Multi-Modal Deep Reinforcement Learning for Indoor Robot Navigation

Final Paper

Sana Moin & Nur Atiqah Zakiah Abdul Khaliq

29.06.2021

Abstract

Cognitive Science research has demonstrated that multi-modal integration has enabled humans to understand their environment better and reduce ambiguity. Unfortunately, this problem still stands in the field of robotics. A standard framework for multi-modal integration has not yet been established. In this paper, we investigate various approaches for multi-modal learning for indoor robot navigation task that utilizes deep reinforcement learning, where we examine how the perception of different modalities can enhance the artificial agent's understanding of the environment, how to train control policies end-to-end, how various challenges around indoor navigation task are tackled and also analyze how well they perform. We will further look at what are their advantages and what challenges do they still face. Finally, we conclude by suggesting future work opportunities based on our analysis.

1 Introduction

Intelligent Robots have become more prominent in this era, and it would be useful to ensure these robots can work autonomously [1, 2]. With that, the first and most fundamental task for these robots is to understand their working environment [3]. Traditional models for speech and vision in robotics are susceptible to misinterpretation because environments are dynamic [4]. Humans can thrive in dynamic environments, by relying on more than one sensory organ to completely understand the context [5]. This brings us to the concept of multi-modal neural networks that can sense the environment using more than one mode and that helps the overall model to understand real-life situations in a better way. Multi-modal integration has enabled humans to understand their environment better and reduce ambiguity, Cognitive Science research has demonstrated [6]. Beyond that, humans require continuous learning as well. For these models to work in real life where they can adapt, they must learn as humans do. Reinforcement learning can

help these models their current state, and also learn from them and adapt. This represents the intelligence required and the type of intelligence that allows robots to work dynamically and independently in practical environments.

1.1 Research Goals

The literature review aims to provide an overview and comparison of the state-of-the-art multi-modal deep reinforcement learning (MMDRL) approaches in recent years. For this literature review, the scope is narrowed down to the task of indoor robot navigation. This literature review could serve as a summary of the techniques and challenges faced in implementing MMDRL in robot control, while also presenting recommendations.

Combining different modalities allows the robot to understand its environment better, this is called modal grounding, and this is similar to how humans process their reality [7]. We aim to look at the combination of various modalities as there are a variety of cues in the home environment, especially when these robots are expected to execute actions based on natural language instructions. We also aim to analyze how instructions, a textual modality, can be grounded with the visual modality.

Much of existing research on robot control has used indoor navigation as a test for how well their implementations [8, 9, 10]. To keep the approaches comparable, this paper narrows down the task in which the robot is trained to perform. This would also allow us to identify advantages from each approach, overarching challenges that occur across the approaches, as well as draw meaningful conclusions about MMDRL.

2 Background

2.1 Multi-Modality

In general, multi-modality refers to the integrated use of information from different senses [11]. Modalities include various sources of information such as smells, visuals like images or videos, or sounds like human speech, environmental sounds, song tunes, etc. Together they create a comprehensive understanding of the surrounding environment [5]. This concept of understanding the environment through different modalities can be leveraged by modern machines as well. When images are augmented with auditory information, for example, it could enhance the learning of machine learning models. This is because these modalities have different statistical properties. For instance, images are represented with their pixel information or through feature extraction whereas audio signals are often represented through spectrograms or fundamental frequencies, for example. To understand the relationship and joint representation of these modalities, multi-modal deep learning

models can be used. These models were introduced by [12] where they presented a series of tasks to show cross-modality feature learning, where better features for one modality can be learned if multiple modalities are present at feature learning time.

2.1.1 Visual

Visual cues for robots can be either videos or images. Videos can be represented as a stack of frames/images. An image is typically represented as a matrix of pixels [13], each representing a color specified as an RGB triplet. Various features of images can be extracted using computer vision techniques and these features can then be learned by different deep neural network models to perform the intended tasks, such as classification.

