Toward Dynamic and Realistic Vision-and-Language Navigation – Addressing Complex Changes in Real-World

February 2025

Yanjun Sun

A thesis for the degree of Ph.D. in Engineering

Toward Dynamic and Realistic Vision-and-Language Navigation – Addressing Complex Changes in Real-World

February 2025

Graduate School of Science and Technology
Keio University

Yanjun Sun

論文要旨

高齢化，危険作業の自動化といった課題に対処するため，物理的な身体を持ち実世界と相互作用するエージェントを開発する Embodied AI が注目されている．その中でも，Vision-and-Language Navigation（VLN）は目的地への移動を行うタスクであり，家庭用ロボットや自動運転への応用が期待されているが，既存の VLN タスクは静的環境を前提としており，動的要因や長期的変化を十分に反映していない．また，都市環境の連続的変化を記録したデータセットも不足している．本研究では，屋外環境での認識とナビゲーションを基に，環境認識の向上と動的環境適応の課題に挑む．具体的には，ランドマーク物体を正確に認識するナビゲーション手法を開発し，長期間にわたる街路変化を記録したデータセットを構築する．さらに，動的な交通や歩行者を考慮したナビゲーションタスクを定義し，エージェントが環境変化に適応する手法を提案する．

第 1 章では， VLN と変化認識の背景と位置付けについて述べる．現実世界の複雑な環境において，従来手法が抱える課題を明確化し，本研究の目的と解決すべき問題を提示する．特に，静的環境を前提とした既存の VLN モデルの限界や，動的および長期的変化への適応能力の欠如に焦点を当てる．第 2 章では，屋外環境におけるナビゲーションタスクにおいて，ランドマーク物体を利用した VLN モデルを提案する．エージェントが環境中の物体情報を活用することで，従来手法によって生じる旋回場所や停止場所の判断ミスを改善し，ナビゲーション精度の向上を実現する．第 3 章では，長期間にわたる屋外環境の変化を認識するための新たなデータセットを構築する．本データセットは，環境の連続的かつ長期的な進化を記録し，従来の静的環境に依存するデータセットの限界を克服することを目指している．さらに，このデータセットが変化領域の分割や記述といった複数のタスクを提案する．第 4 章では，既存の VLN タスクを拡張し，動的な交通状況や天候などの要因を考慮した新たなナビゲーションタスクを定義する．また，これらの動的要因に対応可能な手法を提案し，エージェントがリアルタイムで環境の変化に適応できる能力を強化する．第 5 章では，本研究の成果をまとめ，提案手法およびデータセットが現実世界におけるナビゲーションや変化認識タスクに与える影響について議論する．さらに，今後の課題と展望について述べる．

Abstract

To address challenges such as aging populations and the automation of hazardous tasks, Embodied AI, which develops agents equipped with physical bodies capable of interacting with the real world, has gained significant attention. Among its applications, Vision-and-Language Navigation (VLN) stands out as a task involving navigation to a destination based on natural language instructions. While VLN holds promise for applications such as household robots and autonomous driving, existing tasks assume static environments, failing to adequately reflect dynamic factors or long-term changes. Additionally, datasets capturing continuous changes in urban environments remain insufficient. This study tackles the challenges of improving environmental recognition and understanding and adapting to dynamic environments in outdoor settings. Specifically, it develops a navigation method that accurately recognizes landmark objects and constructs a dataset documenting long-term street changes. Furthermore, it defines navigation tasks that account for dynamic traffic and pedestrians and proposes methods enabling agents to adapt to environmental changes.

Chapter 1 provides an overview of the background and positioning of VLN and change recognition. It highlights the challenges of existing methods in handling complex real-world environments and identifies the objectives and problems this research aims to address, with a focus on overcoming the limitations of static VLN models and their lack of adaptability to dynamic and long-term changes. Chapter 2 proposes a VLN model that leverages landmark objects for outdoor navigation tasks. By utilizing object information in the environment, the model aims to improve the accuracy of navigation, addressing issues such as incorrect decisions at turning and stopping points in existing methods. The effectiveness of the proposed model is validated through evaluations on established benchmark datasets. Chapter 3 focuses on the construction of a novel dataset to recognize long-term changes in outdoor environments. This dataset records continuous and long-term evolution in the environment and aims to overcome the limitations of conventional static datasets. Additionally, this dataset supports multiple tasks, such as change region segmentation and description. Chapter 4 extends existing

VLN tasks by defining a new navigation task that considers dynamic factors like traffic and weather. Methods are proposed to address these dynamic factors, enhancing the agent's ability to adapt to real-time changes in the environment. The effectiveness of the proposed methods is demonstrated through experiments conducted in dynamic scenarios. Chapter 5 summarizes the outcomes of this study and discusses the impact of the proposed methods and datasets on real-world navigation and change recognition tasks. Future challenges and research directions are also outlined.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The concept of Vision-and-Language Navigation (VLN)—a task where intelligent agents interpret natural language instructions to navigate real-world environments —has become an essential focus within Embodied AI. As AI-driven automation becomes more deeply integrated into daily life, VLN holds immense potential for applications such as household assistance, autonomous delivery, and disaster response. For instance, agents equipped with VLN capabilities can guide household robots to specific locations, direct autonomous vehicles through complex urban environments, and assist search-and-rescue robots in interpreting mission-critical instructions. However, real-world navigation presents fundamental challenges, requiring agents to process multimodal inputs, handle fine-grained visual details, and adapt to dynamic environments.

This chapter explores the background, current advancements, and fundamental challenges of VLN, highlighting its pivotal role within Embodied AI. Vision-Language Models (VLMs) and Multimodal Large Language Models (MLLMs) serve as foundational technologies for VLN, enabling agents to align natural language instructions with visual perceptions. Despite their success in controlled settings, these models face significant challenges in three key areas: their limited sensitivity to fine-grained environmental details, the lack of datasets that account for long-term environmental changes, and their inability to adapt to dynamic, evolving environments. To bridge these gaps, this study focuses on three core advancements to improve VLN adaptability and robustness. First, an object-

aware recognition method is introduced to enhance navigation accuracy by leveraging landmark objects along the route, ensuring more precise decision-making. Second, a dataset capturing long-term environmental changes is developed, allowing agents to recognize and adapt to evolving landscapes by understanding temporal consistency across different time frames. Finally, a novel VLN task and adaptive methods are proposed to equip agents with the ability to handle real-time decision-making under dynamic conditions, incorporating multimodal cues to navigate unpredictable environments involving moving traffic, pedestrians, and varying weather conditions.

By addressing these aspects, this research aims to push the boundaries of VLN, making it more robust and applicable to the complexities of real-world scenarios.

1.1 Background

Vision-and-Language Navigation (VLN) requires agents to follow language instructions to understand the environment and navigate through the environment. This task lies at the intersection of natural language understanding, computer vision, and robotics, requiring agents to integrate linguistic instructions with visual perception to navigate complex environments. Figure 1.1 shows an example of a VLN task, allowing agents to navigate in the urban environment according to instructions. This integration makes VLN a unique and challenging task, especially in real-world scenarios characterized by dynamic changes and diverse multimodal inputs.

1.1.1 Current Progress in VLN

Over the past few years, Vision-and-Language Navigation (VLN) has made significant progress, primarily driven by advances in deep learning, multimodal modeling, and the availability of benchmark datasets. Early VLN methods relied heavily on rule-based systems [1], which used predefined heuristics and handcrafted features to interpret navigation instructions. Although these approaches offered initial insights into aligning language and navigation, they were fundamentally

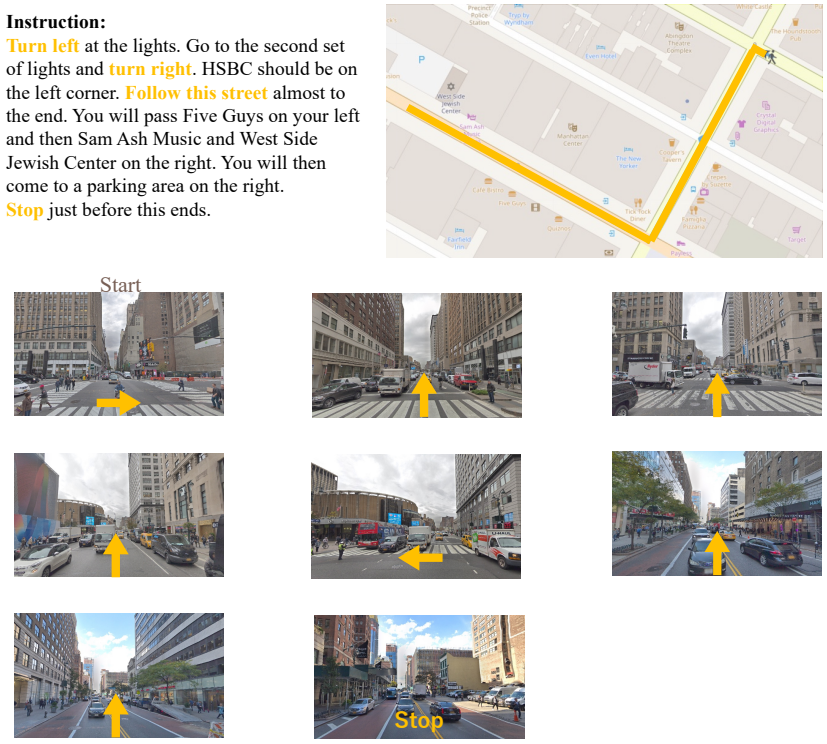


Figure 1.1 An Example of Vision-and-Language Navigation Task.

limited in their scalability and adaptability to diverse and unstructured environments [2, 3, 4].

The introduction of deep learning models, particularly Transformers, significantly improved navigation performance by enhancing the alignment of language and visual observations. Recent approaches leveraging Vision-Language Models (VLMs) and Multimodal Large Language Models (MLLMs) have greatly improved agents' ability to align natural language instructions with visual observations. These models integrate sophisticated visual feature extractors, such as convolutional neural networks (CNNs) [5] or vision transformers (ViTs) [6], with language encoders, enabling agents to interpret and execute complex instructions. For example, VLMs have demonstrated strong performance on benchmark datasets such as Room-to-Room (R2R) [7], which involves indoor navigation, and TOUCHDOWN [2], which focuses on navigation in urban environments.

Regarding modeling techniques, attention mechanisms [8, 9] have played a crucial role in improving the alignment between linguistic instructions and vi-

sual perceptions. By selectively focusing on relevant visual or linguistic features, attention-based models have enhanced agents' understanding of both the environment and the task context. Additionally, reinforcement learning methods [10] have been employed to optimize sequential decision-making, allowing agents to learn robust navigation policies through trial and error.

Despite these advancements, much of the progress has been confined to static or highly controlled environments. The focus on pre-defined routes or instructions in datasets like R2R and TOUCHDOWN has helped standardize evaluation protocols but often fails to reflect the complexities of real-world scenarios. For instance, dynamic factors such as moving pedestrians, changing weather conditions, or evolving traffic patterns are typically absent in these datasets. Similarly, while agents have shown improved generalization to unseen environments within benchmark datasets, their adaptability to evolving or continuous environments remains an open challenge.

Overall, the current trajectory of research in VLN highlights both the potential of deep learning-based methods and the pressing need for approaches that can address real-world complexities. This gap forms the foundation for the research presented in this study, which seeks to enhance navigation accuracy and adaptability by leveraging object-level information and constructing datasets that incorporate dynamic and long-term environmental changes.

1.1.2 Challenges in VLN

To bridge VLN technology into our real life, the transition from static to dynamic and evolving environments introduces several key challenges in Vision-and-Language Navigation (VLN), which must be addressed to enable robust performance in real-world scenarios:

- **Limited Understanding of Fine-Grained Visual Details** Existing VLMs often overlook critical details in their environment, such as small but important objects or subtle changes in object states. This lack of detailed understanding affects their ability to make precise decisions, such as determining accurate turning or stopping points during navigation. Moreover, many

current approaches rely on scene-level feature representations, which can obscure important object-level details.

- **Scarcity of Datasets for Modeling Long-Term Changes** Most existing VLN datasets assume a static world where roads, buildings, and landmarks remain unchanged. However, real-world environments are subject to both short-term variations (e.g., moving vehicles, pedestrians, and dynamic traffic signals) and long-term transformations (e.g., urban development and seasonal shifts). While short-term changes affect real-time decision-making, long-term environmental changes challenge an agent ' s ability to recognize familiar locations over extended periods. Current VLN models lack the capability to recognize temporal consistency across different time frames, leading to failures when navigating in long-term dynamic environments.

- **Over-Reliance on Static Environment Settings** Beyond long-term environmental changes, real-world navigation also involves short-term dynamic factors such as moving vehicles, pedestrians, and changing traffic signals. Current VLN models are often designed for environments with predefined, unchanging routes, limiting their applicability in real-world scenarios where navigation decisions must be made dynamically. Real-world navigation requires agents to adapt in real-time to factors such as moving vehicles, pedestrian activity, or sudden environmental changes, posing significant challenges for existing methods that lack robust adaptability.

Addressing these challenges is essential for advancing VLN beyond controlled settings and enabling its deployment in real-world applications such as autonomous driving, urban navigation, and disaster response. This study seeks to tackle these issues through improved object-aware recognition, datasets capturing both dynamic and long-term environmental changes, and methods enabling real-time adaptability to evolving conditions. To develop truly robust VLN systems, agents must not only make real-time decisions in response to short-term environmental changes but also retain memory of previously visited locations despite long-term transformations. A robust VLN model should be capable of revisiting a route after months or years and recognizing familiar landmarks, even when structural

modifications have occurred. Addressing both short-term dynamics and long-term adaptability is crucial for transitioning VLN from controlled experiments to practical deployment in real-world settings.

1.2 Bridging the Gap: Objectives of This Study

This study aims to bridge the gap between current VLN methods and the demands of real-world, dynamic scenarios by addressing two key challenges: enhancing object-aware recognition and improving adaptability to dynamic environments.

- **Improving Fine-Grained Visual Understanding for Navigation** This study proposes a novel VLN method that leverages object-level information to enhance navigation accuracy. By incorporating features of landmark objects along the route, the method significantly improves the agent's ability to determine precise turning and stopping locations, leading to more accurate navigation decisions. Unlike existing approaches, this object-aware method directly addresses the need for finer-grained environmental understanding during navigation and segmentation.

- **Dataset for Long-Term Adaptability to Environmental Changes** To account for the evolving nature of real-world scenarios, this study introduces a new dataset capturing long-term environmental changes, such as street transformations and infrastructure development. This dataset allows VLN agents to recognize locations even after months or years, despite modifications in the surrounding environment. By incorporating continuous and evolving scenarios, it overcomes the limitations of existing static datasets and offers a robust foundation for studying temporal consistency and long-term scene understanding in navigation.

- **Dataset and Method for Adapting to Dynamic, Short-Term Changes in VLN** This study defines a novel VLN task designed to incorporate real-time dynamic factors, including traffic flows, pedestrian movements, and weather variations. In response to these challenges, new methods are developed to enable agents to adapt to real-time changes effectively. These

methods emphasize real-time decision-making and robust integration of visual and linguistic cues, ensuring safe and efficient navigation in dynamic environments.

1.3 Publication List

The publication list by the author and the related thesis chapters are as follows:

1. Yanjun Sun, Yue Qiu, Yoshimitsu Aoki, and Hirokatsu Kataoka. Outdoor vision-and-language navigation needs object-level alignment. *Sensors*, Vol. 23, No. 13, 2023 (Chapter 2)

2. Yanjun Sun, Yue Qiu, Yoshimitsu Aoki, and Hirokatsu Kataoka. Guided by the way: The role of on-the-route objects and scene text in enhancing outdoor navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5198–5204, 2024 (Chapter 2)

3. Yanjun Sun, Yue Qiu, Mariia Khan, Fumiya Matsuzawa, and Kenji Iwata. The stvchrono dataset: Towards continuous change recognition in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14111–14120, June 2024 (Chapter 3)

4. Yanjun Sun, Yue Qiu, and Yoshimitsu Aoki. Dynamicvln: Incorporating dynamics into vision-and-language navigation scenarios. *Sensors*, Vol. 25, No. 2, 2025 (Chapter 4)

1.4 Structure of This Thesis

Here is the structure of this thesis:

- Chapter 1 provides an overview of the background and positioning of VLN and change recognition. It highlights the challenges of existing methods in handling complex real-world environments and identifies the objectives and problems this research aims to address, focusing on overcoming the

limitations of static VLN models and their lack of adaptability to dynamic and long-term changes.

- Chapter 2 proposes a VLN model that leverages landmark objects for outdoor navigation tasks. By utilizing object information in the environment, the model aims to improve the accuracy of navigation, addressing issues such as incorrect decisions at turning and stopping points in existing methods. The effectiveness of the proposed model is validated through evaluations of established benchmark datasets [11, 12].

- Chapter 3 focuses on the construction of a novel dataset to recognize long-term changes in outdoor environments. This dataset records continuous and long-term evolution in the environment and aims to overcome the limitations of conventional static datasets. Additionally, this dataset supports multiple tasks, such as change region segmentation and description [13].

- Chapter 4 extends existing VLN tasks by defining a new navigation task that considers dynamic factors like traffic and weather. Methods are proposed to address these dynamic factors, enhancing the agent's ability to adapt to real-time changes in the environment. The effectiveness of the proposed methods is demonstrated through experiments conducted in dynamic scenarios [14].

- Chapter 5 summarizes the outcomes of this study and discusses the impact of the proposed methods and datasets on real-world navigation and change recognition tasks. Future challenges and research directions are also outlined.

Chapter 2

Object-Aware Vision-and-Langauge Navigation

This chapter addresses the challenge of improving navigation accuracy in Vision-and-Language Navigation (VLN), particularly in complex environments, as discussed in Chapter 1. An overview of the VLN task is provided, followed by a review of existing works in this field, highlighting their limitations, particularly the lack of focus on object-level information. This oversight often results in navigation failures, such as incorrect turns or stops, especially in complex outdoor environments. To overcome these limitations, the Object-Attention VLN (OAVLN) model is introduced, which enhances navigation accuracy by incorporating object features from the environment. By aligning object-level cues with natural language instructions, OAVLN enables agents to make more precise decisions during navigation. Extensive experiments demonstrate that OAVLN significantly outperforms existing methods in both seen and unseen scenarios.

2.1 Introduction

Enabling robots to navigate real-world environments using natural language instructions is a long-standing goal in AI research. Vision-and-Language Navigation (VLN) tasks aim to achieve this by requiring an agent to interpret linguistic commands, align them with visual perceptions of the environment, and reason

about spatial relations to execute actions that guide it to a destination [2, 7, 15, 16]. This process involves understanding instructions, grounding them in observable environments, tracking the agent's position relative to objects, and dynamically adjusting actions to ensure successful navigation.

Recent advancements in outdoor VLN models have predominantly relied on encoder-decoder frameworks that combine instruction and panoramic visual features to predict navigation actions [2, 3, 4, 8, 17]. However, these approaches often fail to fully leverage object-level information from the environment. A closer examination of generated paths reveals that such models tend to neglect key objects or landmarks referenced in the instructions, which are crucial for human-like navigation. This oversight frequently leads to navigation errors, such as turning or stopping at incorrect locations, thereby hindering their applicability in real-world scenarios.

The challenges posed by this lack of object awareness are well documented. Studies like DiagnoseVLN [18] reveal that agents often prioritize directional cues while neglecting objects explicitly mentioned in instructions. This is contrary to how humans navigate. In unfamiliar settings, humans intuitively rely on landmarks—buildings, objects, or text—as reference points for accurate navigation [19]. For example, as illustrated in Fig. 2.1, a human navigator might turn at a "black iron fence" and stop at the "last blue bike," using these objects as crucial environmental cues.

The success of object-aware models in indoor VLN tasks [20, 21, 22, 23, 24, 25, 26] underscores the importance of integrating object features for navigation. Indoor VLN scenarios typically involve stable and structured environments, making it feasible to leverage specific objects for navigation. However, outdoor environments are inherently more complex and unstructured, requiring models to process diverse visual cues, including natural and man-made landmarks. This complexity underscores the need for robust object-aware VLN models capable of operating effectively in dynamic, real-world conditions.

To address the abovementioned limitations, This chapter proposes a simple yet effective Object-Attention VLN (OAVLN) model that allows the agent to focus more on objects and scene texts to understand the environment better. To evaluate the effectiveness of OAVLN, extensive experiments were conducted on two

Figure 2.1 Objects are important clues in outdoor VLN. Our Object-Attention VLN model is designed to navigate using this information. At viewpoint (b), our agent seeks the 'black iron fence' and turns right. Subsequently, it stops at the viewpoint (c) because it has observed the 'blue bikes.'

widely used outdoor VLN benchmark datasets, Touchdown [2] and map2seq [27]. Comparisons with four baseline models [2, 3, 4, 8] demonstrate that OAVLN consistently outperforms existing approaches across all key metrics, even in unseen scenarios. Qualitative analyses further confirm that the improved performance stems from OAVLN ' s enhanced ability to identify and utilize objects, enabling precise turns and stops in complex environments.

2.2 Related Work

2.2.1 Vision-and-Language Navigation

Vision-and-Language Navigation (VLN) unites visual perception and natural language understanding, requiring agents to interpret linguistic instructions and align them with their surroundings to navigate effectively. Early work in VLN primarily focused on indoor navigation scenarios, exemplified by the R2R benchmark [7], which introduced navigation tasks within the Matterport3D dataset [28] using a multimodal Seq2Seq baseline model. Extensions of R2R introduced multilingual benchmarks like XL-R2R [29] and RxR [15], highlighting the growing demand for linguistic diversity in VLN.

In contrast, outdoor VLN introduces unique challenges due to unstructured and dynamic environments. The Touchdown dataset[2] was the first outdoor VLN benchmark, based on Google Street View[1], featuring complex navigation tasks in real-world urban environments. Subsequent outdoor datasets like StreetLearn [30], Retouchdown [31], StreetNav [32], map2seq [27], and Talk2Nav [33] further expanded the scope of outdoor VLN tasks.

A variety of methods have been developed for these benchmarks:

- RCONCAT [2], the baseline model for Touchdown, uses an LSTM-based architecture to encode trajectories and instructions.
- ARC+l2s [17] cascades action prediction into binary stopping decisions and subsequent direction classification.
- VLN-Transformer[8] enriches navigation data by applying a pre-trained BERT[34] model to external multimodal datasets.
- GA [3] computes fused representations of instructions and images using gated attention mechanisms.
- ORAR [4] enhances navigation performance by introducing junction-type embeddings and heading deltas, which reduce the performance gap between seen and unseen environments.

