ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Schweizer
Paraplegiker
Forschung

Schweizer
Paraplegiker
Zentrum

LAB

Semester Thesis

Studying VLN and The Extension to Crowd-Aware Navigation

Spring Term 2025

Supervised by:
Rashid Alyassi
Diego Paez-Granados
Robert Riener

Author:
Romain Guntz

Declaration of Originality

I hereby declare that the written work I have submitted entitled

Studying VLN and The Extension to Crowd-Aware Navigation

is original work which I alone have authored and which is written in my own words.[1]

Author(s)

Romain	Guntz

Supervisor(s)

Rashid	Alyassi

Supervising lecturer

Robert	Riener

With the signature I declare that I have been informed regarding normal academic citation rules and that I have read and understood the information on 'Citation etiquette' (https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/plagiarism-citationetiquette.pdf). The citation conventions usual to the discipline in question here have been respected.

The above written work may be tested electronically for plagiarism.

Zurich, 10 September 2025
Place and date

Signature

[1] Co-authored work: The signatures of all authors are required. Each signature attests to the originality of the entire piece of written work in its final form.

Intellectual Property Agreement

The student acted under the supervision of Dr. Paez-Granados and contributed to research of his group. Research results of students outside the scope of an employment contract with ETH Zurich belong to the students themselves. The results of the student within the present thesis shall be exploited by ETH Zurich, possibly together with results of other contributors in the same field. To facilitate and to enable a common exploitation of all combined research results, the student hereby assigns his rights to the research results to ETH Zurich. In exchange, the student shall be treated like an employee of ETH Zurich with respect to any income generated due to the research results.
This agreement regulates the rights to the created research results.

1. Intellectual Property Rights

1. The student assigns his/her rights to the research results, including inventions and works protected by copyright, but not including his moral rights ("Urheberpersönlichkeitsrechte"), to ETH Zurich. Herewith, he cedes, in particular, all rights for commercial exploitations of research results to ETH Zurich. He is doing this voluntarily and with full awareness, in order to facilitate the commercial exploitation of the created Research Results. The student's moral rights ("Urheberpersönlichkeitsrechte") shall not be affected by this assignment.
2. In exchange, the student will be compensated by ETH Zurich in the case of income through the commercial exploitation of research results. Compensation will be made as if the student was an employee of ETH Zurich and according to the guidelines "Richtlinien für die wirtschaftliche Verwertung von Forschungsergebnissen der ETH Zürich".
3. The student agrees to keep all research results confidential. This obligation to confidentiality shall persist until he or she is informed by ETH Zurich that the intellectual property rights to the research results have been protected through patent applications or other adequate measures or that no protection is sought, but not longer than 12 months after the collaborator has signed this agreement.
4. If a patent application is filed for an invention based on the research results, the student will duly provide all necessary signatures. He/she also agrees to be available whenever his aid is necessary in the course of the patent application process, e.g. to respond to questions of patent examiners or the like.

2. Settlement of Disagreements

Should disagreements arise out between the parties, the parties will make an effort to settle them between them in good faith. In case of failure of these agreements, Swiss Law shall be applied and the Courts of Zurich shall have exclusive jurisdiction.

Zurich, 10 September 2025
Place and date

Signature

Contents

Preface

Vision–language navigation (VLN) seeks to control mobile robots directly from natural-language instructions. Beyond its scientific appeal, VLN can broaden access to assistive robotics—for example, enabling personal mobility devices to be steered hands-free by users with limited arm mobility. However, most VLN systems are designed for static, map-like settings and fail to cope with the uncertainty and social norms of human-filled spaces. This gap motivates our focus on crowd-aware VLN: interpreting language while forecasting, respecting, and negotiating human motion to achieve safe, socially compliant navigation.

This project was carried out as part of my semester project. I would like to sincerely thank my supervisor, Rashid Alyassi, for his guidance throughout the project and for teaching me rigorous literature review and research practices. I also thank him for introducing me to the world of vision–language navigation research, in which I now have a solid understanding, and for showing me a field that is both fascinating and highly promising for the future. I am also grateful to the SCAI Lab for providing the infrastructure, including their RTX 4090 GPU, which greatly facilitated the training of the models.

Abstract

Vision-Language Navigation (VLN) aims to enable autonomous agents to follow natural language instructions within visually perceived environments. While significant progress has been made, most state-of-the-art VLN systems rely on unrealistic simulation assumptions — such as panoramic camera inputs and predefined navigation graphs — creating a substantial gap between simulated performance and real-world applicability. Furthermore, these systems are typically trained and evaluated in static, human-free environments, rendering them ill-equipped for dynamic and socially complex spaces.

This semester project addresses these critical limitations by focusing on two key areas: monocular VLN in continuous environments and the extension to crowd-aware navigation. First, we implement and evaluate a state-of-the-art monocular VLN model that uses monocular waypoint prediction, 3D feature fields to hallucinate panoramic views, and a topological map for navigation, achieving competitive results on the R2R-CE benchmark. To further improve performance under partial observability, we design a novel recurrent waypoint prediction module that fuses metric map information with current visual observations and language instructions. Separately, we identify a key bottleneck in standard Imitation Learning — namely, its reliance on shortest-path teacher actions, which discourages necessary exploration and recovery behaviors in continuous, partially observable settings. To address this, we lay the groundwork to incorporate Reinforcement Learning as a complementary training paradigm. Finally, we propose extending training into human-populated simulations using platforms like Habitat, to enable the emergence of socially compliant navigation behaviours. Together, these contributions lay the foundation for VLN agents that operate safely, effectively, and with social awareness in real-world environments.

