AI

MDPI

Article

History-Aware Multimodal Instruction-Oriented Policies for Navigation Tasks

Renas Mukhametzianov * and Hidetaka Nambo *

Division of Electrical Engineering and Computer Science, Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa 920-1192, Japan

* Correspondence: renasmukha@gmail.com (R.M.); hide_nambo@staff.kanazawa-u.ac.jp (H.N.)

Abstract: The rise of large-scale language models and multimodal transformers has enabled instruction-based policies, such as vision-and-language navigation. To leverage their general world knowledge, we propose multimodal annotations for action options and support selection from a dynamic, describable action space. Our framework employs a multimodal transformer that processes front-facing camera images, light detection and ranging (LIDAR) sensor's point clouds, and tasks as textual instructions to produce a history-aware decision policy for mobile robot navigation. Our approach leverages a pretrained vision–language encoder and integrates it with a custom causal generative pretrained transformer (GPT) decoder to predict action sequences within a state–action history. We propose a trainable attention score mechanism to efficiently select the most suitable action from a variable set of possible options. Action options are text–image pairs and encoded using the same multimodal encoder employed for environment states. This approach of annotating and dynamically selecting actions is applicable to broader multidomain decision-making tasks. We compared two baseline models, ViLT (vision-and-language transformer) and FLAVA (foundational language and vision alignment), and found that FLAVA achieves superior performance within the constraints of 8 GB video memory usage in the training phase. Experiments were conducted in both simulated and real-world environments using our custom datasets for instructed task completion episodes, demonstrating strong prediction accuracy. These results highlight the potential of multimodal, dynamic action spaces for instruction-based robot navigation and beyond.

Keywords: multimodal transformer; autonomous navigation; imitation learning

check for updates

Academic Editor: Luis Javier Garcia Villalba

Received: 11 February 2025
Revised: 5 April 2025
Accepted: 9 April 2025
Published: 11 April 2025

Citation: Mukhametzianov, R.; Nambo, H. History-Aware Multimodal Instruction-Oriented Policies for Navigation Tasks. *AI* **2025**, *6*, 75. https://doi.org/10.3390/ai6040075

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The field of transformer models has shown remarkable growth in machine learning in general and natural language processing (NLP) in particular [1–3]. Their self-attention mechanism effectively models long-range dependencies and works well with sequence-structured data. Structuring data domains as sequences enables their application to various fields, such as computer vision [4,5] and decision-making [6], as well as their combinations in multimodal tasks [7].

In such conditions, transformer-based language models are promising candidates for robotics and decision-making. Instruction-based systems implemented in robotics are often required in scenarios involving close interactions with humans. The use of natural language, whether verbal or written, makes it accessible to people. Many real environmental tasks require visual observations. In particular, vision-and-language navigation (VLN), which involves guiding robots to complete tasks in real-world environments by interpreting

visual data and natural language instructions, has gained significant attention [8,9] as a fundamental task supporting various robotic operations.

In many real-world environments, the decision-making process relies on previously observed states and performed actions. For example, consider a navigational task given to a mobile robot agent: "Go outside of the office, but if the door is closed, go back to the starting point". The agent must retain a history of either performed actions or observed states to complete the task. History-aware multimodal transformers (HAMTs) [10] have been proposed, which encode sequences of visual and language data with a history of observations effectively replacing long short-term memory (LSTM) [11] for low-level vision-and-language navigation (VLN) task [12].

Despite progress in vision-and-language navigation, some challenges still exist. Low-level navigation requires gathering navigation demonstrations that should demonstrate certain skills in optimal path finding, obstacle avoidance, etc., that leads to large and diverse datasets. Training smart systems to learn these skills is less practical, especially given the availability of reliable semiautomatic LIDAR-based routing systems, which are widely used in mobile robotics. Low-level navigation requires the collection of action and observation history at a certain rate, including visual observations, which results in inefficient memory utilization and constraints on history retention. In mobile applications, computational efficiency is always a priority, so the use of a single graphical processing unit (GPU) is a practical constraint.

The core idea of this work is the integration of a semiautomatic LIDAR-based path-searching system with a vision–language model (VLM) pretrained on visual and textual data, leveraging its rich interconnections, such as visual question answering (VQA), for high-level decision-making. In our approach, prior to navigation, we prepare action options with vision–language multimodal annotations and incorporate dynamic action space adaptiveness. A dynamic action space is necessary for high-level navigation because navigational targets form a unique set in each environment, differing in both description and length.

The key contributions of this work are as follows:

- Integration of a vision–language model (VLM) with a LIDAR-based planner to enhance high-level navigation decision-making.
- A novel dynamic action space approach using an attention score classifier, which enables multiple choices from a set of describable options through a transformer-inspired learnable layer.

As a result, our approach reduces the need for large training datasets while maintaining effective navigation skills. Additionally, our state–action history grows at a slower rate due to high-level decision-making, leveraging both the deep world knowledge of the vision–language model (VLN) and the real-time reliability of the semiautomatic LIDAR-based planner. Furthermore, a dynamic and describable action space allows future advanced smart systems to transition seamlessly not only between different navigational environments but also across various task domains.

2. Related Work

The introduction of large-scale transformer-based models has significantly impacted the field of natural language processing. In the paper "Language models are few-shot learners" [13], the authors demonstrate that language models can generalize to various tasks without extensive task-specific fine-tuning based on a few demonstration examples in few-shot learning settings. They show the ability to combine, in a single prompt, the task description, all possible action descriptions, and a few examples of correct decisions. Text-based descriptions of possible actions are crucial for switching models between different

action spaces and varying numbers of action options. A multimodal (vision–language) approach for describing both the task and action options within a single prompt incorporating images is presented by Dai et al. [14]:

The ability to build state–action sequence policies from such models is limited for tasks where every state of the environment contains an image. To produce a single action, all previous actions taken and every observed state, including images, must be retained as context, leading to rapid memory growth.