[1] https://developers.google.com/maps/documentation/streetview/intro

However, these methods rely heavily on LSTM-based encoder-decoder architectures, which often fail to effectively utilize landmarks and objects specified in instructions, leading to navigation errors in complex outdoor environments. This chapter addresses these limitations by introducing an approach that explicitly integrates object-awareness into the navigation process, enhancing adaptability in unstructured and dynamic settings. While prior research has made significant advances in aligning language and vision for structured environments, the proposed method focuses on capturing object-level features that are critical for outdoor navigation.

2.2.2 Object-Aware VLN

To address the need for fine-grained semantic understanding, particularly with respect to object-level information, vision-and-language pre-trained models such as ViLBERT [35] have been adopted in VLN. These models leverage joint representations of visual and linguistic features, enhancing the alignment of instructions with environmental cues.

Several object-aware VLN models have demonstrated the utility of integrating object features:

- ORIST [36] incorporates object and room features to improve navigation performance.
- OAAM [20] extracts tokens from instructions and encodes object tokens to inform action predictions.
- SOAT [22] combines scene classification networks and object detectors to align distinct visual cues for more effective navigation.

While object-aware models have proven effective in indoor VLN tasks, their applicability to outdoor environments remains limited due to the dynamic and diverse nature of outdoor settings. Indoor navigation typically occurs in static, well-defined environments, whereas outdoor navigation requires agents to handle a wider range of objects, varying spatial layouts, and unstable conditions such as changing weather or lighting.

Outdoor VLN also introduces unique challenges, including recognizing scene-specific features like store names or embedded objects, which are critical for navigation. Addressing these challenges necessitates leveraging scene text recognition and object detection to enhance the agent's understanding of its surroundings. To tackle these complexities, this study extends object-awareness to outdoor navigation by incorporating object features and scene-specific cues such as text and dynamic objects. The proposed Object-Attention VLN (OAVLN) model bridges the gap between object

2.3 Problem Definition for Vision-and-Language Navigation

The Vision-and-Language Navigation (VLN) task involves guiding an agent to navigate within an environment using natural language instructions $X = \{x_1, x_2, \ldots, x_l\}$. The environment is represented as an undirected graph $G = (\mathbb{V}, \mathbb{E})$, where $v \in \mathbb{V}$ denotes nodes and $(v, u) \in \mathbb{E}$ represents labeled edges connecting panoramas. Each node v is associated with a panoramic RGB image, and each edge connects neighboring panoramas at a specified heading angle $\alpha_{(v,u)}$. The agent's state at time t is defined as $s_t = (v_t, \alpha_{(v_{t-1},v_t)})$, where v_t represents the current location of agent, and $\alpha_{(v_{t-1},v_t)}$ is the heading from the previous node. Given instructions X, the agent performs actions $a_t \in \{\text{FORWARD, LEFT, RIGHT, STOP}\}$ and transitions to the next state s_{t+1}. The task is completed when the agent produces a sequence of state-action pairs ending with $a_n = \text{STOP}$, successfully reaching the target destination.

2.4 Preliminary Experiments: What do agents focus on when navigating?

DiagnoseVLN [18] reports that task completion nearly drops to zero when the masking direction word tokens are during testing only and that masking out the object tokens has a weaker impact on task completion rate than the masking di-

Instruction:
Go with traffic, the playground will be on your right. turn left at the first intersection. Turn left again at the next intersection. Green scaffolding will be on your right. Turn left at the next intersection. A fruit market will be on the corner. Look left and stop just pass all the backpacks at the store with a yellow banner.

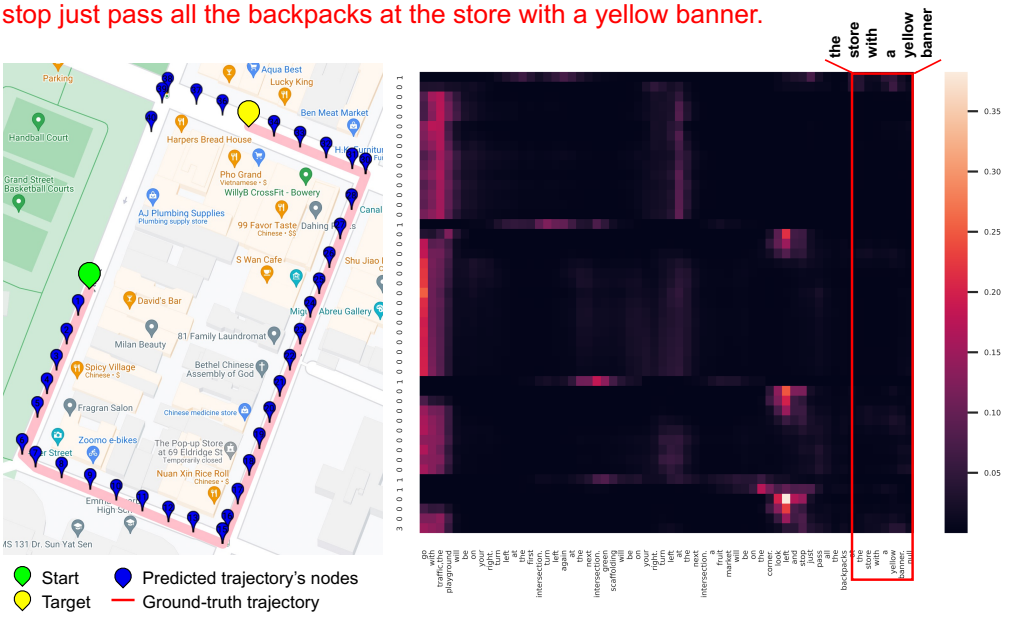


Figure 2.2 Example of visualization of the ORAR model on the Touchdown dataset. The top of this figure is the instruction, and the red text is the distribution of stop location, which ORAR disregarded. Left: trajectory generated by ORAR vs. ground truth. Right: Attention to each token from the instructions during predicting actions.

rection tokens. The authors concluded that direction tokens were more important than object tokens for VLN tasks and suggested that future work explore the use of direction tokens in greater depth. While it may seem counterintuitive, the importance of different types of tokens may vary depending on the specific task and environment being navigated. Object tokens are sometimes more efficient in pinpointing landmarks and deciding turns.

Therefore, to determine which tokens the trained agent paid attention to during outdoor navigation, the generated trajectory was visualized using the Google Map API[2]. A heatmap of instruction attention weights was plotted for the ORAR [4] model. Fig. 2.2 shows an example of the visualization. The x-axis of the heatmap

[2]https://developers.google.com/maps/documentation

represents each token of the instructions, while the y-axis represents the predicted actions of the agent at each timestep. Each grid on this heatmap indicates attention received by a token when the agent predicted the action. The more attention a token receives, the brighter color of the grid. This example shows that the agent was instructed to stop at 'the store with a yellow banner', but ignored this information and turned left at the next junction, eventually stopping at the wrong location to fail navigation. The heatmap shows that the attention weight for 'the store with a yellow banner' was almost zero during navigation. Furthermore, the attention weight of object tokens in the instructions from the test set was analyzed, revealing an average weight of 0.128 for each object token in the instructions. According to the preliminary results reported above, existing outdoor VLN models cannot pay attention to object tokens during navigation, leading to turning or stopping at the wrong location. Additionally, even some non-content words, like 'the', have more attention than object tokens, indicating that the existing model has been learning data biases by ignoring objects.

The findings from our preliminary experiments highlight a critical gap in existing outdoor VLN models: insufficient attention to object tokens, which are essential for accurate navigation in real-world environments. This motivates the development of our Object-Attention VLN (OAVLN) model, designed to prioritize object features and align them with navigation instructions for improved performance in both seen and unseen scenarios. The following sections detail the proposed methodology and its evaluation.

2.5 Proposed Method: Object-Attention Vision-and-Language Navigation (OAVLN)

This section presents the Object-Attention Vision-and-Language Navigation (OAVLN) model, developed to improve navigation accuracy in outdoor Vision-and-Language Navigation (VLN) tasks. The model incorporates object features, scene text, and panorama representations to address the limitations of existing approaches. As shown in Fig. 2.3, the architecture follows a sequence-to-sequence framework with a two-layer decoder that predicts the agent's actions based on multiple input

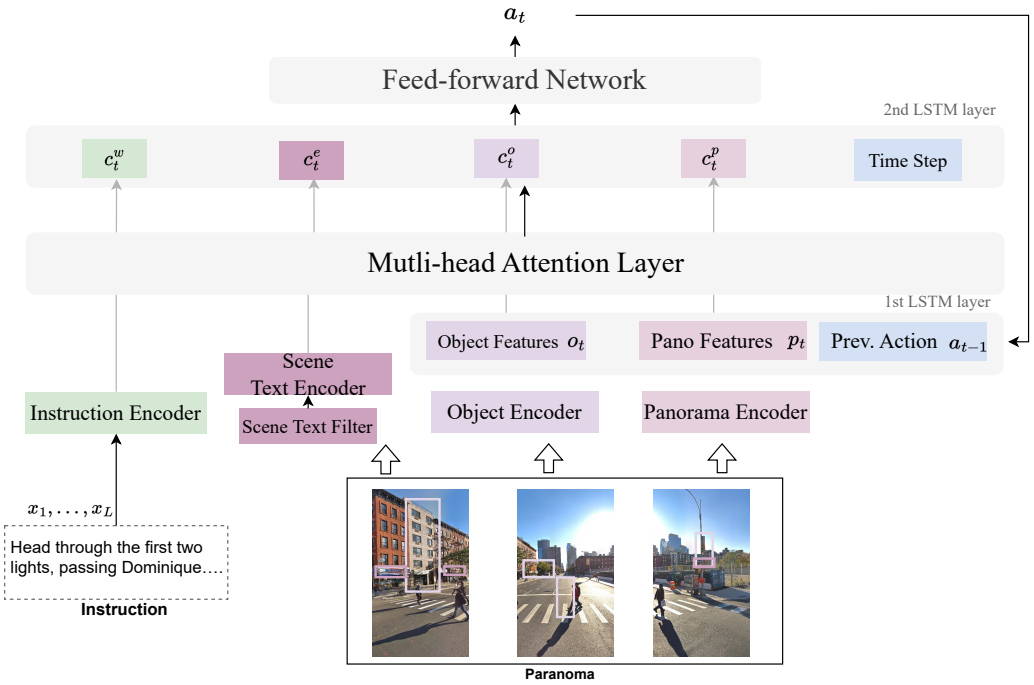


Figure 2.3 Overview of the Object Attention VLN model. The model takes multiple input modalities, including navigation instructions, panoramic features, object features, and scene text. Using these inputs, the model generates contextualized representations of the agent's state at each timestep, considering prior actions, to make informed navigation decisions.

modalities. The Instruction Encoder processes the natural language instructions, converting them into contextualized token embeddings. The Panorama Encoder extracts features from panoramic images, while the Object Encoder focuses on object-level features detected within the scene. Additionally, a Scene Text Encoder filters and encodes textual elements (e.g., store names or signs) from the environment. These modalities are combined via a Multi-Head Attention Layer, which generates context-aware representations of the agent's state. The two-layer decoder, consisting of LSTM layers, predicts the agent ' s next action at each timestep by considering the encoded features and the agent's previous action.

2.5.1 Instruction Encoder.

The instruction encoder embeds and encodes the tokens in the navigation instructions sequence $\mathbf{x} = x_1, ..., x_L$ using a bidirectional LSTM [37]:

$$\hat{x}_i = \text{embedding}(x_i) \tag{2.1}$$

$$((w_1, ..., w_L), z_L^w)) = \text{Bi-LSTM}(\hat{x}_1, ..., \hat{x}_L) \tag{2.2}$$

where $w_1, ..., w_L$ are the hidden representations for each token and z_L^w is the last LSTM cell state.

2.5.2 Panorama Encoder & Object Encoder

At each timestep t, the panorama at the agent's current position is represented by extracted visual features. The 360° panorama is divided into eight projected rectangles, each covering a 60° field of view, ensuring a complete representation of the environment. Among these eight slices, five are selected for further processing: the center slice, which aligns with the agent's heading, and the two slices to its left and right. This selection focuses on the most relevant portions of the panorama for navigation. The selected five slices are fed into a pre-trained ResNet-50 [38] model on ImageNet [39] to extract high-level visual features. Each slice is represented by a feature vector $\bar{p}_t^s$ of size 2,048. To capture object-level information, up to 20 objects are detected within each slice. The features of these objects are extracted using a pre-trained ResNet-101 [38] model on the Visual Genome dataset[3]. The object features from each slice are aggregated into a vector $\bar{o}_t^s$, also with a size of 2,048. By combining panoramic features $\bar{p}_t^s$ and object features $\bar{o}_t^s$, the model gains a holistic understanding of the agent's surroundings, leveraging both high-level scene context and fine-grained object-level details.

2.5.3 Scene Text Filter & Scene Text Encoder.

To improve scene text extraction from low-quality panorama images, a Scene Text Filter was developed. The Object Encoder was utilized to process the entire

[3]http://visualgenome.org/

panorama and identify the 'sign' regions. Scene text recognition was then applied exclusively to these 'sign' regions using the MMOCR [40] model with the SAR [41] architecture for text recognition. Finally, the recognition results were refined by matching them to the closest scene text mentioned in the instructions.

These scene text $\mathbf{e} = e_1, ..., e_M$ were embedded and encoded by a bidirectional LSTM:

$$\hat{e}_i = \text{embedding}(e_i) \tag{2.3}$$

$$((w_1, ..., w_M), z_M^w)) = \textbf{Bi-LSTM}(\hat{e}_1, ..., \hat{e}_L) \tag{2.4}$$

where $w_1, ..., w_M$ are the hidden representations for each token and z_M^w is the last LSTM cell state.

2.5.4 Decoder

The panorama encoder, as described in detail above generates a fixed size representation $\bar{p}_t$ of the sequence of sliced visual representations of the current panorama view, denoted as $\bar{p}_t^1, ..., \bar{p}_t^S$. Similarly, the object encoder emits a fixed size representation $\bar{o}_t$ of the objects in the current panorama and a sequence of sliced view representations $\bar{o}_t^1, ..., \bar{o}_t^S$. The state z_0^{first} of the cell in the first decoder LSTM layer was initialized using z_L^w. The input to the first decoder layer was the concatenation ($\oplus$) of previous action embedding $\bar{a}_{t-1}$, visual representation $\bar{p}_t$ and object features $\bar{o}_t$. The output of the first decoder layer,

$$h_t^{\text{first}} = \text{LSTM}^{\text{first}}([\bar{a}_{t-1} \oplus \bar{p}_t \oplus \bar{o}_t]), \tag{2.5}$$

was then used as the query of multi-head attention [42] over the text encoder. The resulting contextualized text representation c_t^w was then used to attend over the sliced visual representations c_t^p, object representations c_t^o, and scene text encode c_t^e.

The input and output of the second decoder layer were

$$h_t^{\text{second}} = \text{LSTM}^{\text{second}}([\bar{t} \oplus h_t^{\text{first}} \oplus c_t^p \oplus c_t^o \oplus c_t^e]), \tag{2.6}$$

where $\bar{t}$ represents embedded timestep t. The hidden representation h_t^{second} from the second decoder layer goes through a feed-forward network to predict action a_t.

2.6 Experiments

2.6.1 Experimental Setup

Implementation Details. Our framework and baselines were developed in PyTorch [43]. ResNet50 [38] was used for panorama features, while Faster R-CNN [44], pretrained on Visual Genome [45] with ResNet101 [38], was used for object features with an IoU score of 0.6. Scene text was recognized using MMOCR [40]. The object tokens in instructions were summarized with stanza [46], which also optimized scene text recognition by a sequence matching algorithm [47] with 0.8 similarity score. The models were trained using Adam [48] under teacher-forcing, with parameters such as a learning rate of 5e-4, weight decay of 1e-3, batch size of 64, and dropout rates of 0.3. After 150 epochs, the top model was selected from the development set. Instructions and scene texts were converted to byte pair encodings [49] with a 2,000 token vocabulary and embedded at 256. Other embeddings were 256 and 16 in size.

Datasets. The Touchdown [2] and map2seq [27] dataset use urban scenarios to create a large navigation environment based on Google Street View[4]. The environment simulates NYC, comprising 29,641 nodes and 61,319 undirected edges. The touchdown dataset includes 9,326 navigation trajectories, each paired with human-written instructions based on the corresponding panoramas, within 6,525 training, 1,391 development, and 1,409 test samples. The instructions in map2seq instead focused on visual landmarks from OpenStreetMap. map2seq comprises 7,672 navigation instructions, segmented into 6,072 training, 800 development, and 800 test samples. Furthermore, following the approach in ORAR [4], the datasets were split based on the geographic separation of the training and testing areas for the unseen scenario.

Baselines. The proposed model was compared to previous studies on outdoor VLN, including RCONCAT [2], GA [3], VLN-Transformer [8], and ORAR [4]. These models use an LSTM to encode the instruction text and a single-layer decoder LSTM to predict the next action.These models were selected because they do not specifically handle on-the-route object features in detail. By comparing the

[4]https://developers.google.com/maps/documentation/streetview/overview

results with these baseline models, it was demonstrated that incorporating on-the-route object features benefits outdoor VLN.

Metrics. The following metrics were used to evaluate the VLN performance:

- Task Completion (TC): This metric measures the navigation accuracy of the agent to the correct location, which can be either the exact goal panorama or one of its neighboring panoramas.
- Shortest-Path Distance (SPD) [2]: This metric calculates the average distance between the final position of the agent and the goal position in the environment graph.
- Success weighted by Edit Distance (SED): This metric calculates the normalized Levenshtein edit distance [50] between the predicted and ground-truth paths, only awarding points for successful paths.
- Coverage weighted by Length Score (CLS) [51]: This metric measures the similarity between the path of the agent and the ground-truth path.
- Normalized Dynamic Time Warping (nDTW) [52]: This metric measures the cumulative distance between the predicted and ground-truth paths.
- Success-weighted Dynamic Time Warping (SDTW): This metric is the nDTW value calculated only for successful navigations.

2.6.2 Quantitative Results

This section evaluates the performance of the proposed Object-Attention Vision-and-Language Navigation (OAVLN) model in outdoor VLN tasks. The analysis focuses on both seen and unseen scenarios, validating the effectiveness of object features and scene text integration. Comparative results are presented against baseline models, highlighting improvements across various evaluation metrics.

Seen Scenario. Tables 2.1 and 2.2 present a comparison of the OAVLN model and baseline approaches on seen scenarios for the Touchdown and map2seq datasets. The proposed model consistently outperformed baselines across all evaluation

metrics. Notably, OAVLN demonstrated significant improvements in path alignment metrics, such as CLS and sDTW, underscoring the advantage of leveraging object feature attention to enhance instruction-following capabilities and goal achievement rates.

On the map2seq dataset, the OAVLN(+scene text) variant achieved a 6% improvement in goal-oriented metrics (TC and SED) compared to baseline models, indicating its superior ability to utilize objects for precise stopping. In contrast, improvements on the Touchdown dataset were less pronounced, likely due to differences in dataset characteristics, as map2seq instructions emphasize object features more explicitly.

Table 2.1 Navigation results on Touchdown for the seen scenario.

Model	TC↑	SPD↓	SED↑	CLS↑	nDTW↑	sDTW↑
RCONCAT [2]	8.94	22.48	8.55	43.23	18.20	7.98
GA [3]	9.87	20.34	9.42	47.77	21.51	8.92
VLN Transformer [8]	14.90	21.20	14.60	45.40	25.30	14.00
ORAR [4]	24.23	17.30	23.70	56.87	37.20	22.87
Ours (+scene text)	24.77	**15.98**	24.14	59.93	37.64	23.14
Ours (+objects)	**25.90**	16.04	**25.40**	**60.84**	**39.00**	**24.47**

Table 2.2 Navigation results on map2seq for the seen scenario.

Model	TC↑	SPD↓	SED↑	CLS↑	nDTW↑	sDTW↑
RCONCAT [2]	14.62	20.61	14.30	54.18	27.43	13.76
GA [3]	17.88	18.25	17.55	58.56	31.46	17.08
VLN Transformer [8]	17.00	-	-	-	29.50	-
ORAR [4]	43.96	6.93	43.09	82.97	60.43	41.78
Ours (+scene text)	**50.00**	**6.11**	**49.04**	**84.77**	**65.39**	**47.45**
Ours (+objects)	49.00	6.40	48.08	84.28	63.38	46.75

Unseen Scenario. Table 2.3 reports the performance of the OAVLN model in unseen scenarios for the development and test sets of both datasets. While the relative performance improvement compared to baseline models decreased in unseen environments, OAVLN still achieved a 3% increase in TC and nDTW metrics

over existing approaches.

These findings highlight the robustness of OAVLN in unseen scenarios, where it demonstrates the ability to follow instructions and achieve tasks with higher accuracy and reliability. The incorporation of detailed on-the-route object features enables the agent to effectively identify turn and stop locations, even in previously unencountered environments.

Table 2.3 Navigation results for the unseen scenario.

Dataset	**Touchdown**				**map2seq**			
	dev		**test**		**dev**		**test**	
Model	**TC↑**	**nDTW↑**	**TC↑**	**nDTW↑**	**TC↑**	**nDTW↑**	**TC↑**	**nDTW↑**
RCONCAT [2]	2.3	3.9	1.9	3.5	2.0	3.7	2.1	3.8
GA [3]	1.8	3.6	2.2	4.0	1.8	3.9	1.7	4.1
VLN Transformer [8]	2.3	4.7	3.1	5.2	3.6	6.2	3.5	6.1
ORAR [4]	8.50	11.13	**8.76**	11.74	23.88	34.34	22.12	32.69
Ours (+scene text)	9.25	12.83	8.12	11.73	**26.25**	**35.63**	25.25	35.49
Ours (+object)	**10.25**	**13.86**	8.63	**12.12**	25.87	35.27	**25.37**	**36.56**

2.6.3 Qualitative Results

This section presents visualizations and analyses of qualitative examples to evaluate the performance of the Object-Attention VLN (OAVLN) model. The results illustrate improvements in navigating real-world environments compared to the baseline model (ORAR), highlighting OAVLN's enhanced ability to utilize object features and scene text effectively.

Visualization of Trajectories. Figures 2.4 and 2.5 depict trajectories generated by the baseline and OAVLN models in outdoor VLN tasks. The red underlined text in the instructions corresponds to locations where the baseline model failed to make correct navigation decisions. Orange text represents object tokens. By leveraging object features, scene text, and language instructions, OAVLN demonstrates an improved ability to comprehend the environment and make reliable navigation decisions.