Chapter 1

Introduction

Vision-Language Navigation (VLN) is a rapidly growing field at the intersection of robotics, computer vision, and natural language processing. The goal of VLN is to enable autonomous agents to follow natural language instructions in visually perceived and previously unseen environments.

A central challenge in Vision-Language Navigation (VLN) lies in effectively grounding language in visual perception, requiring the alignment of two very different modalities—text and vision—so that the agent can correctly interpret instructions in a dynamic environment. Many existing approaches rely on panoramic visual inputs [1, 2, 3], which are impractical for real-world robots due to their size, cost, and complexity, highlighting the need for VLN agents that can operate with a standard frontal camera. Additionally, the highest-performing methods in VLN often leverage graph-based navigation with access to predefined adjacent nodes [4, 5, 6], which increases the gap between simulation and reality, therefore necessitating agents that can navigate in continuous environments without access to any predefined structure of the environment.

Although Vision-Language Navigation (VLN) enables robots to follow natural language instructions and reach target locations, successful deployment in real-world environments requires more than just geometric path planning. First, the model must remain robust in the presence of humans, ensuring accurate navigation decisions even when people are nearby. Second, the agent needs a dedicated module that allows it to move safely and effectively within human-populated environments, ensuring socially compliant trajectories that respect personal space, avoid collisions, and adhere to human social norms of movement.

In this work, our main contributions include:

- Identifying the bottlenecks of state-of-the-art VLN models.
- Implementing existing solutions for solving VLN tasks with monocular camera and waypoint prediction in continuous environments.
- Designing a Monocular VLN agent.
- Designing a VLN training with crowds.

1.1 Background

Vision-Language Navigation (VLN) is the task of enabling an agent to follow natural language instructions within a visual environment. Given a high-level instruction, such as “go to the coffee table”, the agent must interpret the language in the context of its visual observations and navigate towards the target location.

Dataset
Training VLN agents requires datasets that provide paired natural language instructions and ground truth trajectories. Several datasets have been introduced to support different VLN scenarios. R2R [7] is an indoor navigation dataset containing instruction-trajectory pairs. RxR [8] extends R2R with multilingual instructions, dense spatiotemporal alignments between text and visuals, a larger number of samples, and longer instructions. Reverie [9] focuses on localizing a remote target object specified in the instruction. ALFRED [10] is a household task execution dataset where agents follow natural language instructions to complete multi-step tasks. HANNA [11] addresses interactive object-finding with human assistance, and Touchdown [12] provides data for outdoor navigation. These datasets collectively enable training VLN agents in diverse environments, tasks, and instruction complexities.

The R2R dataset [7] was selected for this work because social navigation typically involves short, high-level instructions, and R2R provides a training set that closely matches this task formulation. R2R leverages the Matterport3D Simulator, which is based on the Matterport3D dataset—a collection of 10,800 panoramic RGB-D images from 90 real-world indoor environments. The dataset contains 21,567 instructions with an average length of 29 words, each paired with a trajectory of 4–6 edges. Trajectories are created by sampling a starting and ending point, computing the shortest path between them, and annotating it with human instructions.

The R2R-CE [13] dataset is an extension of R2R [7] for continuous environments. It retains the same instruction–trajectory pairs as R2R but places agents in continuous 3D spaces where they must execute low-level actions to follow natural language instructions. This removes the assumption of known topology of the environment and provides a more realistic and challenging setting for studying navigation in unknown environments.

Evaluations Metrics
Several metrics are commonly used to evaluate VLN agents. **Success Rate (SR)** measures the percentage of episodes in which the agent reaches the target location within a specified distance, typically 3 meters. **Success weighted by Path Length (SPL)** adjusts SR by the efficiency of the agent's path, taking the ratio of the shortest path to the agent's traveled path. **Navigation Error (NE)** is the average Euclidean distance between the agent's final position and the target. **Trajectory Length (TL)** reports the average length of the paths taken by the agent. **Oracle Success Rate (OSR)** indicates the proportion of episodes where the agent comes within the distance threshold of the target at any point along its trajectory. Finally, **Normalized Dynamic Time Warping (nDTW)** measures the step-wise alignment between the agent's path and the reference path, capturing how closely the agent follows the intended trajectory.

1.2 Related Work

Early Models Early VLN models relied on sequence-to-sequence LSTMs with attention mechanisms to align natural language instructions with visual observations [7, 14]. Later works moved away from RNN-based fixed-size state representations, which struggled to preserve long-term information in extended trajectories, and instead relied on Transformer-based models for text encoding. In [1], The authors introduce a hierarchical history encoder that uses a ViT and a panoramic Transformer to capture spatial relationships within panoramas, followed by a temporal transformer to model dependencies across panoramas. In Recurrent VLN-BERT[15], the paper introduces a recurrent function into the V&L BERT architecture by maintaining and iteratively updating a dedicated "state token" through self-attention with new observations at each time step. Further work [2] encode the trajectory history by processing a sequence of front-view images through a Transformer-based Vision Encoder with self-attention, enabling the model to capture temporal order before fusing it with language via a cross-modal encoder. While these models rely on sequences of images, effective planning requires incorporating memory into a map structure to provide geometrically and spatially informed trajectory representations.

