\mathbf{e}_{A_{t,n}}$, the observation embedding $\mathbf{o}_{t,n}$ for each view n is calculated by:

$$\begin{aligned}\mathbf{o}_{t,n} = \mathrm{Dr}(\mathrm{LN}(\mathrm{LN}(\mathbf{W}_v\mathbf{v}_{t,n}) + \mathrm{LN}(\mathbf{W}_a\mathbf{e}_{A_{t,n}})\\ + \mathbf{e}_{t,n}^{N} + \mathbf{e}_v^{T})),\end{aligned} \tag{1}$$

where $\mathrm{LN}(\cdot)$ and $\mathrm{Dr}(\cdot)$ denote layer normalization (Ba, Kiros, and Hinton 2016) and dropout, respectively. $\mathbf{W}_v$ and $\mathbf{W}_a$ are learnable weights, and $\mathbf{e}_{t,n}^{N}$ and $\mathbf{e}_v^{T}$ denote the navigable embedding and the type embedding, respectively (Chen et al. 2021). The observation feature $\mathbf{o}_t$ is represented by $\mathbf{o}_t = [\mathbf{o}_{t,1}; ...; \mathbf{o}_{t,N_o}]$, where N_o is the number of views. And a hierarchical vision transformer (Chen et al. 2021) is constructed to get the history feature $\mathbf{h}_t = [\mathbf{h}_{t,1}; ...; \mathbf{h}_{t,t-1}]$ for the navigation history H_t.

Then $\mathbf{f}_I$, $\mathbf{o}_t$, and $\mathbf{h}_t$ are fed into a cross-modal transformer encoder $E^c(\cdot)$, resulting in:

$$\tilde{\mathbf{f}}_I^t, \tilde{\mathbf{o}}_t, \tilde{\mathbf{h}}_t = E^c(\mathbf{f}_I, [\mathbf{o}_t; \mathbf{h}_t]). \tag{2}$$

The updated instruction feature $\tilde{\mathbf{f}}_I^t$ and observation feature $\tilde{\mathbf{o}}_t$ are used for action prediction:

$$\mathbf{a}_t = E^a(\tilde{\mathbf{f}}_I^t, \tilde{\mathbf{o}}_t), \tag{3}$$

where $E^a(\cdot)$ is a two-layer fully-connected network. For more model details, refer to (Chen et al. 2021).

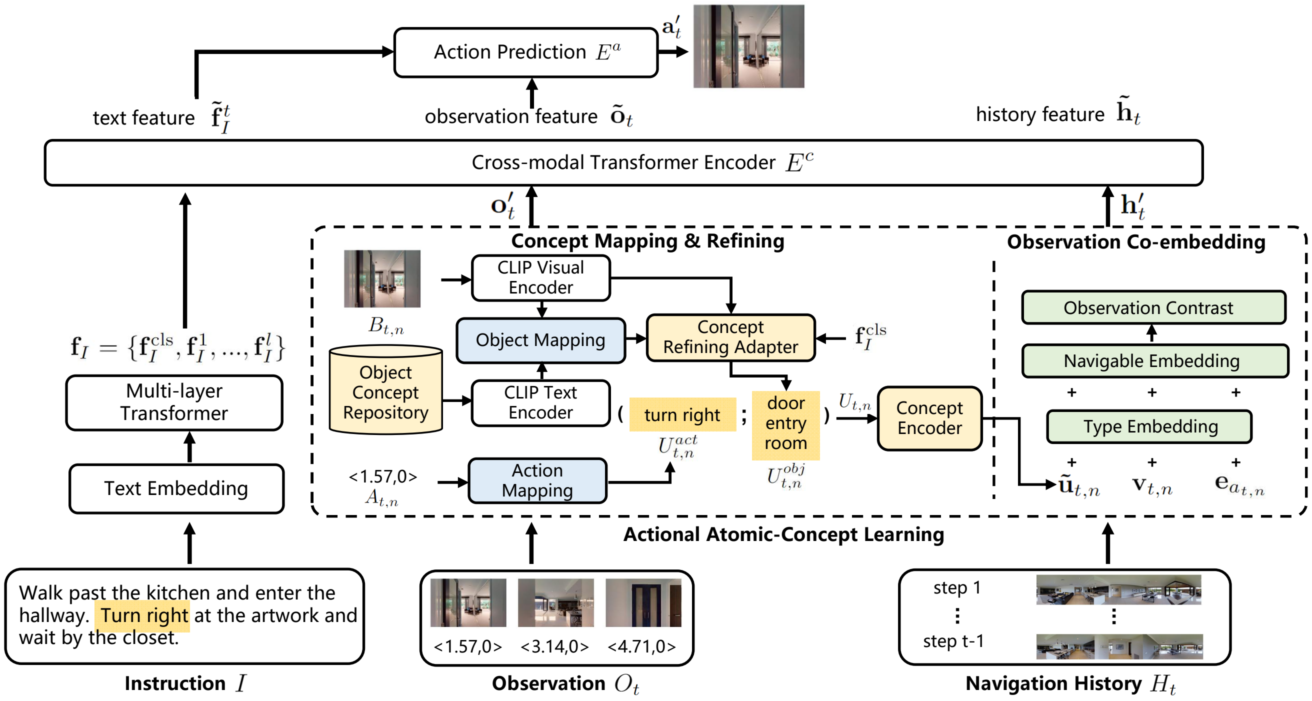
In Eq. 1, HAMT obtains the observation feature $\mathbf{o}_{t,n}$ directly by the pretrained visual feature $\mathbf{v}_{t,n}$ and the raw direction feature $\mathbf{e}_{A_{t,n}}$. In AACL, we map the observations $O_{t,n}$ to actional atomic concepts $U_{t,n}$ formed by language and use $U_{t,n}$ to obtain the new observation feature $\mathbf{o}'_{t,n}$. In this way, the gap between O_t and I can be effectively mitigated to simplify the alignment.