In Figure 2.4, the baseline model fails to execute correct turns due to ignor-

ing landmark references, while OAVLN accurately interprets the instructions and completes the turn successfully. Similarly, Figure 2.5 shows cases where the baseline stops prematurely, missing critical details in the instructions. In contrast, OAVLN effectively aligns object tokens with navigation decisions, ensuring correct stopping points.

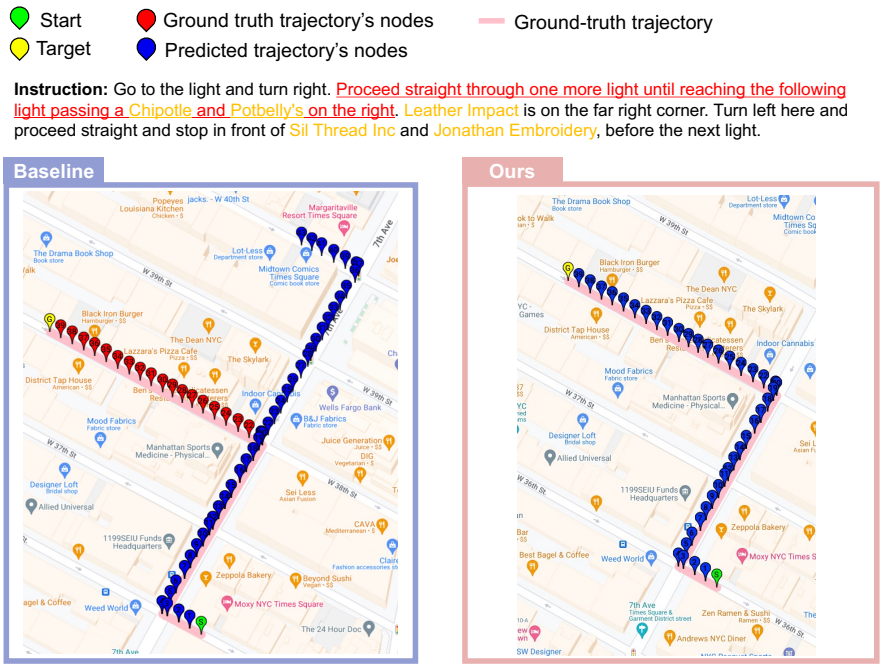
Analysis of Failure Cases. Figures 2.6, 2.7, and 2.8 present examples of failed navigation scenarios for the OAVLN model. These cases provide insights into the limitations and challenges in complex environments.

- **Stopping Near the Goal.** As shown in Figure 2.6, some failures occurred when the model stopped one step away from the goal, accounting for approximately 33% of the failure cases. Despite the failure, these results indicate that OAVLN often navigates very close to the intended destination.
- **Complex Road Conditions.** Figure 2.7 illustrates an instance where the agent chose an incorrect path in an environment with parallel roads. Such failures highlight the challenges of navigating in visually ambiguous conditions.
- **Confusing Instructions.** Figure 2.8 shows failures resulting from unclear or overly complex instructions, which confuse the agent and lead to navigation errors.

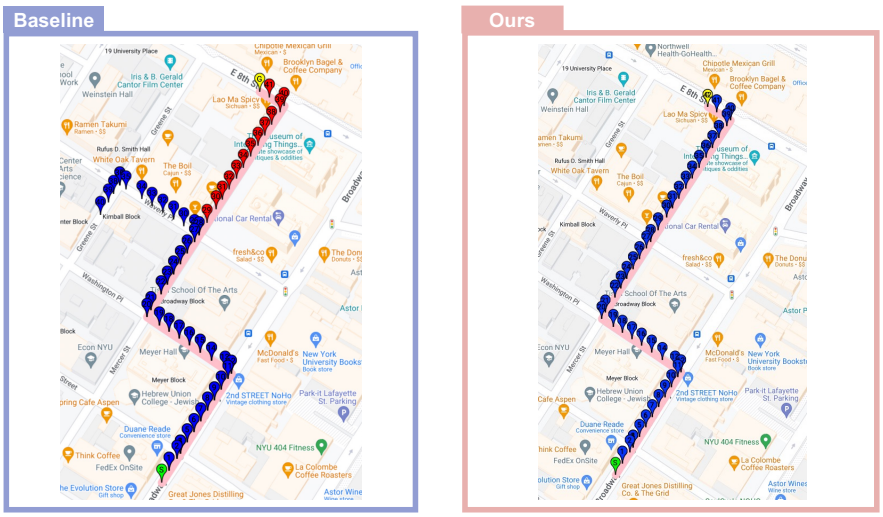
Stop and Turn Accuracy.

Table 2.4 Accuracy of Models at Stop and Turn Locations.

	Touchdown		map2seq	
	Stop	Turn	Stop	Turn
		Seen		
ORAR	72.70%	60.48%	50.89%	6.10%
Ours	**73.51%**	**62.94%**	**59.90%**	**12.00%**
		Unseen		
ORAR	92.26%	26.97%	71.15%	6.34%
Ours	**93.17%**	**28.90%**	**80.08%**	**12.25%**



(a) A case where the baseline model turns at the wrong place.



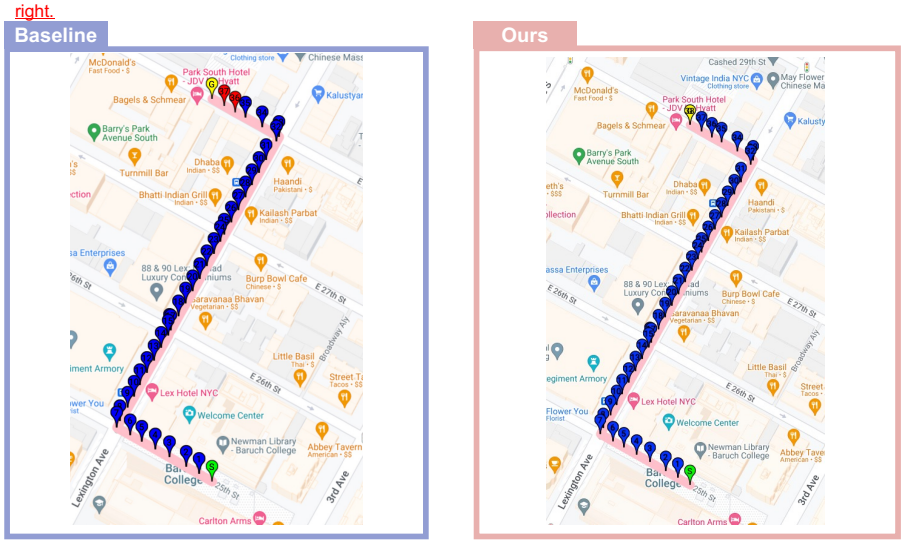
(b) A case where the baseline model turns at the wrong place.

Figure 2.4 Examples of incorrect turns by the baseline model. Left: trajectory generated by ORAR. Right: trajectory generated by the OAVLN model.

Instruction: Head to the first light and make a right. You will pass through two more lights and at the third light you will make a left. There will be a Starbucks on your left once you turn. Head down the street and stop in front of Scarlatto Restaurant. You have gone too far if you hit Hotel Edison.

(a) A case where the baseline model stops at the wrong place.

Instruction: Proceed to the traffic light and should see a library on the corner. Turn right and proceed straight through two more lights. At the third there is a Deccan Spice and Curry in a Hurry on the corners. Turn left here and proceed halfway down the block and stop near Copper Chimney on the left and a large parking area on the right.



(b) A case where the baseline model stops at the wrong place.

Figure 2.5 Examples of incorrect stops by the baseline model. Left: trajectory generated by ORAR. Right: trajectory generated by the OAVLN model.

Table 2.4 presents the stop and turn prediction accuracy of OAVLN compared to the baseline model. Incorrect stops are defined as cases where the agent stops

Instruction: Orient yourself with the red storefront on your right, go forward and make a right at the second intersection, with the blue store sign wrapping around the building on your right. Follow the traffic and make a right at the first intersection, with the green awning now on your left. Go straight and stop just before you pass the last red pole on the building to your left.

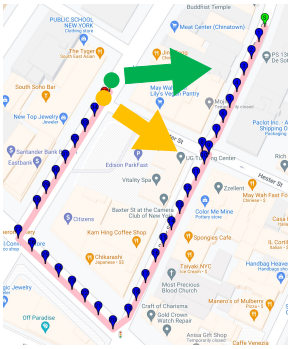


Figure 2.6 Failure case where the OAVLN model stops one step away from the goal.

Instruction: Turn so that the store with the red awning is on your right, and the green construction is on your left. Follow that road to the first intersection, pass through to the next intersection which is just after a short green fenced area. Turn right and follow that road a short ways passing the first available left turn. Stop when you are beside the first tree in the area between the roads on your left.

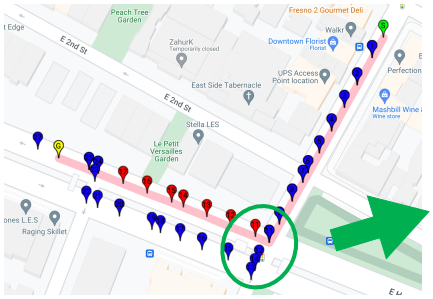


Figure 2.7 Failure case due to complex road conditions. Red arrow: correct path. Blue arrow: chosen path.

Instruction: Turn so the wooden doors are on the left side and you are going the same way as the cars. Go all the way into the intersection before you make a right turn. If you have short white poles on the right side then you turned too soon. After you make the right turn there will be concrete barriers on the left side of you. Go one block then make another right turn. You should have green sign on the left side and a purple sign on the right side. Go a little way down this one way street. When you are almost to a metal overhang on the right that is on a parking area come to a stop.

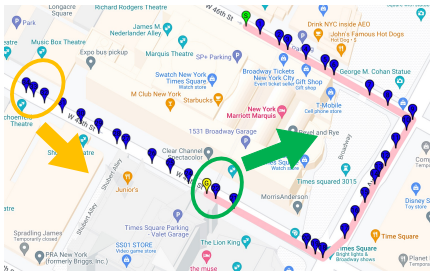


Figure 2.8 Failure case caused by confusing instructions.

within five steps of the goal but at an incorrect location. OAVLN demonstrates a significantly lower failure rate for both stop and turn predictions, highlighting its ability to align navigation actions with object tokens and instructions more effectively.

Token Masking Experiments. To further validate the model's focus on object tokens, token masking experiments were conducted. Object tokens were masked during testing, and the task completion rate was analyzed. As shown in Figure 2.9, OAVLN's task completion rate decreased significantly under masking conditions, indicating its reliance on object tokens for navigation. This contrasts with baseline models, which show less dependence on object features.

Attention heatmaps in Figure 2.10 further illustrate the difference. While ORAR focuses on the initial parts of instructions, OAVLN prioritizes object tokens and their temporal relevance, enhancing its decision-making capabilities.

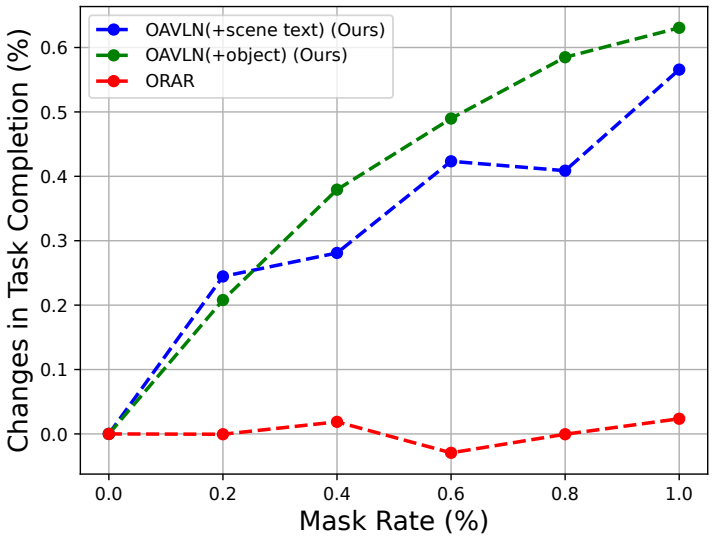
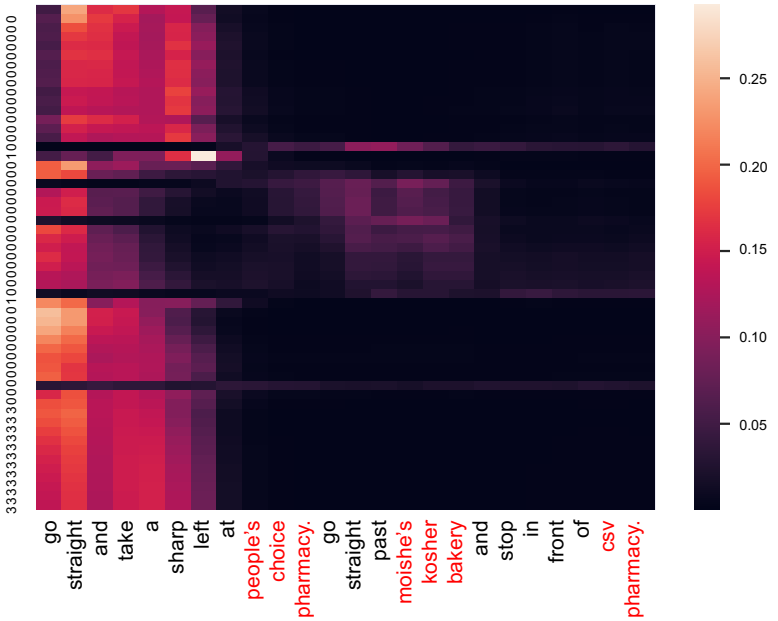


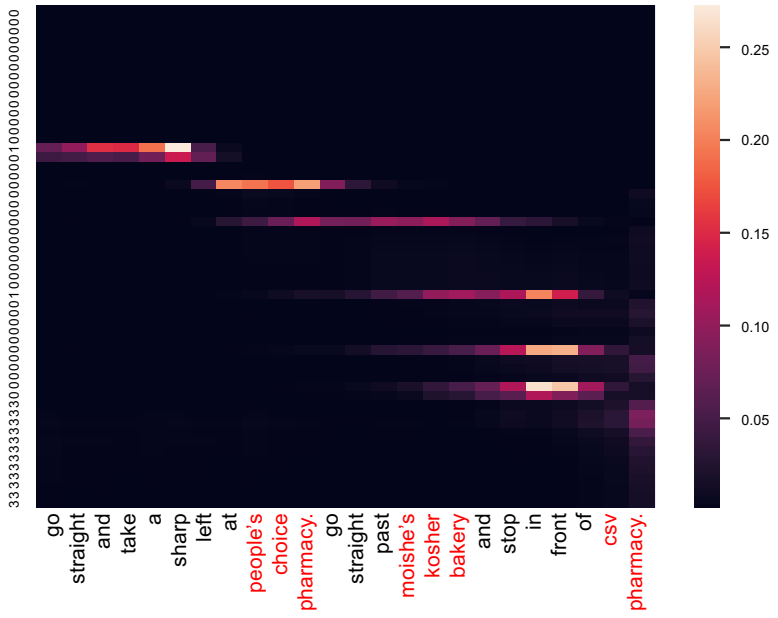
Figure 2.9 Changes in task completion rates when masking object tokens in instructions on the Touchdown dataset (seen scenarios).

2.6.4 Analysis

Our OAVLN works well in a seen scenario and an unseen environment, proving that the on-the-route object feature is helpful for outdoor VLN. The results shown



(a) Heatmap of ORAR



(b) Heatmap of OAVLN

Figure 2.10 Attention heatmap comparison between ORAR and OAVLN. Red text on the x-axis represents object tokens.

below indicated that the 'object feature' and 'scene text' are necessary. It can help the agent focus on on-the-route objects, enabling the agent's localization. Specif-

ically, OAVLN assists the agent in turning and stopping more accurately, which is a more intuitive approach. Therefore, even in unknown locations, OAVLN can use surrounding objects as references to reach the goal. Moreover, our work highlights the importance of leveraging contextual information, such as scene text, in navigation tasks. Our approach could serve as a starting point for future research in this area and inspire the development of more advanced models that can better use the contextual information available in real-world environments.

2.7 Conclusion

This chapter addressed the limitations of current outdoor Vision-and-Language Navigation (VLN) models, particularly their inability to effectively leverage object tokens, which often leads to navigation failures. The proposed Object-Attention VLN (OAVLN) model was introduced to incorporate on-the-route object information, significantly improving outdoor VLN performance. Extensive experiments on two large-scale datasets demonstrated that the OAVLN model consistently outperformed existing methods in both seen and unseen scenarios. Furthermore, qualitative visualizations revealed how the model effectively learns to prioritize object features, resulting in a more nuanced understanding of the environment and enhanced navigation accuracy.

Limitations. While the OAVLN model achieves notable improvements, it is not without limitations. First, the model is susceptible to data biases in both the scene text encoder and the object encoder, which can impact its ability to generalize effectively to novel scenarios. Second, the computational requirements for training and deploying the model are significant, potentially limiting its practical applications in resource-constrained settings. Addressing these challenges will require further research into efficient encoding techniques and strategies to mitigate data biases.

Future Directions. Future research could focus on reducing the computational cost and training time of the model, making it more accessible for practical applications. Additionally, integrating advanced techniques to enhance generalization across diverse environments, such as domain adaptation or robust feature extrac-

tion, could further improve the performance of the model. Exploring alternative datasets that capture a wider variety of environmental dynamics and leveraging more efficient model architectures may also provide promising directions for overcoming the current limitations.

Chapter 3

The STVchrono Dataset: Continuous Change Recognition in Time

In this chapter, the focus is on addressing the lack of datasets for recognizing long-term environmental changes, a key challenge discussed in Chapter 1. Understanding continuous changes in real-world scenarios is essential for applications such as urban planning, environmental science, agriculture, and cultural heritage preservation. Existing datasets primarily emphasize discrete changes between two images, such as object addition or removal, often relying on synthetic or constrained real-world data. These limitations hinder progress in recognizing gradual and continuous changes that unfold over extended time periods.

To bridge this gap, this chapter introduces the STVchrono dataset, a novel benchmark specifically designed for long-term continuous change recognition. The dataset comprises 71,900 photographs spanning 18 years across 50 cities worldwide, offering diverse geographic and temporal coverage. It supports three primary tasks: continual change captioning for image pairs and sequences, and change-aware sequential instance segmentation. These tasks aim to evaluate models' ability to describe and recognize changes over time in real-world scenes.

Extensive experiments are conducted to assess the effectiveness of existing methods, including multimodal Large Language Models (LLMs) and state-of-the-

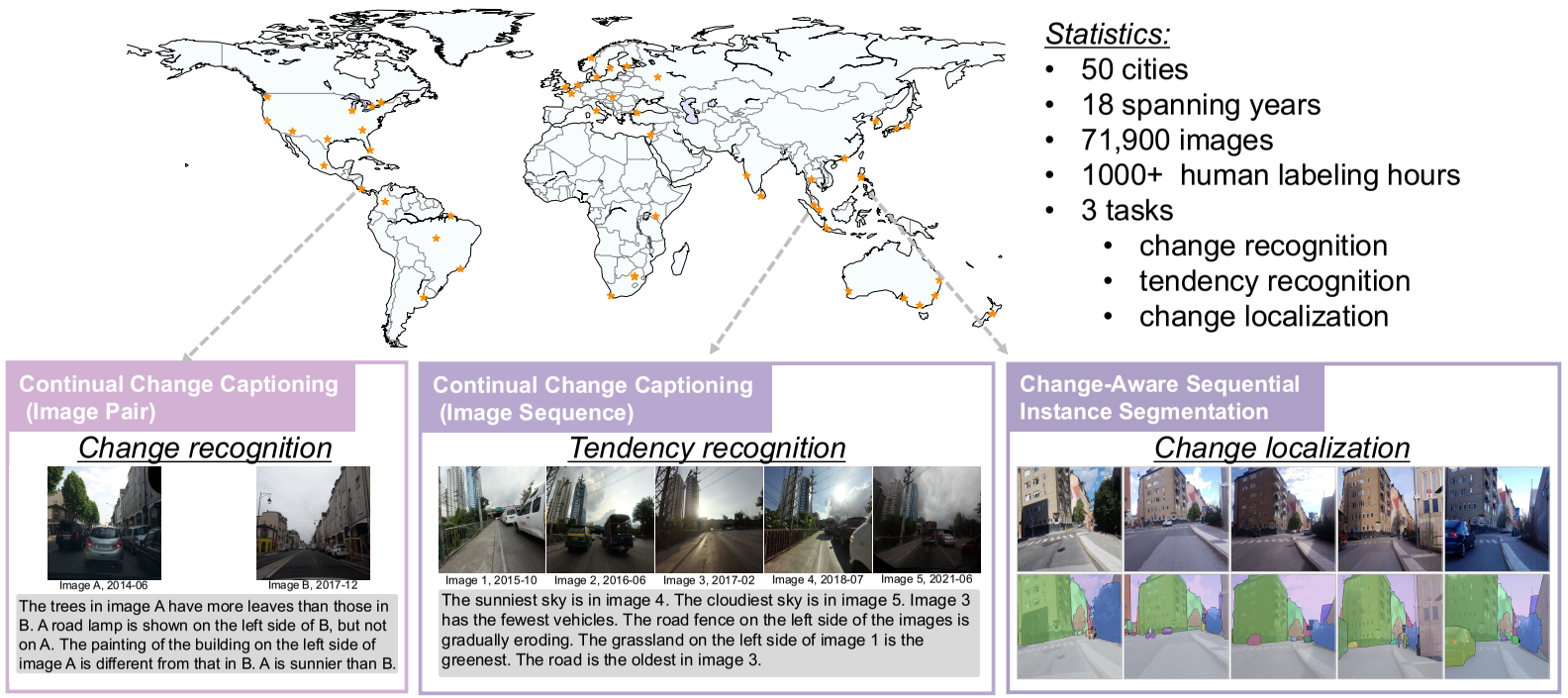


Figure 3.1 Overview of the proposed STVchrono dataset.

art segmentation models, using the STVchrono dataset. The findings highlight the limitations of current approaches in handling real-world continuous changes, underscoring the importance of this new benchmark in advancing research in this domain.

3.1 Introduction

The environment around us evolves continuously due to natural processes, human activities, and technological advancements. Recognizing these changes is pivotal for applications in various fields, such as urban planning, environmental science, agriculture, and cultural heritage preservation. Continuous change recognition can reveal insights into historical patterns, assist in analyzing ongoing trends, and inform future decision-making.