Topological Maps

Topological Maps [16, 17] describe the environment using a graph of nodes and edges. Let the graph-based topological map at step t be $G_t = \{V_t, E_t\}$, which keeps track of all observed nodes along the agent's trajectory Γ'. The nodes V_t are divided into three categories: visited nodes, navigable (unexplored) nodes, and the current node [4, 18, 3]. At each step, the agent adds the current node and its neighboring unvisited nodes to V_{t-1}, updates the edges E_{t-1} accordingly, and refines the visual representations of observed and navigable nodes based on new observations. Topological maps allow long-term planning and backtracking but lack a fine-grained description of the environment, as their representation is a condensed feature vector of the locations.

Metric Maps

Metric Maps are egocentric grids M_t with D-dimensional features in each cell representing the local environment for short-term planning. Early metric maps [19] proposed deep learning architecture that enabled an agent to dynamically build a map of its environment and localize itself within it by using efficient convolution/deconvolution operations on RGBD input. In [20], the authors process RGB images and noisy sensor readings to predict an egocentric top-down obstacle/explored area map, estimates the agent's pose by aligning consecutive egocentric predictions, and then transforms and aggregates this into a global, geocentric 2D spatial map. Later methods [4, 21, 22] rely on depth images to project visual features onto a grid map.

Pre-Training

The Masked Language Modeling (MLM) task in VLN is applied on the outputs of a cross-modal encoder that fuses text and visual features. Each masked word w_m is predicted based on the contextualized representations of the remaining words $w_{\backslash m}$ and the visual trajectory τ:

$$L_{\mathrm{MLM}} = -\mathbb{E}_{(w,\tau)\sim D} \log P_\theta(w_m \mid w_{\backslash m}, \tau) \tag{1.1}$$

The Hybrid Single Action Prediction (HSAP) task predicts the next navigation action given the agent's current state and any additional input modalities, such as a topological map G_t or a metric map M_t. In the pretraining task of [4], these maps are taken from offline expert trajectories and serve as static inputs. The network is trained via cross-entropy with the expert action a_t^*:

$$L_{HSAP} = -\log P_\theta(a_t^* \mid W, G_t, M_t), \tag{1.2}$$

where W represents the instruction input, and G_t and M_t are given as examples of possible map modalities.

The Masked Semantic Imagination (MSI) task predicts the semantics of masked cells. During pretraining, a random subset of cells (15%) is masked to simulate unobserved regions. The model first performs a cross-modal interaction between the instruction W and the partially observed map $M_{t,\setminus m}$, and then predicts the semantic classes S of the masked cells using a multi-label binary cross-entropy loss [4]:

$$\begin{aligned} L_{MSI} = -\mathbb{E}_{(W,\Gamma)\sim D} \sum_{i=1}^{C} \Big[& S_i \log P_\theta(S_i \mid W, M_{t,\setminus m}) \\ & + (1 - S_i) \log(1 - P_\theta(S_i \mid W, M_{t,\setminus m})) \Big] \end{aligned} \tag{1.3}$$

where S is the ground truth semantic label.

Data Augmentation

Data augmentation strategies have been introduced to address the limited diversity of environments and the small size of datasets. Early works [23] relied on a simple LSTM architecture to generate instructions for new trajectories. Following this, [24] the authors consistently mask visual features across all viewpoints within an environment to simulate diverse, unseen scenes. In [25], the authors follow the idea of [24] to generate 4.9M instruction-trajectories pairs to achieve state of the art results on pre-existing models. AirBERT [26] uses in-domain Airbnb data for VLN while [27] pretrains a VLN agent on path-instruction pairs automatically mined from YouTube house tours. Recent work [28] leverages foundation models to generate diverse observation-instruction pairs.

Waypoint Prediction

Recent approaches have focused on using topological maps [18, 29], metric maps [30, 31], or a combination of both [4], combined with pre-training tasks to improve representation learning. A limitation of these methods is that they typically rely on access to adjacent nodes in the map. To address this, alternative models have been proposed that predict surrounding waypoints. In [32], the authors propose a language-conditioned waypoint prediction network that uses GRUs to predict relative polar waypoints from panoramic observations and instructions. Following this, Wang et al. [22] and An et al. [3] build on the approach of Hong et al. [33], using candidate waypoint predictors that leverage Transformer networks over panoramic RGB-D observations to generate heatmaps of accessible navigable positions. However, these models depend on panoramic views for both navigation and waypoint prediction, limiting their applicability in real-world scenarios. This motivates the development of a waypoint predictor and a navigation module that rely exclusively on frontal camera input.

Monocular Waypoint Predictor

In the monocular camera setting, the absence of surrounding views makes it more difficult to predict a heatmap of navigable areas around the agent. For this reason, using a metric map to predict the heatmap is more suitable. In [34], the authors build on the model proposed in [35] and introduce the following architecture. A semantic map is first extracted from the RGB input, which updates the global semantic map and hallucinates unseen regions using a U-Net (see Figure 1.1). Next, an occupancy map derived from the depth input updates the global occupancy map and is similarly processed through a U-Net to hallucinate unseen regions. Finally, both maps are fused, and a final U-Net predicts the navigable heatmaps around the agent.

