AVLEN: Audio-Visual-Language Embodied Navigation in 3D Environments

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent years have seen embodied visual navigation advance in two distinct directions: (i) in equipping the AI agent to follow natural language instructions, and (ii) in making the navigable world multimodal, e.g., audio-visual navigation. However, the real world is not only multimodal, but also often complex, and thus in spite of these advances, agents still need to understand the uncertainty in their actions and seek instructions to navigate. To this end, we present AVLEN – an *interactive agent* for Audio-Visual-Language Embodied Navigation. Similar to audio-visual navigation tasks, the goal of our embodied agent is to localize an audio event via navigating the 3D visual world; however, the agent may also seek help from a human (oracle), where the assistance is provided in free-form natural language. To realize these abilities, AVLEN uses a multimodal hierarchical reinforcement learning backbone that learns: (a) high-level policies to choose either audio-cues for navigation or to query the oracle, and (b) lower-level policies to select navigation actions based on its audio-visual and language inputs. The policies are trained via rewarding for the success on the navigation task while minimizing the number of queries to the oracle. To empirically evaluate AVLEN, we present experiments on the SoundSpaces framework for semantic audio-visual navigation tasks. Our results show that equipping the agent to ask for help leads to a clear improvement in performance, especially in challenging cases, e.g., when the sound is unheard during training or in the presence of distractor sounds.

1 Introduction

Building embodied robotic agents that can harmoniously co-habit and assist humans has been one of the early dreams of AI. A recent incarnation of this dream has been in designing agents that are capable of autonomously navigating realistic virtual worlds for solving pre-defined tasks. For example, in vision-and-language navigation (VLN) tasks [2], the goal is for the AI agent to either navigate to a goal location following the instructions provided in natural language, or to explore the visual world seeking answers to a given natural language question [10, 35, 36]. Typical VLN agents are assumed deaf; i.e., they cannot hear any audio events in the scene – an unnatural restriction, especially when the agent is expected to operate in the real world. To address this shortcoming, SoundSpaces [6] reformulated the navigation task with the goal of localizing an audio source in the virtual scene; however without any language instructions for the agent to follow.

Real-world navigation is not only audio-visual, but also is often complex and stochastic, so the agent must inevitably seek a synergy between the audio, visual, and language modalities for successful navigation. Consider, for example, a robotic agent that needs to find where the *"thud of a falling person"* or the *"intermittent dripping sound of water"* is heard from. On the one hand, such a sounds may not last long and may not be continuously audible, and thus the agent must use semantic

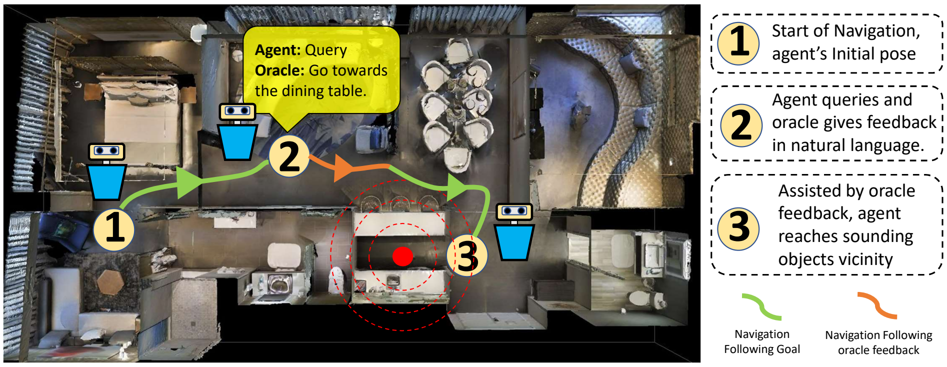

Agent: Query
Oracle: Go towards the dining table.
1
2
3
1 Start of Navigation, agent's Initial pose
2 Agent queries and oracle gives feedback in natural language.
3 Assisted by oracle feedback, agent reaches sounding objects vicinity
Navigation Following Goal
Navigation Following oracle feedback

Figure 1: An illustration of our proposed AVLEN framework. The embodied agent starts navigating from location denoted ① guided by the audio-visual event at ③. At location ②, the learned policy for the agent decides to seek help from an oracle (e.g., because the audio stopped). The oracle provides a short natural language instruction for the agent to follow. The agent translates this instruction to produce a series of navigable steps to move towards the goal ③.

knowledge of the audio-visual modality [5] to reach the goal. On the other hand, such events need to be catered to timely and the agent should minimize the number of navigation mistakes it makes – a situation that can be efficiently dealt with if the agent can seek human help when it is uncertain of its navigation actions. Motivated by this insight, we present AVLEN – a first of its kind embodied navigation agent for localizing an audio source in a realistic visual world. Our agent not only learns to use the audio-visual cues to navigate to the audio source, but also learns to implicitly model its uncertainty in deciding the navigation steps and seeks help from an oracle for navigation instructions, where the instructions are provided in short natural language sentences. Figure 1 illustrates our task.

To implement AVLEN, we build on the realistic virtual navigation engine provided by the Matterport 3D simulator [2] and enriched with audio events via the SoundSpaces framework [6]. A key challenge in our setup is for the agent to decide when to query the oracle, and when to follow the audio-visual cues to reach the audio goal. Note that asking for help too many times may hurt agent autonomy (and is perhaps less preferred if the oracle is a human), while asking too few questions may make the agent explore the scene endlessly without reaching the goal. Further, note that we assume the navigation instruction provided to the agent is in natural language, and thus is often abstract and short (see Figure 1 above), making it difficult to be correctly translated to agent actions (as seen in VLN tasks [2]). Thus, the agent needs to learn the stochasticity involved in the guidance provided to it, as well as the uncertainty involved in the audio-visual cues, before selecting which modality to choose. To address these challenges, we propose a novel multimodal hierarchical options based deep reinforcement learning framework, consisting of learning a high-level policy to select which modality to use for navigation, among (i) the audio-visual cues, or (ii) natural language cues, and two lower-level policies that learn (i) to select navigation actions using the audio-visual features, or (ii) learns to transform natural language instructions to navigable actions conditioned on the audio-visual context. All the policies are end-to-end trainable and is trained offline. During inference, the agent uses its current state, the audio-visual cues, and the learned policies to decide if it needs oracle help or can continue navigation using the learned audio goal navigation policies.