Actional Atomic-Concept Learning

In this section, we describe our AACL in detail, the overview of which is presented in Figure 2. At timestep t, the agent receives multi-modal inputs I, O_t, and H_t similar to HAMT. For each $O_{t,n}$ in O_t containing the single-view image $B_{t,n}$ and the direction $A_{t,n}$, AACL first conducts object concept mapping and atomic action concept mapping to obtain the object concept $U_{t,n}^{obj}$ and the action concept $U_{t,n}^{act}$. And a concept refining adapter is built to re-rank $U_{t,n}^{obj}$ according to the instruction I for better alignment. The actional atomic concept $U_{t,n}$ is then obtained by concatenating $U_{t,n}^{act}$ and $U_{t,n}^{obj}$, and fed to the concept encoder $E^t(\cdot)$ to get the concept feature $\tilde{\mathbf{u}}_{t,n}$. Finally, an observation co-embedding module is constructed to use $\tilde{\mathbf{u}}_{t,n}$ for regularizing the visual feature $\mathbf{v}_{t,n}$ and the directional feature $\mathbf{e}_{A_{t,n}}$ to get new observation features $\mathbf{o}'_{t,n}$. For H_t which contains historical visual observations, we also use AACL to get the enhanced history features $\mathbf{h}'_t$ like O_t. Then $\mathbf{o}'_t = \{\mathbf{o}'_{t,n}\}_{n=1}^{N_o}$, $\mathbf{h}'_t$, and the instruction features $\mathbf{f}_I$ are fed to the cross-modal Transformer encoder $E^c(\cdot)$ for calculating the action predictions $\mathbf{a}'_t$.

Actional Atomic-Concept Mapping

Object Concept Mapping. For each observation $O_{t,n}$ containing the single-view image $B_{t,n}$, we map $B_{t,n}$ to get the object concept $U_{t,n}^{obj}$ based on a pre-built in-domain object concept repository. Benefiting from large-scale language supervision from 400M image-text pairs, CLIP (Radford et al. 2021) has more powerful open-world object recognition ability than conventional image classification or object detection models pretrained on a fixed-size object category set. In this work, we resort to CLIP to conduct the object concept mapping considering its good generalizability. Concretely,



Walk past the kitchen and enter the hallway. Turn right at the artwork and wait by the closet.

Figure 2: Overview of our Actional Atomic-Concept Learning (AACL). At each timestep t, the agent receives the instruction I, the observation O_t, and the navigation history H_t. For each $O_{t,n}$ in O_t containing the single-view image $B_{t,n}$ and the direction $A_{t,n}$, object concept mapping and action concept mapping are conducted based on the concept refining adapter to obtain the actional atomic concept representations $\tilde{\mathbf{u}}_{t,n}$. Then $\tilde{\mathbf{u}}_{t,n}$ is used to regularize the visual representation $\mathbf{v}_{t,n}$ and the directional representation $\mathbf{e}_{A_{t,n}}$ through the observation co-embedding module for making action selection. For simplicity, we omit the learning process of navigation histories H_t, which is similar to that of observations O_t.

the object concept repository $\{U_c^{obj}\}_{c=1}^{N_c}$ is constructed by extracting object words from the training dataset, where N_c is the repository size. And we get the image feature $\mathbf{f}_{B_{t,n}}$ through the pretrained CLIP Image Encoder $E_{\text{CLIP}}^v(\cdot)$:

$$\mathbf{f}_{B_{t,n}} = E_{\text{CLIP}}^v(B_{t,n}). \tag{4}$$

For object concept U_c^{obj}, we construct the text phrase T_c formed as “a photo of a $\{U_c^{obj}\}$”. Then the text feature $\mathbf{f}_{T_c}$ is derived through the pretrained CLIP Text Encoder $E_{\text{CLIP}}^t(\cdot)$:

$$\mathbf{f}_{T_c} = E_{\text{CLIP}}^t(T_c). \tag{5}$$

Then the mapping probability of the image $B_{t,n}$ regarding the object concept U_c^{obj} is calculated by:

$$\mathbf{p}(y = U_c^{obj}|B_{t,n}) = \frac{\exp(\text{sim}(\mathbf{f}_{B_{t,n}}, \mathbf{f}_{T_c})/\tau)}{\sum_{c=1}^{N_c}(\exp(\text{sim}(\mathbf{f}_{B_{t,n}}, \mathbf{f}_{T_c})/\tau))}, \tag{6}$$

where $\text{sim}(\cdot,\cdot)$ denotes the cosine similarity, τ represents the temperature parameter. Considering that a single-view image in the observation usually contains more than one salient object, we extract the top k object concepts (text) having the maximum mapping probabilities conditioned on $B_{t,n}$ as its corresponding object concepts, i.e., $U_{t,n}^{obj} = \{U_{t,n,i}^{obj}\}_{i=1}^k$.

