

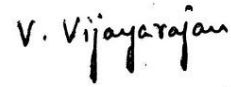
## **CERTIFICATE**

This is to certify that the thesis entitled “**Visual Language Navigation**” submitted by **Oruganti Shiva Charan & 19BCE0545, Scope**, VIT, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering*, is a record of bonafide work carried out by him / her under my supervision during the period, 01.07.2022 to 30.04.2023, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date : 20.05.23



**Signature of the Guide**

**Internal Examiner**

**External Examiner**

**Head of the Department**

**Computer Science and Engineering**

## **ACKNOWLEDGEMENTS**

I would like to acknowledge my professor, Vijayarajan V, for their invaluable guidance, support, and expertise throughout the duration of my capstone project. Their dedication to teaching and commitment to my academic growth has shaped my research and overall learning experience. I am truly grateful for their patience, encouragement, and insightful feedback, which all contributed significantly to the project's success.

I would also like to thank Dr.Ramesh Babu K, the Dean of SCOPE. Their vision and leadership have created an environment that promotes innovation and academic achievement. I am grateful for their encouragement and support in pursuing this capstone project and the department's resources and opportunities.

Furthermore, I am grateful to all of the faculty and teaching staff who shared their knowledge and expertise with me throughout my studies. Their dedication for teaching and commitment to fostering an encouraging academic environment had a significant impact on my understanding of the subject.

I am grateful to my classmates and friends for their insightful comments, support, and encouragement throughout this capstone project. Their cooperation and openness to sharing ideas have enhanced my learning experience and made the journey more enjoyable.

I consider myself extremely fortunate to have had the opportunity to work on this capstone project, and I am grateful to everyone who helped make it possible. Their encouragement and belief in my abilities have been invaluable, and I am grateful for their contribution to my academic and personal development.

Oruganti Shiva Charan  
**Student Name**

## **Executive Summary**

Vision language navigation is a task in Embodied AI research where an agent has to navigate to a location mentioned given language instruction and visual inputs. However, the vision language navigation task poses some challenges, like Visual Complexity, Language Ambiguity, Multimodal Integration, Generalization to Unseen Environments, Dealing with Uncertainty, Zero-shot Learning, Real-time Decision-making, Evaluation Metrics, Human-Agent Interaction, and Long-Term Dependencies. In this project, we concentrate on Human-Agent Interaction and Long-Term Dependencies Challenges for which we introduce HCAM and RLHF mechanisms to our model and training, which enable the agent to think like a human while navigating and storing to infer to make future informed decisions in the same episode as cross episode memory is not tested in this project. We use the Mattersim simulator for an environment with 1.2 TB of data. The dataset comprises 194,400 RGB-D images of 90 building-scale scenes with 21,567 navigation instructions. Our model will be trained using contrastive supervised learning with ground actions, and RLHF techniques will be used as mentioned below. We observe its success in Human-Agent Interaction and Long-Term dependencies and failure in Generalization to Unseen Environments. The metrics used are success rate 42%, Trajectory length -  $x$ , and Navigation error -  $Y$ .

<b>CONTENTS</b>	<b>Page</b>
	<b>No.</b>
<b>Acknowledgement</b>	<b>i</b>
<b>Executive Summary</b>	<b>ii</b>
<b>Table of Contents</b>	<b>iii</b>
<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vi</b>
<b>Abbreviations</b>	<b>vii</b>
<b>Symbols and Notations</b>	<b>viii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Theoretical Background	1
1.2 Motivation	1
1.3 Aim of the Proposed work	2
1.4 Objective(s) of the proposed work	3
<b>2. Literature Survey</b>	<b>4</b>
2.1. Survey of the Existing Models/Work	4
2.2. Summary/Gaps identified in the Survey	9
<b>3. Overview of the Proposed System</b>	<b>11</b>
3.1. Introduction and Related Concepts	11
3.2. Framework, Architecture or Module for the Proposed System	13
<b>4. Proposed System Analysis and Design</b>	<b>18</b>
4.1. Introduction	18
4.2. Requirement Analysis	20
4.2.1.Functional Requirements	
4.2.1.1. Product Perspective	
4.2.1.2. Product features	
4.2.1.3. User characteristics	
4.2.1.4. Assumption & Dependencies	
4.2.1.5. Domain Requirements	
4.2.1.6. User Requirements	
4.2.2.Non-Functional Requirements	26
4.2.2.1. Product Requirements	
4.2.2.1.1. Efficiency	
4.2.2.1.2. Reliability	
4.2.2.1.3. Portability	

4.2.2.1.4.	Usability	
4.2.2.2.	Organizational Requirements	27
4.2.2.2.1.	Implementation Requirements	
4.2.2.2.2.	Engineering Standard Requirements	
4.2.2.3.	Operational Requirements	28
	<ul style="list-style-type: none"><li>• Economic</li><li>• Environmental</li><li>• Social</li><li>• Political</li><li>• Ethical</li><li>• Health and Safety</li><li>• Sustainability</li><li>• Legality</li><li>• Inspectability</li></ul>	
4.2.3.	System Requirements	29
4.2.3.1.	H/W Requirements	
4.2.3.2.	S/W Requirements	
<b>5.</b>	<b>Results and Discussion</b>	<b>30</b>
<b>6.</b>	<b>References</b>	<b>31</b>

## List of Figures

Figure No.	Title	Page No.
1	HCAM attention block	14
2	Where $R$ = chunk relevance, $Q$ = Linear Projection Layer, $S$ = Memory Summary Keys	14
3	where $R_i$ =chunk relevance, MHA=Multi Head Attention, $C_i$ =memory chunks,	14
4	$p[\sigma^1 > \sigma^2]$ = estimated probability of segment 1 preferred than segment 2 and vice versa, where $\sigma^1$ and $\sigma^2$ are the two segments	15
5	cross entropy loss of $p$ loss between predictions and human labels, where $\mu$ is distribution over 1 and 2	15
6	Describes MLM	16
7	Loss function	17
8	Self-supervised learning.	18
9	Loss function for Part - 2	19
10	Training Process	20
11	Dataset making procedure	20
12	PPO-CLIP	21
13	Model Architecture	22
14	Rewards vs steps avg over episodes and multiple runs	30
15	Rewards vs steps avg over episodes and multiple runs	30

## **List of Tables**

<b>Table No.</b>	<b>Title</b>	<b>Page No.</b>
1.	Literature Survey	4
2.	Introduction	18
3.	R2R Validation Seen	29
4.	R2R Validation Unseen	29

## **List of Abbreviations**

VLN	Visual Language Navigation
RLHF	Reinforcement learning using human feedback
HCAM	Hierarchical chunk memory attention
RL	Reinforcement learning
USE	Universal sentence encoder
Vit	Vision transformer
R2R	Room to room
BERT	Bidirectional Encoder Representations from Transformers
NLP	Natural Language Processor



**Symbols and Notations**

R	chunk relevance
Q	Linear Projection Layer
$\sigma_1, \sigma_2$	segments

# 1. INTRODUCTION

## 1.1 Theoretical Background:

Vision-Language Navigation (VLN) is an interdisciplinary research field that aims to develop intelligent agents capable of navigating and understanding natural language instructions in visually realistic environments. The ability to successfully navigate complex environments following language instructions holds great promise for many applications, including robotics, virtual reality, and augmented reality. VLN tasks involve training agents to understand and interpret natural language instructions, comprehend visual information, and effectively navigate dynamic and visually diverse environments. Vision language navigation is a task in the field of Embodied AI research where an agent has to navigate to a point mentioned given a language instruction and visual inputs. Embodied AI is the area of AI that focuses on addressing problems for virtual robots that can move, see, speak, and interact with other virtual robots and the virtual world.