In the work "Do as I can, not as I say: grounding language in robotic affordances" [15], such operations are performed by a large language model (LLM) utilizing a predefined list of skills for robotics. The skills are described textually, and the decision-making LLM decomposes task instructions into a sequence of text-described skills. The probability of the skill option making progress towards the given task completion is estimated based on the LLM's aggregated probability of generating its textual description. The paper demonstrates a describable approach for action annotation and the potential to select from a varying number of action options, estimating them one by one. However, large language models (LLMs) are primarily unimodal, pretrained on language data, and disconnected from visual representations, making them limited for multimodal tasks. This approach is not adaptable for extrapolation to a vision-and-language framework in terms of multimodal action representation.

In contrast, our approach estimates all possible options simultaneously in a single forward pass of the vision–language multimodal model.

Some approaches to multimodal history-aware policies address the issue of growing history [16,17]. These approaches target policies for a 7 DoF gripper manipulator robot in the RLBench [18] virtual environment and real-world implementation. Specifically, in the InstructRL [17] framework, they use a moment-wise encoder to convert every state into a rich representation embedding and a history-aware decoder to predict the next action based on the context of all previous states and actions. For effective gripper manipulation, inverse kinematic automation is utilized, enabling action prediction at a higher level to produce target joint positions, which are executed semiautomatically. The macro-step approach [19] reduces the frequency of decision-making actions and limits the growth of history context sequences.

While utilizing history awareness, these policies have a fixed action space fine-tuned to the robot's inner structure. They do not generalize to navigational tasks, and macro-steps remain a separate consideration for such tasks.

Therefore, for navigation tasks of a mobile robot, shifting to macro-steps is proposed in our work as actions represented by target coordinates on a LIDAR-generated map instead of real-time driving signals. Achieving the target position on the map is taken out of the responsibility of the decision-making system. Moreover, in our work, target points are predefined and described, shifting the task focus from low-level vision–language navigation (VLN) to high-level decision-making.

In our research, we develop compressed history-awareness with a dynamic action space and action annotation capability.

3. Materials and Methods

3.1. Problem Definition

Multimodal Representation. For decision-making policies in navigation tasks, a moment-wise multimodal representation M, adapted for processing by a vision–language multimodal transformer, is used. It contains an image M_I from the robot's front-facing camera and a prompt containing natural language text M_P. Multimodal representation is applied both for states (M^S) and actions (M^A):

- State representation: $M^S = \left(M_I^S, M_P^S\right)$, where:
  M_I^S represents the robot's current front facing observation and local LiDAR sensor's observation.
  M_P^S is a copy of the global navigation task prompt given to the robot at the beginning of the experiment.
- Action representation: $M^A = \left(M_I^A, M_P^A\right)$, where:
  M_I^A provides visual reference to the action option.
  M_P^A contains textual description of the action option.

Predefined environment data. Before navigation begins, we define static environment data:

- Occupancy Grid Map O: A LIDAR-based map of the environment.
- Action Vocabulary: A set of predefined navigation targets represented as:

$$\mathcal{A} = \left\{ \left(x_i, y_i, \theta_i, M_i^A\right) \right\}_{i=1}^{N} \tag{1}$$

where each action option is defined by:

- (x_i, y_i, θ_i): Coordinates and orientation on the map O.
- M_i^A: The multimodal representation of the action option, serving as a reference for the decision-making model.

Multimodal representations of action options in the action vocabulary are not related to any timestamp. The visual part serves as an example of the surrounding environment, while the textual prompt contains a description of the relative location in the environment. Thus, selecting action $a_t \in \mathcal{A}$ means forming a command for navigation planner to move to that corresponding coordinate.

Decision-Making as a Sequence Task. In addition to predefined occupancy map and action vocabulary, in a dynamic environment, observable data are split into either a state S_t or action A_t related to timestamp t. States and actions are resolved in pairs attached to the same timestamp, implying minimal observation change during the time between the captured state and action. In our problem, we define autonomous navigation as a sequential decision-making task, where at a timestamp t, multimodal representation of current state M_t^S and history of all previous states and actions in the episode, denoted as $H_t = \{M_0^S, M_0^A \ldots M_{t-1}^S, M_{t-1}^A\}$, are accessible. Each episode starts with the submission of a textual navigation task, which triggers the first state capture and ends with the selection of an end-of-sequence action option. The number of states and actions in episodes is equal, and the end-of-sequence token has no multimodal representation.

Policy Objective. The goal of our policy is to map a sequence of states and history into a sequence of navigation actions, dynamically selecting the most appropriate action from the given action vocabulary:

$$\pi_\alpha\left(H_t, M_t^S, \mathcal{A}\right) \rightarrow a_t \in \mathcal{A} \tag{2}$$

where:

- π_α is the history-aware decision-making policy parametrized with α;
- $H_t = \{M_0^S, M_0^A \ldots M_{t-1}^S, M_{t-1}^A\}$ is the state–action history up to time t in multimodal representation;
- M_t^S is current state's multimodal representation;
- $\mathcal{A} = \left\{ \left(x_i, y_i, \theta_i, M_i^A\right) \right\}_{i=1}^{N}$ is the action vocabulary;
- a_t is a selected action option with multimodal representation used by policy and coordinates and orientation required by navigation planner, supplied by occupancy grid map.

3.2. State

Each state is an image–prompt pair, containing an observation of the environment and a copy of the given task prompt. The task prompt is a natural language instruction provided to the robot at the beginning of the episode, specifying the navigation goal. It describes the desired destination or sequence of actions the robot should take. The image consists of two concatenated parts: an image from the front camera of the mobile robot and a LIDAR point cloud map of the surroundings with an "arrow" anchor, clarifying the robot's relative localization on the map (Figure 1a). Unlike LIDAR, which provides relative localization context and obstacle profiling, the camera enables contextual perception, allowing the robot to interpret visual cues that influence navigation decisions, especially for the navigational tasks conditioned on visual observations.