Atomic Action Concept Mapping. The atomic action concept $U_{t,n}^{act}$ for $O_{t,n}$ can be derived through its directional information $A_{t,n}$ and the directional information $\tilde{A}_{t-1}$ of the agent's selected action at timestep t-1. We first use six basic actions in VLN tasks to build the predefined atomic action set, i.e., *go up, go down, go forward, go back, turn right,* and *turn left*. Denote $A_{t,n} =< \psi_{t,n}, \theta_{t,n} >$, where $\psi_{t,n} \in [0, 2\pi)$ and $\theta_{t,n} \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ are the heading and the elevation, respectively. Similarly, $\tilde{A}_{t-1} =< \tilde{\psi}_{t-1}, \tilde{\theta}_{t-1} >$, where $\tilde{\psi}_{t-1} \in [0, 2\pi)$ and $\tilde{\theta}_{t-1} \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. We calculate the relative direction of $< \psi_{t,n}, \theta_{t,n} >$ to $< \tilde{\psi}_{t-1}, \tilde{\theta}_{t-1} >$ by:

$$\tilde{\psi}_{t,n} = \psi_{t,n} - \tilde{\psi}_{t-1}, \quad \tilde{\theta}_{t,n} = \theta_{t,n} - \tilde{\theta}_{t-1}. \tag{7}$$

Then we use $< \tilde{\psi}_{t,n}, \tilde{\theta}_{t,n} >$ to obtain $U_{t,n}^{act}$. Following the direction judgement rule in VLN (Anderson et al. 2018), we use $\tilde{\theta}_{t,n}$ first to judge whether $U_{t,n}^{act}$ is “go up” or “go down” by comparing it to zero. Otherwise, $U_{t,n}^{act}$ is further determined through $\tilde{\psi}_{t,n}$. Specifically, if $\tilde{\psi}_{t,n}$ is equal to zero, $U_{t,n}^{act}$ is “go forward”. Otherwise, $U_{t,n}^{act}$ is further determined to be “turn right”, “turn left”, or “go back”. The detailed mapping rule is listed in Table 1.

Concept Refining Adapter

After getting $\{U_{t,n,i}^{obj}\}_{i=1}^k$ and $U_{t,n}^{act}$ for each $O_{t,n}$, the actional atomic concept $\{U_{t,n,i}\}_{i=1}^k$ can be obtained by directly concatenating $U_{t,n}^{act}$ and each $U_{t,n,i}^{obj}$. A direct way to

Elevation $\tilde{\theta}_{t,n}$	Heading $\tilde{\psi}_{t,n}$						
	(-2π, -3π/2]	(-3π/2, -π/2)	[-π/2, 0)	0	(0, π/2]	(π/2, 3π/2)	[3π/2, 2π)
>0	go up						
<0	go down						
0	turn right	go back	turn left	go forward	turn right	go back	turn left

Table 1: Atomic Action Concept Mapping.

obtain the actional atomic concept feature $\mathbf{u}_{t,n}$ is to feed each $U_{t,n,i}$ to the concept encoder $E^t(\cdot)$ and get a weighted sum based on their object prediction probability $\mathbf{p}_i$ by CLIP:

$$\mathbf{u}_{t,n} = \sum_{i=1}^{k} \mathbf{p}_i \cdot E^t(U_{t,n,i}), \tag{8}$$

where $E^t(\cdot)$ is the BERT-like language encoder. However, even if CLIP can extract informative object concepts for each observation, some noisy object concepts may exist and extracting instruction-oriented object concepts would be more useful for alignment and making action decisions. Inspired by (Gao et al. 2021), we propose to construct a concept refining adapter beyond CLIP to refine the object concept under the constraint of the instruction. Given a feature $\mathbf{f}$, the concept refining adapter is written as:

$$A(\mathbf{f}) = \text{ReLU}(\mathbf{f}^T\mathbf{W}_1)\mathbf{W}_2, \tag{9}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable parameters, and $\text{ReLU}(\cdot)$ is the rectified linear unit for activation. Denote the instruction feature as $\mathbf{f}_I = \{\mathbf{f}_I^{cls}, \mathbf{f}_I^1, ...\mathbf{f}_I^l\}$. For the image feature $\mathbf{f}_{B_{t,n}}$ of the single-view image $B_{t,n}$, we obtain the updated image feature $\tilde{\mathbf{f}}_{B_{t,n}}$ by feeding $\mathbf{f}_{B_{t,n}}$ and $\mathbf{f}_I^{cls}$ to $A(\cdot)$:

$$\tilde{\mathbf{f}}_{B_{t,n}} = \alpha \cdot \mathbf{f}_{B_{t,n}} + (1-\alpha) \cdot A([\mathbf{f}_{B_{t,n}}; \mathbf{f}_I^{cls}]), \tag{10}$$

where α servers as the residual ratio to help adjust the degree of maintaining the original knowledge for better performance (Gao et al. 2021), and $[\cdot;\cdot]$ denotes feature concatenation. Denote the top k object concept features obtained by CLIP for the single-view image $B_{t,n}$ as $\{\mathbf{f}_{T_i}\}_{i=1}^k$. We use the updated image feature $\tilde{\mathbf{f}}_{B_{t,n}}$ to get the re-ranking object prediction probability $\tilde{\mathbf{p}}$ of $\{\mathbf{f}_{T_i}\}_{i=1}^k$:

$$\tilde{\mathbf{p}} = \text{Softmax}(\text{sim}(\tilde{\mathbf{f}}_{B_{t,n}}, \mathbf{f}_{T_1}), ..., \text{sim}(\tilde{\mathbf{f}}_{B_{t,n}}, \mathbf{f}_{T_k})). \tag{11}$$

Then we get the refined concept feature $\tilde{\mathbf{u}}_{t,n}$ by replacing $\mathbf{p}$ in Eq. 8 by $\tilde{\mathbf{p}}$.