Closely related to our motivations, a few recent works propose tasks involving interactions with an oracle for navigation, such as [9, 26, 27]. For example, previous works [9, 38] rely on model uncertainty to decide when to query the oracle, where the uncertainty is either quantified in terms of the gap between the action prediction probabilities [9] to be less than a heuristically chosen threshold, or use manually-derived conditions to decide when an agent is lost in its navigation path [27]. In [26], the future actions of the policy of interest are required to be fully observed to identify when the agent is making mistakes and to incorporate this information for identifying when to query. Instead of resorting to heuristics, we propose a data-driven way to learn policies to decide when to query the oracle, these policies thus automatically learning the navigation uncertainty of the various modalities.

To empirically demonstrate the efficacy of our approach, we present extensive experiments on the language-augmented semantic audio-visual navigation (SAVi) task within the SoundSpaces framework for three very challenging scenarios when: (i) the sound source is sporadic, however familiar to the agent, (ii) sporadic but unheard of during training, and (iii) unheard and ambiguous due to the presence of simultaneous distractor sounds. Our results show clear benefits when the agent knows when to query and how to use the received instruction for navigation, as substantiated by improvements in success rate by nearly 3% when using the language instructions directly, or by more than 10% when using the ground truth navigation actions after the query, even when the agent triggering help only 3 times in a long navigation episode.

Before proceeding to detail our framework, we summarize below the main contributions of this paper.

- We are the first to unify and generalize audio-visual navigation with natural language instructions towards building a complete audio-visual-language embodied AI navigation interactive agent.
- We introduce a novel multimodal hierarchical reinforcement learning framework that jointly learns policies for the agent to decide: (i) when to query the oracle, (ii) how to navigate using audio-goal, and (iii) how to use the provided natural language instructions.
- Our approach shows state-of-the-art performances on the semantic audio-visual navigation dataset [5] with 85 large-scale real-world environments with a variety of semantic objects and their sounds, and under a variety of challenging acoustic settings.

2 Related Works

Instruction Following Navigation. There are recent works that attempt to solve the problem of navigation following instructions [2, 17, 25, 16, 21]. The instruction can be of many forms; e.g., structured commands [27], natural language sentences [2], goal images [22], or a combination of different modalities [26]). The task in vision and language navigation (VLN) for example is to execute free-form natural language instructions to reach a target location. To embody this task, several simulators have been used [29, 6, 2] to render real or photo-realistic images and perform agent navigation through a discrete graph [2, 8] or continuous environment [19]. One important aspect of vision and language navigation is to learn the correspondence between visual and textual information. To achieve this, [34] uses cross modal attention to focus on the relevant part of both the modalities. In [23] and [24], an additional module is used to estimate the progress, which is then used as a regularizer. In [13] and [31], augmented instruction-trajectory pairs is used to improve the VLN performance. In [37], long instructions are learned to be decomposed into shorter ones and executing them sequentially (via e.g., navigation). Recently, there are works using Transformer-based architectures for the VLN application [25, 16]. In [16], BERT [11] architecture is used in a recurrent manner maintaining cross-modal state information. These works only consider the language based navigation task. However, AVLEN solves vision-language navigation as a sub-task of original semantic audio-visual navigation task.

Interactive Navigation. Recently, there have been works where an agent is allowed to interact with an oracle or a different agent, receiving feedback, and utilizing this information for navigation [27, 26, 9, 32]. The oracle instructions from existing approaches are limited to ground truth actions [9] and direct mapping of specific number of actions to consecutive phrases [27]. Though [26] uses a fixed set of natural language instructions as the oracle feedback, it is coupled with the target image that the agent will face after completion of the sub-goal task. In Nguyen et al. [26], the agent needs to reach a specific location to query, which may be infeasible practically or sub-optimal if these locations are not chosen properly. Our approach differs fundamentally from these previous works in that we consider free-form natural language instructions and the agent can query the oracle from any navigable point in the environment, making our setup very natural and flexible.

3 Proposed Method

In this section, we will first formally define our task and our objective. This will be followed by details of our multimodal hierarchical reinforcement learning and our training setups.

Problem Setup. Consider an agent in a previously unseen 3D world navigable along a densely-sampled finite grid. At each vertex of this grid, the agent could potentially take one of a subset of actions from an action set $A = \{\text{stop}, \text{move\_forward}, \text{turn\_right}, \text{turn\_left}\}$. Further, the agent is assumed to be equipped with sensors for audio-visual perception via a microphone, and ego-centric RGB and depth cameras. The task of the agent in AVLEN is to navigate the grid from its starting location to find the location of an object that produces a sound (*AudioGoal*), where the sound is assumed to be produced by a static object and is semantically unique, however can be unfamiliar, sporadic, or ambiguous (due to distractors). We assume the agent calls the stop action only at *AudioGoal* that terminates the episode. In contrast to the task in SoundSpaces, an AVLEN agent is also equipped with a language interface to invoke a *query* to an *oracle* under a budget (e.g., a limit on the maximum number of such queries). The oracle responds to the query of the agent via providing a natural language short navigation instruction; this instruction describing (in natural language) an initial *segment* along the shortest path trajectory from the current location of the agent to the goal. For example, for a navigation trajectory given by the actions $\langle \mathrm{move\_forward}, \mathrm{turn\_right}, \mathrm{turn\_right}, \mathrm{move\_forward}, \mathrm{turn\_left}\rangle$, the corresponding language instruction provided by the oracle to the agent could be *"go around the sofa and turn to the door"*. As is clear, to use this instruction to produce navigable actions, the agent must learn to associate the language constructs with objects in the scene and their spatial relations, as well as their connection with the nodes in the navigation grid. Further, given the limited budget to make queries, the agent must learn to balance between when to invoke the query and when to navigate using its audio-visual cues. In the following, we present a multimodal hierarchical options approach to solve these challenges in a deep reinforcement learning framework.