Formally, the agent is given a natural language command  $x = (x_1, x_2, \dots, x_L)$  as input at the start of each episode, where  $L$  is the length of the instruction and  $x_i$  is a single word token. The agent views an initial RGB image  $O$  that is based on its starting pose. The agent must carry out a series of actions  $(s_0, a_0, s_1, a_1, \dots, s_T, a_T)$  each of which results in a new stance  $(s_{t+1} = h(v_{t+1}, t+1, t+1_i))$  and a new observation of the world  $(O_{t+1})$ . When the agent chooses the special stop action, which has been added to the simulator action area, the episode comes to a close. If the action sequence brings the agent close to the desired goal position, the task is successfully accomplished. The most pressing problem in the ongoing research is the ability to generalize a model trained in seen environments to unseen environments.

## 1.2 Motivation

The vision language navigation task poses some challenges like Language Ambiguity is where natural language instructions can be inherently ambiguous, leading to different interpretations and potential confusion for VLN agents. Resolving ambiguity and accurately understanding the intended meaning of instructions poses a significant challenge. Fortunately, the current research models in NLP resolve this ambiguity to a great extent and are extremely useful in training these agents. Visual Complexity: Navigating visually realistic environments requires agents to interpret and comprehend complex image scenes. These scenes may

contain various objects, occlusions, lighting variations, and intricate spatial relationships, making it challenging for agents to extract relevant information for navigation. The current state of models in computer vision like ViT can produce rich features from the images which can and are used in training these agents. Different training procedures are being made to fine-tune these models to use in VLN agents as feature extractors. The next is Multimodal Integration: VLN tasks involve processing and integrating multimodal inputs, including textual instructions and visual information. Effectively fusing these modalities and leveraging their synergies is a non-trivial challenge. The researchers use contrastive pretraining methods, which perform well in understanding the combined representation of image and language, like the CLIP model, which solves the above problem satisfactorily. When we say satisfactorily, improving further on this may not increase the metrics by a significant degree. Other issues like Generalization to Unseen Environments, Dealing with Uncertainty, Zero-shot Learning, Real-time Decision-making, Evaluation Metrics, Human-Agent Interaction, and Long-Term Dependencies must be improved to achieve these ideal agents. One proven way for Zero-shot Learning is to train large parameter models. Dealing with Uncertainty and Real-time Decision-making is not a part of this project as our environment is virtual and assumes these to be given. But these problems arise when we transfer from the virtual to the real world and must be addressed. As the standard defined metrics to measure the success of VLN agents are helpful, they can certainly improve based on the needs of applications. Generalization to Unseen Environments, Long-Term Dependencies and Human-Agent Interaction are the three most critical challenges, and we propose a combination of methods to address these issues.

### **1.3 Aim of the Proposed Work**

This project aims to address the research gaps in Generalization to unseen environments, Long-Term Dependencies, and Human-Agent Interaction. We propose developing a novel model that explicitly targets these challenges and aims to contribute to advancing knowledge in these domains.

Generalization of unseen environments is a crucial aspect of many real-world applications, where the ability of a model to perform well in novel and unfamiliar settings is highly desired. By focusing on this research gap, our model aims to enhance the generalization capabilities of existing systems and improve their performance in scenarios outside their training data.

Long-Term Dependencies refer to the ability of a model to effectively capture and utilize information from past events or inputs over extended periods. This aspect plays a significant role in tasks that require temporal understanding and reasoning. Our project aims to explore and develop techniques that can address the challenges associated with long-term dependencies, enabling models to handle such scenarios better.

Human-Agent Interaction is a critical area where the interaction between humans and intelligent agents is studied. Understanding and improving this interaction is essential for creating efficient and effective human-centric AI systems. We aim to investigate novel approaches to enhance the quality and naturalness of interactions between humans and AI agents, enabling more seamless and intuitive collaborations.

By tackling these research gaps, our project endeavors to contribute valuable insights, methodologies, and advancements to Generalization to unseen environments, Long-Term Dependencies, and Human-Agent Interaction. We believe our proposed model and its findings will contribute to the broader scientific community and pave the way for further developments in these areas."

#### **1.4 Objective of the Proposed Work**

Our project addresses the research gaps in Generalization to unseen environments, Long-Term Dependencies, and Human-Agent Interaction. We propose a model that explicitly targets these challenges. By focusing on the Generalization of Unseen Environments research gap, we hope to improve the present system's generalization capabilities and performance in situations other than their training data. For example, long-term Dependency is crucial in tasks that require temporal comprehension and reasoning. Furthermore, to build effective and efficient human-centric AI systems, it is crucial to comprehend and enhance this relationship. Finally, to enable intuitive agents, we intend to use RLHF to improve the quality of user preferences learned by agents. By addressing these research gaps, we hope to provide valuable insights, approaches and advances to Generalization to Unknown Environments, Long-Term Dependencies, and Human-Agent Interaction.

## 2. LITERATURE SURVEY

### 2.1. Survey of the Existing Models/Work:

Table 1: Literature Survey

Authors and Year	Title	Concept and methodology	Dataset details	Relevant findings	Limitations
Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, Ivan Laptev NeurIPS, 2021	History Aware Multimodal Transformer for Vision-and-Language Navigation	They introduce a History Aware Multimodal Transformer to incorporate a long-horizon history into multimodal decision making. They first train HAMT from beginning to end using a variety of proxy tasks, such as single step action prediction and spatial relation prediction, and then they apply reinforcement learning to further enhance the navigation strategy.	R2R, RxR, R2R-Last, REVE-RIE, CVDN, R4R, R2R-Back.	HAMT achieves new state of the art on all the datasets mentioned	Does not support VLN in continuous actions and has few moral, privacy, or security concerns.

Abhinav Moudgil, Arjun Majumdar, Harsh Agrawal, Stefan Lee, Dhruv Batra NeurIPS, 2021.	SOAT: A Scene- and Object-Aware Transformer for Vision- and- Language Navigation	They propose transformer based VLN agent that uses two different visual encoders. One is scene classification network and other is to detect object. Scene elements provide high-level contextual data that aids in interpretation at the entity level.	R2R, RxR	Improvements of 1.8% absolute in SPL on R2R and 3.7% absolute in SR on RxR.	Use additional object features and perfect images which are not really available in the real world.
Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, Peter Anderson ICCV, 2021	Pathdreamer: A World Model for Indoor Navigation	Introduce Pathdreamer which uses depth observations and predict the future depth image via generating, same with image using cross entropy + MAE loss and multi-SPADE ResBlock and GAN training	R2R	Its performance in generating realistic panoramic images show promise in VLN task	-----

		fashion.			
Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, Stephen Gould CVPR, 2021	VLN BERT: A Recurrent Vision-and-Language BERT for Navigation	They propose recurrent BERT model that is time-aware for use in VLN. They use MLT for encoding language and vision combining with ResNet-152 and Faster-RCNN. They use augmented data.	R2R	The ensemble of many models On "unseen" validation, VLN-BERT increases SR by 3.0 absolute percentage points, resulting in an SR of 73% on the VLN.	-----
Muhammad Zubair Irshad, Chih-Yao Ma, Zsolt Kira ICRA, 2021	Hierarchical Cross-Modal Agent for Robotics Vision-and-Language Navigation	They propose hierarchical cross-model agent and use layered decision making, modularized training and decoupling reasoning with imitation. In high level modules they use CNN and cross modal transformer. In low level they	R2R	It correctly anticipates low-level directives and arrives at the specified goal position. By obtaining a 40% SPL and 46% SR.	Memory is not addressed. As it is important for the task.