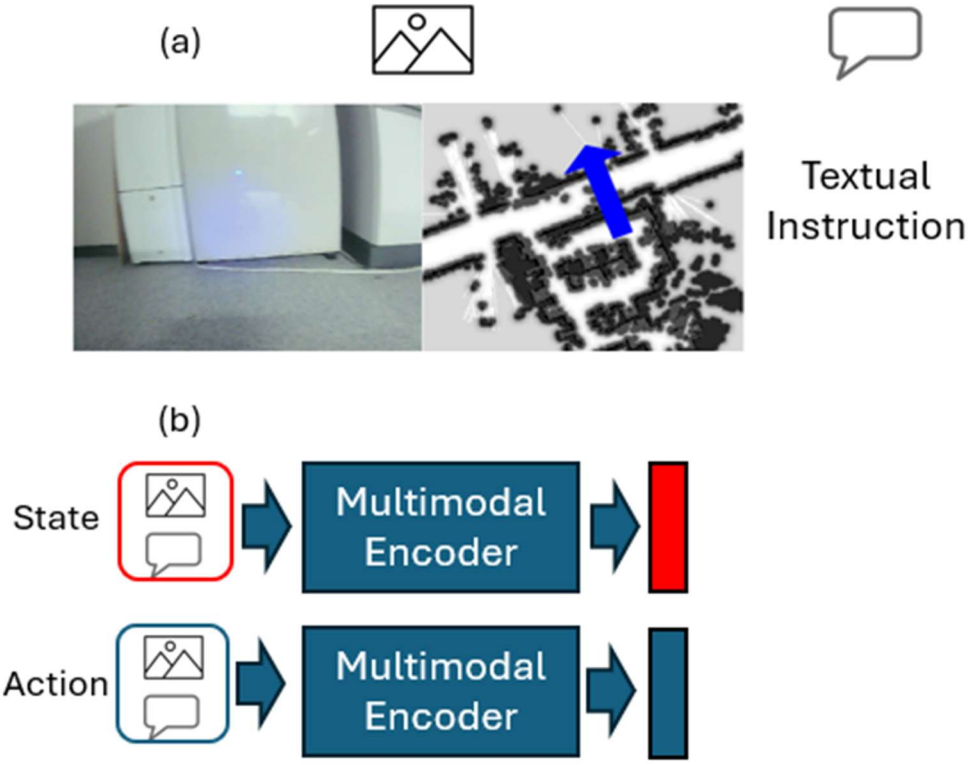

Figure 1. State and action representations. (**a**) Example of a multimodal state or action representation: a front-facing camera image, a LIDAR point cloud map (with an arrow indicating the robot's localization), and a corresponding textual instruction. (**b**) Multimodal transformer encoder processes both state and action representations, producing token embeddings for decision-making.

3.3. Actions

Low-level deterministic navigation. The task of LIDAR-equipped navigation of mobile robots is deterministic and well established. It is beneficial to use classical deterministic methods rather than iterative machine learning models for path planning and motion control for several reasons. The classical approach does not require data collection or trial-and-error iterations with a reward loop associated with achieving the navigational goal or avoiding obstacles. It reduces the dependency on machine learning training for high-level decision-making, thereby lowering dataset requirements. Sequences of states and actions are reduced through the implementation of high-level decision-making. The independent navigation system performs real-time path routing and driving while avoiding obstacles, with an adjustable execution rate.

LIDAR-based pathfinding and navigation are supported within the ROS (robot operating system) [20]. An occupancy map of the environment is generated by the simultaneous

localization and mapping (SLAM) algorithm [21]. The navigation system includes a global planner, a local planner, and a PID (proportional–integral–derivative) controller [22]:

- The global planner (global_planner ROS package) finds a route from the current robot position to the target point on the occupancy map, commonly using the A* algorithm [23]. The A* algorithm is preferred over Dijkstra's algorithm [24] due to its reduced search time. Since Dijkstra's algorithm visits all nodes while computing the total cost from the starting point. A* visits nodes in order of their heuristic function values, which is often the Euclidean distance.
- The local planner (dynamic window approach—DWA) [25] evaluates possible translational and rotational velocities within a limited time window and selects the best option for following the global path while avoiding obstacles.
- The PID (proportional–integral–derivative) motion controller [22] adjusts these velocity commands to ensure smooth execution and correct trajectory deviations before they are translated into differential drive signals for the robot's motors.

Detailed considerations on path planning and motion control within the ROS navigation stack can be found in Zhao et al. [26].

High-level decision-making, where an action is selected as the target waypoint on the map and low-level path-planning and motion control are delegated, incorporates the advances of a modular approach. One such advantage is the flexibility of selecting the low-level algorithm for special needs like instability of control signals and nonlinear environmental disturbances. Advanced techniques of nonlinear robust control, such as multilayer neurocontroller [27] and state-filtered disturbance rejection control (SFDRC) [28], can replace the classical PID controller in future work, enhancing the stability and robustness of robotic navigation systems.

High-level annotated actions. While it is possible to simultaneously search for the route and build the occupancy map using a LIDAR sensor, we specify the following prior annotations for our experiments:

1. Full map of experimental environment surroundings in format of LIDAR point cloud.
2. List of points of interest in the environment—the only action options can be chosen as navigation target. Each action option is annotated with an image from a front camera (visual observation from that point of interest), a LIDAR point cloud with "arrow" anchor depicting relative position and orientation, and prompt (textual description of that point of interest) (Figure 1a).

Thus, we define actions as waypoints on the LIDAR-based occupancy map, annotated with an image and a textual description, ensuring consistency between state and action representation. The distinction is that actions are annotated prior to any experiment, whereas states are observed during the experimental episode.

3.4. Datasets

The reinforcement learning (RL) approach is widely used for robotic tasks involving agent–environment interactions. In this approach, the agent learns by exploring the environment, performing actions, and observing states, along with receiving rewards to estimate action quality.

However, in many robotic applications, the focus shifts to a supervised approach, where the objective is to predict the exact patterns of demonstrated behavior. This approach requires a dataset of state–action sequence episodes, where every action is considered correctly executed. We accept that approach for our paper in the form of behavioral cloning [29]. Considering every action in the demonstration dataset as correct and a target to predict, we do not provide any other quality estimation, such as a reward function.