Problem Formulation. We formulate the AVLEN task as a partially-observable Markov decision process (POMDP) characterized by the tuple $(\mathcal{S}, \mathcal{A}, T, R, \mathcal{O}, P, \mathcal{V}, \gamma)$, where $\mathcal{S}$ represents the set of agent states, $\mathcal{A} = A \cup \{\mathrm{query}\}$ with the navigation actions A defined above combined with an action to query the oracle, $T(s'|s,a)$ is the transition probability for mapping a state-action pair (s,a) to a state s', $R(s,a)$ is the immediate reward for the state-action pair, $\mathcal{O}$ represents a set of environment observations o, $P(o|s',a)$ captures the probability of observing $o \in \mathcal{O}$ in a new state s' after taking action a, and $\gamma \in [0,1]$ is the reward discount factor for long-horizon trajectories. Our POMDP also incorporates a language vocabulary $\mathcal{V}$ consisting of a dictionary of words that the oracle uses to produce the natural language instruction. As our environment is only partially-observable, the agent may not have information regarding its exact state, instead maintains a belief distribution b over $\mathcal{S}$ as an estimate of its current state. Using this belief distribution, the expected reward for taking an action a at belief state b can be written as $R'(b,a) = \sum_{s\in\mathcal{S}} b(s)R(s,a)$. With this notation, the objective of the agent in this work is to learn a policy $\pi : \mathbb{R}^{|\mathcal{S}|} \times \mathcal{A} \to [0,1]$ that maximizes the expected return defined by the value function V^π, while minimizing the number of queries made to the oracle; i.e.,

$$\arg\max_{\pi} V^\pi(b_0) \text{ where } V^\pi(b) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i \left(R'(b_{t+i}, a_{t+i}) - \zeta(t+i)\mathbb{I}(a_{t+i} = \mathrm{query})\right) | b_t = b, \pi\right], \tag{1}$$

where $\mathbb{I}$ denotes the indicator function, and the updated belief $b_{t+1} = \mathrm{update}(o_{t+1}, b_t, a_t)$ defined for state s' as $b_{t+1}(s') = \eta P(o_{t+1}|s', a_t)\sum_{s\in\mathcal{S}} b_t(s)T(s'|s,a_t)$ for a normalization factor $\eta > 0$. The function $\zeta : \mathbb{R}_+ \to \mathbb{R}$ produces a score balancing between the frequency of queries and the expected return from navigation.

At any time step t, the agent (in belief state b_t) receives an observation $o_{t+1} \in \mathcal{O}$ from the environment and selects an action a_t according to a learned policy π; this action transitioning the agent to the new belief state b_{t+1} as per the transition function $T'(b_{t+1}|a_t, b_t) = \sum_{o\in\mathcal{O}} \mathbb{I}(b_{t+1} = \mathrm{update}(o, b_t, a_t))P(o|s_t, a_t)$ while receiving an immediate reward $R'(b_t, a_t)$. As the navigation state space $\mathcal{S}$ of our agent is enormous, keeping a belief distribution on all states might be computationally infeasible. Instead, similar to [5], we keep a history of past K observations in a memory module M, where an observation o_t at time step t is encoded via the tuple $e^o_t = (F^V_t, F^B_t, F^A_{t-1}, p_t)$ comprising neural embeddings of egocentric visual observation (RGB and depth) represented by F^V_t, the binaural audio waveform of the *AudioGoal* heard by the agent represented as a two channel spectrogram F^B_t, and the previous action taken F^A_{t-1}, alongside the pose of the agent p_t with respect to its starting pose (consisting of the 3 spatial coordinates and the yaw angle). The memory M is initialized to an empty set at the beginning of an episode, and at a time step t, is updated as $M = \{e^o_i : i = \max\{0, t-K\}, \dots, t\}$. Apart from these embeddings, AVLEN also incorporates a goal estimation network f_g characterized by a convolutional neural network that produces a step-wise estimate $\hat{g}_t = f_g(F^B_t)$ of the sounding *AudioGoal*; $\hat{g}_t$ consisting of: (i) the (x,y) goal location estimate L_t from the current pose of the agent, and (ii) the goal category estimate $c_t \in \mathbb{R}^C$ for C semantic sounding object classes. The agent updates the current goal estimate combining the previous estimates as $g_t = \lambda\hat{g}_t + (1-\lambda)f_p(g_{t-1}, \Delta p_t)$ where f_p is a linear transformation of g_{t-1} using the pose change Δp_t. We use $\lambda = 0.5$, unless the sound is inaudible in which case it is set to zero. We will use $g \in G \subset \mathbb{R}^{C+2}$ to denote the space of goal estimates.