		don't use CMT.			
Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, Dhruv Batra ECCV, 2020	Improving Vision-and-Language Navigation with Image-Text Pairs from the Web	They introduce a visiolinguistic transformer-based model called VLN-BERT which evaluates compatibility. They initially use ViLBert with self supervised learning and then use the model for the VLN. Use both reinforcement learning and IL.	R2R, REVERIE	They show that VLN-BERT increases SR by 3.0 absolute percentage points on "unseen" validation, resulting in an SR of 73%.	Memory is not addressed. As it is important for the task.
Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, Jianfeng Gao CVPR, 2020.	Towards Learning a Generic Agent for Vision-and-Language Navigation via Pre-training	The pre-trained model offers general representations of visual surroundings and linguistic instructions, and they train on a large number of image-text-action triplets under self-	R2R	It improves the SOAT from 47% to 51% on success rate.	Memory is not addressed. As it is important for the task.



		supervised learning conditions.			
Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, Lei Zhang CVPR, 2019.	Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation	Through reinforcement learning, it imposes cross-modal grounding both locally and globally (RL). A matching critic is used to provide an intrinsic reward to induce global matching between instructions and trajectories.	R2R	significantly outperforms the existing methods, improving the SPL score from 28% to 35%	Memory is not addressed. As it is important for the task.
Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko,	Speaker-Follower Models for Vision-and-Language Navigation	They present the speaker-follower model, in which they synthesise fresh instructions for data augmentation and apply pragmatic reasoning. They use ResNet for visual features	R2R	Achieves a success rate of 53.3% and adding the augmented data improves success rate (SR) from 40.3% to 46.8% on validation seen and from 19.9% to 24.6% on validation unseen.	-----

Dan Klein, Trevor Darrell NeurIPS, 2018.		which is pretrained on Imagenet			
Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, Anton van den Hengel CVPR, 2018.	Vision-and- Language Navigation: Interpreting Visually- Grounded Navigation Instructions in Real Environments	This was the First paper. They introduced the Matterport3D simulator and baseline models like sequence- to-sequence model. Where the language encoding used LSTMs and image used Resnet-152 and CNN. Given R2R dataset.	R2R	They have achieved over 20 % in success rate.	Most of the homes that were photographed are impeccably tidy and spotless. There are extremely few humans and animals in the dataset.

## 2.2 Summary/Gaps Identified

**1. Language Ambiguity:** Natural language instructions can be inherently ambiguous, leading to different interpretations and potential confusion for VLN agents. Resolving the ambiguity and accurately understanding the intended meaning of instructions poses a significant challenge.

**2. Visual Complexity:** Navigating in visually realistic environments requires agents to interpret and comprehend complex visual scenes. These scenes may contain various objects, occlusions, lighting variations, and intricate spatial relationships, making it challenging for agents to extract relevant information for navigation.

**3. Multimodal Integration:** VLN tasks involve processing and integrating multimodal inputs, including textual instructions and visual information. Effectively fusing these modalities and leveraging the synergies between them is a non-trivial challenge.

**4. Generalization to Unseen Environments:** VLN agents must be capable of generalizing their navigation skills to new, unseen environments. However, due to the variability and diversity of environments, ensuring robust generalization remains a significant challenge, especially with limited training data.

**5. Dealing with Uncertainty:** Navigation in real-world environments involves dealing with uncertainty, such as perceptual noise, sensor limitations, or dynamic changes in the environment. Agents must be able to handle such uncertainties and make reliable decisions in uncertain situations.

**6. Long-Term Dependencies:** VLN tasks often require agents to remember and refer back to previous states, actions, or instructions to make informed navigation choices. Capturing and utilizing long-term dependencies effectively is a challenge, particularly when memory and context span over extended periods.

**7. Zero-shot Learning:** VLN agents should be capable of understanding and following instructions in new tasks without explicit training on those specific tasks. Zero-shot learning requires agents to generalize knowledge from previous tasks and adapt it to new scenarios, which is a challenging capability to develop.

**8. Real-time Decision-making:** VLN agents need to navigate efficiently and make decisions in real time, especially in dynamic environments. Balancing the trade-off between speed and accuracy poses a challenge, as agents must act quickly while maintaining safe and effective navigation.

**9. Evaluation Metrics:** Defining appropriate evaluation metrics for VLN tasks is a complex task itself. Metrics should capture the quality of both language understanding and navigation performance, considering factors such as goal-reaching accuracy, path efficiency, and language adherence.

**10. Human-Agent Interaction:** VLN systems often involve interactions between human users and the navigating agent. Ensuring effective communication, understanding user preferences, and handling user feedback in real-time present challenges in creating seamless and intuitive human-agent interactions.

### 3. OVERVIEW OF THE PROPOSED SYSTEM

#### 3.1 Introduction and Related Concepts

##### 3.1.1 HCAM

Regarding Long-Term Dependencies, VLN tasks often require agents to remember and refer to previous states, actions, or instructions to make informed navigation choices. Capturing and utilizing long-term dependencies effectively is a challenge, particularly when memory and context span over extended periods. HCAM introduced by can be experimented upon in this case. The original paper presents a novel approach called "hierarchical memory" HCAM to enhance the decision-making capabilities of RL agents. To evaluate the effectiveness of hierarchical memory, the authors conduct experiments, including Remembering the Ballet, which involves the agent recalling its previous observations and going to that particular dancer after seeing a 12-32 dance episode. Memory and recall were tested in all of these experiments. The results demonstrate that agents equipped with hierarchical memory outperform those without memory or with a single-level memory.

Furthermore, the hierarchical memory allows the agents to generalize their knowledge across different tasks and adapt their decision-making strategies to unseen situations. Overall, the paper presents the concept of hierarchical memory as a valuable mechanism for RL agents to improve their decision-making abilities. By enabling agents to perform mental time travel and leverage past experiences, the hierarchical memory enhances the agents' ability to generalize, adapt, and make informed decisions in dynamic environments. We will be using this mechanism in this paper.

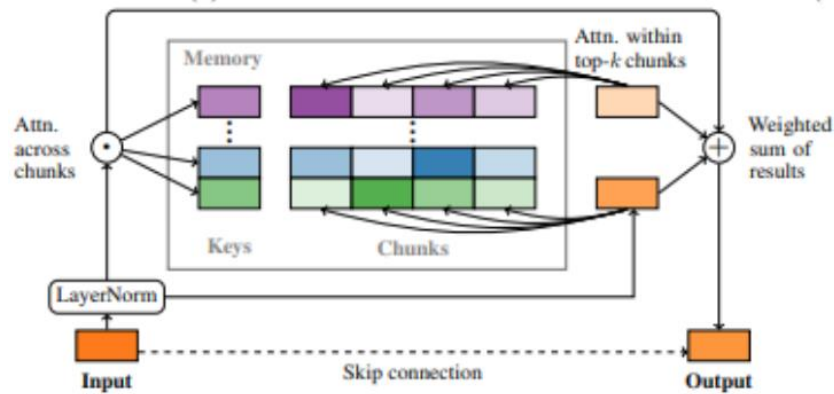


Fig 1: HCAM attention block

$$R = \text{softmax}(Q(\text{normed input}) \cdot S)$$

Fig 2: Where R = chunk relevance, Q= Linear Projection Layer, S= Memory Summary Keys

$$\text{memory query results} = \sum_{i \in \text{top-}k \text{ from } R} R_i \cdot \text{MHA}(\text{normed input}, C_i)$$

Fig 3: where  $R_i$ =chunk relevance, MHA=Multi Head Attention,  $C_i$ =memory chunks,

### 3.1.2 RLHF

The next part of this paper uses RLHF. The concept of Reinforcement Learning with Human Feedback (RLHF) has been explored in various studies, but pinpointing the exact first paper on RLHF can be challenging due to the vast literature in the field. However, one influential early article on RLHF is "Apprenticeship Learning via Inverse Reinforcement Learning" by Pieter Abbeel and Andrew Y. Ng. In this paper, the authors introduce the concept of apprenticeship learning, which combines ideas from inverse reinforcement learning (IRL) and RLHF. The focus is on learning from demonstrations provided by an expert rather than relying solely on trial-and-error exploration. By incorporating human demonstrations, the RL agent can learn from an expert's behavior and acquire skills more efficiently. The paper proposes a mathematical framework for apprenticeship learning, which involves estimating the underlying reward function from expert demonstrations and then using this estimated reward function to train the RL agent.