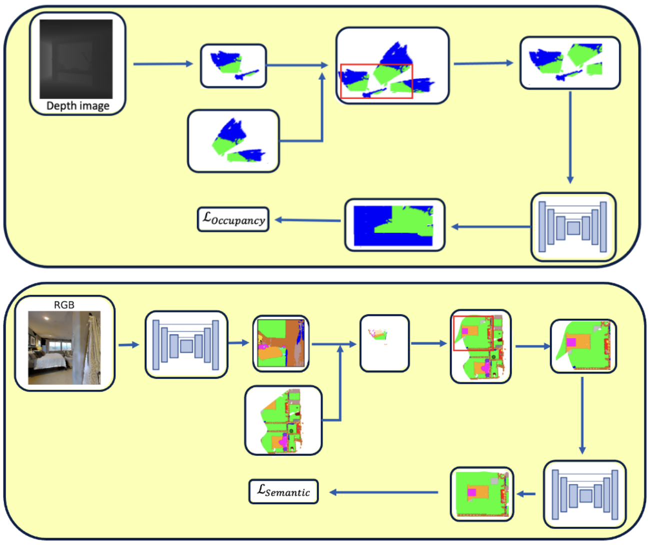

Figure 1.1: Top: Depth input to occupancy map processed through a U-Net to hallucinate unseen regions. Bottom: RGB input to semantic map processed through a U-Net to hallucinate unseen regions. Both maps are then fused to predict navigable areas [34].

3D Feature Fields

In [34], the authors build upon the ideas of [36] to generate panoramic views from monocular input. Monocular RGB images are first processed by a pre-trained CLIP-ViT-B/16 [37] to extract visual features, which are then projected into 3D space using depth maps and camera parameters. For each predicted novel view, an 8×8 feature map R is generated, where each of the 64 subregions corresponds to a portion of the view. To compute the feature for each subregion, rays are sampled from the camera center to the subregion center, and volume densities and latent vectors are predicted and composited via volume rendering. A Transformer-based decoder processes the 64 subregion features in R and integrates them into a single embedding $V \in \mathbb{R}^{1\times512}$, aligned with the CLIP representation. This allows the VLN agent to perceive multiple viewpoints per panorama, combining the forward-facing RGB observation with 11 predicted novel views for navigation.

Monocular Navigation

The authors of [34] explore monocular VLN, and this paper is the one most studied in this project. Their approach first predicts waypoints from monocular input (see Figure 1.1) and then employs a topological map, following [4, 3, 18], where 3D feature fields are used for panoramic reconstruction and navigable node embedding.

Chapter 2

Methodology

2.1 Unrealistic Simulation Assumptions

Most existing VLN models rely on assumptions that limit their applicability in real-world scenarios. Many approaches assume access to neighboring nodes in a predefined navigation graph [4, 18, 5], which is not available outside simulation environments. Among models that predict waypoints around the agent, the majority use panoramic input [3, 22], achieving high performance in simulation but generalizing poorly to real-world settings, where the agent typically has access only to a single frontal RGB-D camera.

In contrast, our work adopts a more realistic formulation: the agent is equipped with a single monocular RGB camera and operates in a continuous environment without a predefined graph structure. The objective is to learn a navigation policy

$$a_t = \pi(o_t, I; \theta),$$

where o_t denotes the visual observation from the monocular camera at time t, I represents the language instruction, and θ are the learnable parameters of the policy.

2.2 Model Architecture

Two papers have been particularly influential in shaping the design of the model architecture. The first, [34], introduces a framework that simulates panoramic view using 3D feature fields. The second, [38], proposes a novel waypoint prediction method that integrates language instructions with metric maps to predict surrounding waypoints, achieving state-of-the-art performance with monocular inputs. In the proposed design, elements from both approaches are fused into a unified pipeline. Given an instruction and monocular RGB-D input, the model first predicts a set of waypoints. 3D feature fields are then applied to encode the information of each node, including out-of-view nodes. Finally, following [3], a Cross-Modal Graph Transformer is employed to predict the long-term navigation goal (see Figure 2.1).

Waypoint Prediction Module

The waypoint prediction module is inspired by [38] (see Figure 2.2). Let the natural language instruction consist of K words. We encode the instruction using the pre-trained BERT model [39], producing a sequence of embeddings:

$$I = \{\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_K\}, \quad \mathbf{l}_i \in \mathbb{R}^{d_l}, \tag{2.1}$$

where $\mathbf{l}_i$ denotes the embedding of the i-th word and d_l is the dimensionality of the word embedding.
The RGB input $\mathbf{r}_t \in \mathbb{R}^{H_0 \times W_0 \times 3}$ is processed using LSeg [40] to obtain a pixel-wise embedding map $\mathbf{E}_t \in \mathbb{R}^{H_0 \times W_0 \times D}$, where D is the embedding dimension. This embedding map is combined with the depth map $\mathbf{d}_t \in \mathbb{R}^{H_0 \times W_0}$ to update the global metric map $\mathbf{M}_t \in \mathbb{R}^{H \times W \times D}$.
To fuse instruction and map information, we employ a Cross-Modal Attention mechanism [41, 42], where the metric map features act as queries and the instruction embeddings serve as keys and values:

$$\mathbf{M}'_t = \mathbf{M}_t + \text{Attention}(\mathbf{Q} = \mathbf{M}_t, \mathbf{K} = I, \mathbf{V} = I), \tag{2.2}$$

where $\mathbf{M}'_t \in \mathbb{R}^{H \times W \times D}$ is the instruction-informed map. The attention operation is the standard scaled dot-product attention [42]:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{2.3}$$