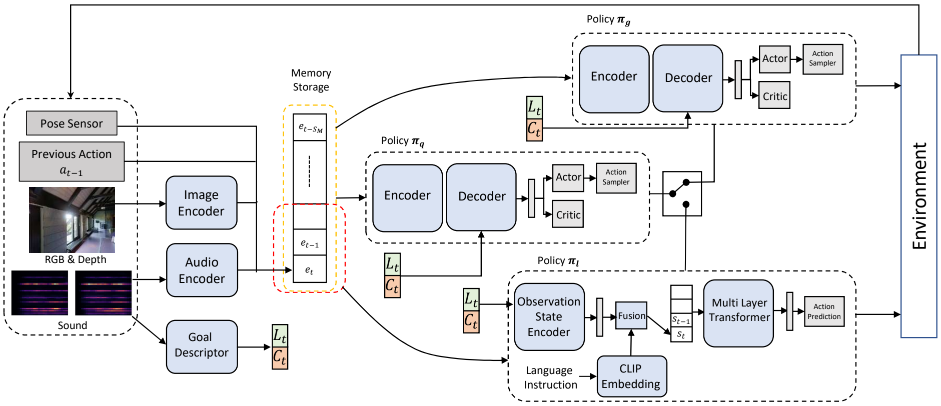


Figure 2: Architecture of our AVLEN pipeline. We show the two-level hierarchical RL policies that the model learns (offline), as well as the various input modalities and the control flow.

Multimodal Hierarchical Deep Reinforcement Learning. As is clear, the diverse input modalities used in AVLEN have distinctly varied levels of semantic granularity, and thus a single monolithic end-to-end RL policy for navigation might be sub-optimal. For example, the natural language navigation instructions received from the oracle might comprise rich details for navigating the scene that the agent need not have to resort to any of the audio-visual inputs for say $\nu > 1$ steps. However, there is a budget on such queries and thus the agent must know when the query needs to be initiated (e.g., when the agent repeats sub-trajectories). Further, from a practical sense, each modality might involve different neural network architectures for processing, can have their own strategies for (pre-)training, involve distinct inductive baises, or incorporate heterogeneous navigation uncertainties.

All the above challenges naturally suggest to abstract the policy learning in the context of *hierarchical options* semi-Markov framework [3, 20, 30] consisting of low-level options corresponding to the navigation using either the *AudioGoal* or the language model, and a high-level policy to select among the options. More formally, an *option* is a triplet consisting of a respective policy ξ, a termination condition, and a set of belief states in which the option is valid. In our context, we assume a *multi-time* option policy for language-based navigation spanning ν navigation steps[1] and a *primitive* policy [30] for *AudioGoal*. We also assume these options may be invoked independent of the agent state. Suppose $\pi_q : \mathbb{R}^{|\mathcal{S}|\times|M|} \times G \times \{\text{query}\} \to [0,1]$ represent the high-level policy deciding whether to query the oracle or not, using the current belief, the history M and the goal estimate g. Further, let the two lower-level policies be: (i) $\pi_g : \mathbb{R}^{|\mathcal{S}|\times|M|} \times G \times A \to [0,1]$, that is tasked with choosing the navigation actions based on the audio-visual features, and (ii) $\pi_\ell : \mathbb{R}^{|\mathcal{S}|\times\nu} \times \mathcal{V}^N \times G \times A \to [0,1]$, that navigates based on the received natural language instruction formed using N words from the vocabulary $\mathcal{V}$, assuming ν steps are taken after each such query. Let R'_g and R'_ℓ denote the rewards (as defined in (1)) corresponding to the π_g and π_ℓ options, respectively, where we have the multi-time discounted cumulative reward (with penalty ζ) for $R'_\ell(b_t, a_t) = \mathbb{E}\left(\sum_{i=t}^{t+\nu-1} \gamma^{i-t} R'(b_i, a_i) | \pi_q = \pi_\ell, a_i \in A\right) - \zeta(t)$, while R'_g is, being a primitive option, as in (1) except that the actions are constrained to A. Then, we have the Bellman equation for using the options given by:

$$V^\pi(b) = \pi_q(\xi_g|b)\left[R'_g + \sum_{o'\in\mathcal{O}} P'(o'|b,\xi_g)V^\pi(b')\right] + \pi_q(\xi_\ell|b)\left[R'_\ell + \sum_{o'\in\mathcal{O}} P(o'|b,\xi_\ell)V^\pi(b')\right], \quad (2)$$

where ξ_g and ξ_ℓ are shorthands for $\xi = \pi_g$ and $\xi = \pi_\ell$, respectively, and $\pi = \{\pi_q, \pi_g, \pi_\ell\}$. Further, P' is the multi-time transition function given by: $P'(o'|b,\xi) = \sum_{j=1}^{\infty}\sum_{s'}\sum_{s}\gamma^j P(s', o', j|s, \xi)b(s)$, where with a slight abuse of notation, we assume $P(s', o', j)$ is the probability to observe o' in j steps using option ξ [30]. Our objective is to find the policies π that maximizes the value function in(2) for $V^\pi(b_0)$. Figure 2 shows a block diagram of the interplay between the various policies and architectural components. Note that by using such a two-stage policy, we assume that the top-level policy π_q implicitly learns the uncertainty in the audio-visual and language inputs as well as the

[1] Otherwise terminated if the stop action is called before the option policy is completed.

predictive uncertainty in the respective low-level options π_g and π_ℓ for reaching the goal state. In the next subsections, we detail the neural architectures for each of these options policies.

Navigation Using Audio Goal Policy, π_g. Our policy network for π_g follows an architecture similar to [5], consisting of a Transformer encoder-decoder model [33]. The encoder sub-module takes in the embedded features e^o from the current observation as well as such features from history stored in the memory M, while the decoder module takes in the output of the encoder concatenated with the goal descriptor g to produce a fixed dimensional feature vector, characterizing the current belief state b. An actor-critic network (consisting of a linear layer) then predicts an action distribution $\pi_g(b,.)$ and the value of this state. The agent then takes an action $a \sim \pi_g(b,.)$, takes a step, and receives a new observation. The goal descriptor network f_g outputs object category c and relative goal location estimation L. Following SAVi [5], we apply off-policy category level predictions and on-policy location estimator.