In our case, the dataset consists of episodes of environment states in the form of a front camera image, a LIDAR point cloud map, and a task in a natural language prompt. Each state in the episode is paired with an action choice from a predefined annotated list. A state always consists of a single bundle of a camera image, a LIDAR map, and a prompt gathered at that moment the next action is expected. Semiautomatic navigation actions may take varying amounts of time, during which no data are gathered.

Gathering such a dataset in practice involves a preparation phase and the collection of demonstration episodes. During preparation, an occupancy map of the surrounding area is collected using a LIDAR sensor, covering the entire area available for navigation. In this environment, a set of points of interest is manually defined as potential navigation targets, with their relative coordinates and poses specified on the occupancy map. Each point of interest (action option) is described using a short textual prompt and visited by the mobile robot once to collect an image from the front camera view.

The collection of dataset episodes starts with sending a task prompt message and capturing the first state's image from the front camera along with the relative pose from the occupancy map. The most appropriate action option is chosen from the described action vocabulary and recorded in the sequence. After performing a navigation step, the robot captures the next state and the process repeats until the end of the sequence, where a special action option is chosen in the case of successful episode completion. The robot moves to the starting point for the next episode, which begins with receiving a task prompt. During dataset gathering, the performed actions are considered correct with respect to the current observation, previously performed actions, prior observations, and action option annotations.

3.5. Dynamic Action Space

For navigation approaches that strongly rely on environment annotation, we must state an additional requirement: dynamic action space adaptivity for our model. This is necessary for updating the annotation of an existing environment, such as adding new action options to the existing action vocabulary. Similarly, different environments should have different annotated action vocabularies with corresponding possible poses to achieve within them. More generally, the ability of a model to flexibly adapt to switching between action spaces is a promising feature for multidomain and multitask learners.

3.6. Model

Our model consists of the following components:

1. Moment-Wise Multimodal Transformer Encoder. This accepts a prompt–image pair and produces a rich representation embedding as a single token. It works uniformly with states and actions represented as image–prompt pairs. From annotated action options, it creates embeddings and forms a so-called action vocabulary (Figure 2a).

Training demonstration episodes are structured as sequences of states and actions. States consist of camera images and surrounding occupancy point clouds captured during semiautomatic navigation to predefined target points on the map, while actions are identifiers of those points in the action annotation list. The moment-wise encoder processes a dataset of state–action sequences and annotated action options to generate embeddings for every action option and state in the episode, representing each as a single token embedding.

We accept vision–language transformers (ViLTs) [30] and FLAVA [31] pretrained for visual question answering (VQA) as a baseline model.

2. History-Aware Causal Decision-Making Transformer. Based on the history of all previous states and actions (and masked future states and actions), this predicts the

next action, agnostic of action options, in the form of a transformer token of the same shape (Figure 2b).

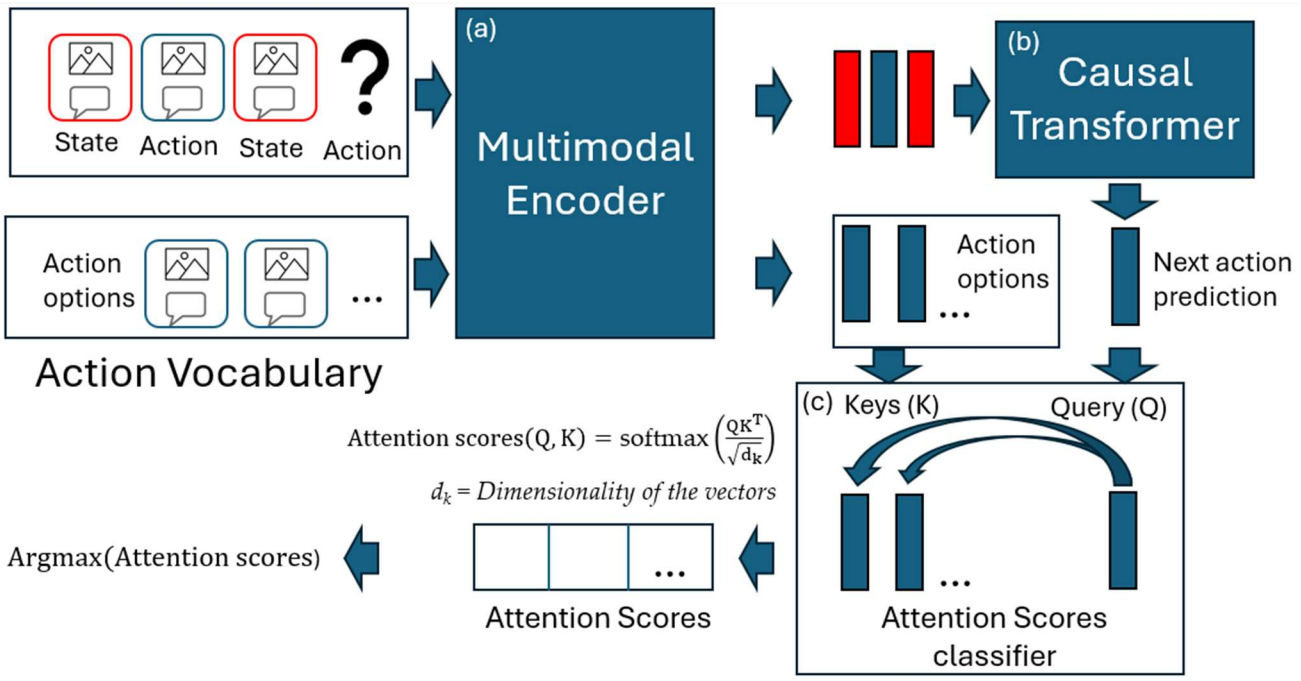

Figure 2. History-aware multimodal policies—model overview. Pipeline of next action selection from state–action sequence and action options vocabulary. (**a**) Multimodal transformer encoder—converts vision–language representations into transformer-token embeddings. (**b**) History-aware causal decision-making transformer—predicts the next action token by the history of state–action tokens. (**c**) Attention score classifier—performs multiple option selection by option-agnostic action prediction and returns preference scores for every given annotated action token.