where d_k is the dimensionality of the key vectors. This allows each map location to focus on instruction-relevant content while retaining its spatial and semantic structure.
Next, a region-based pooling operation is applied to $\mathbf{M}'_t$, yielding a pooled feature vector $\tilde{\mathbf{M}}_t \in \mathbb{R}^{D'}$ that emphasizes spatial regions most relevant to the current instruction. Simultaneously, the RGB and depth images are encoded using the CLIP model [37] to obtain visual embeddings $\mathbf{v}_t^r$ and $\mathbf{v}_t^d$, as illustrated in Figure 2.2.
Finally, the LSTM network [43] integrates sequential information over time. The input to the LSTM at time t is the concatenation of the pooled map features, visual embeddings, and the previous hidden state:

$$\mathbf{h}_t = \text{LSTM}\left(\left[\tilde{\mathbf{M}}_t, \mathbf{v}_t^r, \mathbf{v}_t^d\right], \mathbf{h}_{t-1}\right), \tag{2.4}$$

and the predicted waypoint $\mathbf{y}_t$ is obtained via a linear projection:

$$\mathbf{y}_t = \mathbf{W}_a \mathbf{h}_t + \mathbf{b}_a. \tag{2.5}$$

The LSTM serves as a memory for the agent, maintaining information about previous observations, actions, and partially completed instructions. This allows the agent to predict the next waypoint in a sequential, context-aware manner, integrating current instruction-relevant features with the memory of previous steps.

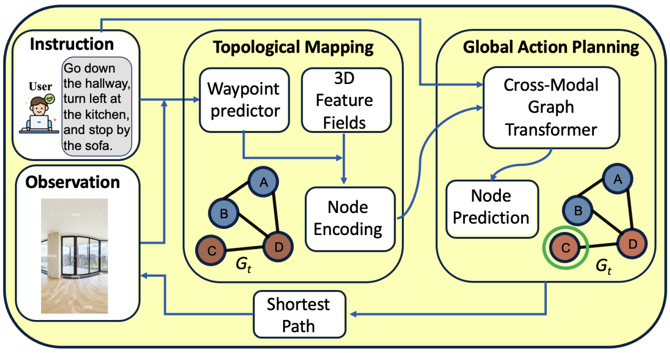

Figure 2.1: Our Model Architecture for VLN

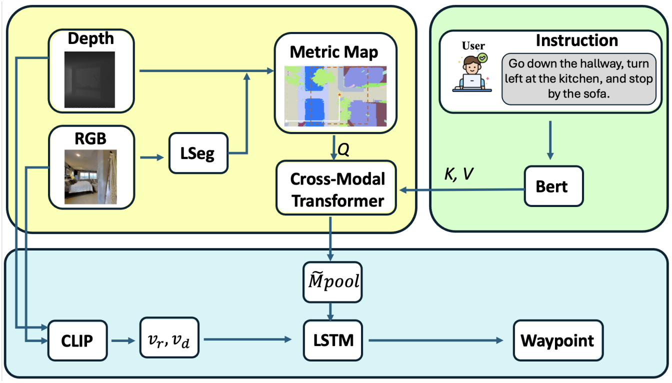

Figure 2.2: Our Waypoint Prediction Module [38]

Waypoint and Panoramic View Encoding

Given the predicted waypoint, the next step is to encode both the waypoint and the agent's current panoramic view (i.e., the current node) to update the topological map of the environment. For the predicted waypoint, we extract features from 12 novel views generated by the 3D feature fields. For the current panoramic node, the forward-facing view is encoded using the CLIP model [37] applied to the monocular RGB image, while the remaining 11 novel views are obtained from the 3D feature fields.

Global Action Planning

To plan actions over the topological map, as in [34], the approach follows a technique inspired by [29] and integrates it into the pipeline. The encoded node features, together with the instruction embeddings, are processed by a multi-layer Graph-Aware Cross-Modal Transformer. Each layer of this Transformer consists of two main components: (i) a Cross-Attention mechanism to model interactions between instruction embeddings and node representations, and (ii) a Graph-Aware Self-Attention mechanism (GASA) that incorporates the structure of the topological map into the attention computation.

The Cross-Modal Attention between nodes and instructions is formally defined as:

$$\mathbf{n}'_i = \mathbf{n}_i + \text{Attention}(\mathbf{Q} = \mathbf{n}_i, \mathbf{K} = I, \mathbf{V} = I), \tag{2.6}$$

where $\mathbf{n}_i$ is the node embedding acting as the query, $I = \{\mathbf{l}_1, \ldots, \mathbf{l}_K\}$ are the instruction embeddings used as keys and values, and the output $\mathbf{n}'_i$ preserves the original node information via the residual connection. W_Q, W_K, W_V denote the learnable projection matrices within the attention mechanism.

Standard self-attention considers only feature similarity among nodes, which can lead to underestimating the relevance of nearby nodes in the graph. To address this, the Graph-Aware Self-Attention incorporates a pairwise distance matrix between nodes, effectively biasing the attention toward nodes that are closer in the topological layout. Formally, the Graph-Aware Self-Attention is computed as:

$$\text{GASA}(\mathbf{X}) = \text{Softmax}\left(\frac{\mathbf{X}W_Q(\mathbf{X}W_K)^\top}{\sqrt{d}} + \mathbf{M}\right)\mathbf{X}W_V, \tag{2.7}$$

where $\mathbf{X}$ denotes the node embeddings, and $\mathbf{M}$ encodes pairwise distances in the graph:

$$\mathbf{M} = \mathbf{E}W_e + b_e, \tag{2.8}$$

with $\mathbf{E}$ being the pairwise distance matrix and W_e, b_e learnable parameters. The outputs of the stacked transformer layers are node embeddings $\hat{\mathbf{n}}_i$ that incorporate both instruction context and topological structure.
Finally, a Feed-Forward Network predicts a navigation score for each node in the current graph:

$$s_i = \mathrm{FFN}(\hat{\mathbf{n}}_i), \tag{2.9}$$

where s_0 represents the stop score.