Navigation Using Language Policy, π_ℓ. When an agent queries, it receives natural language instruction $\mathrm{instr} \in \mathcal{V}^N$ from the oracle. Using instr and the current observation $e^o_t = (F^V_t, F^B_t, F^A_{t-1}, p_t)$, our language-based navigation policy performs a sequence of actions $\langle a_t, a_{t+1}, \dots, a_{t+\nu} \rangle$ as per π_ℓ option, where each $a_i \in A$. Specifically, for any step $\tau \in \langle t, \dots, t+\nu-1 \rangle$, π_ℓ first encodes $\{e^o_\tau, g_\tau\}$ using a Transformer encoder-decoder network T_1[2], the output of this Transformer is then concatenated with CLIP [28] embeddings of the instruction, and fused using a fully-connected layer FC_1. The output of this layer is then concatenated with previous belief embeddings using a second multi-layer Transformer encoder-decoder T_2 to produce the new belief state b_τ, i.e.,

$$b_\tau = T_2\left(\mathrm{FC}_1\left(T_1(e^o_\tau, g_\tau), \mathrm{CLIP}(\mathrm{instr})\right), \left\{b'_\tau : t < \tau' < \tau\right\}\right) \text{ and } \pi_\ell(b_\tau,.) = \mathrm{softmax}(\mathrm{FC}_2(b_\tau)). \tag{3}$$

Learning When-to-Query Policy, π_q. As alluded to above, the π_q policy decides when to query, i.e., when to use π_ℓ. Instead of directly utilizing model uncertainty [9], we use the reinforcement learning framework to be train this policy in an end-to-end manner, guided by the rewards ζ.

Reward Design. For the π_g policy, we assign a reward of +1 to reduce the geometric distance towards the goal, a +10 reward to complete an episode successfully, i.e., calling the stop action near the *AudioGoal*, and a penalty of -0.01 per time step to encourage efficiency. As for the π_ℓ policy, we set a negative reward each time the agent queries from oracle, denoted ζ_q, as well as when the query is made within τ steps from previous query, denoted ζ_f. If the (softly)-allowed number of queries is K, then our combined negative reward function is given by $\zeta_q + \zeta_f$, where

$$\zeta_q(k) = \begin{cases} \frac{k \times (r_{neg} + \exp(-\nu))}{\nu} & k < K \\ r_{neg} + \exp(-k) & k \geq K, \end{cases} \quad \text{and} \quad \zeta_f(j) = \begin{cases} \frac{r_f}{j} & 0 < j < \tau \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where r_{neg} is set to -1.2, and r_f is set to -0.5. As a result, the agent learns when to interact with the oracle directly based on its current observation and history information. In the RL framework, the actor-critic model also predicts the value function of each state. Policy training is done using decentralized distributed proximal policy optimization (DD-PPO).

Policy Training. Learning π_g uses a two-stage training; in the first stage, the memory M is not used, while in the second stage, observation encoders are frozen, and the policy network is trained using both the current observation and the history in M. The training loss consists of (i) the value-function loss, (ii) policy network loss to estimate the actions correctly, and (iii) an entropy loss to encourage exploration. Our language-based navigation policy π_ℓ follows a two stage training as well. The first stage consists of an off-policy training. We re-purpose the fine-grained instructions provided by [15] to learn the language-based navigation policy π_ℓ. The second stage consists of on-policy training. During roll outs in our hierarchical framework, as the agent interacts with the oracle and receives language instructions, we use these instructions with the shortest path trajectory towards the goal to finetune π_ℓ. In both cases, it is trained with an imitation learning objective. Specifically, we allow the agent to navigate on the ground-truth trajectory by following teacher actions and calculate a cross-entropy loss for each action in each step by; given by $-\sum_t a_{t^*} \log(\pi_\ell(b_t, a_t)$ where a^* is the ground truth action and $\pi_\ell(b_t, a_t)$ is the action probability predicted by π_ℓ.

Generating Oracle Navigation Instructions. The publicly available datasets for vision and language tasks contain a fixed number of route and instruction pairs at handpicked locations in the navigation

[2] This is different Transformer from the one used for π_g, however taking g_τ as input to the decoder.

Table 1: Comparison of performances against state of the art in heard and unheard sound settings.

		Heard Sound					**Unheard Sound**				
	Instruction	Success ↑	SPL ↑	SNA ↑	DTG ↓	SWS ↑	Success ↑	SPL ↑	SNA ↑	DTG ↓	SWS ↑
Random Nav.		1.4	3.5	1.2	17.0	1.4	1.4	3.5	1.2	17.0	1.4
ObjectGoal RL		1.5	0.8	0.6	16.7	1.1	1.5	0.8	0.6	16.7	1.1
Gan et al. [14]		29.3	23.7	**23.0**	11.3	14.4	15.9	12.3	11.6	12.7	8.0
Chen et al. [6]		21.6	15.1	12.1	11.2	10.7	18.0	13.4	12.9	12.9	6.9
AV-WaN [7]		20.9	16.8	16.2	10.3	8.3	17.2	13.2	12.7	11.0	6.9
SMT[12]+Audio		22.0	16.8	16.0	12.4	8.7	16.7	11.9	10.0	12.1	8.5
SAVi [5]		33.9	24.0	18.3	8.8	21.5	24.8	17.2	13.2	9.9	14.7
AVLEN (only π_g)		34.5	24.1	18.1	8.8	22.2	22.6	15.8	12.6	9.5	13.1
AVLEN (ours)	✓	**36.1**	**24.6**	19.7	**8.5**	**23.1**	**26.2**	**17.6**	**14.2**	**9.2**	**15.8**
AVLEN (GT)	✓	**48.2**	**34.3**	**26.7**	**7.5**	**36.0**	**36.7**	**24.1**	**18.7**	**8.3**	**26.6**

Table 2: Comparisons in heard and unheard sound settings against varied query-triggering methods.