Observation Co-Embedding

After obtaining the concept feature $\tilde{\mathbf{u}}_{t,n}$ for each single-view observation $O_{t,n}$, we introduce an observation co-embedding module to use $\tilde{\mathbf{u}}_{t,n}$ for bridging multi-modal inputs and calculating the final observation feature $\mathbf{o}'_{t,n}$. At first, we separately embed the visual feature $\mathbf{v}_{t,n}$, the direction feature $\mathbf{e}_{A_{t,n}}$, and the concept feature $\tilde{\mathbf{u}}_{t,n}$ to obtain $\mathbf{o}^v_{t,n}$, $\mathbf{o}^a_{t,n}$, and $\mathbf{o}^u_{t,n}$, respectively, by:

$$\begin{aligned}
\mathbf{o}^v_{t,n} &= \text{Dr}(\text{LN}(\text{LN}(\tilde{\mathbf{W}}_v\mathbf{v}_{t,n}) + \mathbf{e}^N_{t,n} + \mathbf{e}^T_v)),\\
\mathbf{o}^a_{t,n} &= \text{Dr}(\text{LN}(\text{LN}(\tilde{\mathbf{W}}_a\mathbf{e}_{A_{t,n}}) + \mathbf{e}^N_{t,n} + \mathbf{e}^T_v)),\\
\mathbf{o}^u_{t,n} &= \text{Dr}(\text{LN}(\text{LN}(\tilde{\mathbf{W}}_u\tilde{\mathbf{u}}_{t,n}) + \mathbf{e}^N_{t,n} + \mathbf{e}^T_v)),
\end{aligned} \tag{12}$$

where $\tilde{\mathbf{W}}_v$, $\tilde{\mathbf{W}}_a$ and $\tilde{\mathbf{W}}_u$ are learnable weights.

Unlike HAMT that combining different features into one embedding (Eq. 1), we keep the separate embeddings as in Eq. 12 such that a new observation contrast strategy can be performed. Concretely, the view embedding $\mathbf{o}^v_{t,n}$ and the direction embedding $\mathbf{o}^a_{t,n}$ are summed as the visual embedding $\mathbf{o}^V_{t,n} = \mathbf{o}^v_{t,n} + \mathbf{o}^a_{t,n}$. Then $\mathbf{o}^V_{t,n}$ in each single-view observation $O_{t,n}$ is forced to stay close to the paired concept embedding $\mathbf{o}^u_{t,n}$ while staying far away from the concept embeddings $\bar{\mathbf{o}}^u_{t,n}$ in other single-view observations in O_t:

$$\begin{aligned}
\mathcal{L}_\text{c} = -\sum_t\sum_n \log(& e^{\text{sim}(\mathbf{o}^V_{t,n}, \mathbf{o}^u_{t,n})/\tau} /\\
& (e^{\text{sim}(\mathbf{o}^V_{t,n}, \mathbf{o}^u_{t,n})/\tau} + \sum_{\bar{\mathbf{o}}^u_{t,n}} e^{\text{sim}(\mathbf{o}^V_{t,n}, \bar{\mathbf{o}}^u_{t,n})/\tau})),
\end{aligned} \tag{13}$$

where τ is the temperature parameter. By observation contrast, the discrimination of each single-view observation can be effectively enhanced and the semantic gap between observations and instructions can be largely mitigated with the help of the actional atomic concept. To fully merge the information for each observation, we use $\mathbf{o}'_{t,n} = \mathbf{o}^V_{t,n} + \mathbf{o}^u_{t,n}$ to obtain the final observation feature $\mathbf{o}'_{t,n}$.

Action Prediction

Similar to the observation feature $\mathbf{o}'_t = \{\mathbf{o}'_{t,n}\}_{n=1}^{N_o}$ (N_o is the number of views), the history feature $\mathbf{h}'_t$ is obtained for H_t through AACL. With $\mathbf{o}'_t$, $\mathbf{h}'_t$, and the instruction feature $\mathbf{f}_I$, the action $\mathbf{a}'_t$ can be obtained from the cross-modal transformer encoder $E^c(\cdot)$ and the action prediction module $E^a(\cdot)$ (see Eq. 2 and Eq. 3). Following most existing VLN works (Tan, Yu, and Bansal 2019; Hong et al. 2021; Chen et al. 2021), we combine Imitation Learning (IL) and Reinforcement Learning (RL) to train VLN agents. Let the imitation learning loss be $\mathcal{L}_\text{IL}$ and the reinforcement learning loss be $\mathcal{L}_\text{RL}$. The total training objective of AACL is:

$$\mathcal{L} = \mathcal{L}_\text{RL} + \lambda_1\mathcal{L}_\text{IL} + \lambda_2\mathcal{L}_\text{c}, \tag{14}$$

where λ_1 and λ_2 are balance parameters.

