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

Anonymous Author(s)
Affiliation
Address
email

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

1 Qualitative Navigation Video Results

We have included a Powerpoint slide deck as well as raw videos demonstrating our algorithm in action. Please have a look at the slides and the associated README.txt for details on how to view these videos and the slides.

Below, we provide a brief overview of the included videos. **Note that the raw videos will need VLC player to hear the audio properly**. Following are the video names in the raw video set from five example episodes, utilizing our proposed AVLEN approach:

- *pa4otMbVnkk_22608_cushion_spl0.88.mp4*
- *pa4otMbVnkk_15330_picture_spl1.00.mp4*
- *pa4otMbVnkk_14394_picture_spl0.14.mp4*
- *jtcxE69GiFV_2648_cabinet_spl1.00.mp4*
- *fzynW3qQPVF_22767_cabinet_spl0.31.mp4*

Working and Failure Cases: Among the five episodes, *jtcxE69GiFV_2648_cabinet_spl1.00.mp4* completes the episode without querying any language instruction. In all other videos, agent queries and utilizes language instructions for navigation.

Comparison With Alternatives: We have also provided videos corresponding to scene *pa4otMbVnkk* and episode 15330, where

- *pa4otMbVnkk_15330_picture_spl0.00_savi.mp4*: uses only audio goal policy π_g.
- *pa4otMbVnkk_15330_picture_spl0.00_jask.mp4*: utilizes Model Uncertainty to decide when-to-query.

2 Model Architecture

Model architecture for query policy π_q. Our policy network for π_q follows an architecture similar to [1], consisting of a Transformer encoder-decoder model [5]. The encoder sub-module takes in the embedded features e_t 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 (here, action is selection of lower-level policy) $\pi_q(b, .)$ and the value of this state corresponding to selecting the option policy. Then the agent selects lower-level policy by $\pi_q(b, .)$.

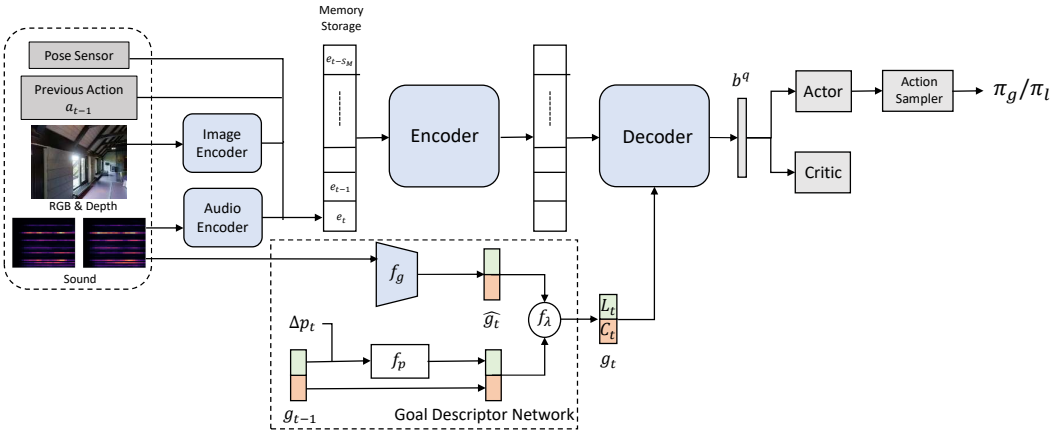


Figure 1: Network architecture for option selection/ query policy π_q

Model architecture for goal-based navigation policy π_g. Our policy network for π_g follows an architecture similar to [1], consisting of a Transformer encoder-decoder model [5]. The encoder

sub-module takes in the embedded features e_t 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 step by action $a \sim \pi_g(b, .)$.

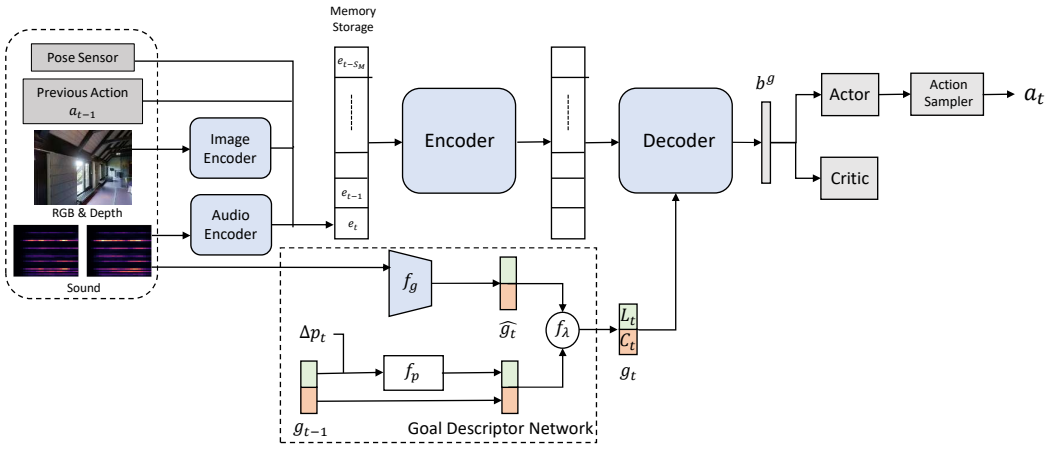


Figure 2: Network architecture for goal-based navigation policy π_g. The model architecture is similar to option selection/query policy π_q. However, the action space is different for these two policies.