Real-world changes may include different spatial and temporal changes in the natural landscape (e.g. water volume in the river), urban infrastructure (e.g. road width), weather conditions (e.g. season change), or population dynamics (e.g. type of human activities). What matters the most is the continuous and dynamic nature of all these change types. Recognition of continuous changes can provide valuable insights into past historical events, support current trend analysis, and facilitate future planning.

Currently available tasks related to scene change understanding focus on change detection and change description. While the target of change detection is to find changed regions within a scene, change description deals with the generation of language captions for the detected changes. The existing change datasets [53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63] mostly focus on recognizing discrete changes between paired images or 3D point clouds, overlooking the importance of continuous, gradual changes, occurring over long time periods. Additionally, these datasets either include synthetic data [59, 60, 61, 62] (artificially generated images from simulated environments) or concentrate on simplified real-world scenes (like tabletop rearrangement in [53]). Thus, these datasets are not suitable for understanding the real-world continuous changes.

To address the above-mentioned limitations, this chapter proposes a novel

benchmark STVchrono (STreet View chrono) dataset. STVchrono is designed to facilitate the understanding of long-term continuous changes in the real world. To capture continuous outdoor changes, the Mapillary [1] platform was utilized for data collection. Specifically, 71,900 photographs of 50 different cities over a span of 18 years (2006 to 2023) were collected. The chosen 50 cities vary in location (spread across various continents) and encompass different landscape types (urban and rural areas).

The STVchrono dataset is suitable to facilitate three change understanding tasks (Figure 3.1): continual change captioning for image pairs and image sequences, and change-aware sequential instance segmentation (for change recognition). The aim of continual change captioning for an image pair is to describe the content of the change between a pair of images, taken in the same location but at two different times. These changes may include variations in color, age, volume, or condition for 10 object types (Table 3.2). Another type of continual change captioning task deals with the longer image sequences (3-6 images) taken over a span of several years. This task involves evaluating the degree of the change, its progression over time, and visible trends (Table 3.2). The primary objective of the change-aware sequential instance segmentation task is to identify and track object instances within a set of 5 images, taken over different time intervals in the same location.

Extensive experiments using STVchrono evaluate state-of-the-art methods for change detection and captioning, including multimodal Large Language Models (LLMs). While LLMs demonstrate superior performance in describing changes compared to traditional methods, they still fall short of human-level accuracy. Similarly, segmentation methods tested on the change-aware sequential instance segmentation task reveal considerable room for improvement. These results underscore the complexity of continuous change recognition and highlight the need for further advancements in this area.

By providing a robust benchmark for evaluating models on real-world continuous change recognition, the STVchrono dataset aims to bridge the gap between current methods and the demands of real-world applications, driving progress in

[1] https://www.mapillary.com/app/

this critical area of research.

3.2 Related Works

3.2.1 Change Understanding Datasets

Currently, available change understanding datasets primarily concentrate on two main tasks: the detection of changed regions within a scene and the linguistic description of the change content. The KTH Meta-rooms [64] and tvtable [53] datasets facilitate change detection in the robotics field between pairs of 3D point clouds of indoor rooms and tabletop surfaces, correspondingly. The Change3D [55], Panoramic Change Detection [56], and SOCD [57] datasets are suitable for the street-view scene recognition. While [55] consists of 3D point cloud pairs, [56] and [57] works in 2D and use semantic masks and bounding boxes for change detection, correspondingly. Another set of four datasets was recently proposed for 2D change detection: COCO-Inpainted, Synthtext-Change, Kubric-Change, and VIRAT-STD [58]. The 3DCD [54], EGY-BCD [65], and ChangeNet [66] datasets, aim for change detection in satellite remote sensing.

The CLEVR-Change [59] and CLEVR-Multi-Change [60] datasets focus on captioning single and multiple changes in synthetic image pairs, whereas the TRANCE [61] and OVT [62] datasets represent changes and their temporal orders using triples and graphs. Real-world datasets include Spot-the-Diff [63] (surveillance) and LEVIR-CC [67] (aerial imagery). Research also covers change detection in multi-view images [68] and 3D point clouds [69, 70]. Additionally, [71] introduced a Visual Room Rearrangement task, where agents rearrange a room to its original layout by interacting with changed objects.

Existing datasets for change understanding often focus solely on detecting or describing discrete changes in static image pairs. In contrast, our STVchrono dataset captures continuous, gradual changes over time using sequences of 2-6 images and is created from historical photographs of 50 different cities around the world (Table 3.1).

Table 3.1 Comparision of the STVchrono against existing change detection (top ten rows) and change description (four middle rows) datasets.

Dataset	Environment	# change pair	# city	Time span	Sequence length	Real image	Discrete change	Continuous change	Human-labeled caption	Change detection
Meta-rooms [64]	indoor	588	-	days	2	✓	✓	✗	✗	✓
Change3D [55]	outdoor	866	1	4 years	2	✓	✓	✓	✗	✓
SOCD [57]	outdoor	15,000	-	-	2	✗	✓	✗	✗	✓
COCO-Inpainted [58]	in- & out-door	60,000	-	-	2	✓	✓	✗	✗	✓
Synthtext-Change [58]	outdoor	5,000	-	-	2	✗	✓	✗	✗	✓
Kubric-Change [58]	outdoor	1,605	-	-	2	✓	✓	✗	✗	✓
VIRAT-STD [58]	in- & out-door	1,000	-	hours	2	✓	✓	✗	✗	✓
3DCD [54]	satellite	472	1	7 years	2	✓	✓	✓	✗	✓
EGY-BCD [65]	satellite	6,091	1	8 years	2	✓	✓	✓	✗	✓
ChangeNet [66]	satellite	31,000	100	9 years	6	✓	✓	✓	✗	✓
CLEVR-Change [59]	table	79,606	-	-	2	✗	✓	✗	✗	✗
CLEVR-Multi-Change [60]	table	60,000	-	-	2	✗	✓	✗	✗	✓
Spot-the-Diff [63]	outdoor	13,192	1	hours	2	✓	✓	✗	✓	✗
LEVIR-CC [67]	satellite	10,077	1	15 years	2	✓	✓	✓	✓	✗
STVchrono (our)	outdoor	19,400	50	18 years	2-6	✓	✓	✓	✓	✓

3.2.2 Change Understanding Methods

State-of-the-art change captioning methods, such as DDLA [63], DUDA [59], MCCFormers [60], M-VAM [72] and CLIP4IDC [73] compute differences either at the pixel- [63] or feature-level [59] or use transformers [60, 72, 73] to correlate image pairs. Models like VARD-Trans [74] and SCORER [75] focus on identifying consistent features in images with viewpoint shifts. In the field of change detection task, two recent studies [58] target identification of change regions with viewpoint differences. [58] introduces a co-attention-based approach for identifying correspondences between image pairs, while [58] relies on depth map generation for image correlations. Despite numerous existing methods, most of them focus on 2D image pairs or 3D data pairs and overlook serial-image change recognition. Recent studies highlight the potential of LLMs in context reasoning, but their application in change recognition remains unexplored. Our data delves into change recognition in serial images, encompassing captioning, change region detection, and the usage of LLMs in this realm.

3.2.3 Image Sequence Recognition Datasets

Alongside change understanding, various datasets support image pair or image sequence recognition tasks: NLVR [76] and NLVR2 [77] for difference reason-

ing, and GeneCIS [78] and VisualDNA [79] for image similarity. Similar to STVchrono, Mapillary [80] and [81] datasets utilize image sequences taken over different time periods for place recognition and robust aerial place representation, correspondingly. SatlasPretrain [82] is a temporal and spatial remote sensing dataset for remote sensing image analysis. In contrast, STVchrono focuses on identifying and describing regions of long-term continuous change.

3.2.4 Instance Segmentation Methods

Instance segmentation aims to identify and outline distinct objects in visual content through pixel masks. The Mask R-CNN [**?**] method improved upon Faster R-CNN [83] by adding mask prediction. Subsequent methods like MaskFormer [84] incorporated transformer technology to enhance accuracy. Recent studies, such as Mask2Former [85], Mask DINO [86], and UNINEXT [87] have merged instance, semantic, and panoptic segmentation into unified models for simultaneous segmentation across various levels. Mask2Former has been adapted to 3D masked attention for video instance segmentation [85], while Ying *et al.* [88] propose the CTVIS method by adding a memory bank to maintain consistency across frames. Alternatives like Seq2Former [89] and DVIS [90] have developed trackers to preserve temporal continuity in image-level segmentation results. Our work introduces a change-aware instance segmentation for image sequences, tracking the evolution of natural scenes over years, thus extending beyond the typical short-term focus of existing video segmentation methods.

3.3 The STVchrono Dataset

The STVchrono dataset uniquely localizes and describes details of ongoing, extensive changes across space and time, going beyond the discrete changes (such as add, delete, or move) identified by current datasets (Table 3.1). It addresses both easily labeled discrete changes and complex continuous gradual shifts, which are hard to quantify with labels. Encompassing shifts in weather patterns, seasonal transitions, vehicular movement, and city architecture, the STVchrono dataset captures the dynamics of the real-world environments (Figures 3.1 and 3.2, This

(a) Seasonal changes

(b) Vegetation growth

(c) Building exterior changes

(d) Building construction stages

(e) Road and building maintenance and development

Figure 3.2 Different change types contained in the STVchrono dataset.

dataset encompasses a wide range of changes, including natural changes (e.g. (a) and (b)), as well as changes related to infrastructure and construction (e.g. (c), (d), and (e)).), supporting three distinct tasks related to these changes:

- **Continual Change Captioning (Image Pair)** aims at the recognition of the change details between 2 images taken at 2 distinct time periods (Figure 3.1, left). Examples of such changes can include the appearance of new cars, the removal of road signs, or a change in a building color.
- **Continual Change Captioning (Image Sequence)** focuses on the change tendencies over a sequence of 3-6 images taken at different time periods (Figure 3.1, middle). It offers insights into patterns, progressions, and trends over an extended time period, like the growth of plants.
- **Change-Aware Sequential Instance Segmentation** is suitable for the detection, understanding, and tracking of the change regions (Figure 3.5), ensuring a comprehensive analysis of the change dynamics for the specific object instances over a long time period.

3.3.1 Image Collection

The STVchrono dataset was collected using the Mapillary API. Mapillary was chosen for its repository of images from diverse global locations captured over many years, enabling an in-depth analysis of temporal historical changes. Specifically, images were selected for 50 different cities, spanning 18 years: from 2006 to 2023 (Figure 3.1). OpenStreetMap [2] was employed to determine the boundaries of each city and then randomly sampled 300 to 1,000 latitude and longitude coordinates within these city limits. In the preliminary phase of the dataset creation, all available images for these coordinates (each with a resolution of 640x640 pixels) were retrieved. Subsequently, images containing projection-related distortions that hindered annotators from recognizing changes, as well as coordinates yielding fewer than two images, were excluded. The resulting dataset comprises 71,900 photographs. Depending on the specific caption or detection task, relevant images were handpicked and manually annotated from the preliminary collection.

[2] https://nominatim.openstreetmap.org/search

Table 3.2 Annotation guidelines for the continual change captioning.

Subject	Attributes	Dataset example
Weather	Conditions, brightness, color	Image A is sunny, while image B is cloudy. **(IP, distinction)**
Tree	Growth pattern, color, volume, presence/absence	The tree on the right side becomes progressively thicker. **(IS, tendency)**
Building	Construction stages, age, cleanliness, heights, exterior alterations	Image 1 has the newest building on the left side. **(IS, superlative)**
Road	Age, cleanliness, width and volume, number and presence/absence of roads, cars, and traffic signs	In Images 1 and 2, a road is visible on the left; in Images 3 and 4, it disappears. **(IS, similarity)**
Lawn / Grassland	Color variations, volume, growth rates, transitions, presence/absence	The lawn on the right side looks greener in image B than in image A. **(IP, distinction)**
Soil / Land	Color variations, volume, transitions, presence/absence	The land on the right side of the sidewalk turned into a lawn from images 2 to 5. **(IS, tendency)**
River	Color variations, volume, transitions, presence/absence	The river is the cleanest in image 3. **(IS, superlative)**
Road fence	Age, color, cleanliness, height, presence/absence	The fence gate is not visible in image 1 but is present in images 2 and 3. **(IS, similarity)**
Human	Number, type and nature of activities, presence/absence	In image A, someone walks on the road; in image B, someone sits on a bench. **(IP, distinction)**
Animal	Number, type and nature of activities, presence/absence	There is a cat on the road in Image 3, but it is absent in the other images. **(IS, similarity)**

3.3.2 Continual Change Captioning (Image Pair)

The goal of this task is to describe in detail the visual differences between two street view images taken at 2 different time periods. For this purpose, 15,000 image pairs (a total of 30,000 images) were selected from the STVchrono set of 71,900 images. Crowdsourcing platforms were employed to gather human annotations in English. Each image pair received three to eight descriptive sentences detailing the dominant changes from one human annotator, while another annotator verified the effectiveness of these descriptions. An image pair annotation was approved only after the validation received from the second annotator.

Considering the possibility of numerous changes between two images, the focus was placed on 10 dominant subjects commonly found in street view images, such as "weather" and "tree", to be featured in the change captions. For each subject, the annotations specifically addressed the **distinction** in various aspects, including color, age, volume, or condition. The comprehensive annotation guidelines are presented in Table 3.2. The image pair task involves comparing two images, labeled A and B, to identify attribute distinctions. The image sequence task requires analyzing a series of 3-6 images to detail tendencies, superlatives, and similarities. The series start with the earliest image, designated as Image A and number 1 (IP: image pair; IS: image sequence). Additionally, annotators were instructed to report dominant changes that go beyond the guidelines, allowing for a more open-ended approach to change recognition. Figure 3.3 shows an example of continual change captioning (image pair).

3.3.3 Continual Change Captioning (Image Sequence)

The objective of the continual change captioning (image sequence) task is to narrate the progression of changes observed in a series of 3-6 images, captured at the same location over several years. From the 71,900 images in the STVchrono dataset, 19,800 images were utilized, divided into 4,400 sequences. These images were grouped into four categories, each containing 1,100 sequences with 3, 4, 5, and 6 images, respectively. Human annotators were asked to focus on the same 10 change aspects identified in the continual change captioning (image pair)

Image A, 2015-11 Image B, 2016-07

Ground truth:
- A is cloudy, B is sunny and more bluish.
- The trees on the left side in B are more flourish than in A.
- B has more cars and humans than A.
- A woman is wearing with cloth on the left side of B but not in A.

Figure 3.3 An example of continual change captioning (image pair) in STVchrono

task. Each image sequence received annotations, which were then validated by two separate annotators. The annotations were directed to capture the **tendency, superlative, and similarity** in color, age, volume, or condition across various change aspects, as outlined in Table 3.2. Figure 3.4 shows an example of continual change captioning (image sequence).

3.3.4 Change-Aware Sequential Instance Segmentation

The central goal of the consistent sequential instance segmentation task is to identify and track specific subject instances, within image sequences, captured at the same location over different time intervals. 520 sequences were selected, representing a variety of cities and coordinates. Each sequence includes five images taken at different times (yielding a total of 2,600 images). Human annotators manually marked the instance regions and labels for each image. This task is particularly crucial for monitoring long-term trends such as the increase or decrease in vegetation, changes in river width, and the construction or demolition of buildings. A key challenge of this task is maintaining consistent instance labels for the same subjects despite their transformations over time. Labels were pro-

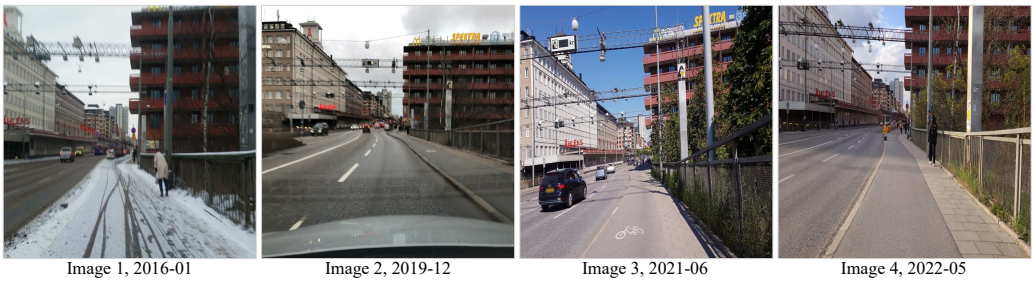


Image 1, 2016-01 | Image 2, 2019-12 | Image 3, 2021-06 | Image 4, 2022-05

Ground truth:

- Image 1 has snow but others do not.
- Image 3 has the sunniest sky.
- There are people shown in Images 1, 2, and 4, but not in 3.
- The trees in Image 3 have more leaves.
- Images 3 and 4 have grassland on the right side.
- The road fence on the right side of Image 4 shows white because of the sunshine.

Figure 3.4 An example of continual change captioning (image Sequence) in STVchrono

Figure 3.5 Two examples of image sequences (top) and their annotations (bottom) for the change-aware sequential instance segmentation task. Objects with consistent IDs share the same segmentation mask colors within each sequence.

vided for 12 subject categories, including vehicle (car_bus), building, tree, road, sky, lawn/grassland, soil/land, road fence, motorbike, bicycle, human, and animal. Two examples illustrating the task are shown in Figure 3.5.

3.3.5 Dataset Statistics

To ensure a comprehensive representation of street view changes, 50 different cities were selected from around the globe for our STVchrono dataset image collection. The distribution encompasses 14 cities in Asia, 13 in Europe, 8 in North America, 6 in South America, 6 in Oceania, and 4 in Africa. Istanbul was included in both the Asian and European tallies because of its transcontinental position. The dataset was split by cities into train and test sets, with ratios of 38/12 for image pair and sequence captioning, and 22/8 for the segmentation task. For the two change caption tasks, the dataset boasts a vast range of vocabulary due to the fully human-annotated sentences. Specifically, the total vocabulary encompasses 1,223 unique words, with an average of 35.98 words per caption for the image pair task, and 50.65 words per caption for the image sequence task.

A comparative analysis of the STVchrono dataset with existing datasets is summarized in Table 3.1. The STVchrono dataset is the first of its kind to capture ongoing changes on a global scale (50 cities) and to consider the trends within sequences of images (2-6 images). It facilitates not only the detection of changes but also the recognition of change content through detailed human-labeled sentences.

Word Distribution I analyze the word distribution in the captions of two setups for the continual change captioning tasks: image pair (Figure 3.6, left) and image sequence (Figure 3.6, right) using WordCloud visualization [3]. Both setups feature a wide variety of words. The image pair task requires identifying differences between two images, leading to captions that include a greater number of comparative words, such as “brighter”, “greener”, “cleaner”, and “different”. Conversely, the image sequence task involves recognizing trends, superlatives, and similarities across image sequences. As a result, the dataset contains a higher frequency of relative terms like “newest”, “thickest”, “clearest”, and “gradually”.

Sentence Length Distribution The sentence length (change caption length per

[3] https://amueller.github.io/word_cloud

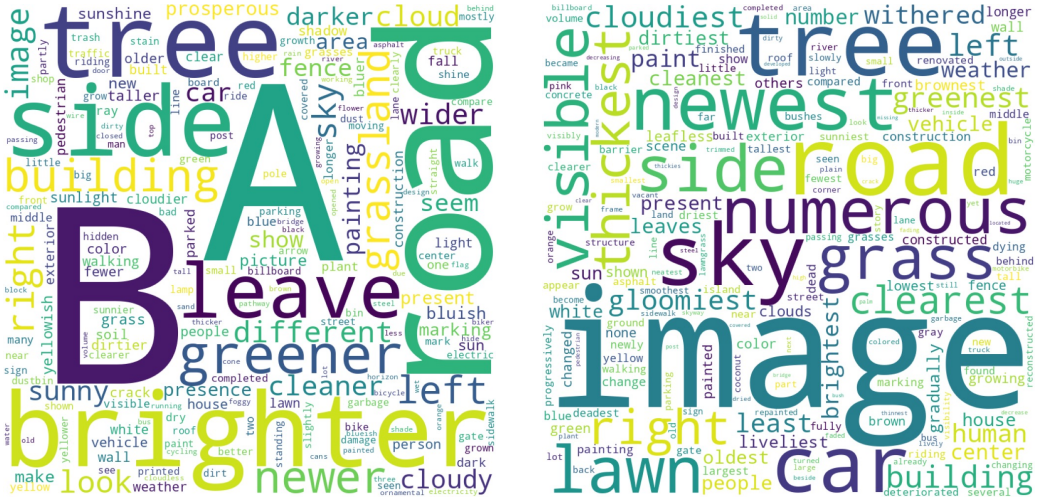


Figure 3.6 Wordcloud visualization of the continual change captioning (image pair) task (left) and the continual change captioning (image sequence) task (right) of the STVchrono dataset.

dataset instance) distribution for the two continual change captioning tasks is presented in Figure 3.7. Both dataset setups exhibit a long-tailed distribution. Specifically, the image pair task has an average sentence length of 35.98, while the image sequence task, involving more images, has an average sentence length of 50.65. Due to the minimum sentence number requirement set for sequences (three for 3-image and 4-image sequences, and five for 5-image and 6-image sequences), there are two distinct peaks in the sentence length distribution for the continual change captioning (image sequence). Additionally, both datasets feature a significant number of instances with longer sentences, offering a wide array of detailed changes in the image pairs and sequences for model training and evaluation.

Time Deltas Distribution Figure 3.8 describes the distribution of time deltas (spanned years of each dataset instance) for the three tasks of the STVchrono dataset. All three tasks encompass instances with a wide range of time deltas.

3.4 Experiments

This section evaluates the proposed STVchrono dataset by benchmarking existing state-of-the-art methods across its continual change captioning (image pair and

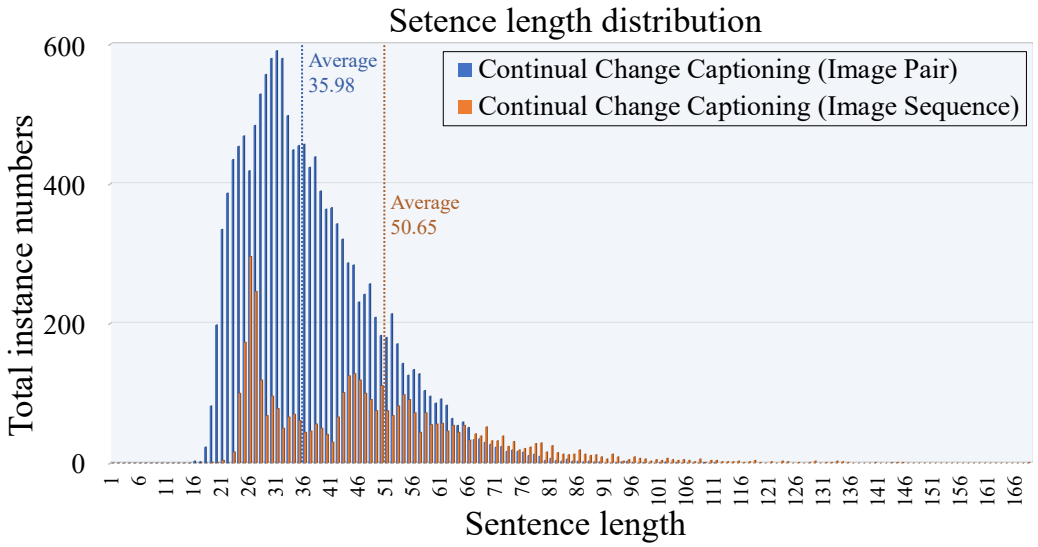


Figure 3.7 Sentence length distribution of two continual change captioning tasks of the STVchrono dataset.