Experiments

Experimental Setup

Datasets. We evaluate AACL on several popular VLN benchmarks with both fine-grained instructions (R2R (Anderson et al. 2018)) and high-level instructions (REVERIE (Qi et al. 2020) and R2R-Last (Chen et al. 2021)). R2R includes 90 indoor scenes with 7189 trajectories. The dataset is split into train, val seen, val unseen, and

Method	Val Seen				Val Unseen				Test Unseen			
	TL	NE ↓	SR ↑	SPL ↑	TL	NE ↓	SR ↑	SPL ↑	TL	NE ↓	SR ↑	SPL ↑
Seq2Seq (Anderson et al. 2018)	11.33	6.01	39	-	8.39	7.81	22	-	8.13	7.85	20	18
RCM+SIL(train) (Wang et al. 2019)	10.65	3.53	67	-	11.46	6.09	43	-	11.97	6.12	43	38
EnvDropout (Tan, Yu, and Bansal 2019)	11.00	3.99	62	59	10.70	5.22	52	48	11.66	5.23	51	47
PREVALENT (Hao et al. 2020)	10.32	3.67	69	65	10.19	4.71	58	53	10.51	5.30	54	51
ORIST (Qi et al. 2021)	-	-	-	-	10.90	4.72	57	51	11.31	5.10	57	52
VLN↻BERT (Hong et al. 2021)	11.13	2.90	72	68	12.01	3.93	63	57	12.35	4.09	63	57
HAMT (Chen et al. 2021) (baseline)	11.15	2.51	76	72	11.46	3.62	66	61	12.27	3.93	65	60
HAMT+AACL (ours)	11.31	2.53	76	72	12.09	3.41	69	**63**	12.74	3.71	66	**61**
DUET (Chen et al. 2022) (baseline)	12.32	2.28	79	**73**	13.94	3.31	72	60	14.73	3.65	69	59
DUET+AACL (ours)	13.32	**2.15**	**80**	72	15.01	**3.00**	**74**	61	15.47	**3.38**	**71**	59

Table 2: Comparison with the SOTA methods on R2R.

Method	Val Unseen						Test Unseen					
	TL	SR ↑	OSR↑	SPL ↑	RGS↑	RGSPL↑	TL	SR ↑	OSR↑	SPL ↑	RGS↑	RGSPL↑
RCM (Wang et al. 2019)	11.98	9.29	14.23	6.97	4.89	3.89	10.60	7.84	11.68	6.67	3.67	3.14
SMNA (Ma et al. 2019)	9.07	8.15	11.28	6.44	4.54	3.61	9.23	5.80	8.39	4.53	3.10	2.39
FAST-MATTN (Qi et al. 2020)	45.28	14.40	28.20	7.19	7.84	4.67	39.05	19.88	30.63	11.60	11.28	6.08
SIA (Lin, Li, and Yu 2021)	41.53	31.53	44.67	16.28	22.41	11.56	48.61	30.80	44.56	14.85	19.02	9.20
VLN↻BERT (Hong et al. 2021)	16.78	30.67	35.02	24.90	18.77	15.27	15.86	29.61	32.91	23.99	16.50	13.51
HAMT (baseline)	14.08	32.95	36.84	30.20	18.92	17.28	13.62	30.40	33.41	26.67	14.88	13.08
HAMT+AACL (ours)	14.08	34.17	38.54	29.70	20.53	17.69	13.30	35.52	39.57	31.34	18.04	15.96
DUET (baseline)	22.11	46.98	51.07	**33.73**	32.15	**23.03**	21.30	52.51	56.91	36.06	31.88	22.06
DUET+AACL (ours)	**23.77**	**49.42**	**53.93**	33.54	**33.31**	22.49	21.88	**55.09**	**59.92**	**37.08**	**33.17**	**22.55**

Table 3: Navigation and object grounding performance on REVERIE.

Method	Val Seen		Val Unseen	
	SR↑	SPL↑	SR↑	SPL↑
EnvDrop	42.8	38.4	34.3	28.3
VLN↻BERT	50.2	45.8	41.6	37.3
HAMT (baseline)	53.3	50.3	45.2	41.2
HAMT+AACL (ours)	**54.2**	**51.1**	**47.2**	**42.1**

Table 4: Comparison on R2R-Last.

test unseen sets with 61, 56, 11, and 18 scenes, respectively. REVERIE replaces the fine-grained instructions in R2R with high-level instructions which mainly target at object localization. R2R-Last only uses the last sentence of the original R2R instruction as the instruction.

Evaluation Metrics. We adopt the common metrics used in previous works (Chen et al. 2021; Anderson et al. 2018; Qi et al. 2020) to evaluate the model performance: 1) Navigation Error (NE) calculates the average distance between the agent stop position and the target viewpoint, 2) Trajectory Length (TL) is the average path length in meters, 3) Success Rate (SR) is the ratio of stopping within 3 meters to the goal, 4) Success rate weighted by Path Length (SPL) makes the trade-off between SR and TL, 5) Oracle Success Rate (OSR) calculates the ratio of containing a viewpoint along the path where the target object is visible, 6) Remote Grounding Success Rate (RGS) is the ratio of performing correct object grounding when stopping, and 7) Remote Grounding Success weighted by Path Length (RGSPL) weights RGS by TL. 1)–4), 3)–4), and 2)–7) are used for evaluation on R2R, R2R-Last, and REVERIE, respectively.