This component consists of self-attention transformer layers with a causal mask, which hides future (later in sequence) tokens for attention computation at every position. It is initialized without pretrained checkpoints, and the number of attention layers is treated as a training hyperparameter. After passing through the layers, we generally consider each token as a prediction of the next one. In our case, we use only the predictions of the next actions. We utilize the GPT (generative pretrained transformer) architecture [32] as a causal decision-making transformer.

3. Attention score classifier. From action option-agnostic prediction, this selects the most relevant token in action vocabulary. Although this is still a classification task, it differs from traditional methods because the number of action classes is dynamic (Figure 2c). Every action prediction and action option from vocabulary is a numeric vector–transformer-produced embedding. We propose using attention scores with action agnostic prediction as a query (Q) and stacked action vocabulary embeddings as a key matrix (K) (Equation (3)). Attention scores are part of transformer's attention mechanism [1] and enable self-attention to work with various input sequence length. We utilize this ability for handling various multiple-option choices. Attention scores, being a vector of numbers with the length of keys vectors, can be used as "digits" for classification. This approach is trainable if queries (Q) and keys (K) have trainable parameters (weights), and it is scalable, allowing the insertion of self-attention layers [1] before the attention score classifier.

Practical implementation of the dynamic action space in the proposed framework includes separate on-demand calls to the moment-wise encoder for action vocabulary to

form single-vector embeddings for action options from their multimodal vision–language annotations. The call for new action option embeddings may be triggered by any change in the action vocabulary, including modification of an action option annotation, the addition or removal of action options, or a full change of the action annotation when switching to a new environment. During the training phase, action embeddings may be recomputed to update the representation at arbitrary frequency. The current approach provides flexibility in the usage of the framework: no changes in the action space introduce architectural changes to the model or introduce new trainable parameters, demonstrating its adaptability to dynamic action spaces.

$$\text{Attention scores}(Q, K) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{3}$$

Here:

Q = Query matrix
K = Key matrix
d_k = Dimensionality of the key vectors (a scalar)

4. Experimental Setup

Navigation experiments were conducted in two domains.

1. Gazebo virtual environment—simulation of robot and the student office. A simulation of the robot and the student office. In the ROS-based virtual environment engine Gazebo, we built a robot simulation that includes simplified body joints, differential drivers, a camera, and LIDAR sensors (Figure 3). For the environment, we used a LIDAR occupancy grid gathered in the student office to represent inelastic walls. The environment utilized the default physics engine of Gazebo and was modified with standard library visual and physical objects, such as a fridge and a shelf.
2. A real-world environment—the student office room. In this domain (Figure 4), the Xiaor Geek mobile robot (Figure 5) equipped with a camera and LIDAR and running on an ROS (robot operating system) was used. The robot was connected to a remote server via Wi-Fi, sending sensor observations (camera image stream, LIDAR point cloud occupancy map, and odometry measurements) and receiving control signals as differential driver voltages.

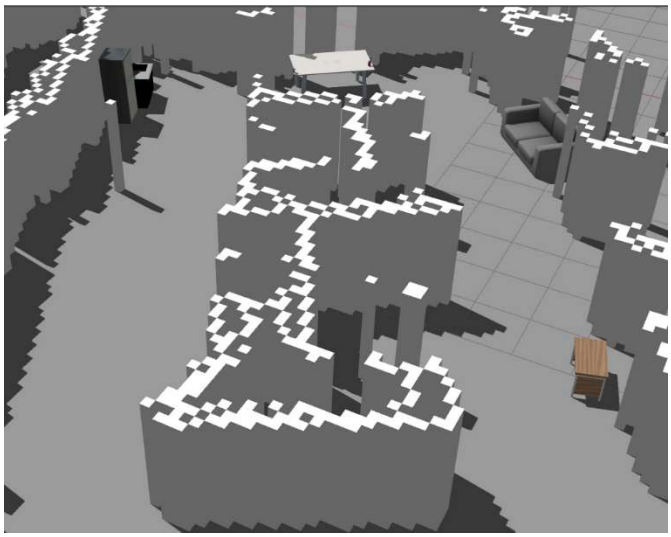

Figure 3. Gazebo virtual simulation environment for navigation tasks with the mobile robot.

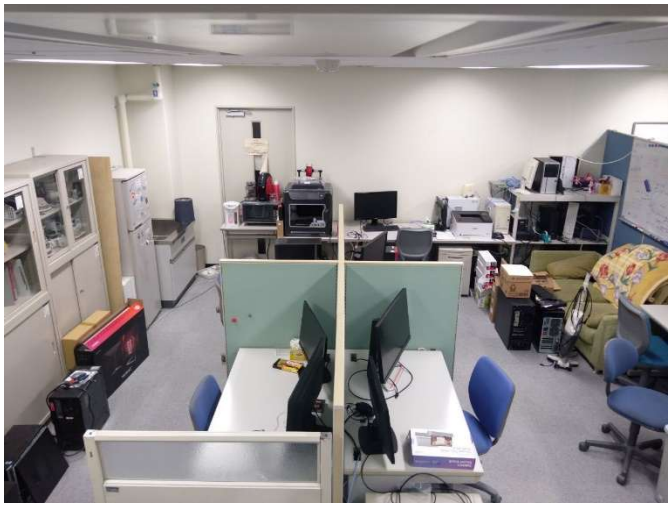

Figure 4. Real-world environment for navigation tasks with the mobile robot.



Figure 5. XiaoR Geek LIDAR ROS mobile robot.

The attention score classifier output logits are treated as multiclass classification logits and optimized minimizing the categorical cross-entropy loss function $\mathcal{L}$, defined as:

$$\mathcal{L} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{4}$$

where:

- y_i is 1 for true label and 0 for false (one-hot encoding);
- $\hat{y}_i$ is predicted probability of i being a true class;
- N is total number of classes.