2.1.2 Audio

Audio in the environment can be speech, music, animal sounds, car honking, etc. Audio signals are electronic representations of sound waves. Sound is longitudinal waves that travel through a medium such as air, and are made up of compressions and rarefactions. For processing such sounds, digital signal processing techniques can be applied, for operations like filtering or enhancing sounds. Processing techniques also allow a researcher to capture sound features such as fundamental frequencies and power to name a few [14]. Through Natural Language Processing (NLP), Automatic speech recognition systems such as Siri or Alexa have also been developed. Furthermore, signal processing is applied not just to comprehend what is said but also to understand it in intelligent ways, such as being context-aware and also mapping semantic meaning. In this paper, we will see how it is possible to map sound to objects to facilitate robot navigation.

2.1.3 Textual

Audio instructions are usually converted into their textual forms, or at times the textual inputs are given directly. NLP can be used to understand the instructions based on the task in hand, which could be for example asking a robot to do a certain task or follow a certain path. Word embeddings are used to represent the words in the text.

2.2 Reinforcement Learning

Reinforcement learning (RL) is a type of machine learning technique that allows an intelligent agent to learn in an interactive environment by trial and error using feedback from its actions and experiences [15]. This feedback is the cumulative reward that the agent receives, and RL tries to maximize this reward. This task requires both exploration and exploitation, so the RL algorithm needs to find the right balance between them. For this, an optimal policy has to be built that can

efficiently map an agent's state to actions [16].
There are three main approaches to implement RL algorithms:

1. Value-based: a value function is maximized and a long-term return of current states is expected from the agent.

2. Policy-based: action performed in every state is used to gain maximum reward in the future. It can be deterministic or stochastic.

3. Model-based: a virtual model is created in a specific environment for learning.

Mathematically, the RL problem is represented by the Markov Decision Process whose transition functions are unknown to the agent and learned through Q-learning [16]. Q-learning uses a value-based method of supplying information to inform which action the agent is supposed to take, i.e., for a robot to learn an action policy.

Figure 1 shows the basic idea of RL. An agent performs an action in the environment at time step t and receives the observation and reward for that given action. Using these, the policy learns how to maximize the reward received and the process continues.

2.2.1 Reward functions

A reward is a scalar feedback signal from the environment when the agent performs an action on it [16]. It indicates how well an agent is doing and how the agent ought to behave. A reward function ranks the behaviors and the learning agent has to find the behavior with the highest rank. A positive reward encourages certain actions, while negative discourages. During training, an agent updates its policy based on the rewards received for different state-action combinations. Reward functions can be continuous, discrete, or a mix of the two [18].

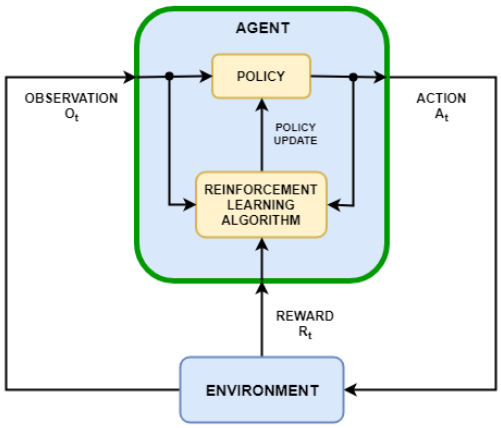

Figure 1: RL basic overview. Adapted from [17]

2.2.2 Policy

A policy comprises the mappings of the suggested actions that the agent should take for all the possible states based on the observation from the environment [16]. Reinforcement learning agents estimate policies and value functions using function approximators called actor and critic representations respectively. The actor represents the policy that selects the best action to take, based on the current

observation. The critic represents the value function that estimates the expected cumulative long-term reward for the current policy [19].