Baselines. In this work, we choose two strong baseline agents, HAMT (Chen et al. 2021) and DUET (Chen et al. 2022) to verify AACL's effectiveness. In HAMT, a hierarchical transformer is adopted for storing historical observations and actions. In contrast, DUET keeps track of all visited and navigable locations through a topological map.

Implementation Details. We implement our model using the MindSpore Lite tool[2]. The batch size is set to 8, 8, 4 on R2R, R2R-Last, and REVERIE, respectively. The temperature parameter τ is set to 0.5. The loss weight λ_1 is set to 0.2 on all datasets, and the loss weight λ_2 is set to 1, 1, and 0.01 on R2R, REVERIE, and R2R-Last, respectively. The residual ratio in Eq. 10 is set to 0.8 empirically. During object concept mapping, we remain top 5 object predictions for each observation. The learning rate of the concept refining adapter is set to 0.1.

Quantitative Results

Comparison with the State-of-the-Arts (SOTAs). Table 2[3], Table 3, and Table 4 present the performance comparison between the SOTA methods and AACL, where we can find that AACL establishes new state-of-the-art results in most metrics on R2R, REVERIE and R2R-Last. These results show that AACL is useful not only when the instructions are fine-grained but also when the instruction information is limited, demonstrating that the proposed actional atomic concepts can effectively enhance the observa-

[2] https://www.mindspore.cn/

[3] The original value 2.29 of NE under Val Unseen in HAMT is a typo, which is actually 3.62 confirmed by the author.

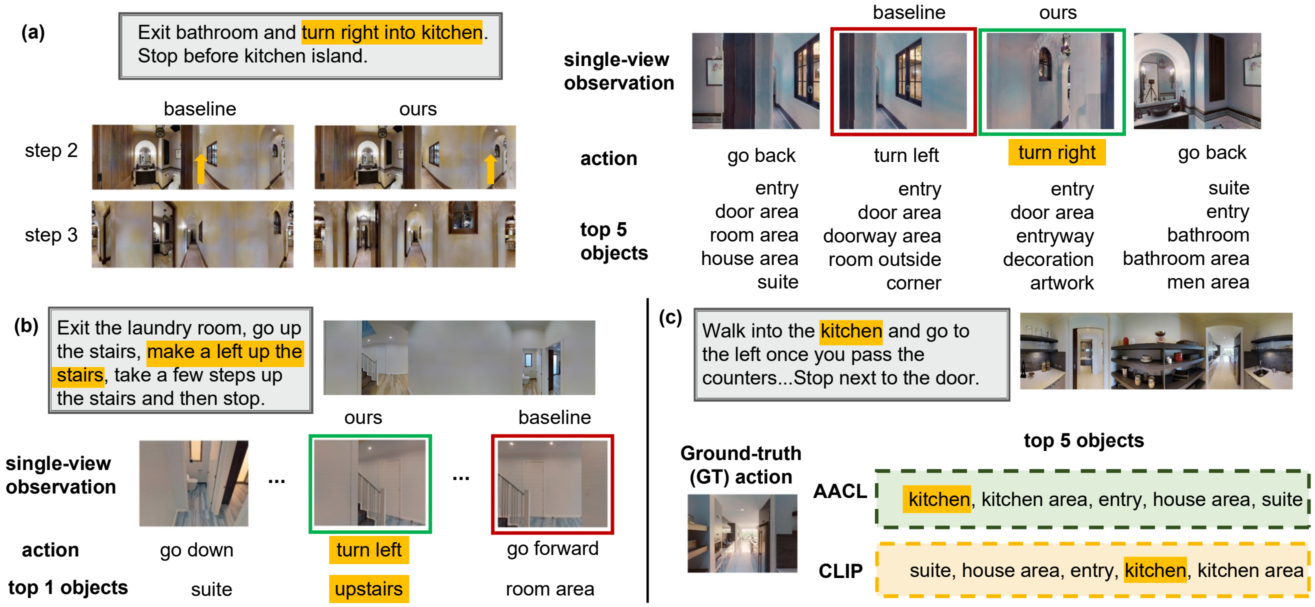


Figure 3: Visualization examples of action selection ((a) and (b)) and object concept mapping ((c)). In (a) and (b), the baseline is HAMT (Chen et al. 2021). The green boxes denote the correct actions and the red boxes denote the wrong ones.

Method	Val Unseen		
	NE↓	SR↑	SPL↑
separate embedding	3.66	66.67	61.19
w/o contrast	3.42	67.94	61.48
w/o refine	3.45	67.82	62.33
full model	**3.41**	**68.54**	**62.96**

Table 5: Ablation Study on R2R. The baseline agent we choose is HAMT (Chen et al. 2021).

tion features, simplify their alignment to the linguistic instruction features, and therefore improve the agent performance. Moreover, we can find that AACL consistently outperforms the two strong baselines on these three benchmarks especially under Unseen scenarios, showing that AACL can be used as a general tool for the multi-modal alignment.