		Heard Sound					**Unheard Sound**				
	Instruction	Success ↑	SPL ↑	SNA ↑	DTG ↓	SWS ↑	Success ↑	SPL ↑	SNA ↑	DTG ↓	SWS ↑
Random	✓	32.5	21.1	16.1	8.93	21.8	23.5	14.8	11.5	9.9	14.3
Uniform	✓	33.2	22.4	17.8	9.1	22.0	22.1	14.6	11.5	9.8	13.3
Model Uncertainty	✓	34.2	24.0	19.5	8.7	20.5	24.9	16.1	13.5	9.3	15.2
AVLEN (ours)	✓	**36.1**	**24.6**	**19.7**	**8.5**	**23.1**	**26.2**	**17.6**	**14.2**	**9.2**	**15.8**

grid. However, in our setup, a navigating agent can query an oracle at any point in the grid.To this end, we assume the oracle knows the shortest path trajectory $\mathrm{s\_path}$ from the current agent location to the *AudioGoal*, and from which the oracle select a segment consisting of n observation-action pairs (we use $n = 4$ in our experiments), i.e., $\mathrm{s\_path} = \big\langle (o_0, a_0), (o_1, a_1), \dots, (o_{n-1}, a_{n-1}) \big\rangle$. With this assumption, we propose to mimic the oracle by a *speaker* model [13], which can generate a distribution of words $P^S(w|\,\mathrm{s\_path})$[3]. The observation and action pairs are sequentially encoded using an LSTM encoder, $\big\langle F_0^S, F_1^S, \dots, F_n^S \big\rangle = \mathrm{SpeakerEncoder}(\mathrm{s\_path})$ and decoded by another LSTM predicting the next word in the instruction by: $w_t = \mathrm{SpeakerDecoder}(w_{t-1}, \big\langle F_0^S, F_1^S, \dots, F_n^S \big\rangle)$. The instruction generation model is trained using the available (instruction, trajectory) pairs from the VLN dataset [2]. We use cross entropy loss and teacher forcing during training.

4 Experiments and Results

Dataset. We use the SoundSpaces platform [6] for simulating the world in which our AVLEN agent conducts the navigation tasks. Powered by Matterport3D scans [4], SoundSpaces facilitates a realistic simulation of a potentially-complex 3D space navigable by the agent along a densely sampled grid with 1m sides. The platform also provides access to panoramic ego-centric views of the scene in front of the agent both as RGB and as depth images, while also allowing the agent to hear realistic binaural audio of acoustic events in the 3D space. To benchmark our experiments, we use the semantic audio-visual navigation dataset from Chen et al. [5] built over SoundSpaces. This dataset consists of sounds from 21 semantic categories of objects that are visually present in the Matterport3D scans. The object-specific sounds are generated at the location of the Matterport3D objects. In each navigation episode, the duration of the sounds are variable and is normal-distributed with a mean 15s and deviation 9s, clipped for a minimum 5s and maximum 500s [5]. There are 0.5M/500/1000 episodes available in this dataset for train/val/test splits respectively from 85 Matterport3D scans.

Evaluation Metrics. Similar to [5], we use the following standard navigation metrics for evaluating our performances on this dataset: i) *success rate* for reaching the *AudioGoal*, ii) *success weighted by inverse path length* (SPL) [1], iii) *success weighted by inverse number of actions* (SNA) [7], iv) *average distance to goal* (DTG), and v) *success when silent* (SWS). SWS refers to the fraction of successful episodes when the agent reaches the goal after the end of the acoustic event.

Implementation Details. Similar to prior works, we use RGB and depth images, center-cropped to 64×64. The agent receives binaural audio clip as 65×26 spectrograms. The memory size for π_g and π_q is 150 and for π_ℓ is 3. All the experiments consider maximum $K = 3$ allowed queries (unless otherwise specified). For each query, the agent will take $\nu = 3$ navigation steps in the environment using the natural language instruction. We use a vocabulary with 1621 words. Training uses ADAM [18] with learning rate 2.5×10^{-4}. Refer to the Appendix for more details.

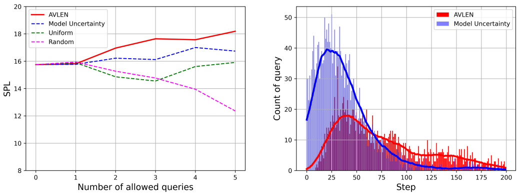
Experimental Results and Analysis. The main objective of our AVLEN agent in the semantic audio-visual navigation task is to navigate towards a sounding object in an unmapped 3D environment when

[3] $\mathrm{s\_path}$ is approximated in the discrete Room-to-Room [2] environment and then used to generate instruction.

Figure 3: (a) Comparison of AVLEN performances against baselines and when-to-query approaches in the *presence of distractor sound*, (b) Performance (SPL) comparison against varying the number of allowed queries, and (c) Distribution of queries triggered against the time steps in episodes.

	Instruction	Success ↑	SPL ↑	SNA ↑	DTG ↓	SWS ↑
Chen et al. [6]		4.0	2.4	2.0	14.7	2.3
AV-WaN [7]		3.0	2.0	1.8	14.0	1.6
SMT[12]+Audio		4.2	2.9	2.1	14.9	2.8
SAVi [5]		11.8	7.4	5.0	13.1	8.4
Random	✓	11.6	6.6	4.8	12.9	7.8
Uniform	✓	11.6	6.8	5.1	13.3	7.7
Model Uncertainty	✓	12.4	6.7	5.0	12.8	8.4
AVLEN (only π_g)		11.8	7.4	5.0	13.1	8.4
AVLEN (ours)	✓	**14.0**	**8.4**	**5.9**	**12.8**	**11.1**
AVLEN (GT)	✓	**24.4**	**15.3**	**11.3**	**11.3**	**21.5**