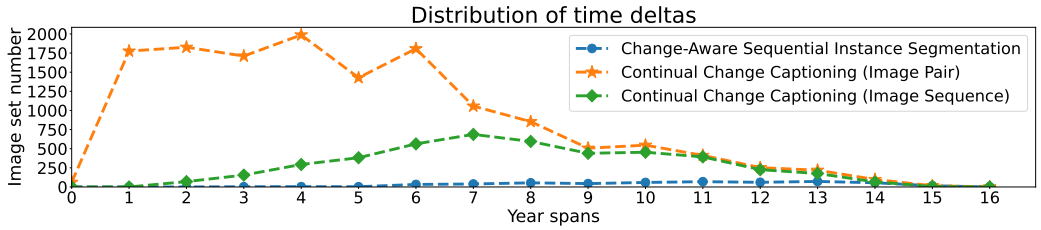


Figure 3.8 Distribution of the time deltas of the STVchrono dataset.

image sequence) and change-aware sequential instance segmentation tasks. Experiments were designed to assess the performance of both traditional and multimodal LLM-based approaches while highlighting the effectiveness of the dataset in facilitating continuous change recognition.

3.5 Experiments

This section evaluates the proposed STVchrono dataset by benchmarking existing state-of-the-art methods across its continual change captioning (image pair and image sequence) and change-aware sequential instance segmentation tasks. Experiments were designed to assess the performance of both traditional and multimodal LLM-based approaches while highlighting the effectiveness of the dataset in facilitating continuous change recognition.

3.5.1 Baseline Methods

Continual Change Captioning. I evaluated the performance of five state-of-the-art change captioning methods: DUDA [59], MCCFormers-D, MCCFormers-S [60], CLIP4IDC [73], and VARD-Trans [74], on both continual change captioning tasks (image pair and image sequence). To explore the potential of multimodal Large Language Models (LLMs) in this domain, two recent LLM-based methods were also included: OpenFlamingo [91] and BLIP2 [92] combined with GPT4 [93].

Change-Aware Sequential Instance Segmentation. As no existing methods specifically target change-aware sequential instance segmentation, two state-of-the-art video instance segmentation models were adapted: Mask2Former [85] and CTVIS [88]. These models were modified to track object instances, such as roads, trees, or buildings, across sequential images instead of videos. Experiments were conducted using various backbones, including ResNet50 [94], ResNet101 [94], and Swin Transformer (SwinT-S, SwinT-L) [95], to evaluate their performance.

3.5.2 Implementation Details

Out-of-the-box implementations of DUDA [59], MCCFormers-D, MCCFormers-S [60], and CLIP4IDC [73] were used and VARD-Trans [74] for the continual change captioning (image pair) task. For the continual change captioning (image sequence) task, MCCFormers-S and CLIP4IDC were explored, as both methods allow the sequential input. The initial learning rate was set as 10^{-4} and adopted the Adam optimizer. All methods were trained for 80 epochs for captioning tasks and 50 epochs for the segmentation task. For evaluation of OpenFlamingo [91] and BLIP2 [92] + GPT4 [93], different prompts were designed as following.

3.5.3 Prompt Design

Specifically, detailed prompts were designed for use with OpenFlamingo and BLIP2 + GPT4 in continual change captioning tasks.

OpenFlamingo. Custom prompts for OpenFlamingo were designed based on its official image captioning templates. Figure 3.9 illustrates the prompt for image pairs, while Figure 3.10 shows the design for image sequences. Experiments varied the number of input examples (3, 5, 10, 15, and 20) and output formats, which included:

- complete sentences (e.g., *"B building is clearer than A. B grassland is greener than A. Road B is newer than Road A. B is darker than A."*).
- itemized structures (e.g., *'building': ['item': 'Old and new', 'answer': 'B building is clearer than A.'], 'human': [], 'grassland': ['item': 'Color', 'answer': 'B grassland is greener than A.'], 'road': ['item': 'Old and new', 'answer': 'Road B is newer than Road A.'], 'road fence': [], 'tree': [], 'weather': ['item': 'Light and darkness', 'answer': 'B is darker than A.']*).

BLIP2 + GPT4. For BLIP2 and GPT4, in-context examples (Fig. 3.11) for BLIP2 were combined with GPT4's system messages. Since GPT4 cannot directly process images, BLIP2 was used to extract visual attributes such as color

Prompt for OpenFlamingo

Example

<image>This is image A and taken at {prompt_imgA_year} .
<image> This is image B and taken at {prompt_imgB_year}.
The changes of A and B are: {prompt_cap} <|endofchunk|>.

Query

<image> This is image A and taken at {query_imgA_year} .
<image> This is image B and taken at {query_imgB_year}.
The changes of A and B are:

Figure 3.9 Prompt design for OpenFlamingo (image pair).

Prompt for OpenFlamingo

Example

<image> This is image 1 and taken at {prompt_img1_year} .
<image> This is image 2 and taken at {prompt_img2_year}.
<image> This is image 3 and taken at {prompt_img3_year}.
...
The changes tendency of 1, 2, 3 … are: {prompt_cap} <|endofchunk|>.

Query

<image> This is image 1 and taken at {query_img1_year} .
<image> This is image 2 and taken at {query _img2_year}.
<image> This is image 3 and taken at {query _img3_year}.
...
The changes tendency of 1, 2, 3 … are:

Figure 3.10 Prompt design for OpenFlamingo (image sequence).

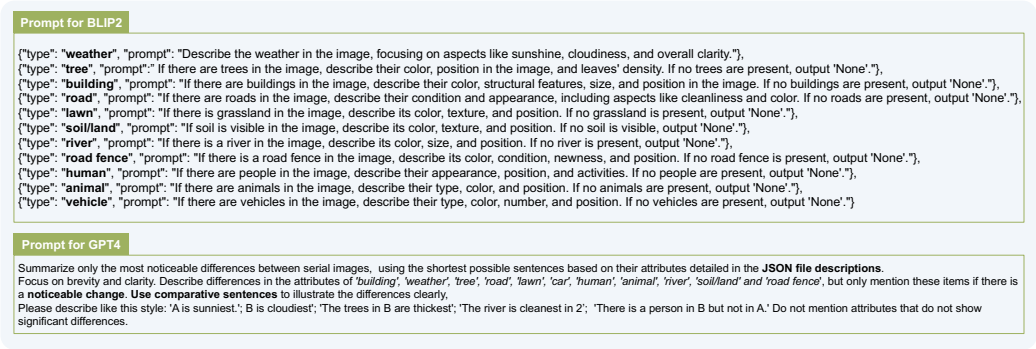


Figure 3.11 Prompt design for BLIP2 + GPT4.

and age, storing the results in JSON files. These JSON files were subsequently parsed by GPT4 to generate comprehensive change descriptions.

3.5.4 Evaluation Metrics

For evaluation of the generated change captions, standard captioning metrics were employed, including BLEU4 [96] and CIDEr [97], to assess the similarity between generated and reference captions. Additionally, GPT4 [93] evaluation was utilized to emphasize meaning similarity over sentence structures. The number of sentences in the STVchrono dataset's ground truth captions is limited to 3-8 reference captions per image sequence. As this number might not be enough to describe all the changes within the image sequence, human ratings were further implemented to manually assess the accuracy and coverage of the generated captions. Accuracy is calculated as the proportion of correct change descriptions relative to total changes, while coverage is the average number of correctly captured changes per image sequence. Human ratings were provided for the randomly sampled 100 sequences for each evaluated method. The standard Average Precision (AP) metric was used for the evaluation of the generated instance segmentation masks.

3.5.5 Results of Continual Change Captioning

Image Pair The comparison of the selected baseline models and multimodal LLMs for this task is presented in Tab. 3.3 and Figure 3.12. Among all baselines

(DUDA, MCCFormers -D and -S, CLIP4IDC, and VARD-Trans), CLIP4IDC achieves the highest BLEU4, CIDEr, and GPT4 scores, with 28.5, 69.5, and 32.4 points respectively. This performance is attributed to the large dataset size on which the model was pre-trained. OpenFlamingo and BLIP2+GPT4 show relatively low BLEU4 and CIDEr scores, while obtaining higher scores on GPT4 and human ratings. This is because these methods do not undergo a training process, tending to predict sentences with structures that differ from the ground truth sentences. In Figure 3.12, all methods capture only one to two changes. The highest human rating results come from CLIP4IDC and OpenFlamingo, but their best accuracy score of 47.8 and coverage score of 1.85 are extremely low, indicating that the models struggle to recognize changes within the images from the STVchrono dataset correctly.

Table 3.3 Change description evaluation on continual change captioning (image pair).

Methods	BLEU4↑	CIDEr↑	GPT4↑	Accuracy↑	Coverage↑
DUDA [59]	21.7	39.1	26.3	32.7	1.1
MCCFormers-D [60]	22.4	52.7	29.8	39.8	1.34
MCCFormers-S [60]	25.4	51.3	26.8	35.9	1.28
CLIP4IDC [73]	**28.5**	**69.5**	32.4	**47.8**	1.74
VARD-Trans [74]	16.4	19.4	21.9	28.3	1.0
OpenFlamingo [91]	7.8	37.3	**37.9**	43.8	**1.85**
BLIP2 [92] + GPT4 [93]	4.2	16.1	33.1	37.5	1.52
Human	21.2	50.8	40.0	94.4	3.58

Image Sequence Experimental results for this task are in Table 3.4 (averaged for all sequences from 3 to 6 images). As for the image pair continual change captioning task, multimodal LLM-based methods exhibited lower scores in BLEU4 and CIDEr but achieved better GPT4 scores and human ratings.IDEr, but achieved better GPT4 score and human ratings. Specifically, BLIP2 + GPT4 scored the highest in GPT4, while OpenFlamingo averaged nearly 1.10 changes detected with higher accuracy (34.4 points). Figure 3.14 presents OpenFlamingo and BLIP2 + GPT4 correctly identifying changes. Compared to change recognition from image pairs, all methods demonstrated reduced performance, when recognizing changes from

Image A, 2015-11

Image B, 2016-07

Ground truth: B is sunnier than A. The trees in B have more leaves than A. Road A is newer than B because A has new markings. A has more cars than B. There is a bicycle on the left side of A, but not on B.

MCCS: B is sunnier than A. There is a building on the left side of A but not on B. Road B is newer than A. A has more leaves than B.

CLIP4IDC: Road B is newer than A. A has more cars than B. A has more humans than B.

OpenFlamingo: Road B is newer than A, due to its new markings. B has more trees, and is brighter than A, aligning with B being sunnier.

BLIP2+GPT4: B is sunnier than A. A has trees, in contrast to B. Both A and B feature buildings.

(a)

Image A, 2014-09

Image B, 2018-08

Ground truth: B has more numerous cars than A. There is a red mailbox on the left side of B, but not on A. A person is walking on A, but not on B. A is sunnier than B because there are tree shadows on road A, but not on B. Road A is cleaner than B.

MCCS: B is greener than A. A and B paintings are different for the same building on the left side. Road B is newer than A. A is sunnier than B.

CLIP4IDC: B has more cars than A. B is more bluish than A. A has more leaves than B.

OpenFlamingo: Road A is newer than B. Road B is wider than A. Road A is cleaner than B.

BLIP2+GPT4: B is sunnier than A. B has more people than A. The trees on B are less healthy than on A. The building on B is more modern than A. The road on B is busier than on A, showing more cars than A. The grass on B is lusher than on A.

(b)

Figure 3.12 Experimental results of the existing methods in continual change captioning (image pair). Changes correctly retrieved are highlighted in blue.

image sequences. Figure 3.13 depicts BLEU4 and GPT4 scores for varying sequence lengths. BLEU4 scores drop with the length increase, attributed to lengthier ground truth captions and diminished model efficiency in grasping complex structures. GPT4 scores stabilize, indicating a consistent complexity level in recognizing the change trends across 3 to 6 images. The performance gap compared to human accuracy highlights a deficiency in identifying temporal transitions in sequences, even for the advanced multimodal LLMs.

Table 3.4 Change description evaluation on continual change captioning (image sequence).

Methods	BLEU4↑	CIDEr↑	GPT4↑	Accuracy↑	Coverage↑
MCCFormers-S [60]	19.5	**39.3**	13.7	22.7	0.67
CLIP4IDC [73]	**20.0**	26.0	9.5	13.0	0.48
OpenFlamingo [91]	11.2	23.4	20.9	**34.4**	**1.10**
BLIP2 [92] + GPT4 [93]	4.9	7.5	**30.3**	21.3	1.02
Human	24.3	39.7	40.2	89.8	4.62

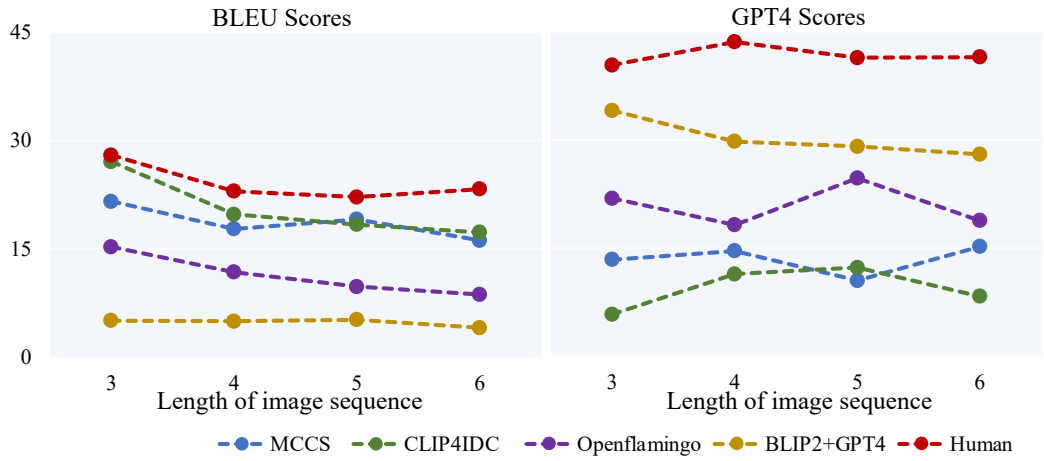


Figure 3.13 Experimental results on dataset examples with different sequence lengths (image numbers).

Prompt Design. Regarding the OpenFlamingo prompts, it was observed that among the two output formats, complete sentences slightly outperformed the itemized output. Therefore, complete sentences were used as the output in the main

Image 1, 2014-08 Image 2, 2018-05 Image 3, 2019-07

Ground truth: The sunniest sky is in image 2, the gloomiest sky is in image 1. Image 3 has the thickest leaves. Image 2 has the most numerous cars. There is a construction site in image 1 but not in other images.

MCCS: There are the most numerous cars in image 1. The trees in image 1 are the thickest.

CLIP4IDC: The road in image 2 is newer than image 1. Image 1 road is newer than road 2.

OpenFlamingo: The sky is the clearest in image 1, which contrasts with the description of it being cloudiest. The road is the newest in image 1. Cars are most numerous in image 2.

BLIP2+GPT4: Image 1 is the cloudiest, image 2 is the sunniest. There is a road in image 3 but not in images 1 and 2. Image 3 has a road fence, and the others do not.

(a)

Image 1, 2016-04 Image 2, 2017-07 Image 3, 2017-10 Image 4, 2018-04

Ground truth: Image 4 has the most bluish sky. Trees in images 2 and 3 have leaves, but trees in images 1 and 4 are withered. Images 1, 3, and 4 have people, but image 2 does not.

MCCS: Image 3 is the sunniest. Image 4 has the most numerous trees.

CLIP4IDC: Road 4 is the newest. Road 2 is newer than road 3.

OpenFlamingo: The lawn grass in image 2 is the brownest. The cars in image 2 are the most numerous. The clearest sky is in image 3.

BLIP2+GPT4: Image 1 has a city street and bike lane. The road in image 1 is better. There are no vehicles in image 1. Image 2 is grassier than image 1. Image 3 has a better road condition than Image 2. Image 4 is brighter than Image 3.

(b)

Image 1, 2015-04 Image 2, 2016-01 Image 3, 2016-10 Image 4, 2019-04 Image 5, 2020-07

Ground truth: Image 5 is the cloudiest. The tree in the middle of image 3 has the most leaves. Image 1 shows an orange traffic sign, but others do not. The lawn on the right side of images 3 and 5 is green, while in images 1, 2, and 4, the lawn is withered. The cars are the least numerous in image 5.

MCCS: Image 3 has the most numerous cars. Image 2 has the sunniest sky.

CLIP4IDC: Road 2 is wider than road 4. Road 3 is the newest.

OpenFlamingo: The sky is the clearest in image 3. The sky is the clearest in image 1. The road is the clearest in image 2.

BLIP2+GPT4: Images 2 and 3 have fewer clouds than Image 1. Trees only appear in images 1, 4, and 5. Images 2 and 3 don't have buildings. All roads are in good condition. Grassland is present in all images. Only images 1 and 5 show a car and a human. The fence is newer in images 3 and 5 than in image 1.

(c)

Image 1, 2015-10 Image 2, 2019-07 Image 3, 2019-07 Image 4, 2019-08 Image 5, 2020-03 Image 6, 2020-05

Ground truth: Image 6 has the most sunniest sky. The trees in image 5 are the most withered. The cars in image 4 are the fewest numerous. There are tree shadows on the road in image 6 but not in other images.

MCCS: Image 4 is the sunniest. The building in image 1 is the newest. Image 5 has the most numerous cars.

CLIP4IDC: Image 4 has the newest road. Image 6 road is the newest.

OpenFlamingo: Image 3's trees are the densest. Image 1 has the newest road. Image 4 is crowded with people, the most of all. Image 3 has more cars than any other.

BLIP2+GPT4: Image 2, unlike image 1, includes both trees and buildings. It's sunnier in image 2 than in image 1. Only image 3 has a road, unlike images 4, 5, and 6. More sunlight is found in image 3 than in image 1. Image 5 shows clearer skies than Image 1. Image 6 is similar to image 5 in weather.

(d)

Figure 3.14 Experimental results of the existing methods in continual change captioning (image sequence).

experiments. Moreover, initially, there was a concern that OpenFlamingo might not handle multiple images effectively. This led to trials, in which images were concatenated horizontally, before being input into OpenFlamingo. However, inputting images separately was found to yield more effective results, and this approach was adopted for all experiments. The results of experiments with varying example numbers are shown in Table 3.5. In shorter sequences (like image pairs or 3/4-image sequences) the model often replicated prompt language, leading to minimal BLEU4 and CIDEr scores. Thus, 3-example experiments for these sequences were excluded. Additionally, experiments with larger example numbers for longer sequences were omitted due to the limitations of the input token length.

Table 3.5 Change description evaluation on continual change captioning tasks using OpenFlamingo.

Number of examples	**image pair** BLEU4/CIDEr	**3-image sequence** BLEU4/CIDEr	**4-image sequence** BLEU4/CIDEr	**5-image sequence** BLEU4/CIDEr	**6-image sequence** BLEU4/CIDEr
3	-	-	-	11.0/6.7	**8.8/8.8**
5	7.7/31.3	11.5/30.6	**11.8/32.4**	**9.8/12.2**	4.8/7.0
10	6.5/19.3	**14.4/40.0**	9.7/17.9	7.5/5.0	-
15	9.4/27.2	12.9/29.4	11.7/24.4	-	-
20	**7.8/37.3**	11.5/30.6	-	-	-

Regarding BLIP2, experiments were conducted with two different BLIP2 base models (BLIP2-opt-2.7b and BLIP2-flan-t5-xl), setting the number of BLIP2 tokens to 10, 15, 20, 25, and 30. Additionally, the GPT4 prompts were adjusted based on the output from BLIP2 to generate grammatically similar sentences to the ground truths. Through experiments with continual change caption tasks using image pairs and 3-image sequence data, the design of the prompts for BLIP2 and GPT4 was finalized. It was determined that a BLIP2 token count of 25 was optimal. A comparative analysis of the opt- and flan-based models, detailed in Table 3.6, revealed that the opt-based model generally outperformed the flan model.

Figure 3.15 Examples of the change-aware sequential instance segmentation results (from top to bottom: input images; ground truth; results from Mask2Former and CTVIS). Objects with the consistent IDs share the same mask colors within each sequence.

Table 3.6 Change description evaluation on continual change captioning tasks using BLIP2+GPT4.

Base model	**image pair** BLEU4/CIDEr	**3-image sequence** BLEU4/CIDEr	**4-image sequence** BLEU4/CIDEr	**5-image sequence** BLEU4/CIDEr	**6-image sequence** BLEU4/CIDEr
Blip2-opt-2.7b	**4.2/16.1**	**5.1/12.4**	5.0/4.3	**5.2/6.3**	**4.1/6.8**
Blip2-flan-t5-xl	3.9/8.6	4.9/7.6	**5.2/4.6**	4.1/3.8	3.1/2.8

Methods	Backbone	AP	AP50	AP75
Mask2Former [85]	ResNet50 [94]	4.60	6.73	4.52
	ResNet101 [94]	4.64	6.29	4.70
	SwinT-S [95]	6.02	8.34	6.47
	SwinT-L [95]	6.46	9.52	6.32
CTVIS [88]	ResNet50	5.86	7.82	6.37
	SwinT-L	**7.08**	**10.42**	**7.00**

Table 3.7 Evaluation on the change-aware sequential instance segmentation task (SwinT-S, -L: swintransformer small, large).