Ablation Study. Table 5 gives the ablation study of AACL. "separate embedding" means using the separate embedding scheme (Eq. 12) only for the visual feature and the directional feature. By comparing the results between "separate embedding" and "w/o contrast", we can find that the direct introduction of actional atomic concepts under the separate embedding strategy can already improve the navigation performance (1.27% increase on SR), showing their effectiveness for enhancing the observation features. By comparing the results between "w/o contrast" and "w/o refine", we can observe that the proposed observation contrast strategy can effectively regularize the observation representation and improve the performance (0.85% increase on SPL). The comparison between "w/o refine" and "full model" further shows the effectiveness of the concept refining adapter, demonstrating that the instruction-oriented object concept extraction can facilitate better cross-modal alignment.

Qualitative Results

Figure 3 visualizes some results of action decision and object concept mapping. We can find that by introducing the actional atomic concepts, the agent is able to perform better cross-modal alignment for improving action decisions. In Figure 3(a), although the candidate observations do not contain the visual appearance of "kitchen", with the help of the actional atomic concepts, AACL successfully chooses the right action whose paired action concept matches the one mentioned in the instruction ("turn right"), while the baseline selects the wrong one. In Figure 3(b), with the actional atomic concept, AACL successfully chooses the correct action asked in the instruction. In Figure 3(c), we can observe that the probability of "kitchen" of AACL is higher than that of CLIP for the GT action (top-1 vs. top-4), showing that the concept refining adapter enables more instruction-oriented object concept extraction, which is useful for selecting correct actions.

Conclusion

In this work, we propose Actional Atomic-Concept Learning, which helps VLN agents demystify the alignment in VLN tasks through actional atomic concepts formed by language. During navigation, each visual observation is mapped to the specific actional atomic concept through the VLN environment and CLIP. A concept refining adapter is constructed to enable the instruction-oriented concept extraction. An observation co-embedding module is introduced to use concept features to regularize observation features. Experiments on public VLN benchmarks show that AACL achieves new SOTA results. Benefiting from these human-understandable actional atomic concepts, AACL shows excellent interpretability in making action decision.

Acknowledgements

This work was supported in part by National Key R&D Program of China under Grant No. 2020AAA0109700, National Natural Science Foundation of China (NSFC) under Grant No.61976233, Guangdong Outstanding Youth Fund (Grant No. 2021B1515020061), Guangdong Natural Science Foundation under Grant 2017A030312006, Shenzhen Fundamental Research Program (Project No. RCYX20200714114642083, No. JCYJ20190807154211365), and the Fundamental Research Funds for the Central Universities, Sun Yat-sen University under Grant No. 22lgqb38. We gratefully acknowledge the support of MindSpore, CANN (Compute Architecture for Neural Networks), and Ascend AI Processor used for this research.

References

Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sunderhauf, N.; Reid, I.; Gould, S.; and van den Hengel, A. 2018. Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments. In *CVPR*.

Ba, J.; Kiros, J. R.; and Hinton, G. E. 2016. Layer Normalization. *ArXiv*, abs/1607.06450.

Chen, H.; Suhr, A.; Misra, D. K.; Snavely, N.; and Artzi, Y. 2019. TOUCHDOWN: Natural Language Navigation and Spatial Reasoning in Visual Street Environments. In *CVPR*.

Chen, S.; Guhur, P.-L.; Schmid, C.; and Laptev, I. 2021. History Aware Multimodal Transformer for Vision-and-Language Navigation. In *NeurIPS*.

Chen, S.; Guhur, P.-L.; Tapaswi, M.; Schmid, C.; and Laptev, I. 2022. Think Global, Act Local: Dual-scale Graph Transformer for Vision-and-Language Navigation. In *CVPR*.

Chen, Y.-C.; Li, L.; Yu, L.; Kholy, A. E.; Ahmed, F.; Gan, Z.; Cheng, Y.; and Liu, J. 2020. UNITER: UNiversal Image-TExt Representation Learning. In *ECCV*.

Dai, W.; Hou, L.; Shang, L.; Jiang, X.; Liu, Q.; and Fung, P. 2022. Enabling Multimodal Generation on CLIP via Vision-Language Knowledge Distillation. In *ACL*.

Desai, K.; and Johnson, J. 2021. VirTex: Learning Visual Representations from Textual Annotations. In *CVPR*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv*, abs/1810.04805.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Fried, D.; Hu, R.; Cirik, V.; Rohrbach, A.; Andreas, J.; Morency, L.-P.; Berg-Kirkpatrick, T.; Saenko, K.; Klein, D.; and Darrell, T. 2018. Speaker-Follower Models for Vision-and-Language Navigation. In *NeurIPS*.

Fu, T.-J.; Wang, X. E.; Peterson, M. F.; Grafton, S. T.; Eckstein, M. P.; and Wang, W. Y. 2020. Counterfactual Vision-and-Language Navigation via Adversarial Path Sampler. In *ECCV*.

Gao, P.; Geng, S.; Zhang, R.; Ma, T.; Fang, R.; Zhang, Y.; Li, H.; and Qiao, Y. J. 2021. CLIP-Adapter: Better Vision-Language Models with Feature Adapters. *ArXiv*, abs/2110.04544.

Hao, W.; Li, C.; Li, X.; Carin, L.; and Gao, J. 2020. Towards Learning a Generic Agent for Vision-and-Language Navigation via Pre-Training. In *CVPR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

Hong, Y.; Wu, Q.; Qi, Y.; Rodriguez-Opazo, C.; and Gould, S. 2021. VLN BERT: A Recurrent Vision-and-Language BERT for Navigation. In *CVPR*.