RLHF was initially introduced as an approach to address the challenges of sparse reward signals and accelerate the learning progress in RL. Traditional RL methods rely on trial-and-error exploration to learn optimal policies, which can be time-consuming and inefficient. RLHF leverages human expertise to provide additional guidance and feedback to the RL agent, helping it learn faster and perform better. In RLHF, human feedback can take various forms, such as demonstrations, preferences, or direct evaluations. Demonstrations involve a human expert showcasing desired behaviors or providing example trajectories for the RL agent to learn from. Preferences involve the comparison of different actions or trajectories to indicate preferred choices. Direct evaluations provide explicit feedback or reward signals to guide the learning process. For example, we generate multiple trajectories, manually compare them in two pairs, and select the most successful one.

$$\hat{P}[\sigma^1 \succ \sigma^2] = \frac{\exp \sum \hat{r}(o_t^1, a_t^1)}{\exp \sum \hat{r}(o_t^1, a_t^1) + \exp \sum \hat{r}(o_t^2, a_t^2)}.$$

Fig 4 :  $p[\sigma^1 > \sigma^2]$ =estimated probability of segment 1 preferred than segment 2 and vice versa,  
where  $\sigma^1$  and  $\sigma^2$  are the two segments

$$\text{loss}(\hat{r}) = - \sum \mu(1) \log \hat{P}[\sigma^1 \succ \sigma^2] + \mu(2) \log \hat{P}[\sigma^2 \succ \sigma^1].$$

Fig 5: cross entropy loss of p loss between predictions and human labels, where  $\mu$  is distribution over 1 and 2

### 3.2 Framework, Architecture or Module for the Proposed System (with explanation)

#### METHODOLOGY ADAPTED

##### Part 1:

**Training of encoders:** Bidirectional Encoder Representations from Transformers and universal sentence encoder with self-supervised learning and pre-training

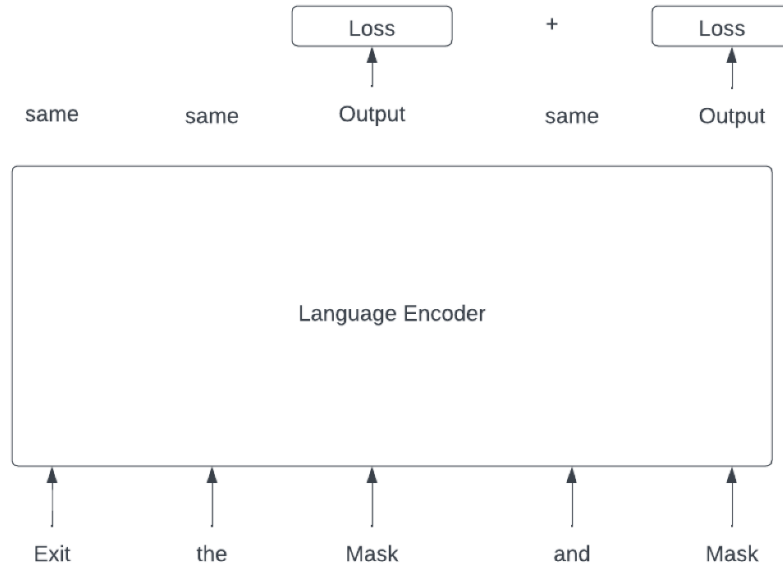


Fig 6: Describes MLM

In part one we train our language encoder and image encoders. We use BERT and the universal sentence encoder. In BERT for first model, we collect all the instructions in train dataset and pretrain the BERT with Masked language modelling loss given in Fig 7. For the model two we take the pretrained model and train with self-supervised setting that is the loss function is the difference between the average of all the tokens and the CLS token outputs. The intuition is to capture all the information in the given text. This process is shown in Fig8.

The images features are taken from ViT transformer.

$$L_{MLM}^{(x)} = -\frac{1}{|M_x|} \sum_{i \in M_x} \log P(x_i / x_{\setminus M_x})$$

where:

$x_{\setminus M_x}$  represents masked version of  $x$

$M_x$  represents set of masked token positions in  $x$

Fig 7: Loss function

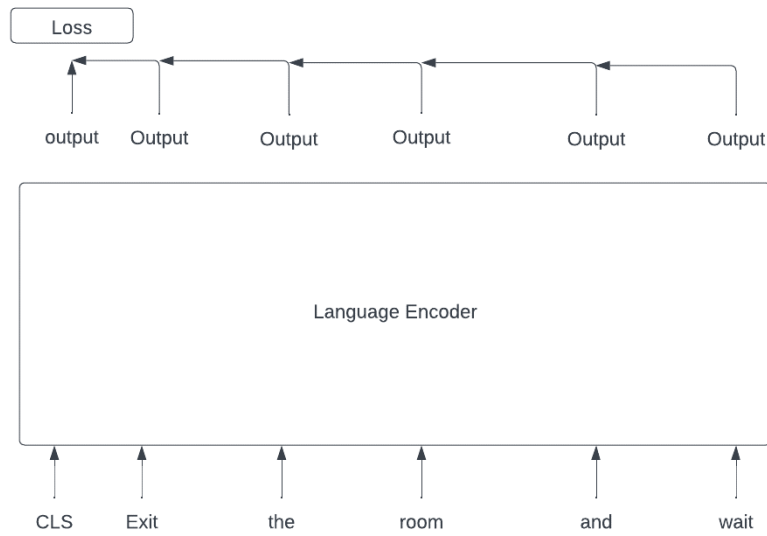


Fig 8: Self-supervised learning.

## Part 2:

Whole network training with contrastive supervised learning:

We add the other network to the encoders trained above and freeze the weights of the encoders to avoid overfitting. We train the whole network using contrastive supervised learning in which we create a dataset with vision language pair and action variable. we create this dataset by iteratively interacting with the simulator and following the tight path at every step. One dataset is set action variable equal to 1 when the action is forward and other actions as 0. The goal is to make all the pairs with similar actions get near in the output vector space and which are not similar be far apart the loss in doing so is mentioned in Fig 9. The second type of the dataset is to continuous perform all actions and categorize all actions not only one and zero but all forward and left and so on. These actions numbers are taken into the dataset and similar actions are labels 1 and others are zero same loss function is used for this also. In

training we take 2 pairs and compare the action in order to determine whether to make them close or apart them.