(a) Comparisons under distractors



(b) Query sensitivity

(c) Query distribution

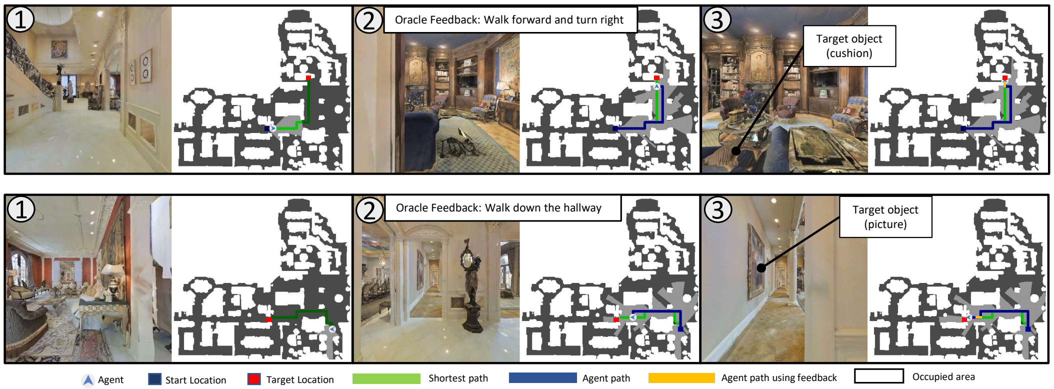


Figure 4: Two qualitative results from AVLEN's navigation trajectories. We show egocentric views and top down maps for three different viewpoints in agent's trajectory. The agent starts from ①, receives oracle help in ②, navigates to the goal in ③.

the sound is sporadic. Since we are the first to integrate oracle interaction through natural language instruction in this problem setting, we compare with against existing state-of-the-art semantic audio-visual navigation approaches, namely Gan et al. [14], Chen et al. [6], AV-WaN [7], SMT [12] + Audio, and SAVi [5]. Following the standard protocol of [5] and [6], we evaluate performance of the the same trained model on two different sound settings: i) *heard sound*, in which the sounds used during test are heard by the agent during training, and ii) *unheard sound*, in which the train and test sets use distinct sounds. In both experimental settings, the test environments are unseen.

Table 1 provides the results of our experiments using heard and unheard sounds. The table shows that "AVLEN (ours)" – which is our full model based on language feedback – shows a $+2.2\%$ and $+1.6\%$ absolute gain in success rate and success-when-silent (SWS) respectively, compared to the best performing baseline SAVi [5] for heard sound. Moreover, we obtain 1.4% and 1.1% absolute gain in success rate and SWS respectively for unheard sound compared to the next best method, SAVi. Our results clearly demonstrate that the agent is indeed able to use the short natural language instructions for improving the navigation. To further ascertain this observation, we also ran an experiment where the language instructions are cut off (i.e., π_q always selecting π_g policy); the results – marked as "only π_g" in Table 1 – attest to this observation showing a drop in the performances.

A natural question to ask in this setting is: *Why are the improvements not so dramatic, given the agent is receiving guidance from an oracle?* Generally, navigation based on language instructions is a challenging task in itself ([2, 25]) since language incorporates strong inductive biases and usually spans large vocabularies; as a result the action predictions can be extremely noisy, imprecise, and misguiding. However, the key for improved performance is to identify *when to query*. Our experiments show that AVLEN is able to identify when to query correctly (also see Table 2) and thus improve performance. To further substantiate this insight, we designed an experiment in which the agent is provided the ground truth navigation actions as feedback (instead of providing the corresponding language instruction) whenever a query is triggered. The results of this experiment – marked AVLEN (GT) in Table 1 – clearly show an improvement in success rate by nearly +15% for heard sounds and +12% for unheard sounds, suggesting future work to consider improving π_ℓ.

Navigation Under Distractor Sounds. We also consider the presence of distractor sound while navigating towards a particular unheard sound as provided in SAVi [5]. In this setting, the agent must know which sound its target is. Thus, a one hot encoding of the target is also provided as an input to the agent, if there are multiple sounds in the environment. Presence of distractor sounds makes the navigation task even more challenging, and intuitively, the agents interaction ability would come useful. This insight is clearly reflected in Figure 3a, where AVLEN shows 2.2% and 2.7% higher success rate and SWS respectively compared to baseline approaches. Figure 3a shows that AVLEN outperforms other query procedures as well and the performance difference is more significant compared to when there was no distractor sound.

Analyzing When-to-Query. To evaluate if AVLEN is able to query at the appropriate moments, we designed potential baselines that one could consider for deciding when-to-query and compare their performances in Table 2. Specifically, the considered baselines are: i) *Uniform*: queries after every 15 steps, ii) *Random*: queries randomly within first 50 steps of each episode, and iii) *Model Uncertainty (MU)* based on [9]: queries when the softmax action probabilities of the top two action predictions of π_g have an absolute difference are less than a predefined threshold (≤ 0.1). Table 2 shows our results. Unsurprisingly, we see that Random and Uniform perform poorly, however using MU happen to be a strong baseline. However, MU is a computationally expensive baseline as it requires π_g forward pass to decide when to query. Even so, we observe that compared to all the baselines, AVLEN shows better performance in all metrics. To further understand this, in Figure 3c we plot the distribution of the episode time step when the query is triggered for the unheard sound setting. As is clear, MU is more likely to query in the early stages of the episode, exhausting the budget, while AVLEN learns to distribute the queries throughout the steps, suggesting our hierarchical policy π_q is learning to be conservative, and judicial in its task. Interestingly we also find reasonable overlap between the peaks of the two curves, suggesting that π_q is considering the predictive uncertainty of π_g implicitly.