3.5.6 Results of Change-Aware Sequential Instance Segmentation

The comparison of the chosen baseline models for the change-aware sequential instance segmentation task is present in Table 3.7. Among the two baselines, the CTVIS method achieved the highest Average Precision (AP) score across all thresholds (7.08 AP, 10.42 AP50, 7.00 AP75), when used with the SwinT-L backbone. Notably, even with the adoption of more extensive backbones like SwinT-L, the scores were not significantly improved. Examples of the generated instance segmentation masks for the chosen baseline models are present in Figure 3.15. Both Mask2Former and CTVIS exhibited low accuracy in identifying buildings with changing viewpoints, and in segmenting small regions like cars and humans. This is attributed to the unique challenges the STVchrono dataset poses, which include significant appearance changes due to factors like: construction, traffic, weather, seasons, and varying camera angles. These factors distinguish STVchrono from the typical tasks such as video instance segmentation, highlighting its complexity. The results underscore the need for ongoing innovation and the development of new approaches to improve robustness in the change-aware

sequential instance segmentation.

3.6 Conclusion

This chapter addresses the challenge of modeling long-term environmental changes in real-world scenarios by introducing STVchrono, a novel benchmark dataset for continuous change recognition. Continuous, long-term change is a pervasive and essential characteristic of real-world observations, with critical applications in domains such as urban analysis, agriculture, and cultural heritage preservation. However, most existing research in change recognition focuses on short-term, discrete changes and relies heavily on synthetic datasets constrained to two-image observation pairs, limiting their applicability to dynamic and evolving environments.

To bridge this gap, STVchrono provides a comprehensive benchmark designed to support research in long-term continuous change recognition. Comprising street view images from 50 cities worldwide over an 18-year span, the dataset facilitates evaluations across tasks such as paired image change captioning, sequential image change description, and change-aware instance segmentation. By emphasizing gradual and long-term changes, STVchrono challenges models to move beyond static or short-term scenarios and adapt to dynamic temporal transitions observed in real environments.Experiments conducted with STVchrono revealed a substantial performance gap between current state-of-the-art methods, including multimodal Large Language Models (LLMs), and human capabilities. While advanced LLMs showed promise in understanding change trends, their ability to capture nuanced, dynamic changes remains limited, underscoring the need for further methodological innovations.

Limitation. However, STVchrono has its limitations. These include uneven city data distribution and a restricted diversity of changes, particularly those related to weather variations and time-of-day shifts. Addressing these gaps will require expanding the dataset to include a broader range of visual changes and more detailed linguistic descriptions. Furthermore, existing methods for change recognition often treat change description and region detection as separate tasks.

Developing integrated approaches that seamlessly combine change detection with adaptive captioning remains an open challenge and a promising direction for future research.

Future Directions. Future work can address these limitations in several ways. First, expanding the dataset to include a broader range of changes, such as weather variations and temporal lighting shifts, could improve the diversity and robustness of change recognition models. Enhancing linguistic annotations to provide more detailed and context-aware descriptions of changes would also support more comprehensive evaluations. Furthermore, developing integrated approaches that combine change detection with adaptive captioning could enable seamless recognition and description of changes in dynamic environments.

Chapter 4

Dynamic Vision-and-Language Navigation

Real-world navigation is inherently dynamic, with agents needing to respond to changing conditions such as moving vehicles, pedestrian activities, fluctuating traffic signals, and varying weather. Traditional Vision-and-Language Navigation (VLN) tasks, constrained by static environments, fail to capture these complexities, making them inadequate for real-world applications where adaptability is critical.

This chapter addresses the third and final challenge outlined in Chapter 1: enabling agents to navigate dynamically evolving environments. To meet this challenge, the Dynamic Vision-and-Language Navigation (DynamicVLN) task is introduced, incorporating scenarios that require adaptive decision-making in the face of dynamic elements. Additionally, a baseline model, DynaNav, is proposed to establish benchmarks for this task. DynaNav features a Dynamic Detection Module to recognize environmental changes and adapt its navigation strategy, ensuring both instruction-following and situational awareness. This chapter explores the limitations of static VLN, introduces the DynamicVLN task and dataset, and evaluates the performance of the proposed baseline, laying the foundation for robust, adaptable navigation systems capable of real-world deployment.

4.1 Introduction

The Vision-and-Language Navigation(VLN) [7] task requires the agent to be able to navigate in environments based on visual inputs and natural language instructions. This capability is crucial for various real-world applications, including household robots, caregiver assistance, navigation aids for visually impaired individuals, disaster area assessment, and delivery services. There is a strong societal demand for advancements in this technology, as it has the potential to transform everyday life and support critical operations. To support these applications, researchers have proposed a variety of VLN datasets that capture diverse navigation challenges, including the Room-to-Room (R2R) [7] for the indoor setting and Touchdown [2] explores outdoor urban navigation, Specialized datasets such as ALFRED [16] introduce tasks that combine navigation with object interaction, further enriching the scope of VLN research.

However, these datasets were designed based on static environments, where the objects and layout remain unchanged, failing to encapsulate the unpredictable nature of the real world. The absence of dynamic elements such as moving cars, pedestrians, fluctuating traffic lights, and variable weather conditions limits the applicability of these datasets for preparing agents to navigate in environments that closely mimic daily scenarios.

To address this, this chapter introduces the Dynamic Vision-and-Language Navigation (DynamicVLN) task. DynamicVLN expands on traditional VLN by incorporating the unpredictability of real-life scenarios, challenging systems to adapt to changes in their surroundings while following navigation instructions. DynamicVLN is structured around four key scenarios: vehicles, pedestrians, traffic signals, and weather conditions. Each of these elements introduces variability that requires agents to dynamically adjust their behavior, closely mirroring humans' challenges in everyday navigation. In response to these elements, agents may encounter numerous situations where they must decide whether to perform a *temporal stop* or to continue action according to instructions. As shown in Figure 4.1, at timestep T, while the instruction indicates 'turn right,' the agent must instead briefly stop to avoid a collision with an oncoming vehicle, exemplifying the need for adaptive decision-making in dynamic environments. This scenario

Figure 4.1 In traditional VLN tasks, agents predict actions based only on instructions, without accounting for real-time environmental changes. In Dynamic VLN tasks, however, agents must consider both instructions and dynamic elements, such as moving vehicles. For example, although the instruction here directs the agent to "turn right," the agent must temporarily stop to yield to an oncoming car, adapting its actions to avoid a potential accident.

highlights the challenges addressed by DynamicVLN, which requires agents to respond to their surroundings while adhering to navigation instructions dynamically.

Scenarios like this are among the 10 types of dynamic variations incorporated into the DynamicVLN dataset, designed to evaluate agents' adaptability across diverse real-world challenges. For example, in vehicle-related scenarios, an agent must navigate around sudden stops by cars or adjust its path in response to vehicles merging into its lane. Pedestrian scenarios test the agent's ability to safely navigate around individuals crossing the street unexpectedly or moving in unpredictable patterns. In scenarios involving traffic signals, agents are required to

interpret changes in traffic lights, making split-second decisions that ensure compliance with traffic laws while progressing toward their goal. Weather scenarios introduce visual and physical challenges, such as reduced visibility due to fog or altered road conditions caused by rain or snow, requiring agents to modify their navigation strategies to maintain safety and efficiency. This chapter constructed DynamicVLN using the CARLA simulator [98] and automatically generated instructions with GPT-4 [93], resulting in a dataset of 11,261 unique navigation instances. Each of the 10 dynamic scenarios—vehicles, pedestrians, traffic signals, and weather conditions- is divided between cases requiring a 'temporal stop' and those that do not. This balanced design ensures comprehensive coverage of dynamic change detection, providing a robust foundation for training and evaluating navigation systems in real-world-like environments.

Along with the DynamicVLN task, this chapter introduces DynaNav, a baseline model designed to address its distinct challenges effectively. At the core of DynaNav is a Dynamic Detection Module, which recognizes dynamic elements within the environment. This module enables DynaNav to discern when to execute a 'temporal stop' or proceed, ensuring effective and adaptive navigation in dynamic scenarios. In summary, the contribution of our work is four-fold:

- This chapter introduces Dynamic Vision-and-Language Navigation (DynamicVLN), a novel task that incorporates dynamic real-world scenarios such as moving vehicles, pedestrians, fluctuating traffic signals, and varying weather conditions, addressing the limitations of traditional static VLN tasks.

- This chapter constructs the DynamicVLN dataset, comprising 11,261 navigation instances across ten dynamic scenarios. Data collection was automated using the CARLA simulator, and captions were generated automatically by GPT-4, ensuring both realism and diversity.

- This chapter proposes DynaNav, a baseline model equipped with a Dynamic Detection Module, enabling agents to recognize dynamic elements and make context-aware decisions, such as when to execute a 'temporal stop.'

4.2 Related Works

4.2.1 Vision-and-Language Navigation Dataset

Vision-and-Language Navigation (VLN) tasks require agents to navigate environments using natural language instructions. Existing VLN datasets encompass a variety of scenarios. For indoor navigation, the Room-to-Room (R2R) dataset [7], based on Matterport3D [28], serves as a foundational benchmark. It has been extended by Room-Across-Room (RxR) [15] and XL-R2R [29], which include multilingual instructions. These datasets are designed for simple and structured navigation tasks. For outdoor navigation, the Touchdown dataset [2], utilizing Google Street View[1], provides a benchmark for navigating complex urban environments. Similarly, StreetLearn [30], Retouchdown [31], StreetNav [32], Talk2Nav [33], map2seq [27] and VLN-VIDEO [99] focus on urban navigation tasks, with VLN-VIDEO augmenting navigation performance using driving videos. In terms of object interaction, the ALFRED dataset [16], built upon AI2-THOR 2.0 [100], emphasizes complex tasks requiring agents to interact with objects while navigating indoor environments. For aerial navigation, the AerialVLN dataset [101], based on the AirSim [102], introduces challenges that require agents to interpret instructions and navigate elevated perspectives. Some datasets focus on task-specific navigation; for instance, CARLA-NAV [103] explores the grounding of navigable regions corresponding to textual descriptions, while DOROTHIE [104] highlights dialogue-based navigation in dynamic environments, Although these datasets address dynamic elements in autonomous driving settings, they do not encompass sudden situations and lack the dynamic elements found in real-world scenarios. In contrast, our DynamicVLN dataset incorporates realistic, dynamic changes along navigation routes, providing a comprehensive benchmark for evaluating agent adaptability in dynamic and fluctuating settings.

[1] https://developers.google.com/maps/documentation/streetview

Table 4.1 Comparison of various Vision-and-Language Navigation datasets highlighting environment type, data source, presence of dynamic elements, use of automatic annotation, and primary task focus.

Dataset	Environment	Data Source	Dynamic Elements	Automatic Annotation	Emergent Adaptation	Complex Navigation Conditions
Room-to-Room [7]	indoor	Matterport3D	✗	✗	✗	Structured, static
Room-Across-Room [15]	indoor	Matterport3D	✗	✗	✗	Structured, static
VLN-CE [105]	indoor	Matterport3D	✓	✗	✗	Continuous navigation
ALFRED [16]	indoor	AI2-THOR 2.0	✓	✗	✗	Object interactions
Touchdown [2]	outdoor	Google Street View	✗	✗	✗	Urban navigation
map2seq [27]	outdoor	Google Street View	✗	✓	✗	Urban navigation
AerialVLN [101]	outdoor	AirSim	✓	✗	✗	Aerial navigation
CARLA-NAV [103]	outdoor	CARLA	✓	✗	✗	Grounded navigation
DOROTHIE [104]	outdoor	CARLA	✓	✗	✗	Dialogue-based navigation
VLN-VIDEO [99]	outdoor	Google Street View	✗	✓	✗	Urban navigation
DynamicVLN (our)	outdoor	CARLA	✓	✓	✓	emergent adaptation

4.2.2 Approach for Vision-and-Language Navigation

With the introduction of numerous Vision-and-Language Navigation (VLN) benchmarks, various methods exist to enhance navigation performance. Initially, many approaches employed sequence-to-sequence (seq2seq) models, integrating images and instructions into pre-trained models [6, 34, 94], which have great understanding for images and instructions, then utilizing cross-modal attention mechanisms for action prediction [7, 106]. Transformer-based models have further advanced VLN performance. ORIST [36] introduced object-level and word-level inputs to learn fine-grained relationships across textual and visual modalities, enabling more precise decision-making. VLN-BERT [9] extended this approach by developing a recurrent vision-and-language BERT, which incorporates historical states to enhance sequential decision-making in navigation tasks. SOTA [22] proposed a scene- and object-aware transformer, emphasizing context-specific understanding by focusing on relevant objects and environmental details. Given the challenges of environmental understanding in VLN tasks, researchers have also incorporated fine-grained attention mechanisms to improve agents' comprehension of their surroundings. For example, previous work [11, 12] leveraged landmarks along the route to divide the navigation path into smaller segments, using these landmarks as references to enhance navigation performance by providing clear intermediate goals. Recently, LLM-based approaches have emerged as a promising direction

in VLN. For instance, VELMA [107] employs large language models to generate natural language explanations, helping agents reason about their navigation steps. VirtuWander [108] introduced a method for augmenting VLN tasks with detailed descriptions generated by LLMs, improving interpretability and task success. MapGPT [109] integrates GPT-based language generation to facilitate map-based navigation, showcasing the potential of LLMs to handle complex, multi-modal navigation tasks.

This chapter expands upon traditional VLN methods by introducing the Dynamic Vision-and-Language Navigation (DynamicVLN) task, which incorporates dynamic, real-world elements into navigation scenarios. Unlike existing models, DynamicVLN explicitly challenges agents to adapt to dynamic elements, such as vehicles, pedestrians, traffic signals, and weather changes, requiring real-time decision-making and enhanced contextual understanding. The proposed DynaNav model further advances the field by introducing a Dynamic Detection Module, enabling agents to recognize dynamic elements and make context-aware decisions, addressing gaps in current research.

4.2.3 Large Language Model for Dataset Generation.

Recent advancements in Large Language Models (LLMs) have significantly enhanced data annotation processes. Traditional methods often involve manual labeling, which is time-consuming and costly. Application for synthetic data generation has garnered significant attention due to their remarkable capability to understand and generate human-like text based on input prompts and a few examples [110]. Early works utilized LLMs primarily for generating textual data, particularly in the natural language processing (NLP) field, enabling tasks such as text classification [111], summarization [112], and translation with minimal human intervention [113]. Building on this foundation, LLMs have also been applied in multimodal contexts. For instance, LLaVA [114] demonstrated how language-only models like GPT-4 could generate multimodal instruction-following data by combining vision encoders with LLMs to create datasets for visual understanding tasks. Moreover, LLMs have been leveraged to augment existing datasets to boost task performance. AttrPrompt [115] utilized LLMs to enhance attribute-based rea-

soning tasks, while other studies employed LLMs to expand datasets with diverse, high-quality examples [116]. Beyond general-purpose datasets, there has been a growing focus on using LLMs to generate domain-specific datasets [117, 118]. This chapter also leverages GPT-4 [93] to automatically generate diverse and contextually rich instructions for the DynamicVLN dataset. By integrating GPT-4 with the CARLA simulator [98], this study bridges the gap between synthetic data generation and real-world scenarios. This approach not only automates the dataset creation process but also ensures high-quality, diverse, and context-aware navigation instructions, addressing the limitations of traditional manual annotation and static datasets.

4.3 DynamicVLN Dataset

4.3.1 Task Definition

DynamicVLN is a task that challenges an agent to navigate through an environment based on natural language instructions $X = \{x_1, x_2, \ldots, x_l\}$ while dynamically adapting to changes within that environment to reach a specified target location. The instructions consist of a sequence of l word tokens, each represented by x_i. The environment is structured as an undirected graph, with nodes $v \in \mathbb{V}$ representing specific locations connected by labeled edges $(v, u) \in \mathbb{E}$, where u denotes an adjacent location to v. Each node is linked to an RGB image, providing visual context, and edges denote possible navigation paths with heading angles $\alpha_{(v,u)}$ between images.

At any given time t, the agent's state is $s_t \in \mathcal{S}, s_t = (v_t, \alpha_{(v_{t-1},v_t)})$, incorporating the current panoramic view v_t and the heading angle $\alpha_{(v_{t-1},v_t)}$ from the previous state to the current one. To navigate, the agent performs actions from the set $a_t \in \{\text{FORWARD, LEFT, RIGHT, TEMPORAL STOP, STOP}\}$, where the temporal stop action is introduced to allow brief pauses in response to dynamic events such as moving pedestrians or vehicles.

The objective in DynamicVLN is for the agent to generate a sequence of state-action pairs $\langle (s_1, a_1), (s_2, a_2), \ldots, (s_n, a_n) \rangle$, culminating in a $a_n = \text{STOP}$ action that indicates the goal location has been reached, as defined by the instructions.

Including the *temporal stop* action enhances the agent's ability to navigate more effectively by adapting to real-time changes in the environment, thus making DynamicVLN a more realistic and challenging task that mirrors the complexities of navigating dynamic, real-world scenarios.

4.3.2 Scenario Design

This section designes four dynamic element types to realistically simulate complex navigation environments: vehicles, pedestrians, traffic signals, and weather. The specific scenarios were introduced for each type that require the agent to make adaptive decisions. For example, these scenarios often necessitate a temporal stop to avoid potential accidents, such as yielding to cross-traffic or halting for a pedestrian crossing. Including these dynamic elements adds complexity to the navigation task, challenging the agent ' s ability to adapt and make real-time decisions in unpredictable environments. Table 4.2 summarizes the scenarios in DynamicVLN, including whether a temporal stop is required based on the presence of dynamic elements. Figure 4.2 shows examples in DynamicVLN when the vehicle needs to temporarily stop at timestep T during driving for each type of dynamic element.

4.3.3 Dataset Collection

Driving scenarios were designed using the open-source CARLA simulator [98] to simulate realistic driving conditions and collect images and driving data of vehicle navigation. Specifically, CARLA ' s pre-defined town maps (Town 1, 2, 3, 4, 5, 6, and 10HD) were utilized as the environment. To construct navigation routes, waypoints were first sampled at 5-meter intervals across the map, defining a set of potential starting locations. Then, for each waypoint, a route of 25-50 waypoints was generated using CARLA ' s GlobalRoutePlanner, which allows for automatic pathfinding across the road network. This ensures the generated paths follow realistic road structures. Additionally, to introduce complexity, each navigation route contained at least one turning intersection, mimicking the design of traditional VLN datasets. A vehicle equipped with a front-facing cam-

Table 4.2 In DynamicVLN, each dynamic element corresponds to specific scenarios. The preferred action often involves a 'temporal stop' or adjusting the original navigation action to ensure safety and optimal performance.

Dynamic Element	Scenario	Required Action
Vehicle	Approaching ambulances or fire engines	Temporal stop and yield
	A car comes alongside	Maintain distance or yield
	Vehicles from different laneways	Yield at intersections
	The vehicle ahead stops suddenly	Temporal stop
	A vehicle changes lanes abruptly	Adjust trajectory
Pedestrian	Pedestrians appear alongside	Maintain caution or stop
	Pedestrians appear in the crosswalk	Temporal stop
	Children running into the road	Emergency stop
Traffic Condition	Changes in traffic signals	Stop for red or proceed on green
	Road signs (e.g., stop signs, yield)	Follow traffic rules
	Congested traffic ahead	Adjust speed or stop
Weather	Poor visibility due to rain, snow, or fog	Reduce speed and proceed cautiously
	Strong winds affecting vehicle stability	Adjust speed and maintain control
	Slippery roads due to ice or rain	Temporal stop or proceed with caution

era sensor was deployed to drive along each static route. The vehicle followed the predefined paths, and the camera captured an image at every waypoint. At each waypoint, navigation actions, including forward, right, left, temporal stop, and stop, were determined based on changes in yaw angle and vehicle speed to reflect real navigation decisions. Landmark information was extracted from the CARLA map within a 3-meter radius of each waypoint, capturing relevant traffic signs, signals, and road markers to provide additional scene context. While static routes provide a structured baseline, real-world navigation requires adaptability to unpredictable dynamic elements. To simulate this, dynamic routes were

(a) Temporal stop due to oncoming car

(b) Temporal stop due to pedestrian crossing

(c) Temporal stop due to stop traffic sign

(d) Temporal stop due to low visibility

Figure 4.2 Example of a temporal stop under each dynamic element types setting.

created by introducing moving vehicles, pedestrians, and environmental changes along the predefined static routes. Using CARLA ' s Traffic Manager, dynamic entities were added with randomized behaviors. Moving vehicles were generated to simulate real-world road conditions, pedestrians were introduced near crosswalks and sidewalks, requiring agents to detect and adapt to human movement, and traffic signals were programmed to change dynamically, forcing agents to adjust to signal variations. Additionally, obstacle vehicles were strategically placed to block the agent ' s path at certain waypoints, requiring adaptive responses such as temporal stops before proceeding.

The final dataset consists of images captured at every waypoint along both static and dynamic routes, recorded navigation actions based on route yaw changes and vehicle speed, landmark annotations extracted within 3 meters of each waypoint, and dynamic event labels indicating the presence of moving vehicles, pedestrians, or obstacles. By incorporating both static and dynamic navigation scenarios, this dataset serves as a comprehensive benchmark for evaluating agents' real-world adaptability in VLN.

4.3.4 Instruction Generation

Based on the data collected from CARLA, a pipeline was proposed to automatically generate high-quality navigation instructions using GPT-4. In this work, the aim was to train the agent to adapt to sudden events; therefore, the temporal stop action was not included in the instructions. This pipeline employs two LLM agents: an **Instruction Generator** and an **Instruction Supervisor**. As shown in Figure 4.3, the pipeline begins by feeding the route overview, the sequence of actions at each waypoint, and the landmarks encountered along the route into the **Instruction Generator**. The generator produces an initial navigation instruction formatted similarly to traditional VLN instructions. Next, the generated instruction and a simplified action list are passed to the **Instruction Supervisor**. The supervisor evaluates the alignment between the simplified action list and the generated instruction, checking for missing actions, incorrect sequencing, or unnecessary additions. Based on its analysis, the supervisor refines the instruction to ensure accuracy and consistency with the action list. This iterative process ensures that the final instruction is coherent, aligned with the route's actions, and accurately reflects the dynamic elements of the environment.

Specifically, Figure 4.4 and Figure 4.5 present the prompts used by the Instruction Generator and the Instruction Supervisor during the instruction creation process. The Instruction Generator receives structured input detailing the route overview, actions, and landmarks, generating an initial instruction. The Instruction Supervisor takes the simplified action list and the initial instruction as input and performs a consistency check, identifying necessary corrections before refining the final instruction.