The loss function for each sample is:

$$\text{loss}(x, y) = \begin{cases} 1 - \cos(x_1, x_2), & \text{if } y = 1 \\ \max(0, \cos(x_1, x_2) - \text{margin}), & \text{if } y = -1 \end{cases}$$

Fig 9: Loss function for Part - 2

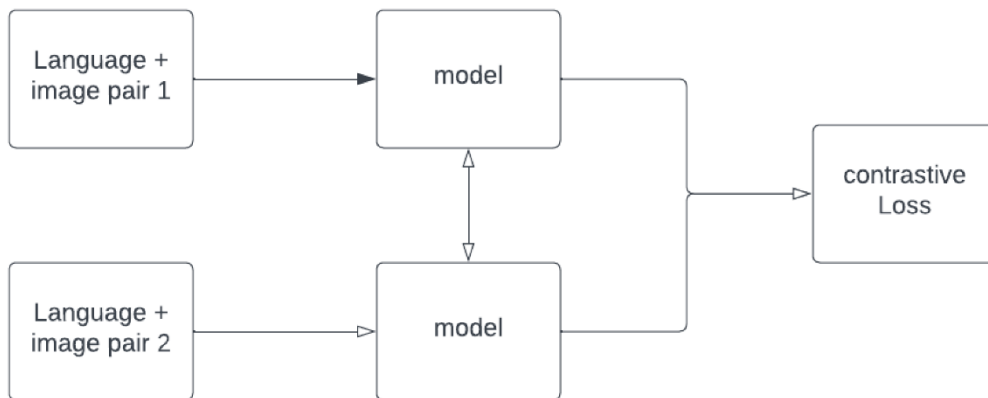


Fig 10: Training Process

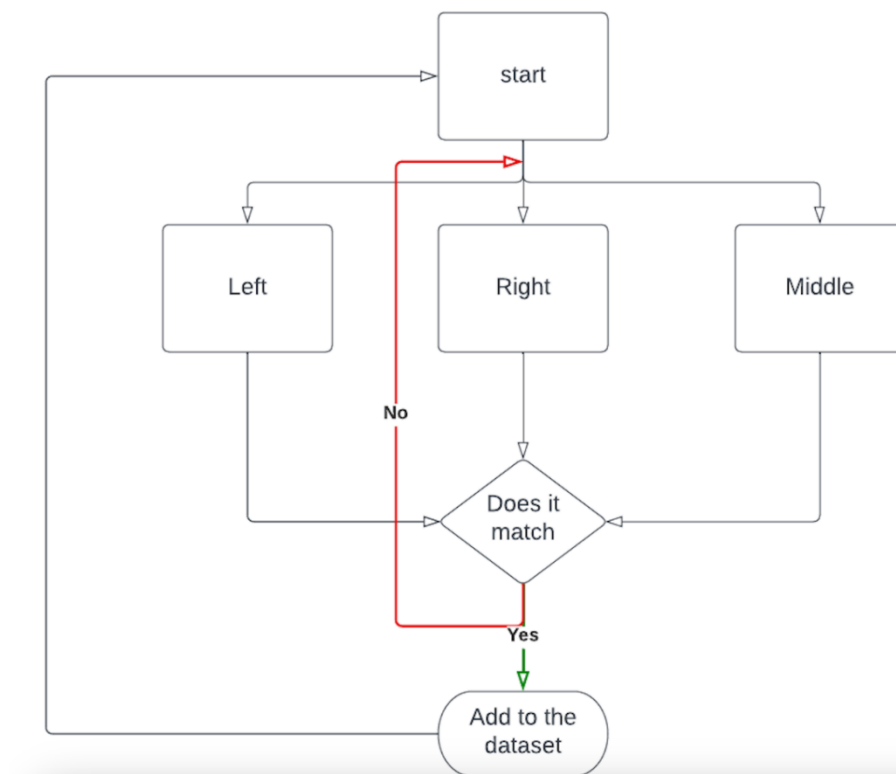


Fig 11: Dataset making procedure



### Part 3:

Training of network with Reinforcement learning (Proximal Policy Optimization (PPO)): -

After completing the Part 1 and Part 2 training, we freeze the weights of all the layers in the previous parts. We add the MLP layers to predict which actions to be taken. We then interact with the Environment and train the network to learn low-level actions. We use PPO algorithm to train the network this algorithm is mentioned below.

- 
- 1: Input: initial policy parameters  $\theta_0$ , initial value function parameters  $\phi_0$
  - 2: **for**  $k = 0, 1, 2, \dots$  **do**
  - 3:   Collect set of trajectories  $\mathcal{D}_k = \{\tau_i\}$  by running policy  $\pi_k = \pi(\theta_k)$  in the environment.
  - 4:   Compute rewards-to-go  $\hat{R}_t$ .
  - 5:   Compute advantage estimates,  $\hat{A}_t$  (using any method of advantage estimation) based on the current value function  $V_{\phi_k}$ .
  - 6:   Update the policy by maximizing the PPO-Clip objective:
$$\theta_{k+1} = \arg \max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^T \min \left( \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), \quad g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right),$$

typically via stochastic gradient ascent with Adam.
  - 7:   Fit value function by regression on mean-squared error:
$$\phi_{k+1} = \arg \min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^T \left( V_{\phi}(s_t) - \hat{R}_t \right)^2,$$

typically via some gradient descent algorithm.
  - 8: **end for**
- 

Fig 12: PPO-CLIP

## ARCHITECTURE

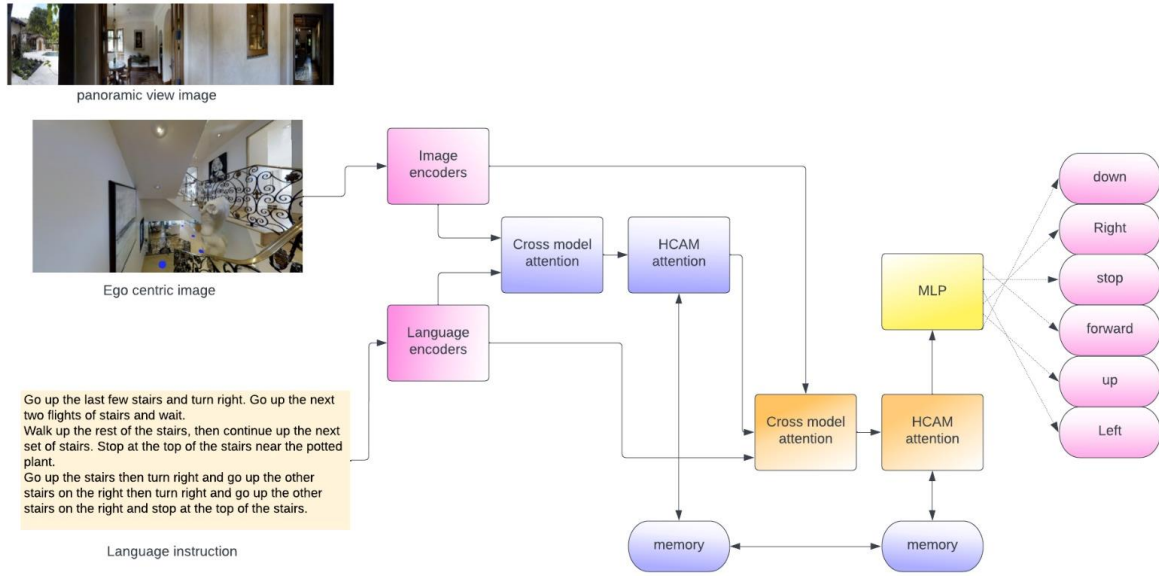


Fig13: Model Architecture

First the Instructions will be inputted to language encoder and the incoming egocentric RGB image will be inputted to image encoder/precomputed features. The output vector representations from USE and VIT will be concatenated and sent to HCAM attention module where the query is the concatenated representation and the memory has the previous representations, these will be key and values for the attention operations. The output of the HCAM attention and the image, language representation will be sent to cross model attention with the RGB representation. The output of this is then send to MLP to get select the final actions as shown in the figure.