Model architecture for language-based navigation policy π_ℓ. When an agent queries, it receives natural language instruction $\text{instr} \in \mathcal{V}^N$ from the oracle. Using instr and the current observation e_t, our language-based navigation policy performs a sequence of actions $\langle a_t, a_{t+1}, \ldots, a_{t+\nu} \rangle$, where each $a_i \in A$. Specifically, for any step $\tau \in \langle t, \ldots, t+\nu-1 \rangle$, π_ℓ first encodes $\{e_\tau, g_\tau\}$ using a Transformer encoder-decoder network T_1 (Observation state encoder), the output of this Transformer is then concatenated with CLIP [4] 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 (history of belief information) using a second multi-layer Transformer encoder-decoder T_2 to produce the new belief state b_τ, i.e.,

$$b_\tau = T_2\left(\text{FC}_1\left(T_1(e_\tau, g_\tau), \text{CLIP}(\text{instr})\right), \left\{b_{\tau'} : t < \tau' < \tau\right\}\right) \text{ and } \pi_\ell(b_\tau, .) = \text{softmax}(\text{FC}_2(b_\tau)).$$

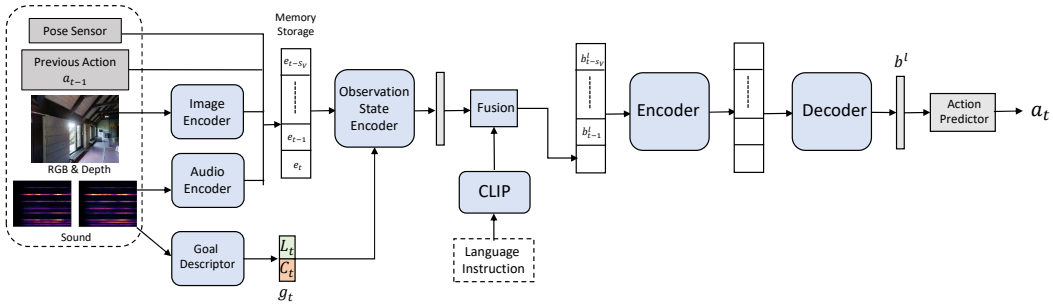


Figure 3: Network architecture for language-based navigation policy π_ℓ

3 Implementation Details

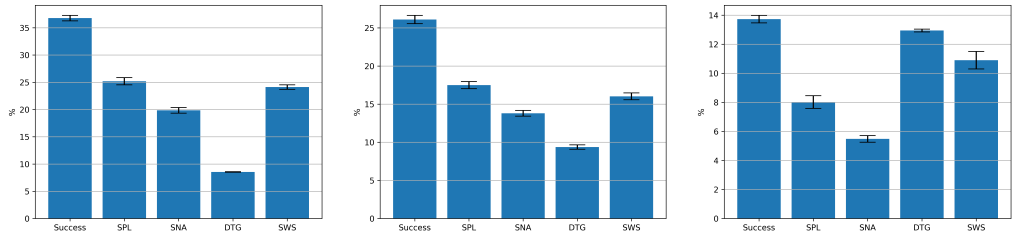
Training query policy π_q. Similar to prior works, we use RGB and depth images, center-cropped to 64×64. The agent receives binaural audio clip as 65×26 spectrograms. The memory size for π_g and π_q is $S_M = 150$. All the experiments consider maximum $K = 3$ allowed queries (unless otherwise

specified). For each query, the agent will take $\nu = 3$ navigation steps in the environment using the natural language instruction. π_q policy training uses ADAM [3] with learning rate 2.5×10^{-4}. Goal descriptor network uses 1×10^{-3} learning rate. The policy was rolled out for 150 steps and updated with each collected experience for two epochs. We use $\sim 22M$ steps to train π_q.

Training goal-based navigation policy π_ℓ. Similar to π_q, 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_ℓ is $S_V = 3$. Agent is allowed to take $\nu = 3$ navigation steps in the environment using the natural language instruction. π_ℓ policy training uses ADAM [3] with learning rate 1×10^{-4}. The policy was (pre-)trained using repurposed vision-language dataset for ~ 8 epochs.