2.3 Training Process

Current VLN training exhibits limitations in both Imitation Learning and crowd-aware navigation. To address these bottlenecks, this work introduces two complementary solutions: Reinforcement Learning and training in human-populated simulations.

Supervision via Teacher Actions in VLN
In most VLN frameworks, training relies on querying the *teacher action* from the dataset. The teacher action is defined as the step that leads along the shortest path from the agent's current node to the goal node.
This strategy works well in the standard R2R setting, where the ground-truth trajectory is indeed the shortest path between the start and goal nodes. As long as the agent remains on this path, the teacher action provides an accurate supervision signal.
However, two important limitations arise:

1. **Off-Trajectory Behavior:** When the agent deviates from the ground-truth trajectory, the shortest-path teacher action is no longer aligned with the behavior required to follow the instruction. In such cases, the agent may need to backtrack or deliberately explore in order to re-align with the instruction, rather than immediately following a new shortest path to the goal.

2. **Monocular vs. Panoramic Settings:** Even when the agent is on the ground-truth trajectory, differences arise between panoramic and monocular observation spaces. In panoramic settings, the teacher action remains valid because the agent has full 360° information and can confidently commit to that move. In monocular settings, however, the agent only perceives a limited forward-facing view. Here, it may need to *explore first* (e.g., by rotating or scanning the environment) before deciding on a navigation step. A strict shortest-path teacher action discourages such exploration, forcing the agent to mimic behavior that is not feasible given its perceptual constraints.

Reinforcement Learning as a Solution
To address the limitations of shortest-path supervision, we adopt Reinforcement Learning (RL) as a complementary training paradigm. Unlike Imitation Learning from teacher actions, RL directly optimizes navigation policies with respect to task-specific rewards, such as successful goal-reaching. This allows the agent to learn recovery strategies when deviating from the demonstration path, as well as to balance exploration and exploitation in settings with partial observability (e.g., monocular input).

Crowd-Aware VLN
The ultimate goal of VLN is deployment in real-world environments, which are inherently populated by humans. However, during training, agents are placed in

human-free simulated environments, creating a generalization gap: agents trained in such settings may fail to behave robustly when humans are present.
To address this, recent advances in simulation platforms such as Habitat [44] enable the integration of human agents into training environments. Leveraging these capabilities makes it possible to develop crowd-aware VLN agents.

Chapter 3

Results

3.1 Results

Monocular VLN Evaluation

In the context of extending VLN to social navigation, the model from [34] was trained and evaluated on the R2R-CE dataset as part of this project. This model, referred to as Sim-to-Real Transfer in Table 3.1, introduces a waypoint predictor module with 3D feature fields to reconstruct panoramic inputs, combined with a topological map. The waypoint predictor builds on the approach discussed in Section 1.2 (Monocular Waypoint Predictor) of the Related Work. The results of this implementation, evaluated on the R2R-CE dataset, are presented in Table 3.1, compared to other monocular methods on the Val Unseen split.

Methods	NE	OSR	SR	SPL
Seq2Seq [7]	7.37	40	32	30
CM^2[35]	7.02	41.5	34.3	27.6
WS-MGMap[45]	6.28	47.6	38.9	34.3
Sim-to-Real Transfer[34]	5.96	56.0	45.4	30.2
OVL-MAP[38]	4.62	64	60	50
NaVid[46]†	5.47	49.1	37.4	35.9
NaVILA [47]†	5.37	57.6	49.7	45.5
StreamVLN[48]†	5.43	62.5	52.8	47.2

Table 3.1: Comparison of methods on the Val Unseen split of the R2R-CE dataset. Methods marked with † are video-based vision-language-action (VLA) methods using monocular input.

3.2 Discussion

First, we observe that the Sim-to-Real Transfer via 3D Feature Fields for Vision-and-Language Navigation model [34] already outperforms previous monocular navigation methods [7, 35, 45] by approximately 6.5%, highlighting the effectiveness of its architecture. Since the implemented model shares a similar waypoint predictor with the CM^2 model [35], which does not use topological maps for navigation, this further underscores the impact of incorporating 3D feature fields within a topological map. Nevertheless, despite these gains, the model achieves only a 45.4% success rate, which remains relatively low and emphasizes the need for further research to close the gap toward robust real-world navigation.

During the course of this semester project, a new method, OVL-MAP [38], was published in April 2025, significantly surpassing all existing approaches, as shown in Table 3.1. Unfortunately, this paper was discovered only toward the end of the project, and no code repository was publicly available, making a full reimplementation infeasible. To address this, the project proposes a model design that combines the strengths of [34] and [38], leveraging both approaches to achieve competitive performance.

Additionally, Table 3.1 includes recent video-based Vision-Language-Action (VLA) methods—StreamVLN, NaVILA, and NaVid—that utilize monocular input. These methods employ a completely different architecture, relying entirely on Large Language Models for navigation. While they are included in the table for transparency and to provide context with state-of-the-art methods, their architecture is not directly comparable to the approaches studied in this project. These methods will be discussed further in the future works section at the end of this report.