For machine learning computations, we utilized two Nvidia RTX A4500 GPU units. For training with the AdamW optimizer, we used cycles of learning rate cosine annealing with a decay factor of 10 over 100 epochs, followed by a constant learning rate. These rounds are repeatable, with learning rates dropping by a factor of 10 and initial rates set to 10^{-6} and 10^{-7}. The warmup phase uses a linear increment to the initial learning rate over 10 epochs. For the baseline model, we used the multimodal encoder ViLT [30] with a token dimension (d) of 768. For the state–action decoder, we used GPT [32] without pretraining.

State and action tokens were assigned separate modality-encoding vectors and distinct 1D positional encodings.

5. Results

In the Gazebo virtual environment, we gathered a dataset of 100 episodes of simple navigational tasks such as "go to the fridge." The virtual environment was designed to closely resemble the real-world domain, with the LIDAR occupancy map demonstrating an almost-perfect match to the real-world environment, while visual complexity was kept minimal. In this domain, two points of interest were defined as action options: "position facing the fridge and rack" and "position outside the lab facing the hallway."

The primary objective in this initial stage of experimentation was to achieve 100% accuracy in both training and validation, which served as a benchmark for potential baseline model selection. This goal was intended to establish the feasibility of a small GPT-based decision-making policy in controlled, low-variability conditions before extending the approach to more complex real-world environments.

For the experiments we used two baseline encoders trained on visual question answering (VQA):

- **ViLT** (Vision-and-Language Transformer) [30]—a lightweight and efficient vision–language transformer that directly processes image–text pairs without region-based feature extraction using 16*16 pixels image patches embedded with trainable linear projections together with language BERT-like embeddings within the same lightweight transformer.
- **FLAVA** (Foundational Language and Vision Alignment) [31]—A multimodal transformer that learns strong representations from both unimodal (unpaired images and text) and multimodal (image–text pairs) data. FLAVA consists of separate image and text encoders, both based on transformer architectures, followed by a multimodal transformer encoder that fuses the unimodal representations for integrated reasoning.

In the real-world domain, we prepared five distinct action options corresponding to target points of interest and a dataset of 77 episodes of complex navigation tasks, such as "Go to the fridge, then go to the right hall outside the lab. If there is a pink cone, turn left into the hall." For both baseline encoders, we added our decision-making causal transformer and attention score classifier with random weight initialization. The results for different numbers of self-attention layers in decision-making GPT are shown in Table 1.

Table 1. Real-world domain (77 episodes) camera–LIDAR instruction-based training results.

Encoder Model	ViLT		FLAVA	
	Training Accuracy, %	Validation Accuracy, %	Training Accuracy, %	Validation Accuracy, %
GPT layers				
6	86	77	87	87
7	84	86	91	88
8	84	86	90	87

For the investigation of potential ability of the sim-to-real transfer approach to achieve better convergence in the real-world setting, we trained our model in the same real-world domain after pretraining in a virtual environment. The results of that approach are shown in Table 2.

Table 2. Real-world domain (77 episodes) camera–LIDAR instruction-based training with virtual environment Gazebo (100 episodes) pretraining.

Encoder Model	ViLT		FLAVA	
	Training Accuracy, %	Validation Accuracy, %	Training Accuracy, %	Validation Accuracy, %
GPT layers				
6	86	84	91	89
7	86	87	96	94
8	87	87	89	87

We measured the GPU VRAM usage during the training phase for both ViLT and FLAVA baseline encoders with the same variety of GPT layers used and batch size of one (Table 3). Increasing the batch size primarily adds VRAM usage due to additional data size, but the memory required by the model's weights and gradients remains constant. The addition of more GPT layers results in an approximately linear increase in VRAM consumption, as expected (Table 3). While VRAM usage can be theoretically estimated, we conducted empirical measurements with Nvidia's nvidia-smi tool to capture the combined impact of model activations, attention computations, and memory overhead, ensuring accurate insights into the practical feasibility of training on resource-constrained GPUs. The video memory usage demonstrates that the proposed training pipeline can be executed on systems with a single 8 GB GPU, which aligns with widely available consumer-grade GPUs.

Table 3. GPU video memory usage for training phase, batch size 1.

Encoder Model	ViLT	FLAVA
	GPU VRAM Usage, MB	GPU VRAM Usage, MB
GPT layers		
6	3987	6601
7	4127	6733
8	4287	6879

6. Ablation Study

The classical classification approach of multilayer perceptron (MLP) is not appropriate for tasks with a changing number of classes, as long as its weights are tied to specific class outputs, making it unable to adapt dynamically. The closest comparison to our proposed attention score method is the cosine similarity metric, which also relies on the dot product operation between vectors. In our case, this dot product occurs in the form of a query vector compared against a key matrix, where each key represents a candidate action embedding.

The key difference lies in how the query and keys are formed. While cosine similarity uses fixed embeddings, the attention score mechanism utilizes trainable parameters for generating both the query and key vectors. This feature potentially allows the attention score method to adaptively learn relationships between predictions and dynamically changing action options, which is not achievable with static similarity metrics. To validate this, we performed an ablation study by replacing the attention score mechanism with cosine similarity on the same classification task using the ViLT baseline model (Table 4). The results clearly demonstrate the limitations of cosine similarity: while it achieves acceptable performance in the simpler virtual environment with only two action options, its performance drops significantly in the real-world dataset with five action options.

Table 4. Attention scores vs. Cosine Similarity as multiple option classification layer.

GPT Layers	Dataset	Action Options	Attention Scores		Cosine Similarity	
			Training Accuracy, %	Validation Accuracy, %	Training Accuracy, %	Validation Accuracy, %
6	Virtual env. Gazebo 100 episodes	2	100	100	100	100
8	Real env. 77 episodes	5	84	86	55	50

7. Discussion

7.1. Trainable Parameter Distribution

To estimate the trainable parameters in our proposed model, we suggest a conservative approximation focused on the quadratic scaling of parameters in a standard transformer layer. Specifically, the dominant part of the parameter growth can be approximated as $\mathbf{12d^2}$, where **d** is the hidden dimension size:

$\mathbf{4d^2}$ accounts for the self-attention mechanism (query, key, and value projections, along with output projection),

$\mathbf{8d^2}$ represents the parameters for the two fully connected layers in the position-wise feedforward network, with the hidden layer typically using a dimension size of 4d.