## 4. PROPOSED SYSTEM ANALYSIS AND DESIGN

### 4.1. Introduction

Table 2: Introduction

Parameter name	Parameter type	Default	Description
exp-name	str	-	the name of this experiment
seed	int	1	seed of the experiment
torch-deterministic	lambda x: bool(strtobool(x)),	True	if toggled, `torch.backends.cudnn.deterministic=False`
cuda	lambda x: bool(strtobool(x))	True	if toggled, cuda will be enabled by default
track	lambda x: bool(strtobool(x))	False	if toggled, this experiment will be tracked with Weights and Biases
wandb-project-name	str	VLNRL	the wandb's project name
wandb-entity	str	None	the entity (team) of wandb's project
capture-video	lambda x: bool(strtobool(x))	False	whether to capture videos of the agent performances (check out `videos` folder)
env-id	str	CartPole-v1	the id of the environment
learning-rate	float	2.5e-4	the learning rate of the optimizer
num-steps	int	100	the number of steps to run in each environment per policy rollout
anneal-lr	lambda x: bool(strtobool(x))	True	Toggle learning rate annealing for policy and value networks
gamma	float	0.99	the discount factor gamma
gae-lambda	float	0.95	the lambda for the general advantage estimation
num-minibatches	int	4	the number of mini-batches

update-epochs	int	2	the K epochs to update the policy
norm-adv	lambda x: bool(strtobool(x))	True	Toggles advantages normalization
clip-coef	float	0.2	the surrogate clipping coefficient
clip-vloss	lambda x: bool(strtobool(x))	True	Toggles whether or not to use a clipped loss for the value function, as per the paper.
ent-coef	float	0.01	coefficient of the entropy
vf-coef	float	0.5	coefficient of the value function
max-grad-norm	float	0.5	the maximum norm for the gradient clipping
target-kl	float	None	the target KL divergence threshold
new_reward_m	bool	False	Should use the new reward module
same_agent	bool	False	Should use the previously trained agent
init	bool	False	Should run update loop or not
env_batch_size	int	10	Batch size used for environment
batch_size	Int	args.env_batch_size *args.num_steps	Batch size used for update 1
minibatch_size	int	args.batch_size // args.num_ minibatches	Batch size used for update 2

## **4.2. Requirement Analysis**

### **4.2.1. Functional Requirements**

1. Collection of Training Data: For RL training, the system should be able to store training data, including expert demonstrations and human comments.
2. Integration of RL Agents: The system should be able to connect with existing RL algorithms and frameworks, allowing for the training and evaluation of RL agents based on the gathered data.
3. Human Feedback Mechanism: During training or execution, the system should be able to integrate feedback from users—the RL agent—allowing them to guide and correct the agent's behavior.
4. Playback of Expert Demonstrations: During training or execution, the system should be able to include feedback from users—the RL agent—allowing them to direct and adjust the agent's conduct.
5. State Representation: Given the multimodal nature of RLHF tasks, including vision, language, or other modalities, the system should manage environment state representation in a way suitable for RL training.
6. Integration of Reinforcement Signals: The system should provide tools and criteria for evaluating the performance of RL agents trained with human feedback, such as work completion, efficiency, and convergence speed.
7. Memory Mechanisms: The system should incorporate human input into the RL training process, effectively blending environmental reinforcement signals with human feedback direction.
8. Performance Evaluation: The system should provide tools and criteria for evaluating the performance of RL agents trained with human feedback, such as work completion, efficiency, and convergence speed.

#### **4.2.1.2 Product Perspective**

The research presented in this project represents an initial step towards achieving the original authors' final vision, enabling robots to navigate, speak, listen, communicate, and ultimately perform tasks as directed by humans. The overarching goal of this vision is to create intelligent and autonomous robotic systems that can seamlessly interact with humans naturally and intuitively.

By focusing on the specific navigation domain in conjunction with vision and language processing, the researchers aim to address one aspect of this broader vision. They recognize

that for robots to become capable agents in human environments truly, they must possess the ability to understand and interpret natural language instructions, perceive their surroundings through visual inputs, and navigate accordingly.

The research presented in this project on vision-and-language navigation (VLN) has significant implications for various real-world applications. Here are some potential applications that can benefit from the integration of vision, language, and navigation capabilities in robotic systems:

- **Personal Assistants:** Robots equipped with vision-and-language navigation capabilities can serve as personal assistants in homes, offices, or other environments. They can receive natural language instructions from users and navigate to perform tasks such as fetching objects, providing information, or assisting with daily activities.
- **Healthcare Facilities:** Robots with VLN capabilities can assist in healthcare settings by navigating through hospital wards or clinics and following instructions from medical staff or patients. They can deliver medications, transport medical equipment, or support patients and healthcare providers.
- **Retail Environments:** Robots that can understand and navigate using natural language instructions can assist customers in retail stores. They can provide guidance, locate products, and answer customer queries, enhancing the shopping experience.
- **Hospitality Industry:** Robots with VLN capabilities can act as concierge services in hotels or resorts. They can guide guests to their rooms, provide information about facilities, and recommend nearby attractions or services.
- **Smart Homes:** Integrating VLN capabilities into home automation systems allow robots to navigate through the home, performing tasks such as adjusting lighting, temperature control, or managing other intelligent devices based on spoken instructions from residents.
- **Industrial Settings:** Robots with VLN capabilities can be employed in warehouse environments for tasks such as inventory management, picking and placing items, or navigating complex environments for logistics purposes.
- **Emergency Response:** In disaster-stricken areas or emergencies, robots with VLN capabilities can navigate challenging environments, assisting search and rescue missions and relaying information to human responders.

#### **4.2.1.3 Product Features**

##### **Personal Assistants:**

- **Natural Language Interaction:** Robots can understand and respond to spoken instructions, allowing intuitive communication with users.
- **Task Execution:** Robots can navigate to perform various tasks based on user instructions, such as fetching objects or providing information.
- **Personalized Assistance:** Robots can learn and adapt to user preferences, providing customized assistance.

##### **Healthcare Facilities:**

- **Efficient Navigation:** Robots can navigate complex healthcare environments, reducing the burden on healthcare staff.
- **Medication Delivery:** Robots can securely deliver medications to patients, minimizing errors and improving efficiency.
- **Patient Support:** Robots can provide assistance and companionship to patients, enhancing their well-being and comfort.

##### **Retail Environments:**

- **Enhanced Customer Service:** Robots can guide customers, provide product information, and improve the shopping experience.
- **Inventory Management:** Robots can navigate through aisles, assisting in inventory tracking and restocking processes.

##### **Hospitality Industry:**

- **Concierge Services:** Robots can navigate hotels or resorts, providing guests with information, recommendations, and personalized assistance.
- **Efficient Guest Support:** Robots can handle guest inquiries, freeing up staff resources for more complex tasks.

##### **Smart Homes:**

- **Home Automation:** Robots can navigate through homes, adjusting intelligent devices and providing seamless control over various systems.
- **Voice-Activated Assistance:** Users can interact with robots through natural language,

enabling convenient and hands-free control over home automation.