Sensitivity to Allowed Number of Queries, ν. To check the sensitivity AVLEN for different number of allowed queries, we consider a set of allowed query number $\{2, 3, 4, 5\}$ and evaluate performance. Figure 3b shows the SPL metric for allowed queries $\in \{2, 3, 4, 5\}$ in presence of unheard sound. As expected, increasing the number of queries suggest an increase in the SPL for methods with AVLEN demonstrating a clear advantage.

Qualitative Results. Figure 4 provides two example episodes of semantic audio-visual navigation using AVLEN. Please refer to the **supplementary materials for more visualizations and details**.

5 Conclusions

In this paper, we looked at the novel task of audio-visual-language navigation in a realistic virtual world, enabled by the SoundSpaces simulator. The agent, visualy navigating the scene to localize an audio goal, is also equipped with the possibility of asking an oracle for help. We modeled the problem as one of learning a multimodal hierarchical reinforcement learning policy, with a two-level policy model: higher-level policy to decide when to ask questions, and lower-level policies to either navigate using the audio-goal or follow the oracle instructions. We presented experiments using our proposed framework; our results show that using the proposed policy allows the agent achieve higher success rate for semantic audio-visual navigation task.

6 Limitations and Societal Impacts

Limitations. AVLEN requires to have a strong language based navigation policy (π_ℓ) to assist the semantic audio-visual navigation task. Moreover, the agent can query at any location in the 3D environment and we rely on a pretrained speaker model to generate instructions. Therefore, in some of the cases the instruction itself is very noisy and may hamper the performance of the entire system. Also, this work only considers English annotations and MatterPort 3D dataset resembles North American houses.

Societal Impacts. The ability to interact with oracle/human using natural language instructions to solve difficult tasks (e.g., navigation in this paper) is of great importance from a human-machine interaction standpoint, or enabling partial-autonomy to a robotic agent with occasional guidance. Such a system will definitely push forward the automation process of tasks that are challenging for humans to do. On the other side, it may negatively affect worker employment. Also, training our dataset would need navigation environments for the agent to explore and learn, which could potentially bring in biases into the agent's behaviour.

References

[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.

[3] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[5] Changan Chen, Ziad Al-Halah, and Kristen Grauman. Semantic audio-visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15516–15525, 2021.

[6] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *European Conference on Computer Vision*, pages 17–36. Springer, 2020.

[7] Changan Chen, Sagnik Majumder, Ziad Al-Halah, Ruohan Gao, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Learning to set waypoints for audio-visual navigation. In *International Conference on Learning Representations*, 2021.

[8] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547, 2019.

[9] Ta-Chung Chi, Minmin Shen, Mihail Eric, Seokhwan Kim, and Dilek Hakkani-tur. Just ask: An interactive learning framework for vision and language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2459–2466, 2020.

[10] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2018.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 538–547, 2019.

[13] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 31, 2018.

[14] Chuang Gan, Yiwei Zhang, Jiajun Wu, Boqing Gong, and Joshua B Tenenbaum. Look, listen, and act: Towards audio-visual embodied navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9701–9707. IEEE, 2020.

[15] Yicong Hong, Cristian Rodriguez-Opazo, Qi Wu, and Stephen Gould. Sub-instruction aware vision-and-language navigation. *arXiv preprint arXiv:2004.02707*, 2020.

[16] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1643–1653, June 2021.

[17] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6741–6749, 2019.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020.

[20] Tuyen P Le, Ngo Anh Vien, and TaeChoong Chung. A deep hierarchical reinforcement learning algorithm in partially observable markov decision processes. *Ieee Access*, 6:49089–49102, 2018.

[21] Chong Liu, Fengda Zhu, Xiaojun Chang, Xiaodan Liang, Zongyuan Ge, and Yi-Dong Shen. Vision-language navigation with random environmental mixup. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1644–1654, 2021.

[22] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR, 2020.

[23] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. *arXiv preprint arXiv:1901.03035*, 2019.

[24] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6732–6740, 2019.

[25] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *European Conference on Computer Vision*, pages 259–274. Springer, 2020.

[26] Khanh Nguyen and Hal Daumé III. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 684–695, Hong Kong, China, November 2019. Association for Computational Linguistics.

[27] Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12527–12537, 2019.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[29] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[30] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[31] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. *arXiv preprint arXiv:1904.04195*, 2019.

[32] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR, 2020.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[34] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019.

[35] Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. Embodied question answering in photorealistic environments with point cloud perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6659–6668, 2019.

[36] Licheng Yu, Xinlei Chen, Georgia Gkioxari, Mohit Bansal, Tamara L Berg, and Dhruv Batra. Multi-target embodied question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6309–6318, 2019.

[37] Wang Zhu, Hexiang Hu, Jiacheng Chen, Zhiwei Deng, Vihan Jain, Eugene Ie, and Fei Sha. Babywalk: Going farther in vision-and-language navigation by taking baby steps. *arXiv preprint arXiv:2005.04625*, 2020.

[38] Yi Zhu, Yue Weng, Fengda Zhu, Xiaodan Liang, Qixiang Ye, Yutong Lu, and Jianbin Jiao. Self-motivated communication agent for real-world vision-dialog navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1594–1603, 2021.

NeurIPS Paper Checklist

- Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? (Yes)
- Have you read the ethics review guidelines and ensured that your paper conforms to them? (Yes)
- Did you discuss any potential negative societal impacts of your work? (Yes)
- Did you describe the limitations of your work? (Yes)
- Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? (Instructions will be provided in supplementary material)
- Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? (All details are provided in supplementary material)
- Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? (Will be provided in the supplementary materials.)
- Did you include the amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? (Provided in supplementary material)
- If your work uses existing assets, did you cite the creators? (Yes)
- Did you mention the license of the assets? (Not required, it's public dataset)
- Did you include any new assets either in the supplemental material or as a URL? (No)
- Did you discuss whether and how consent was obtained from people whose data you're using/curating? (Not applicable)
- Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? (Not applicable)