For simplicity, we ignore smaller contributions, such as the **2d** parameters for layer normalization and potential projections of image embeddings or token embeddings. Additionally, optimizations like parameter sharing or mixed precision computations are not included in this estimation.

Both encoder models, ViLT and FLAVA, utilize BERT-style vocabulary projections from textual tokens into **d**-dimensional embeddings, but FLAVA has trainable projections. To reflect this, we include an additional term of **V * d**, where **V** is the vocabulary size of language embeddings, to account for the FLAVA's trainable parameter estimation.

In our experiments, the hidden dimension size (**d**) is 768 and vocabulary size **V** is 30,000. The dimension size (d) of 768 was proposed in the original transformer paper [1] as an empirical compromise between computational cost and representational capacity of embeddings. The same dimension size **d** was used both in FLAVA and ViLT and was adopted in our paper for causal transformer and attention score classifier to ensure consistency. The vocabulary size **V** of 30000 tokens is derived from BERT's original WordPiece [33] tokenization, ensuring compatibility with pretrained models.

Using our conservative parameter estimation method, we calculate the trainable parameters for the baseline models and our proposed components as follows:

- **ViLT:** transformer with 12 attention layers (Table 5).
- **FLAVA:** FLAVA consists of three components: 12 layers for vision, 12 layers for natural language, and 6 multimodal layers together with vocabulary projections.
- **Decision-Making GPT:** For our history-aware GPT decision-making component, six-, seven-, and eight-layer options were used.
- **Attention Score Classifier:** The attention score classifier adds a lightweight $\mathbf{2d^2}$ parameter contribution.

Table 5. Trainable parameter estimation for different parts of model with d (dimension size) = 768, V (BERT-based vocabulary size) = 30,000, and number of trainable parameters, shown in millions (M).

Model Part	ViLT	FLAVA	Decision-Making GPT	Attention Scores
Estimation	$12*12d^2$	$30*12d^2 + Vd$	$\{6, 7, 8\}*12d^2$	$2d^2$
Parameters	84.9 M	235.3 M	{42.5 M, 49.5 M, 56.6 M}	1.2 M

The estimation of free parameters highlights the computational efficiency of our proposed framework. While baseline models like ViLT and FLAVA require 84.9 M and 235.3 M parameters, respectively, our decision-making GPT remains lightweight with 42.5 M–56.6 M parameters, depending on the number of layers. The attention score classifier adds a compact overhead of 1.2 M parameters, maintaining a minimal impact on resource usage while having a significant impact on results.

7.2. Inference Time and Real-Time Feasibility

We conducted an empirical analysis of inference latency in the real-world environment during the forward pass of the framework, focusing on its real-time deployment feasibility. For ViLT-based models, the forward pass time was consistently under 0.23 s, while for FLAVA-based models, it remained under 0.35 s. Notably, increasing the number of GPT layers (from six to eight, as tested in our other experiments) did not lead to significant changes in latency. Similarly, increasing the state–action sequence length to 100 elements in embedding form resulted in no measurable slowdown. However, if full multimodal representations (image + text) were retained instead of compressed embeddings, a linear increase in latency was observed, approximately 0.07 s per additional state–action pair. For example, with the ViLT encoder, a history of 10 such multimodal elements results in a forward pass time that can exceed 1 s. Since full multimodal retention is used during dataset collection and not required during policy inference, this overhead does not impact deployment. These results demonstrate the computational efficiency of our approach for real-time use in robotics settings.

7.3. Scalability

The scalability of our approach is relevant for both scaling up and scaling down the model to fit different computational requirements. Scaling up allows for the use of larger, more advanced vision–language models as baselines, which improves world knowledge and overall quality. Scaling down is essential for deployment in onboard mobile robot equipment, where computational efficiency is a priority. Our key addition, the attention score classifier, is inherently scalable due to its use of the transformer attention mechanism. Since the classification logic is based on attention, it maintains flexibility with respect to the dimension size of the transformer. This ensures that changes to the baseline model's size or complexity are naturally absorbed by the classifier. The number of GPT layers can also be adjusted.

7.4. Limitations

The experiments presented in the paper used relatively small datasets, with 77 episodes in the real-world environment and 100 episodes in the virtual domain. We consider the results as a proof of concept, and future work will require larger datasets to validate the stability of the performance and support cross-domain generalization.

Our model is designed for instruction-based decision-making, but it is limited in its ability to support two-way interaction, such as answering with text or forming a dialogue. This limitation comes from our choice to use a vision–language multimodal encoder-only model as the baseline. The main reason for this design choice is the need to encode both states and actions as single tokens, which simplifies the process of building a sequential state–action history. However, most vision–language models capable of generating text responses, like encoder–decoder or decoder-only models, use multiple token embeddings instead of a single token. These multiple embeddings are usually comparable by the number to the input sequence of the encoder, making it difficult to fit them into our simple token-based approach.

Another limitation of our approach is the manual description of action options in terms of their number, position on the map, and text descriptions. This process requires human input during the preparation stage. In future work, this limitation could be addressed by enabling the system to autonomously identify points of interest during the exploration phase, as the robot gathers the LIDAR occupancy map. This limitation is specific to the applied task of mobile robot navigation and does not necessarily affect more general decision-making policies, where action options and their dynamic changes are handled differently.

8. Conclusions

Our work introduces a scalable framework for decision-making tasks with dynamically varying action spaces based on accepting a widely trained vision–language multimodal encoder for moment-wise rich single-embedding representation forming, with the addition of a history-aware causal transformer with a proposed attention score classifier block. Efficiency is demonstrated in the context of instruction-based navigation for autonomous mobile robots. To address the challenge of dynamic class numbers, we designed the attention score mechanism, a trainable classification layer capable of efficiently selecting actions based on adaptive relationships between predictions and dynamically annotated options.

The experiments in both simulated and real-world environments validate the effectiveness of our approach. Comparative evaluations using ViLT and FLAVA as baseline multimodal transformers highlight the superior performance of FLAVA within the constraints of a single 8 GB VRAM system, showcasing the method's practical feasibility. Additionally, our ablation study reveals the critical role of trainable attention scores in achieving robust performance over static similarity metrics like cosine similarity. We believe that the approach of describable and dynamic action spaces is not limited by our application and is crucial for developing multitasking or switching-domain systems.