#### Industrial Settings:

- **Warehouse Automation:** Robots with VLN capabilities can navigate and perform tasks such as inventory management, reducing human effort and increasing efficiency.
- **Precision and Accuracy:** Robots can navigate complex industrial environments precisely, enhancing operational effectiveness.

#### Emergency Response:

- **Search and Rescue:** Robots equipped with VLN capabilities can navigate hazardous environments, assisting in locating and rescuing individuals.
- **Real-time Communication:** Robots can relay crucial information to human responders, facilitating coordinated emergency response efforts.

#### **4.2.1.3. User characteristics**

- **Personal Assistants:** Users who require assistance with daily tasks, such as individuals with mobility impairments, elderly individuals, or busy professionals seeking convenience.
- **Healthcare Facilities:**
  1. Medical staff, including doctors, nurses, and caregivers, can benefit from robotic assistance in patient care and hospital operations.
  2. Patients may need support and interaction from robots, such as those in long-term care facilities or hospitals.
- **Retail Environments:** Shoppers may benefit from guidance, recommendations, and information robots provide during their shopping experience.
- **Hospitality Industry:** Hotel guests can benefit from the personalized assistance, information, and recommendations provided by robots acting as concierge services.
- **Smart Homes:** Homeowners or residents who desire seamless control and automation of various home devices and systems.
- **Industrial Settings:** Warehouse managers and staff who can utilize robots with VLN capabilities for efficient inventory management and logistics operations.
- **Emergency Response:** Emergency responders like search and rescue teams can benefit from robotic assistance in hazardous environments.



#### **4.2.1.4. Assumption & Dependencies**

##### **Assumptions:**

- **Availability of Reliable Sensor Data:** The proposed system assumes accurate and reliable sensor data, such as visual and audio inputs, to enable practical perception and interpretation of the environment and user instructions.
- **Adequate Network Connectivity:** The system assumes a stable network connection to facilitate real-time communication and data exchange between the robotic system and other components, such as cloud-based services or remote human operators.
- **Well-Defined Language Instructions:** The system assumes that users provide unambiguous natural language instructions that the robotic system can easily interpret.

##### **Dependencies:**

- **Robust Natural Language Processing (NLP) Techniques:** The successful interpretation of natural language instructions depends on the availability of advanced NLP techniques, including semantic parsing, language understanding, and dialogue management, to accurately extract meaning and intent from user instructions.
- **High-Quality Vision Processing:** The system relies on sophisticated computer vision algorithms and technologies to process visual inputs and extract relevant environmental information, enabling effective navigation and perception.
- **Development of Reliable Navigation Algorithms:** The successful navigation of robots in various environments depends on the availability of robust navigation algorithms, obstacle avoidance techniques, and path-planning strategies tailored to the specific application domain.
- **Access to Comprehensive and Representative Training Data:** The development and training of the proposed system require access to diverse and comprehensive training data, including annotated language instructions, corresponding visual data, and navigation trajectories, to ensure effective learning and generalization in unseen environments.
- **Availability of Adequate Hardware and Computing Resources:** The system's performance may depend on the availability of suitable hardware, such as powerful processors, memory, and storage capabilities, as well as efficient computing resources for training and inference tasks.

#### **4.2.1.5. Domain Requirements**

- **Robust Perception:** The system should be able to perceive and interpret visual and auditory inputs accurately, allowing for a practical understanding of the environment and user instructions.
- **Natural Language Understanding:** The system should have advanced natural language processing techniques to comprehend and interpret user instructions in various linguistic contexts, considering nuances, semantics, and intent.
- **Adaptive Learning and Generalization:** The system should possess the capability to learn from interactions and adapt to different environments, enabling generalization to unseen scenarios and improving performance over time.
- **Efficient Navigation and Path Planning:** The system should be able to navigate through complex environments, plan optimal paths, and avoid obstacles in real time, ensuring safe and efficient movement.

#### **4.2.1.6. User Requirements**

- **Intuitive Interaction:** Users should be able to communicate with the system using natural language instructions, enabling seamless and intuitive interaction without the need for complex command structures or technical knowledge.
- **Accuracy and Reliability:** Users expect the system to accurately interpret their instructions and reliably perform the requested tasks, ensuring high performance and minimizing errors.
- **Personalization and Customization:** The system should be able to adapt to individual user preferences and provide personalized assistance, tailoring its responses and actions to the specific needs and requirements of each user.
- **User-Friendly Interface:** The system should have a user-friendly interface that is easy to understand and navigate, allowing users to provide instructions, monitor system status, and access relevant information effortlessly.
- **Real-Time Responsiveness:** Users expect the system to respond promptly and provide real-time feedback during interactions, creating a seamless and natural conversational experience.
- **Privacy and Security:** Users require assurance that their personal information and interactions with the system are protected and handled securely, ensuring privacy and data confidentiality.

## **4.2.2. Non-Functional Requirements**

### **4.2.2.1. Product Requirements**

#### **4.2.2.1.1. Efficiency (in terms of Time and Space)**

- **Efficiency: Schedule Efficiency:** The system should be tuned to minimize response and processing times, ensuring that activities and instructions are completed quickly and on time.
- **Space Efficiency:** The system should use computational and memory resources wisely, lowering resource usage while maintaining performance.

#### **4.2.2.1.2. Reliability**

- **Stability and robustness:** The system should be able to deal with a wide range of circumstances and environmental conditions without degrading performance or creating system breakdowns.
- **Fault Tolerance:** The system should be able to tolerate failures, errors, or disruptions, and methods for recovering from defects and keeping the system running reliably should be in place.

#### **4.2.2.1.3. Portability**

- **Platform Independence:** The system should be designed to be platform-independent, allowing for easy deployment on a wide range of hardware combinations and operating systems.
- **Compatibility:** To allow for easy integration and interoperability, the system must be interoperable with existing robotic platforms and frameworks.

#### **4.2.2.1.4. Usability**

- **The system's interface** should be user-friendly, with clear instructions, intuitive controls, and informative feedback, allowing users to interact quickly and efficiently.
- **Learnability:** The system should be simple to learn and use, requiring little training or technical knowledge for users to take advantage of its features.

#### **4.2.2.2. Organizational Requirements**

##### **4.2.2.2.1. Implementation Requirements**

Scalability:

- The system should be designed to handle fluctuating workloads and user demand, enabling effective scaling and deployment across various environments and usage scenarios.
- To allow seamless deployment and integration into current IT ecosystems, the system should be compatible with popular deployment platforms and infrastructure, such as cloud services, edge devices, or on-premises servers.

Resource Optimization:

- The system should be optimized to make the best use of computational resources while reducing hardware requirements and optimizing performance for cost-effective deployment.

Real-Time limitations:

- If real-time responsiveness is required, the system must adhere to time limitations in order to respond to user commands and environmental changes in a timely and correct manner.

##### **4.2.2.2.2. Engineering Standard Requirements**

- Code Documentation: The system's codebase should be well-documented, adhering to accepted documentation practices, allowing developers and maintainers to easily understand and modify the system.
- Modularity and maintainability: The system should be built with a modular architecture to make maintenance, updates, and problem patches easier.
- Version Control: Version control systems should be used to manage the system's source code, allowing developers to collaborate, track changes, and manage codebases.

##### **4.2.2.3. Operational**

- Economic: The proposed vision-and-language navigation system should be economically viable and provide value to stakeholders. By improving navigation efficiency and reducing human intervention, the system can save costs associated with manual navigation and increase productivity in various industries such as robotics,

automation, and logistics.