Chapter 4

Summary

4.1 Conclusion

The motivation of this work was to develop intelligent agents capable of language-instructed navigation. To this end, we first conducted a study of the state of the art in VLN, identifying key bottlenecks such as reliance on known environment topologies and panoramic observations, both of which introduce a gap between simulation and real-world deployment. We then focused on VLN in monocular settings and continuous environments, and implemented the state-of-the-art model [34] (prior to the publication of [38]) within the R2R-CE framework. This model leverages a hallucinated metric map as a waypoint predictor and employs 3D feature fields to approximate panoramic inputs.

After analyzing its performance, we identified limitations in terms of success rate and designed a new VLN architecture. In this architecture, the waypoint prediction module is replaced by a recurrent representation that integrates the metric map with the current observation. This modification was motivated by the impressive results achieved by [38]. Furthermore, two broader challenges were highlighted: (i) the limitations of Imitation Learning, and (ii) the absence of human presence in training environments. To address these, Reinforcement Learning was introduced as a complementary training paradigm, and the Habitat simulator was selected to enable human presence in training scenarios.

Overall, this work demonstrates that robust VLN remains in its early stages, with many open challenges to address. Nevertheless, the field is highly promising, and continued research is likely to yield significant advances toward deploying socially and environmentally robust navigation agents in the real world.

4.2 Discussion of Future Work

Video-based Large Language Models

One area of research demonstrating highly efficient navigation is Video-based Large Language Models (VLMs). These models have attracted significant attention because, unlike the methods studied in this semester project, they typically do not rely on odometry, depth information, or explicit maps. Early works [47], building on the ideas of [49], propose a model consisting of three modules: a vision encoder, a projector, and an LLM that makes navigational decisions. Specifically, at each time step, the agent receives a sequence of RGB frames from its monocular camera. These frames are encoded into visual tokens using the vision encoder, which are then projected into a shared embedding space aligned with language tokens. Simultaneously, the natural language instruction is tokenized into instruction tokens.

The observation and instruction tokens are concatenated and fed into the LLM, which determines the navigation action based on both.

In [46], the authors use the same architecture but add cross-modal attention to obtain instructed visual tokens in addition to the visual tokens. The follow-up work [50] incorporates a metric map, encodes it into language tokens, and feeds it along with the observational and instruction tokens to the LLM. Building on previous methods, [48] introduces a hybrid SlowFast context strategy to enable real-time, long-horizon navigation. Its key innovations are: (1) a sliding-window KV cache that reuses recent dialogue states for low-latency action generation, avoiding costly recomputation; and (2) a slow-updating memory that compresses past visual observations via 3D-aware, voxel-based token pruning at test time.

These methods reach state-of-the-art performance in VLN but face limitations. First, they struggle to reason over extended historical trajectories. Second, inference is typically offloaded to a remote server (e.g., using an RTX 4090 GPU, as in [48]), introducing communication latency of 0.2–1.0 s on top of the 0.27 s inference time—even for this optimized architecture.

Remaining Work

Future work will focus on implementing the designed model described in Section 2.2 (Model Architecture), training it using Reinforcement Learning, and simulating it in Habitat with human agents. Additionally, Video-based Large Language Models represent a promising direction for vision-and-language navigation that should be further explored.

Bibliography

[1] S. Chen, P.-L. Guhur, C. Schmid, and I. Laptev, "History aware multimodal transformer for vision-and-language navigation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 5834–5847, 2021.

[2] Y. Qiao, Y. Qi, Y. Hong, Z. Yu, P. Wang, and Q. Wu, "HOP: History-and-order aware pre-training for vision-and-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15 418–15 427.

[3] D. An, H. Wang, W. Wang, Z. Wang, Y. Huang, K. He, and L. Wang, "ETP-NAV: Evolving topological planning for vision-language navigation in continuous environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[4] D. An, Y. Qi, Y. Li, Y. Huang, L. Wang, T. Tan, and J. Shao, "BevBERT: Multimodal map pre-training for language-guided navigation," *arXiv preprint arXiv:2212.04385*, 2022.

[5] X. Zhang, Y. Xu, J. Li, Z. Hu, and R. Hong, "Agent journey beyond RGB: Unveiling hybrid semantic-spatial environmental representations for vision-and-language navigation," *arXiv preprint arXiv:2412.06465*, 2024.

[6] L. Wang, Z. He, J. Tang, R. Dang, N. Wang, C. Liu, and Q. Chen, "A dual semantic-aware recurrent global-adaptive network for vision-and-language navigation," *arXiv preprint arXiv:2305.03602*, 2023.

[7] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, A. Van Den Hengel *et al.*, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3674–3683.

[8] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," *arXiv preprint arXiv:2010.07954*, 2020.

[9] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Y. Wang, C. Shen, and A. Van Den Hengel, "REVERIE: Remote embodied visual referring expression in real indoor environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9982–9991.

[10] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, D. Fox *et al.*, "ALFRED: A benchmark for interpreting grounded instructions for everyday tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 740–10 749.

[11] K. Nguyen and H. Daumé III, "HELP, Anna! Visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning," *arXiv preprint arXiv:1909.01871*, 2019.

[12] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. 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 (CVPR)*, 2019, pp. 12 538–12 547.

[13] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the Nav-Graph: Vision-and-language navigation in continuous environments," in *European Conference on Computer Vision (ECCV)*. Cham: Springer International Publishing, 2020, pp. 104–120.

[14] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[15] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo, and S. 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)*, 2021, pp. 1643–1653.

[16] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, "Neural topological SLAM for visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 12 875–12 884.

[17] N. Savinov, A. Dosovitskiy, and V. Koltun, "Semi-parametric topological memory for navigation," *arXiv preprint arXiv:1803.00653*, 2018.