4.3.5 Data Statistics

Following the pipeline introduced above, a total of 11,261 routes were collected. Among these, 2,786 routes are associated with dynamic elements of vehicles, 1,680 with pedestrian activities, 3,370 with traffic conditions, and 601 with weather-related scenarios. These represent various dynamic factors encountered during navigation. However, the occurrence of a dynamic factor does not necessarily re-

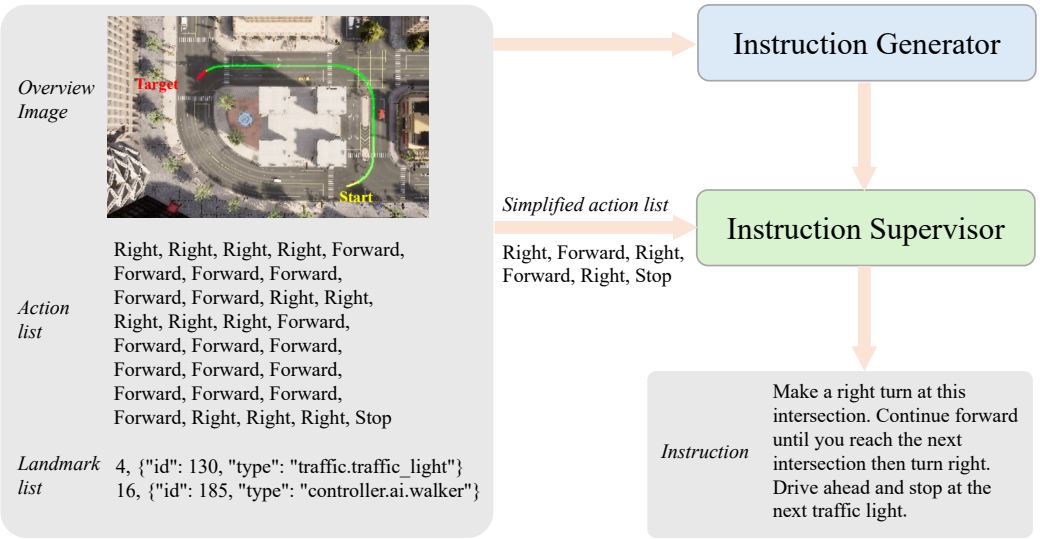


Figure 4.3 Pipeline of instruction generation for DynamicVLN.

Prompt: You are an annotator helping generate instructions for a vision-and-language navigation task. Below is the action list that describes the vehicle's actions at specific points along the route: *<action list>* Along this route, the vehicle may encounter the following landmarks : *<landmark list>* .
Given the action list, landmark list, and *overview image* of the route (with the green line as the route, the red mark as the target, and the yellow mark as the start)Key points for generating instructions:

1. The overview image contains route colors, start, and stop markers, which are not visible while driving. Do not include these markers in the instructions. Use simple phrases like "stop at the place near the intersection or landmark.
2. Accurate landmark references: Only refer to landmarks mentioned in the list. Use their respective indexes accurately.
3. Treat consecutive 'left' or 'right' actions as a single turn at the same location. Use phrases like 'make a left turn' or 'make a right turn' without adding phrases like 'make a series of turns.'
4. Do not add extra actions: Instructions should only include actions that are explicitly mentioned in the action list. Do not introduce new turns or stops.

Please provide a global summaries navigation instruction similar to these instruction examples.:

1. Go straight then turn right at the first intersection, you will see a stop sign. Stop at the place near to the intersection.
2. Drive straight, and at the third intersection, make a left turn. Continue forward until you reach the bus stop on the right. Then stop at the next crosswalk.
3. Head straight along the route until you see the tall building with a red sign. Turn right at that point, proceed forward, and make another right turn at the next traffic light. Stop when you reach the intersection.

You don't need to generate too detailed instruction and list them, just use one or two or three simple sentence to describe the whole route. Please ensure that the generated instruction: Does not include 'make a series of turns.' Accurately reflects the given actions and landmarks without introducing additional details.

Figure 4.4 The Instruction Generator processes the route overview, action list, and landmarks to generate an initial navigation instruction.

Prompt: You are a supervisor tasked with verifying the accuracy of navigation instructions generated for vision-and-language navigation tasks. And If you verified the instruction with some mistakes, please modify it. You will be given:
1. A simplified action list representing the core sequence of actions: *<simplified_action_list>*.
2. A generated instruction: *<generated instruction>*, that describes the route based on overview of route.

Here are key points for verification:
1. Please check the generated instruction follows the exact order of the simplified action list. The instruction should describe the actions in sequence.
2. Avoid Series Turns: The Instruction should not include phrases like 'make a series of right turns. or 'make a series of left turns.'. If such phrases are found, replace them with 'make a left turn' or 'make a series of right turns.'
3. No Extra Actions: The instruction should not add extra turns, stops, or actions that are not present in the simplified action list.
4. No Extra Landmarks: The instruction should not add extra landmarks (except intersection, traffic sign, traffic light, etc.).

Finally, Output only the modified instruction, not the details of verification.

Figure 4.5 The Instruction Supervisor refines the initial instruction by ensuring alignment with the simplified action list and correcting any discrepancies.

quire the vehicle to stop temporally. In many cases, vehicles may need to stop at multiple points along the route before they can proceed, rather than at a single temporal stop. To better understand these dynamics, the distribution of temporal stops within these routes was analyzed. Each dynamic route involves between 2 and 7 temporal stop actions, reflecting the complexity of real-world scenarios where vehicles must repeatedly adjust to changing conditions. Furthermore, each dynamic instance has 2-7 temporal stops. Figure 4.6 illustrates the distribution of routes based on the number of temporal stops they contain, providing insight into the frequency and variation of such actions across all collected routes.

4.4 Proposed Method: DynaVLN

This section introduces the DynaVLN model, designed to detect dynamic events and improve navigation performance in dynamic environments, building upon traditional VLN models. As illustrated in Figure 4.7, DynaVLN consists of four key components: the *Image Encoder*, *Instruction Encoder*, *Dynamic Event Detector*, and *Action Predictor*. At each decoding timestep, DynaVLN computes a visual representation of the agent's current and previous states in the environment, integrating previously predicted actions, instruction features, and dynamic event detection results to predict the next action.

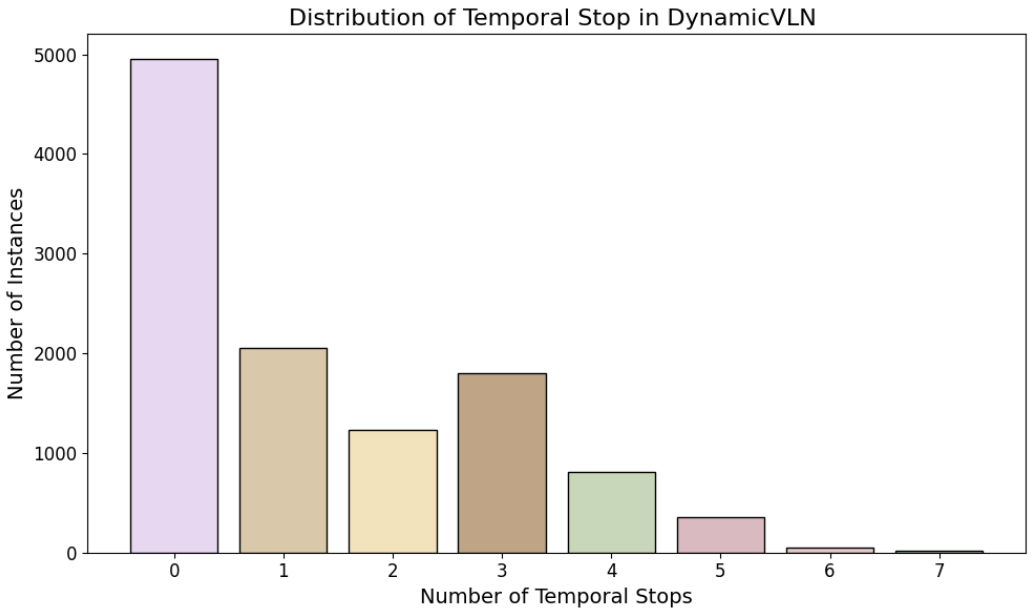


Figure 4.6 Distribution of Routes Based on the Number of Temporal Stops. This figure shows the frequency of routes containing different numbers of temporal stops, reflecting the complexity and variability of dynamic scenarios in the collected dataset.

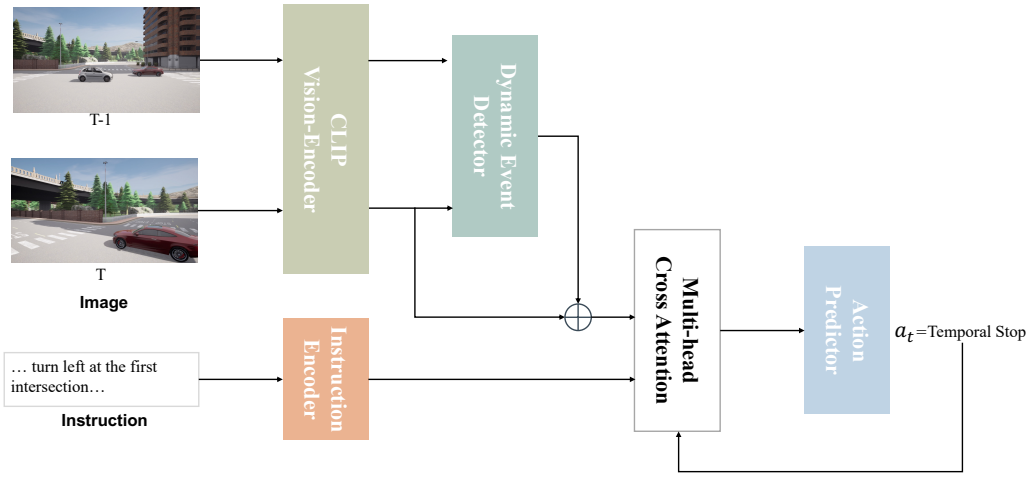


Figure 4.7 Overview of the proposed DynaVLN model. At each decoding timestep, the CLIP Vision Encoder processes the current and previous images (T and $T-1$) to extract visual representations of the environment. The Instruction Encoder encodes the navigation instructions to provide linguistic context. The Dynamic Event Detector identifies dynamic elements, such as moving vehicles or pedestrians, in the visual scene. These outputs, combined through a Multi-Head Cross Attention mechanism, are used by the Action Predictor to generate the next action (a_t), including temporal stops, ensuring safe and effective navigation in dynamic environments.

4.4.1 Model Details

Image Encoder. At each timestep t, the agent captures an image view of its surroundings. The visual representation of the current agent position is computed by extracting features from the panorama using a pre-trained CLIP Vision Encoder [119]. This step provides a robust and compact visual embedding $\mathbf{v_t}$ for navigating and detecting changes during the driving process.

Instruction Encoder. The instruction encoder processes the natural language navigation instructions $\mathbf{x} = \{x_1, x_2, ..., x_L\}$, where L denotes the number of tokens in the instruction sequence. Following the approach introduced in Section 2.5.1, each token x_i is embedded and encoded using a bidirectional LSTM [37]:

$$\hat{\mathbf{x}}_\mathbf{i} = \text{embedding}(x_i) \tag{4.1}$$

$$(\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_L}), \mathbf{z_L^w} = \text{Bi-LSTM}(\hat{\mathbf{x}}_\mathbf{1}, \hat{\mathbf{x}}_\mathbf{2}, \ldots, \hat{\mathbf{x}}_\mathbf{L}), \tag{4.2}$$

where w_i represents the hidden representation of token x_i, and z_L^w is the final cell state of the LSTM. These outputs capture both local (token-level) and global (sequence-level) contextual information from the navigation instructions.

As in Section 2.5.1, the Instruction Encoder provides a robust representation of linguistic input, ensuring consistency and adaptability across various VLN tasks. The use of a bidirectional LSTM enables the model to encode contextual dependencies in both forward and backward directions, which is essential for understanding complex navigation instructions.

Dynamic Event Detector To enable the agent to intelligently handle unforeseen dynamic events without explicit instructions, a *Dynamic Event Detector* (DED) was incorporated, and an *Attention Modulation Mechanism* was proposed that dynamically adjusts the model's focus between visual and language features based on the detected event. This mechanism allows the model to autonomously decide actions, including *temporal stop*, to address sudden changes in the environment.

At each timestep t, the DED processes the current visual feature $\mathbf{v_t}$ and the previous visual feature $\mathbf{v_{t-1}}$ to compute the event signal e_t, representing the likelihood of a dynamic event. Formally, the event signal is defined as follows: Formally, the event signal is defined as follows:

$$e_t = \sigma(\text{MLP}([\mathbf{v_t} - \mathbf{v_{t-1}}])) \tag{4.3}$$

where $\sigma(\cdot)$ is the sigmoid activation, and MLP denotes a lightweight multi-layer perceptron. $e_t \in [0, 1]$ indicates the intensity of the detected dynamic event.

The event signal e_t is then used to modulate the attention mechanism, dynamically altering the interaction between the visual feature $\mathbf{v_t}$ and the instruction feature sequence $\{\mathbf{w_1}, \ldots, \mathbf{w_L}\}$. The modulation affects both the Query ($\mathbf{q_t}$) and the Key/Value representations ($\mathbf{k_t}, \mathbf{v_k}$) as follows:

$$\mathbf{q_t} = \text{Linear}_q([\mathbf{v_t}; e_t]) \tag{4.4}$$

$$\mathbf{k_t}, \mathbf{v_k} = \text{Linear}_k([\mathbf{w_1}, \ldots, \mathbf{w_L}]) \cdot (1 - e_t), \quad \text{Linear}_v([\mathbf{w_1}, \ldots, \mathbf{w_L}]) \cdot (1 - e_t) \tag{4.5}$$

where $[\mathbf{v_t}; e_t]$ denotes the concatenation of the visual feature and the event signal. The Key and Value representations are scaled by $(1 - e_t)$, reducing the influence of language instructions when a significant dynamic event is detected ($e_t \to 1$).

The attention [42] output $\mathbf{f_t}$ is computed via the scaled dot-product attention mechanism:

$$\text{Attention}(\mathbf{q_t}, \mathbf{k_t}, \mathbf{v_k}) = \text{softmax}\left(\frac{\mathbf{q_t}\mathbf{k_t}^\top}{\sqrt{d_k}}\right)\mathbf{v_k} \tag{4.6}$$

where d_k is the dimensionality of the Key vectors. The resulting feature $\mathbf{f_t}$ incorporates the adjusted contribution of visual and language features, dynamically weighted by the presence or absence of a detected event.

Action Predictor. Finally, the attention output $\mathbf{f_t}$ is concatenated with the event signal e_t and passed to the Action Predictor to generate the next action:

$$a_t = \text{ActionPredictor}([\mathbf{f_t}; e_t], a_{t-1}) \tag{4.7}$$

where a_t is the predicted action (*e.g.*, *move forward*, *turn left*, or *stop*) and a_{t-1} is the previous action.

4.4.2 Loss Function

To train the proposed model effectively, a loss function was designed that jointly optimizes the performance of the *Dynamic Event Detector* (DED) and the *Action Predictor*.

Dynamic Event Detector Loss. The DED outputs an event signal $e_t \in [0, 1]$, representing the probability of a dynamic event occurring at timestep t. The ground truth event label $y_t \in \{0, 1\}$ indicates whether a dynamic event is present. The detection loss is formulated as a binary cross-entropy loss:

$$\mathcal{L}_{\text{DED}} = -\frac{1}{N}\sum_{t=1}^{N}\Big[y_t \log(e_t) + (1 - y_t)\log(1 - e_t)\Big], \tag{4.8}$$

where N is the total number of samples. This loss ensures that the DED module learns to predict event probabilities that align with the ground truth labels.

Action Prediction Loss. The *Action Predictor* generates the probability distribution over possible actions $\{a_1, a_2, ..., a_C\}$, where C is the number of action classes (*e.g.*, *move forward*, *turn left*, *stop*). The ground truth action label is denoted as $a_t^{\text{true}} \in \{1, ..., C\}$. A categorical cross-entropy loss was used to optimize

the predicted action distribution:

$$\mathcal{L}_{\text{Action}} = -\frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{C} 1\left[a_t^{\text{true}} = i\right] \log P\left(a_t = i \mid f_t, e_t, a_{t-1}\right), \tag{4.9}$$

where $P(a_t = i | f_t, e_t, a_{t-1})$ is the predicted probability of action i, and $\mathbb{1}[\cdot]$ is the indicator function.

Joint Loss. To jointly optimize both components of the model, the above losses were combined into a single objective:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{DED}} + \lambda_2 \mathcal{L}_{\text{Action}}, \tag{4.10}$$

where λ_1 and λ_2 are balancing weights that control the relative importance of the two loss terms.

4.5 Experiments

This section presents extensive experiments on our DynamicVLN dataset to evaluate our DynaVLN models' performance on the outdoor VLN task and emergency accident adaptation.

4.5.1 Implementation Details

Data Processing. A clustering-based approach was adopted to group spatially proximate waypoints into unified nodes, constructing a node-based navigation graph for each town map for Vision-and-Language Navigation (VLN) tasks. Specifically, all waypoints extracted from the routes, including those used during navigation, were collected and clustered based on their spatial proximity. The DBSCAN algorithm [120], which is effective for identifying arbitrarily shaped clusters without requiring a predefined number of clusters, was utilized for clustering. Each waypoint was treated as a three-dimensional point (x, y, z) in Cartesian coordinates, and waypoints within a distance threshold ($\epsilon = 0.5$ meters) were grouped into the same cluster. After clustering, each cluster was assigned a unique node ID. The navigation graph was then constructed by treating each cluster as a node and connecting nodes adjacent to the same route.

Model Details. The proposed framework and baseline models were implemented using PyTorch [2]. A pretrained CLIP vision encoder with a ViT backbone was employed to extract visual features of size 512 from images at each waypoint. Navigation instructions were tokenized into byte pair encodings (BPE) using a vocabulary size of 2,000 tokens. The instruction tokens were lower-cased and embedded into vectors of size 256. For the loss function, the objectives of navigation performance and dynamic event detection were balanced using weights $\lambda_1 = 0.5$ and $\lambda_2 = 1.0$. This prioritization ensures the agent achieves reliable navigation while accurately detecting dynamic events. In scenarios where ground truth event labels y_t are unavailable, pseudo-labels were generated heuristically. These heuristics included abrupt changes in the visual scene (e.g., the appearance of obstacles) or significant deviations in the agent's planned actions (e.g., an unexpected stop or turn). This allows the model to handle dynamic events in complex environments adaptively. Since each town map is unique, it was necessary to construct a separate navigation graph for each town and conduct training individually for each map. However, this section only presents experimental results using the Town05 map. Town05 was selected because it contains the largest number of routes, with a total of 2,611 routes. These routes were randomly split into training, development, and test sets in a 7:2:1 ratio, resulting in 1,828 routes for training, 522 for development, and 261 for testing.

4.5.2 Baseline Models

ORAR [4] is a VLN model proposed for outdoor VLN task. This model uses an LSTM to encode the instruction text and an LSTM to decode the multimodal features and predict the following action. ORAR was selected because it is a model without collision detection and only uses whole images and instruction features for decision-making. By comparing the results with ORAR, it was demonstrated that collision detection enhances adaptability to unforeseen events.

[2] https://pytorch.org/

4.5.3 Metrics

The following metrics are used to evaluate VLN performance and are introduced in Chapter 2.6 in detail: (1) Task Completion (TC),(2) Shortest-Path Distance (SPD), (3) Success weighted by Edit Distance (SED), (4) Coverage weighted by Length Score (CLS), (5) Normalized Dynamic Time Warping, (6) Success weighted Dynamic Time Warping (SDTW).

4.5.4 Results

A comprehensive evaluation of the experimental results was provided to assess DynaVLN's performance compared to the baseline ORAR across various metrics. These results highlight the proposed model's strengths and limitations, particularly in handling dynamic navigation environments. The experimental results are summarized in Table 4.3.

DynaVLN demonstrates higher TC, indicating its ability to navigate more effectively and reach target destinations more reliably. In terms of SPD, DynaVLN maintains a closer adherence to the target trajectory, reflecting its capability to navigate with greater precision. For SED, DynaVLN achieves better alignment with the ground-truth action sequences, highlighting its accuracy in predicting correct actions during navigation. Conversely, ORAR achieves a higher CLS, suggesting stronger coverage of the navigation path but potentially less responsiveness to dynamic changes. Regarding temporal alignment, ORAR performs better in static environments, as indicated by its nDTW score. However, DynaVLN excels in dynamic scenarios, with a significantly higher sDTW score, showcasing its ability to adapt and succeed in environments with dynamic changes. Overall, these results suggest that while ORAR is well-suited for structured and static environments, DynaVLN offers superior adaptability and robustness in dynamic and complex real-world navigation tasks.

Table 4.3 Quantitative results comparing **ORAR** and **DynaVLN** on navigation performance metrics. Higher TC and CLS scores indicate better trajectory completion and coverage length, respectively. Lower SPD, SED, nDTW, and sDTW scores indicate better alignment with ground truth trajectories.

Method	TC ↑	SPD ↓	SED ↑	CLS ↑	nDTW ↑	sDTW ↑
ORAR	1.65	24.02	1.08	16.35	4.15	1.05
DynaVLN	2.74	23.65	1.41	15.07	3.26	2.10

4.6 Conclusion

This chapter addresses the challenge of navigating dynamic environments by introducing the Dynamic Vision-and-Language Navigation (DynamicVLN) task. The proposed task and dataset aim to bridge the gap between traditional VLN tasks, which predominantly focus on static environments, and the complexities of real-world scenarios. By incorporating dynamic elements such as moving vehicles, pedestrians, fluctuating traffic signals, and varying weather conditions, DynamicVLN introduces a more realistic framework for training and evaluating navigation agents. DynamicVLN expands the scope of VLN research by requiring agents to dynamically adapt to changes in their surroundings while following natural language instructions. For instance, agents must decide whether to perform a *temporal stop* or proceed, depending on the presence of obstacles or changes in the environment. The proposed DynaNav model integrates visual and linguistic inputs with a Dynamic Detection Module to enhance real-time adaptability, providing a robust baseline for future advancements in this area.

Limitation. Despite these contributions, there are several limitations in the current work. The DynamicVLN dataset, while comprehensive in its incorporation of dynamic scenarios, is inherently limited by the simulated environment provided by the CARLA simulator. Real-world datasets may introduce additional challenges, such as irregular pedestrian behavior, diverse weather patterns, and unseen environmental complexities, which are not fully captured in the current dataset. A critical limitation lies in the discrete nature of the DynamicVLN setting. While the dataset includes dynamic elements, the navigation framework

relies on waypoint-based navigation, where actions are determined at predefined points along the route. This approach, however, cannot fully represent the necessity for a *temporal stop* between two waypoints or highlight the agent ' s ability to make real-time decisions based on continuously changing conditions. To better mimic real-world navigation, a continuous spatial setting is essential, where the agent ' s decisions are influenced by seamless transitions in the environment rather than discrete, waypoint-to-waypoint actions. Furthermore, the DynaNav model has only been evaluated within the constraints of the DynamicVLN dataset. This raises questions about its generalizability to other dynamic datasets or real-world environments with more complex scenarios. Some dynamic elements in the dataset, such as traffic signals and vehicle movements, are simulated with predefined patterns, potentially oversimplifying the unpredictability of real-world conditions and leading to overfitting to dataset-specific dynamics.