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.-T.; Parekh, Z.; Pham, H.; Le, Q. V.; Sung, Y.-H.; Li, Z.; and Duerig, T. 2021. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *ICML*.

Khandelwal, A.; Weihs, L.; Mottaghi, R.; and Kembhavi, A. 2022. Simple but Effective: CLIP Embeddings for Embodied AI. In *CVPR*.

Ku, A.; Anderson, P.; Patel, R.; Ie, E.; and Baldridge, J. 2020. Room-Across-Room: Multilingual Vision-and-Language Navigation with Dense Spatiotemporal Grounding. In *EMNLP*.

Li, G.; Duan, N.; Fang, Y.; Gong, M.; and Jiang, D. 2020a. Unicoder-VL: A Universal Encoder for Vision and Language by Cross-Modal Pre-Training. In *AAAI*.

Li, L. H.; Yatskar, M.; Yin, D.; Hsieh, C.-J.; and Chang, K.-W. 2019. VisualBERT: A Simple and Performant Baseline for Vision and Language. *ArXiv*, abs/1908.03557.

Li, X.; Yin, X.; Li, C.; Zhang, P.; Hu, X.; Zhang, L.; Wang, L.; Hu, H.; Dong, L.; Wei, F.; Choi, Y.; and Gao, J. 2020b. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *ECCV*.

Liang, X.; Zhu, F.; Li, L.; Xu, H.; and Liang, X. 2022. Visual-Language Navigation Pretraining via Prompt-based Environmental Self-exploration. In *ACL*.

Lin, B.; Zhu, Y.; Chen, Z.; Liang, X.; zhuo Liu, J.; and Liang, X. 2022. ADAPT: Vision-Language Navigation with Modality-Aligned Action Prompts. In *CVPR*.

Lin, X.; Li, G.; and Yu, Y. 2021. Scene-Intuitive Agent for Remote Embodied Visual Grounding. In *CVPR*.

Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*.

Ma, C.-Y.; jiasen lu; Wu, Z.; AlRegib, G.; Kira, Z.; richard socher; and Xiong, C. 2019. Self-Monitoring Navigation Agent via Auxiliary Progress Estimation. In *ICLR*.

Moudgil, A.; Majumdar, A.; Agrawal, H.; Lee, S.; and Batra, D. 2021. SOAT: A Scene- and Object-Aware Transformer for Vision-and-Language Navigation. In *NeurIPS*.

Nguyen, K.; and Daumé, H. 2019. Help, Anna! Visual Navigation with Natural Multimodal Assistance via Retrospective Curiosity-Encouraging Imitation Learning. In *EMNLP*.

Qi, Y.; Pan, Z.; Hong, Y.; Yang, M.-H.; van den Hengel, A.; and Wu, Q. 2021. The Road to Know-Where: An Object-and-Room Informed Sequential BERT for Indoor Vision-Language Navigation. In *ICCV*.

Qi, Y.; Pan, Z.; Zhang, S.; van den Hengel, A.; and Wu, Q. 2020. Object-and-Action Aware Model for Visual Language Navigation. In *ECCV*.

Qi, Y.; Wu, Q.; Anderson, P.; Wang, X.; Wang, W. Y.; Shen, C.; and van den Hengel, A. 2020. REVERIE: Remote Embodied Visual Referring Expression in Real Indoor Environments. In *CVPR*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*.

Rao, Y.; Zhao, W.; Chen, G.; Tang, Y.; Zhu, Z.; Huang, G.; Zhou, J.; and Lu, J. 2022. DenseCLIP: Language-Guided Dense Prediction with Context-Aware Prompting. In *CVPR*.

Sariyildiz, M. B.; Perez, J.; and Larlus, D. 2020. Learning Visual Representations with Caption Annotations. In *ECCV*.

Shen, S.; Li, L. H.; Tan, H.; Bansal, M.; Rohrbach, A.; Chang, K.-W.; Yao, Z.; and Keutzer, K. 2022. How Much Can CLIP Benefit Vision-and-Language Tasks? In *ICLR*.

Song, H.; Dong, L.; Zhang, W.; Liu, T.; and Wei, F. 2022. CLIP Models are Few-Shot Learners: Empirical Studies on VQA and Visual Entailment. In *ACL*.

Subramanian, S.; Merrill, W.; Darrell, T.; Gardner, M.; Singh, S.; and Rohrbach, A. 2022. ReCLIP: A Strong Zero-Shot Baseline for Referring Expression Comprehension. In *ACL*.

Tan, H.; Yu, L.; and Bansal, M. 2019. Learning to Navigate Unseen Environments: Back Translation with Environmental Dropout. In *NAACL-HLT*.

Wang, H.; Wu, Q.; and Shen, C. 2020. Soft Expert Reward Learning for Vision-and-Language Navigation. In *ECCV*.

Wang, X.; Huang, Q.; Celikyilmaz, A.; Gao, J.; Shen, D.; Wang, Y.-F.; Wang, W. Y.; and Zhang, L. 2019. Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation. In *CVPR*.

Zhu, F.; Zhu, Y.; Chang, X.; and Liang, X. 2020. Vision-Language Navigation With Self-Supervised Auxiliary Reasoning Tasks. In *CVPR*.