4 Performance Error Analysis

To check the consistency of performance of our proposed AVLEN, we consider running the experiment with four different random seeds. Figure 4 illustrates the standard deviation error bars for the experiments. We observe that the variance of performance for different experiments are insignificant. Standard deviation for success rate is 0.50, 0.53 and 0.26 respectively for heard sound, unheard sound and distractor sound. For all other metrics, the variance os also low.

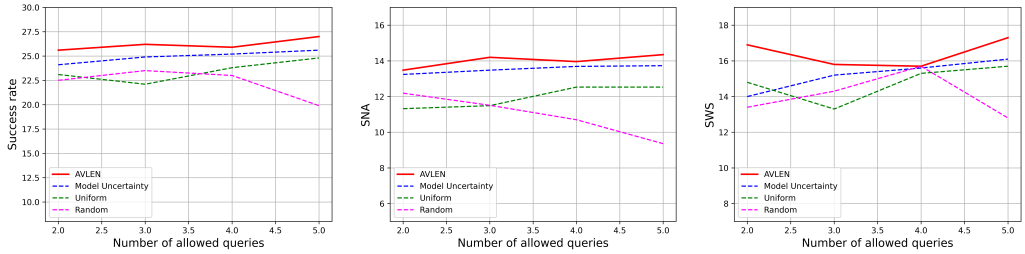


(a) Heard sound (b) Unheard sound (c) Distractor sound

Figure 4: Performance error analysis

5 Sensitivity to Allowed Number of Queries

To check the sensitivity AVLEN for different number of allowed queries, we consider a set of allowed query number $\nu = \{2, 3, 4, 5\}$ and evaluate performance. Figure 5 shows the success rate, SNA and SWS metric for allowed queries $\in \{2, 3, 4, 5\}$ in presence of unheard sound. For the the metrics, AVLEN retains an advantage over other approaches.

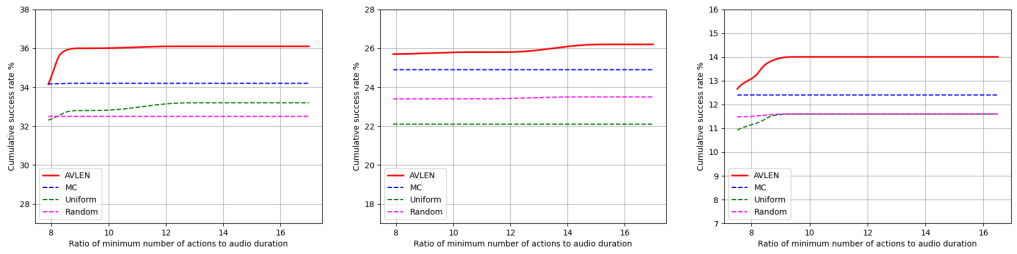


(a) Query sensitivity (SR) (b) Query sensitivity (SNA) (c) Query sensitivity (SWS)

Figure 5: Sensitivity to the number of queries ν to the oracle that AVLEN can make. The results are for the unheard sound sceneario. Please see the main paper for plots on the success rate.

6 Robustness to Silence Duration

Figure 6 shows the cumulative success of different approaches. The x axis represents the silent ratio (ratio of the minimum number of actions required to reach the goal to the duration of audio). A point (x, y) on this plot means the fraction of successful episodes with ratios up to x among all episodes is y. When this ratio is greater than 1, no agent can reach the goal before the audio stops. The greater this ratio is, the longer the fraction of silence, and hence the harder the episode. We observe that AVLEN results in higher cumulative success when sound is silent for longer period.



(a) Heard sound (b) Unheard sound (c) Distractor sound

Figure 6: Robustness to silence duration analysis

7 Vision-Language Navigation Performance

In our setting, an agent receives natural language instruction when it queries. It needs to "comprehend" this instruction properly and should take navigation steps grounded on this instruction. To analyze if π_ℓ (the language policy) takes navigation steps well-grounded on the instruction, we created a VLN test-set of $7,031$ short instruction-trajectory pairs. These short trajectories aligns/overlaps with segments of test-set trajectories from semantic audio-visual navigation dataset. We analyzed the performance of **VLN-b**: trained on repurposed fine-grained instruction from [2], **VLN-f**: fine-tuned π_ℓ with collected trajectory-instruction pairs in AVLEN training, and **VLN-b (w/o instruction)** (language instruction masked) in the VLN test-set. In Table 1, evaluation metric $step - n$ reflects the percentage of episodes that took n sequential steps correctly. Table 1 shows that there is a significant drop in performance if the language is masked out (removed), which indicates π_ℓ predictions are grounded on the instruction. Also, fine-tuning π_ℓ policy with collected trajectory-instruction pairs in an online manner helps improve the performance.

	$Step-1$	$Step-2$	$Step-3$
VLN-b (w/o instruction)	51.3	22.2	17.0
VLN-b	62.8	47.3	37.8
VLN-f	**65.9**	**55.5**	**45.3**

Table 1: Vision-language navigation performance.

References

[1] 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.

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

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

[4] 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.

[5] 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.