Future Directions. Looking ahead, future work should focus on addressing the discrete nature of the DynamicVLN task by transitioning from waypoint-based navigation to a continuous spatial framework. This would enable agents to make real-time decisions, such as executing a *temporal stop*, in response to seamlessly changing environmental conditions. Additionally, expanding the dataset to include more diverse and complex dynamic scenarios, such as irregular pedestrian behaviors or erratic vehicle movements, would better reflect the unpredictability of real-world environments. Improving the adaptability of the DynaNav model is another key direction. Advanced reinforcement learning techniques or continual learning approaches could enhance the model ' s ability to generalize to unseen environments. Real-world validation of navigation models, particularly in urban settings, is essential to ensure robustness under realistic conditions. Finally, integrating more nuanced dynamic interactions, such as multi-agent coordination or simultaneous dynamic elements, could further advance the study of dynamic navigation systems.

Chapter 5

Conclusion

This study has made significant advancements in Vision-and-Language Navigation (VLN) by addressing critical gaps in adaptability and performance under real-world conditions. Through the development of novel methods and datasets, this work contributes to enabling agents capable of navigating dynamic, unpredictable environments and understanding long-term environmental changes. In alignment with the initial objectives, the contributions span three key areas.

Chapter 2 focused on the development of the OAVLN model to address the overlooked importance of object tokens in outdoor VLN tasks. By incorporating object information from on-route landmarks, OAVLN demonstrated superior navigation performance in both seen and unseen environments, as validated through extensive experiments on two large-scale datasets. The model's ability to prioritize relevant objects during navigation illustrates its effectiveness in improving scene understanding and action decisions. However, limitations remain regarding data biases and computational demands, paving the way for future efforts to enhance generalization and efficiency.

Chapter 3 introduced the STVchrono dataset to bridge the gap in real-world change recognition, emphasizing long-term continuous variations in urban and natural environments. This aspect is crucial for VLN, where agents must recognize familiar locations despite infrastructure changes, seasonal shifts, or urban development. Without modeling long-term changes, VLN systems risk failing in real-world applications. STVchrono enables sequential change description and

instance segmentation, helping agents align current observations with past experiences for robust localization. While it sets a new benchmark, challenges like uneven data distribution and limited change diversity remain. Future research should integrate change modeling with VLN to enhance adaptability in dynamic, evolving environments.

Chapter 4 addressed the challenge of navigating unpredictable environments through the introduction of the DynamicVLN task. By incorporating dynamic elements such as traffic and weather variations, this work established a realistic dataset for studying navigation under real-world conditions. The proposed DynaNav model represents a promising step toward enabling agents to make real-time decisions using multimodal inputs. Future developments will enhance DynaNav's adaptability and safety in complex, dynamic scenarios.

This study underscores the importance of bridging the gap between static assumptions in traditional VLN tasks and the dynamic, evolving nature of real-world environments. Moving forward, future work will focus on reducing the computational overhead of VLN models to enable practical deployment in resource-constrained environments. Datasets like STVchrono and DynamicVLN will be expanded to cover broader geographic, temporal, and contextual variations, ensuring that models trained on these datasets can generalize to unseen conditions. Additionally, efforts will be directed toward developing integrated frameworks that combine navigation and change recognition tasks, enabling seamless adaptation to dynamic environments.

In summary, this study lays a strong foundation for advancing VLN and its applications in real-world scenarios. By addressing key challenges in navigation, scene understanding, and dynamic adaptability, the contributions of this research pave the way for the development of robust, adaptable systems capable of navigating and reasoning in complex, real-world environments. Future advancements in VLN, including continuous navigation frameworks and real-world validations, will further enhance the ability of intelligent agents to operate safely and efficiently in diverse conditions.

Acknowledgement

本研究は，著者が慶應義塾大学大学院理工学研究科後期博士課程に在学中、産業技術総合研究所人工知能研究センターにリサーチアシスタントとして在職中に，青木義満教授と Qiu Yue 研究員のご指導のもとで行われました．この場を借りて，本研究に関わったすべての皆様に心より感謝申し上げます。

まず，本論文の主査であり，指導教員である慶應義塾大学理工学部青木義満教授に深く感謝いたします．青木義満教授には本研究の立ち上げから本論文の執筆に至るまで，多くの指導をしていただきました．また，研究室の同期，後輩，先輩，秘書さんの皆様にも深く感謝いたします．さらに，産業技術総合研究所で共に研究活動を行った同期や同僚，秘書の方々にも多くの助力をいただきましたことを心より御礼申し上げます．博士課程において得られた経験と知識は，筆者の人生において計り知れない価値を持つものと確信しております．

本論文の審査にあたり，副査を快くお引き受けくださった慶應義塾大学理工学部の田中敏幸教授，村田真悟准教授，吉岡健太郎講師にも深く感謝申し上げます．

最後に，常に支えてくれた家族に対し，心より感謝申し上げます．特に，両親のおかげで日本への留学という機会をいただき，この 7 年間において研究に打ち込むとともに，人生の視野を広げる多くの貴重な経験を得ることができました．また，パートナーである Yang Shaoyu 君，家族として共に時間を過ごしている大福君およびスイカ君にも，常に癒されて，心から感謝いたします．

そして，何よりも，自分自身に感謝したいと思います．21 年間の学生生活を通じて，多くの課題に向き合い，学び，少しずつ成長することができました．この経験をこれからの人生でも大切にしていきたいと思います．皆様の支えがあったからこそ，この論文を完成させることができました．

References

[1] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: connecting language, knowledge, and action in route instructions. In *Proceedings of the 21st National Conference on Artificial Intelligence*, p. 1475–1482. AAAI Press, 2006.

[2] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12530–12539, 2019.

[3] Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. Gated-attention architectures for task-oriented language grounding. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[4] Raphael Schumann and Stefan Riezler. Analyzing generalization of vision and language navigation to unseen outdoor areas. In *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 7519–7532, 2022.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, Vol. 521, No. 7553, p. 436, 2015.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias

Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[7] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3674–3683, 2018.

[8] Wanrong Zhu, Xin Wang, Tsu-Jui Fu, An Yan, Pradyumna Narayana, Kazoo Sone, Sugato Basu, and William Yang Wang. Multimodal text style transfer for outdoor vision-and-language navigation. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, p. 1207–1221, 2021.

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

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

[11] Yanjun Sun, Yue Qiu, Yoshimitsu Aoki, and Hirokatsu Kataoka. Outdoor vision-and-language navigation needs object-level alignment. *Sensors*, Vol. 23, No. 13, 2023.

[12] Yanjun Sun, Yue Qiu, Yoshimitsu Aoki, and Hirokatsu Kataoka. Guided by the way: The role of on-the-route objects and scene text in enhancing outdoor navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5198–5204, 2024.

[13] Yanjun Sun, Yue Qiu, Mariia Khan, Fumiya Matsuzawa, and Kenji Iwata. The stvchrono dataset: Towards continuous change recognition in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14111–14120, June 2024.

[14] Yanjun Sun, Yue Qiu, and Yoshimitsu Aoki. Dynamicvln: Incorporating dynamics into vision-and-language navigation scenarios. *Sensors*, Vol. 25, No. 2, 2025.

[15] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 4392–4412, 2020.

[16] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10740–10749, 2020.

[17] Jiannan Xiang, Xin Wang, and William Yang Wang. Learning to stop: A simple yet effective approach to urban vision-language navigation. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[18] Wanrong Zhu, Yuankai Qi, Pradyumna Narayana, Kazoo Sone, Sugato Basu, Xin Wang, Qi Wu, Miguel Eckstein, and William Yang Wang. Diagnosing vision-and-language navigation: What really matters. In *Proceedings of Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, p. 5981–5993, 2022.

[19] Edgar Chan, Oliver Baumann, Mark Bellgrove, and Jason Mattingley. From objects to landmarks: The function of visual location information in spatial navigation. *Frontiers in Psychology*, Vol. 3, p. 304, 2012.

[20] Yuankai Qi, Zizheng Pan, Shengping Zhang, Anton van den Hengel, and Qi Wu. Object-and-action aware model for visual language navigation. In *Proceedings of The European Conference on Computer Vision (ECCV)*, 2020.

[21] Chen Gao, Jinyu Chen, Si Liu, Luting Wang, Qiong Zhang, and Qi Wu. Room-and-object aware knowledge reasoning for remote embodied referring expression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3064–3073, 2021.

[22] Abhinav Moudgil, Arjun Majumdar, Harsh Agrawal, Stefan Lee, and Dhruv Batra. Soat: A scene- and object-aware transformer for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, pp. 7357 – 7367, 2021.

[23] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12684–12694, 2021.

[24] Ronghang Hu, Daniel Fried, Anna Rohrbach, Dan Klein, Trevor Darrell, and Kate Saenko. Are you looking? grounding to multiple modalities in vision-and-language navigation. In *Proceedings of the Association for Computational Linguistics (ACL)*, p. 6551–6557, 2019.

[25] Yubo Zhang, Hao Tan, and Mohit Bansal. Diagnosing the environment bias in vision-and-language navigation. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 890–897, 2020.

[26] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *Proceedings of The European Conference on Computer Vision (ECCV)*, pp. 259 – 274, 2020.

[27] Raphael Schumann and Stefan Riezler. Generating landmark navigation

instructions from maps as a graph-to-text problem. In *Proceedings of the Association for Computational Linguistics (ACL)*, p. 489–502, 2021.

[28] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *Proceedings of International Conference on 3D Vision (3DV)*, pp. 667–676, 2017.

[29] An Yan, Xin Eric Wang, Jiangtao Feng, Lei Li, and William Yang Wang. Cross-lingual vision-language navigation, 2019.

[30] Piotr Mirowski, Matthew Koichi Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Koray Kavukcuoglu, Andrew Zisserman, and Raia Hadsell. Learning to navigate in cities without a map. In *Advances in Neural Information Processing Systems*, Vol. 31, p. 2424–2435, 2018.

[31] Harsh Mehta, Yoav Artzi, Jason Baldridge, Eugene Ie, and Piotr Mirowski. Retouchdown: Releasing touchdown on StreetLearn as a public resource for language grounding tasks in street view. In *EMNLP-SpLU*, p. 56–62, 2020.

[32] Karl Hermann, Mateusz Malinowski, Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, and Raia Hadsell. Learning to follow directions in street view. *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, Vol. 34, pp. 11773–11781, 2020.

[33] Arun Balajee Vasudevan, Dengxin Dai, and Luc Van Gool. Talk2nav: Long-range vision-and-language navigation with dual attention and spatial memory. *IJCV*, Vol. 129, No. 1, 2021.

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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: Hu-*

man Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, June 2019.

[35] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, Vol. 32, pp. 13–23, 2019.

[36] Yuankai Qi, Zizheng Pan, Yicong Hong, Ming-Hsuan Yang, Anton van den Hengel, and Qi Wu. The road to know-where: An object-and-room informed sequential bert for indoor vision-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1655–1664, 2021.

[37] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. In *Artificial Neural Networks: Formal Models and Their Applications*, pp. 799–804, 2005.

[38] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2961–2969, 2017.

[39] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

[40] Zhanghui Kuang, Hongbin Sun, Zhizhong Li, Xiaoyu Yue, Tsui Hin Lin, Jianyong Chen, Huaqiang Wei, Yiqin Zhu, Tong Gao, Wenwei Zhang, Kai Chen, Wayne Zhang, and Dahua Lin. MMOCR: A comprehensive toolbox for text detection, recognition and understanding. *CoRR*, Vol. abs/2108.06543, , 2021.

[41] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *Pro-*

ceedings of the Association for the Advancement of Artificial Intelligence (AAAI), Vol. 33, pp. 8610–8617, 2019.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[45] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[46] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 101–108, 2020.

[47] John W. Ratcliff and David E. Metzener. Pattern matching: The gestalt approach. *Dr. Dobb's Journal*, Vol. 13, No. 7, p. 46, July 1988.

[48] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, Vol. abs/1412.6980, , 2014.

[49] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 1715–1725, 2016.

[50] V Levenshtein. Leveinshtein distance, 1965.

[51] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. In *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 1862–1872, 2019.

[52] Gabriel Ilharco Magalhaes, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. In *NeurIPS Visually Grounded Interaction and Language (ViGIL) Workshop*, 2019.

[53] Evan Herbst, Peter Henry, Xiaofeng Ren, and Dieter Fox. Toward object discovery and modeling via 3-d scene comparison. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2623–2629, 2011.

[54] V Coletta, V Marsocci, and R Ravanelli. 3dcd: A new dataset for 2d and 3d change detection using deep learning techniques. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, p. 1349–1354, 2022.

[55] Tao Ku, Sam Galanakis, Bas Boom, Remco C Veltkamp, Darshan Bangera, Shankar Gangisetty, Nikolaos Stagakis, Gerasimos Arvanitis, and Konstantinos Moustakas. Shrec 2021: 3d point cloud change detection for street scenes. *Computers & Graphics*, pp. 192–200, 2021.

[56] Ken Sakurada and Takayuki Okatani. Change detection from a street image pair using cnn features and superpixel segmentation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[57] Kento Doi, Ryuhei Hamaguchi, Yusuke Iwasawa, Masaki Onishi, Yutaka Matsuo, and Ken Sakurada. Detecting object-level scene changes in images with viewpoint differences using graph matching. *Remote Sensing*, Vol. 14, No. 17, p. 4225, 2022.

[58] Ragav Sachdeva and Andrew Zisserman. The change you want to see. In *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*, pp. 3993–4002, 2023.

[59] Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust change captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4625–4633, 2019.

[60] Yue Qiu, Shintaro Yamamoto, Kodai Nakashima, Ryota Suzuki, Kenji Iwata, Hirokatsu Kataoka, and Yutaka Satoh. Describing and Localizing Multiple Changes with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1971–1980, 2021.

[61] Xin Hong, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. Transformation driven visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6903–6912, 2021.

[62] Yue Qiu, Yanjun Sun, Fumiya Matsuzawa, Kenji Iwata, and Hirokatsu Kataoka. Graph representation for order-aware visual transformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22793–22802, 2023.

[63] Harsh Jhamtani and Taylor Berg-Kirkpatrick. Learning to describe differences between pairs of similar images. In *Empirical Methods in Natural Language Processing*, p. 4024–4034, 2018.

[64] Rareş Ambruş, Nils Bore, John Folkesson, and Patric Jensfelt. Meta-rooms: Building and maintaining long term spatial models in a dynamic world.

In *International Conference on Intelligent Robots and Systems*, pp. 1854–1861, 2014.

[65] Shimaa Holail, Tamer Saleh, Xiongwu Xiao, and Deren Li. Afde-net: Building change detection using attention-based feature differential enhancement for satellite imagery. *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2023.

[66] Deyi Ji, Siqi Gao, Mingyuan Tao, Hongtao Lu, and Feng Zhao. Changenet: Multi-temporal asymmetric change detection dataset. *arXiv preprint arXiv:2312.17428*, 2023.

[67] Chenyang Liu, Rui Zhao, Hao Chen, Zhengxia Zou, and Zhenwei Shi. Remote sensing image change captioning with dual-branch transformers: A new method and a large scale dataset. *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 60, pp. 1–20, 2022.

[68] Yue Qiu, Yutaka Satoh, Ryota Suzuki, Kenji Iwata, and Hirokatsu Kataoka. 3d-aware scene change captioning from multiview images. *International Conference on Intelligent Robots and Systems*, pp. 4743–4750, 2020.

[69] Yue Qiu, Yutaka Satoh, Ryota Suzuki, Kenji Iwata, and Hirokatsu Kataoka. Indoor scene change captioning based on multimodality data. *Sensors*, Vol. 20, No. 17, p. 4761, 2020.

[70] Yue Qiu, Shintaro Yamamoto, Ryosuke Yamada, Ryota Suzuki, Hirokatsu Kataoka, Kenji Iwata, and Yutaka Satoh. 3d change localization and captioning from dynamic scans of indoor scenes. In *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*, pp. 1177–1185, 2023.

[71] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5922–5931, 2021.

[72] Xiangxi Shi, Xu Yang, Jiuxiang Gu, Shafiq Joty, and Jianfei Cai. Finding it at another side: A viewpoint-adapted matching encoder for change captioning. In *Proceedings of The European Conference on Computer Vision (ECCV)*, pp. 574–590, 2020.

[73] Zixin Guo, Tzu-Jui Wang, and Jorma Laaksonen. CLIP4IDC: CLIP for image difference captioning. In *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 33–42, 2022.

[74] Yunbin Tu, Liang Li, Li Su, Junping Du, Ke Lu, and Qingming Huang. Adaptive representation disentanglement network for change captioning. *IEEE Transactions on Image Processing*, pp. 2620 – 2635, 2023.

[75] Yunbin Tu, Liang Li, Li Su, Zheng-Jun Zha, Chenggang Yan, and Qingming Huang. Self-supervised cross-view representation reconstruction for change captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2805–2815, 2023.

[76] Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. A corpus of natural language for visual reasoning. In *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 217–223, 2017.

[77] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the Association for Computational Linguistics (ACL)*, p. 6418–6428, 2019.

[78] Sagar Vaze, Nicolas Carion, and Ishan Misra. Genecis: A benchmark for general conditional image similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6862–6872, 2023.

[79] Benjamin Ramtoula, Matthew Gadd, Paul Newman, and Daniele De Martini. Visual dna: Representing and comparing images using distributions of neuron activations. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition (CVPR), pp. 11113–11123, 2023.

[80] Frederik Warburg, Soren Hauberg, Manuel Lopez-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2626–2635, 2020.

[81] Utkarsh Mall, Bharath Hariharan, and Kavita Bala. Change-aware sampling and contrastive learning for satellite images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5261–5270, 2023.

[82] Favyen Bastani, Piper Wolters, Ritwik Gupta, Joe Ferdinando, and Aniruddha Kembhavi. Satlaspretrain: A large-scale dataset for remote sensing image understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16772–16782, 2023.

[83] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.

[84] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. pp. 17864 – 17875, 2021.

[85] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1290–1299, 2022.

[86] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3041–3050, 2023.

[87] Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Ping Luo, Zehuan Yuan, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15325–15336, 2023.

[88] Kaining Ying, Qing Zhong, Weian Mao, Zhenhua Wang, Hao Chen, Lin Yuanbo Wu, Yifan Liu, Chengxiang Fan, Yunzhi Zhuge, and Chunhua Shen. Ctvis: Consistent training for online video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 899–908, 2023.

[89] Junfeng Wu, Yi Jiang, Song Bai, Wenqing Zhang, and Xiang Bai. Seqformer: Sequential transformer for video instance segmentation. In *Proceedings of The European Conference on Computer Vision (ECCV)*, pp. 553–569, 2022.

[90] Tao Zhang, Xingye Tian, Yu Wu, Shunping Ji, Xuebo Wang, Yuan Zhang, and Pengfei Wan. Dvis: Decoupled video instance segmentation framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1282–1291, 2023.

[91] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

[92] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, 2023.

[93] OpenAI. Gpt-4 technical report, 2024.

[94] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.

[95] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, 2021.

[96] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 311 – 318, 2002.

[97] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4566–4575, 2015.

[98] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

[99] Jialu Li, Aishwarya Padmakumar, Gaurav Sukhatme, and Mohit Bansal. Vln-video: Utilizing driving videos for outdoor vision-and-language navigation. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, pp. 18517–18526, 03 2024.

[100] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

[101] Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. Aerialvln: Vision-and-language navigation for uavs. In *International Conference on Computer Vision (ICCV)*, pp. 15384–15394, 2023.

[102] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *International Symposium on Field and Service Robotics*, 2017.

[103] Kanishk Jain, Varun Chhangani, Amogh Tiwari, K Madhava Krishna, and Vineet Gandhi. Ground then navigate: Language-guided navigation in dynamic scenes. *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4113–4120, 2023.

[104] Ziqiao Ma, Benjamin VanDerPloeg, Cristian-Paul Bara, Yidong Huang, Eui-In Kim, Felix Gervits, Matthew Marge, and Joyce Chai. DOROTHIE: Spoken dialogue for handling unexpected situations in interactive autonomous driving agents. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 4800–4822. Association for Computational Linguistics, December 2022.

[105] Jacob Krantz, Erik Wijmans, Arjun Majundar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision and language navigation in continuous environments. In *European Conference on Computer Vision (ECCV)*, pp. 104–120, 2020.

[106] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *Neural Information Processing Systems (NeurIPS)*, pp. 3318 – 3329, 2018.

[107] Raphael Schumann, Wanrong Zhu, Weixi Feng, Tsu-Jui Fu, Stefan Riezler, and William Yang Wang. Velma: Verbalization embodiment of llm agents for vision and language navigation in street view. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, pp. 18924 – 18933, 2023.

[108] Zhan Wang, Lin-Ping Yuan, Liangwei Wang, Bingchuan Jiang, and Wei Zeng. Virtuwander: Enhancing multi-modal interaction for virtual tour

guidance through large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2024.

[109] Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee Wong. MapGPT: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 9796–9810. Association for Computational Linguistics, August 2024.

[110] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, Vol. 33, pp. 1877–1901, 2020.

[111] Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. Text classification via large language models, 2023.

[112] Yiming Wang, Zhuosheng Zhang, and Rui Wang. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 8640–8665. Association for Computational Linguistics, July 2023.

[113] Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. Multilingual machine translation with large language models: Empirical results and analysis, 2023.

[114] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, Vol. 36, pp. 34892–34916, 2023.

[115] Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. Large language model as attributed training data generator: a tale of diversity and bias. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23. Curran Associates Inc., 2024.

[116] John Chung, Ece Kamar, and Saleema Amershi. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 575–593, July 2023.

[117] Minju Seo, Jinheon Baek, James Thorne, and Sung Ju Hwang. Retrieval-augmented data augmentation for low-resource domain tasks, 2024.

[118] Dong-Ho Lee, Jay Pujara, Mohit Sewak, Ryen W White, and Sujay Kumar Jauhar. Making large language models better data creators. In *The 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 15349–15360, 2023.

[119] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[120] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, p. 226–231. AAAI Press, 1996.