- **Environmental:** The system's ability to navigate autonomously and efficiently can contribute to reducing energy consumption and carbon emissions. By optimizing routes and minimizing unnecessary movements, the system can help conserve energy and promote environmental sustainability.
- **Social:** The vision-and-language navigation system can have significant social implications by enhancing human-robot interaction and enabling robots to understand and follow natural language instructions. In addition, it can facilitate communication and collaboration between humans and robots, opening up possibilities for assistance in various domains such as healthcare, hospitality, and home automation.
- **Political:** The political implications of the system lie in its potential impact on policies and regulations regarding robotics, artificial intelligence, and automation. For example, as technology advances, policymakers may need to address liability, privacy, and ethical considerations associated with autonomous robotic systems.
- **Ethical:** Ethical considerations are crucial in developing and deploying the vision-and-language navigation system. Ensuring the system respects user privacy, maintains data confidentiality, and avoids discriminatory behaviors are essential ethical requirements. Additionally, designing the system to prioritize human safety and well-being is paramount.
- **Health and Safety:** The system should adhere to health and safety regulations to prevent harm to users, operators, and the environment. It should incorporate mechanisms to avoid collisions, operate safely, and handle emergencies effectively. Robust fail-safe mechanisms and regular maintenance practices should be implemented to ensure the system's safe operation.
- **Sustainability:** By optimizing navigation routes and reducing resource consumption, the system can contribute to sustainability efforts. It can minimize unnecessary travel, optimize energy usage, and reduce the overall environmental impact, aligning with the goal of sustainable development.
- **Legality:** The system should comply with relevant laws and regulations governing robotics, artificial intelligence, data protection, and privacy. In addition, it should ensure proper consent, lawful data collection and usage, and adherence to intellectual property rights.
- **Inspectability:** The system should be designed to allow for inspectability and

transparency. This includes providing access to system logs, algorithms, and data for decision-making, enabling audits, and ensuring accountability.

### **4.2.3. System Requirements**

#### **4.2.3.1. H/W Requirements:**

- Apple silicon M2
- Intel i7
- NVIDIA T4 GPU X 1
- 32 CPUs
- 250 GB RAM
- 2TB Disk

#### **4.2.3.2. S/W Requirements:**

- VSCODE
- Ubuntu 18.04
- Mac os x/windows 11
- Python 3.6+
- PyTorch, Torch vision, Pandas, PIL, Transformers, pickle, NumPy, matter sim, TQDM, PyTorch lightning, datasets.

## 5. Results and Discussion

Table 3: R2R Validation Seen

Methods	TL	NE	SR
Random	9.58	9.45	16
Seq2Seq	11.33	6.01	39
SF	-	3.36	66
PRESS	10.57	4.39	58
EnvDrop	11.00	3.99	62
AuxRN	-	3.33	70
PREVALENT	10.32	3.67	69
RelGraph	10.13	3.47	67
RecBERT	11.13	2.90	72
HCAM-HF (ours)	13.05	4.84	52

Table 4: R2R Validation Unseen

Methods	TL	NE	SR
Random	9.77	9.23	16
Seq2Seq	8.39	7.81	22
SF	-	6.62	35
PRESS	10.36	5.28	49
EnvDrop	10.70	5.22	52
AuxRN	-	5.28	55
PREVALENT	10.19	4.71	58
RelGraph	9.99	4.73	57
RecBERT	12.01	3.93	63
HCAM-HF (ours)	18.52	5.86	42



Fig 14: Rewards vs steps avg over episodes and multiple runs



Fig15: Rewards vs steps avg over episodes and multiple runs

We used three metrics to evaluate the model Success rate measures the percentage of navigation tasks in which the agent successfully reaches the target location specified by the given natural language instruction. Navigation error quantifies the discrepancy between the agent's final location and the target location. Trajectory length is a valuable metric as it provides insights into the efficiency and optimality of the agent's navigation. We use the Mattersim simulator for an environment with 1.2 TB of data. The dataset comprises 194,400 RGB-D images of 90 building-scale scenes with 21,567 navigation instructions. After preprocessing the data, it is randomly shuffled and seed put in. Then, the reward model is trained using the output of the primary model. Finally, they are compared in pairs. In every cycle, four things happen: two epochs of the value and action network and two epochs for the reward model, a simple MLP concatenated with image and language.

The model scores around TL - 13.05, NE - 4.84, and SR - 52 for R2R validation-seen. It has surpassed two models and was very close to the PRESS model in terms of success rate. The navigation error is 4.84, which indicates the agent has improved in long-term dependencies. As one can see, the trajectory length is much more than NE. They usually will be near each other, which indicates the agent is exploring more and still getting near under 3 m of goal.



The model did not perform better than the state of the art as the middle model network needs to be more significant to associate those many relations. Finally, we still need to address the Generalize to unseen environments, as the metrics for R2R Validation Unseen are TL - 18.52, NE - 5.86 and SR - 42. To evaluate our approach, we choose the PyTorch framework to train the Model:

1. We ran a hyperparameter search on the learning rate, revealing 0.0004 to be suitable.
2. We analyzed batch sizes starting from 10 to 16 and found that size 10 gives the best performance, whereas the batch size of 16 resulted in complete hardware capacity utilization.
3. The best-performing version used 64 as batch size and  $2 \times (0.0004)$  as the learning rate.

## 6. REFERENCES

- [1] Chen, S., Guhur, P. L., Schmid, C., & Laptev, I. (2021). History aware multimodal transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 34, 5834-5847.
- [2] Moudgil, A., Majumdar, A., Agrawal, H., Lee, S., & Batra, D. (2021). Soat: A scene-and object-aware transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 34, 7357-7367.
- [3] J. Y. Koh, H. Lee, Y. Yang, J. Baldridge and P. Anderson, "Pathdreamer: A World Model for Indoor Navigation," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021, pp. 14718-14728, doi: 10.1109/ICCV48922.2021.01447.
- [4] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo and S. Gould, "VLN $\cup$ BERT: A Recurrent Vision-and-Language BERT for Navigation," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 1643-1653, doi: 10.1109/CVPR46437.2021.00169.
- [5] M. Z. Irshad, C. -Y. Ma and Z. Kira, "Hierarchical Cross-Modal Agent for Robotics Vision-and-Language Navigation," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021, pp. 13238-13246, doi: 10.1109/ICRA48506.2021.9561806.

- [6] Majumdar, A., Shrivastava, A., Lee, S., Anderson, P., Parikh, D., & Batra, D. (2020, August). Improving vision-and-language navigation with image-text pairs from the web. In *European Conference on Computer Vision* (pp. 259-274). Springer, Cham.
- [7] W. Hao, C. Li, X. Li, L. Carin and J. Gao, "Towards Learning a Generic Agent for Vision-and-Language Navigation via Pre-Training," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 13134-13143, doi: 10.1109/CVPR42600.2020.01315.
- [8] Landi, F., Baraldi, L., Cornia, M., Corsini, M., & Cucchiara, R. (2021). Multimodal attention networks for low-level vision-and-language navigation. *Computer Vision and Image Understanding*, 210, 103255.
- [9] F. Zhu, Y. Zhu, X. Chang and X. Liang, "Vision-Language Navigation With Self-Supervised Auxiliary Reasoning Tasks," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 10009-10019, doi: 10.1109/CVPR42600.2020.01003.
- [10] X. Wang et al., "Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 6622-6631, doi: 10.1109/CVPR.2019.00679.
- [11] Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L. P., & Darrell, T. (2018). Speaker-follower models for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 31.
- [12] P. Anderson et al., "Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 3674-3683, doi: 10.1109/CVPR.2018.00387.