[18] S. Chen, P. L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev, "Think global, act local: Dual-scale graph transformer for vision-and-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 16 537–16 547.

[19] J. F. Henriques and A. Vedaldi, "MapNet: An allocentric spatial memory for mapping environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8476–8484.

[20] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural SLAM," *arXiv preprint arXiv:2004.05155*, 2020.

[21] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," *arXiv preprint arXiv:2210.05714*, 2022.

[22] J. Wang, M. Sun, and Y. Zhou, "Multi-Map Fusion for Vision-and-Language Navigation in Continuous Environment," in *2024 7th International Conference on Information Communication and Signal Processing (ICICSP)*. IEEE, Sep. 2024, pp. 228–232.

[23] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L. P. Morency, T. Darrell *et al.*, "Speaker-Follower Models for Vision-and-Language Navigation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[24] H. Tan, L. Yu, and M. Bansal, "Learning to navigate unseen environments: Back translation with environmental dropout," *arXiv preprint arXiv:1904.04195*, 2019.

[25] Z. Wang, J. Li, Y. Hong, Y. Wang, Q. Wu, M. Bansal, Y. Qiao *et al.*, "Scaling data generation in vision-and-language navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 12 009–12 020.

[26] P. L. Guhur, M. Tapaswi, S. Chen, I. Laptev, and C. Schmid, "AirBERT: In-domain pretraining for vision-and-language navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 1634–1643.

[27] K. Lin, P. Chen, D. Huang, T. H. Li, M. Tan, and C. Gan, "Learning vision-and-language navigation from YouTube videos," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 8317–8326.

[28] Z. Wei, B. Lin, Y. Nie, J. Chen, S. Ma, H. Xu, and X. Liang, "Unseen from seen: Rewriting observation-instruction using foundation models for augmenting vision-language navigation," *arXiv preprint arXiv:2503.18065*, 2025.

[29] H. Wang, W. Wang, W. Liang, C. Xiong, and J. Shen, "Structured scene memory for vision-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8455–8464.

[30] Z. Wang, X. Li, J. Yang, Y. Liu, and S. Jiang, "GridMM: Grid memory map for vision-and-language navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 15 625–15 636.

[31] R. Liu, W. Wang, and Y. Yang, "Volumetric environment representation for vision-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16 317–16 328.

[32] J. Krantz, A. Gokaslan, D. Batra, S. Lee, and O. Maksymets, "Waypoint models for instruction-guided navigation in continuous environments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 162–15 171.

[33] Y. Hong, Z. Wang, Q. Wu, and S. Gould, "Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15 439–15 449.

[34] Z. Wang, X. Li, J. Yang, Y. Liu, and S. Jiang, "Sim-to-real transfer via 3D feature fields for vision-and-language navigation," *arXiv preprint arXiv:2406.09798*, 2024.

[35] G. Georgakis, K. Schmeckpeper, K. Wanchoo, S. Dan, E. Miltsakaki, D. Roth, and K. Daniilidis, "Cross-modal map learning for vision and language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15 460–15 470.

[36] Z. Wang, X. Li, J. Yang, Y. Liu, J. Hu, M. Jiang, and S. Jiang, "Lookahead exploration with neural radiance representation for continuous vision-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 13 753–13 762.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, I. Sutskever *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*. PMLR, Jul. 2021, pp. 8748–8763.

[38] S. Wen, Z. Zhang, Y. Sun, and Z. Wang, "OVL-MAP: An Online Visual Language Map Approach for Vision-and-Language Navigation in Continuous Environments," *IEEE Robotics and Automation Letters*, 2025.

[39] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jun. 2019, pp. 4171–4186.

[40] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," *arXiv preprint arXiv:2201.03546*, 2022.

[41] L. Ye, M. Rochan, Z. Liu, and Y. Wang, "Cross-modal self-attention network for referring image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 502–10 511.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[44] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra, "Habitat 2.0: Training home assistants to rearrange their habitat," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[45] P. Chen, D. Ji, K. Lin, R. Zeng, T. Li, M. Tan, and C. Gan, "Weakly-supervised multi-granularity map learning for vision-and-language navigation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 149–38 161, 2022.

[46] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, H. Wang *et al.*, "NavID: Video-based VLM Plans the Next Step for Vision-and-Language Navigation," *arXiv preprint arXiv:2402.15852*, 2024.

[47] A. C. Cheng, Y. Ji, Z. Yang, Z. Gongye, X. Zou, J. Kautz, X. Wang *et al.*, "NavILA: Legged Robot Vision-Language-Action Model for Navigation," *arXiv preprint arXiv:2412.04453*, 2024.

[48] M. Wei, C. Wan, X. Yu, T. Wang, Y. Yang, X. Mao, J. Pang *et al.*, "StreamVLN: Streaming Vision-and-Language Navigation via SlowFast Context Modeling," *arXiv preprint arXiv:2507.05240*, 2025.

[49] Y. Li, C. Wang, and J. Jia, "Llama-VID: An Image is Worth 2 Tokens in Large Language Models," in *European Conference on Computer Vision (ECCV)*. Springer Nature Switzerland, Sep. 2024, pp. 323–340.

[50] L. Zhang, X. Hao, Q. Xu, Q. Zhang, X. Zhang, P. Wang, R. M. Xu *et al.*, "A Novel Memory Representation via Annotated Semantic Maps for VLM-Based Vision-and-Language Navigation," *arXiv preprint arXiv:2502.13451*, 2025.