Author Contributions: R.M. contributed to the methodology, software development, and the writing of the original draft. H.N. provided supervision and contributed to the review and editing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by JST SPRING (grant JPMJSP2135).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data, including source code and datasets, presented in the study are openly available. The source code is available in the following GitHub repository: github.com/Renass/phd_xiaor_project. The training pipeline used in the experiments is available at https://github.com/Renass/phd_xiaor_project/blob/v1.0/camera-lidar_experiment/models/renas10_train_all_experimental.py accessed on 13 December 2024; Datasets are accessible at https://doi.org/10.5281/zenodo.14433668 accessed on 13 December 2024—Virtual Environment Gazebo 100 episodes; https://doi.org/10.5281/zenodo.14434222 accessed on 13 December 2024—Real office environment 77 episodes.

Acknowledgments: The authors presented preliminary results of this work at APIEMS 2024. This paper substantially extends and refines the concepts introduced in that conference presentation. Notable advancements include new baselines, experimental analysis, enhanced theoretical contributions, schematics, and computational efficiency analysis.

Conflicts of Interest: The authors declare no conflicts of interest. The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
2. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186.
3. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. PaLM: Scaling language modeling with pathways. *J. Mach. Learn. Res.* **2023**, *24*, 1–113.
4. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. In Proceedings of the 9th International Conference on Learning Representations (ICLR), Vienna, Austria, 3–7 May 2021.
5. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollar, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 16000–16009.
6. Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Adv. Neural Inf. Process. Syst. (NeurIPS)* **2021**, *34*, 15084–15097.
7. Tsai, Y.-H.H.; Bai, S.; Liang, P.P.; Kolter, J.Z.; Morency, L.-P.; Salakhutdinov, R. Multimodal Transformer for Unaligned Multimodal Language Sequences. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 6558–6569.
8. Hong, Y.; Wu, Q.; Qi, Y.; Rodriguez-Opazo, C.; Gould, S. VLN BERT: A recurrent vision-and-language BERT for navigation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 1643–1653.
9. Chen, H.; Suhr, A.; Misra, D.; Snavely, N.; Artzi, Y. 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), Long Beach, CA, USA, 15–20 June 2019; pp. 12538–12547.
10. Chen, S.; Guhur, P.-L.; Schmid, C.; Laptev, I. History Aware Multimodal Transformer for Vision-and-Language Navigation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 19485–19497.
11. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
12. Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sünderhauf, N.; Reid, I.; Gould, S.; Van Den Hengel, A. 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), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3674–3683.
13. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst. (NeurIPS)* **2020**, *33*, 1877–1901.
14. Dai, W.; Li, J.; Li, D.; Tiong, A.M.H.; Zhao, J.; Wang, W.; Li, B.; Fung, P.; Hoi, S. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. *arXiv* **2023**, arXiv:2305.06500.
15. Ahn, M.; Brohan, A.; Brown, N.; Chebotar, Y.; Cortes, O.; David, B.; Finn, C.; Fu, C.; Gopalakrishnan, K.; Hausman, K.; et al. Do as I can, not as I say: Grounding language in robotic affordances. In Proceedings of the 6th Conference on Robot Learning (CoRL), Auckland, New Zealand, 14–18 December 2022.
16. Guhur, P.-L.; Chen, S.; Garcia, R.; Tapaswi, M.; Laptev, I.; Schmid, C. Instruction-driven history-aware policies for robotic manipulations. In Proceedings of the Conference on Robot Learning (CoRL), Auckland, New Zealand, 14–18 December 2022.
17. Liu, H.; Lee, L.; Lee, K.; Abbeel, P. Instruction-following agents with multimodal transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–20 June 2022.
18. James, S.; Ma, Z.; Rovick Arrojo, D.; Davison, A.J. RLBench: The Robot Learning Benchmark & Learning Environment. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3019–3026.
19. James, S.; Davison, A.J. Q-Attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4480–4487.
20. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5.
21. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [CrossRef]
22. Ziegler, J.G.; Nichols, N.B. Optimum settings for automatic controllers. *Trans. Am. Soc. Mech. Eng.* **1942**, *64*, 759–768. [CrossRef]
23. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
24. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]

25. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [CrossRef]
26. Zhao, J.; Liu, S.; Li, J. Research and implementation of autonomous navigation for mobile robots based on SLAM algorithm under ROS. *Sensors* **2022**, *22*, 4172. [CrossRef] [PubMed]
27. Yang, G.; Yao, J. Multilayer neurocontrol of high-order uncertain nonlinear systems with active disturbance rejection. *Int. J. Robust Nonlinear Control* **2024**, *34*, 2972–2987. [CrossRef]
28. Yang, G. State filtered disturbance rejection control. *Nonlinear Dyn.* **2025**, *113*, 6739–6755. [CrossRef]
29. Florence, P.; Lynch, C.; Zeng, A.; Ramirez, O.A.; Wahid, A.; Downs, L.; Wong, A.; Lee, J.; Mordatch, I.; Tompson, J. Implicit Behavioral Cloning. In Proceedings of the 5th Conference on Robot Learning (CoRL), PMLR, London, UK, 8–11 November 2021; Volume 164, pp. 158–168.
30. Kim, W.; Son, B.; Kim, I. ViLT: Vision-and-language transformer without convolution or region supervision. In Proceedings of the 38th International Conference on Machine Learning (ICML), PMLR, Virtual, 18–24 July 2021; Volume 139, pp. 5583–5594.
31. Singh, A.; Hu, R.; Goswami, V.; Couairon, G.; Galuba, W.; Rohrbach, M.; Kiela, D. FLAVA: A foundational language and vision alignment model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 15638–15650.
32. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. *Improving Language Understanding by Generative Pre-Training*; OpenAI: San Francisco, CA, USA, 2018.
33. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv* **2016**, arXiv:1609.08144.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.