2.3 Indoor Robots

Many robots are designed for use in indoor settings such as households, warehouses, stores, etc. Common examples are vacuum cleaning robots, Amazon warehouse robots, and elderly assistance robots. They can be easily connected using the indoor network systems through WiFi. However, household environments tend to have limited space and often dynamic elements such as people walking. As such, it is essential for these robots to be good at their specific tasks but also have a general sense of their environment. They should have strong obstacle detection mechanisms in place and be gentle around sensitive elements. Moreover, the mechanism for receiving the instructions should be robust. For example, if the instructions are through speech, then the sounds in indoor environments can obstruct the clarity of processing the sound and hence the interpretation of instructions given. The robot navigation task itself is also a crucial component because the robot may take a very long time to perform an operation if the chosen path is not optimal.

The systems developed for indoor robots should handle several aspects of the issues that these robots face. In this literature survey, we will discuss how some approaches that aim to design robust robots that can do well in indoor environments with the use of MMDRL techniques to achieve cross-modal grounding.

2.4 Robot Navigation Task

Robotic control is the system that contributes to the movement of robots. This involves the mechanical aspects and program systems that makes it possible to control robots. In this paper, we would like to see how robots' tasks can be accomplished when they understand their environment better by using sight, sound, and textual instructions.

Robot Navigation Task is defined in this literature survey as a task in which robots are required to move within their environment from the start position to a target position, where the goal is. Combining visual and auditory signals for the Robot Navigation Task can be challenging, especially because auditory signals beyond acoustic ones from the sonar signals are not typically used for navigation. In this paper, the auditory signals that will be discussed include spoken language and environmental sounds such as dogs barking or telephone ringing, for example.

The Robot Navigation task can typically be broken down into subtasks - Path Planning, Collision Prevention, Search Algorithm, and Map Building [20]. Additionally, the environment greatly affects the choice of technique used in each subtask. To counter this problem, in this literature survey only end-to-end archi-

tectures will be explored. Hence, the action a robot takes will be dependent on the reinforcement learning component that outputs an action policy based on the information it receives from multiple modalities. Many researchers have used deep reinforcement learning on sensor information to solve navigation tasks [21, 22, 23, 24, 25] with great success.

2.4.1 Traditional methods for RL for Robot Navigation

Before delving into MMDRL for robot navigation, it is necessary to be familiar with the traditional methods used to understand how MMDRL can contribute further. Mobile robots are required to perceive objects in their surroundings for target recognition and obstacle avoidance. Traditional methods typically use one modality which is primarily vision and mobile robots are typically equipped with a laser range finder and cameras [26] which could be applied in a landmark-based framework [27]. This is occasionally simplified further through the use of artificial landmarks such as RFID tags. Landmark-based frameworks rely on these landmarks referenced to construct an internal map for localization. With one modality, reliance on these landmarks is necessary because small positional and directional errors tend to accumulate.

Several approaches incorporate deep learning through the use of Convolutional Neural Networks to recognize these landmarks and calculate the distance [28]. Since this task would require a robot to navigate in a 3D space, 2D laser rangefinders are typically used for depth sensing and internal map reconstruction. In one of such approaches, researchers used a Rapidly Exploring Random Tree to build a safe traversable map of the indoor environment [29]. However, 2D laser rangefinders may not capture enough information because their frame is vertically limited, and as such, they cannot detect objects above or below its line of scan.

Generally, integrating information from multiple modalities could improve the robustness of an indoor navigation system. Especially for humanoid robots, a multi-modal system would emulate typical human behavior better as compared to relying only on the vision system as humans integrate multiple modalities to facilitate perception [6].

3 Related Work

Naturally, there has been an increasing interest in approaches that fuse information from two modalities to learn robot action policies, which have been met with success [30, 31]. Several types of approaches were explored, and for the purposes of this literature survey, we focus on two approaches that fused sight and sound signals and one approach that fused sight and textual information using a deep reinforcement learning network.

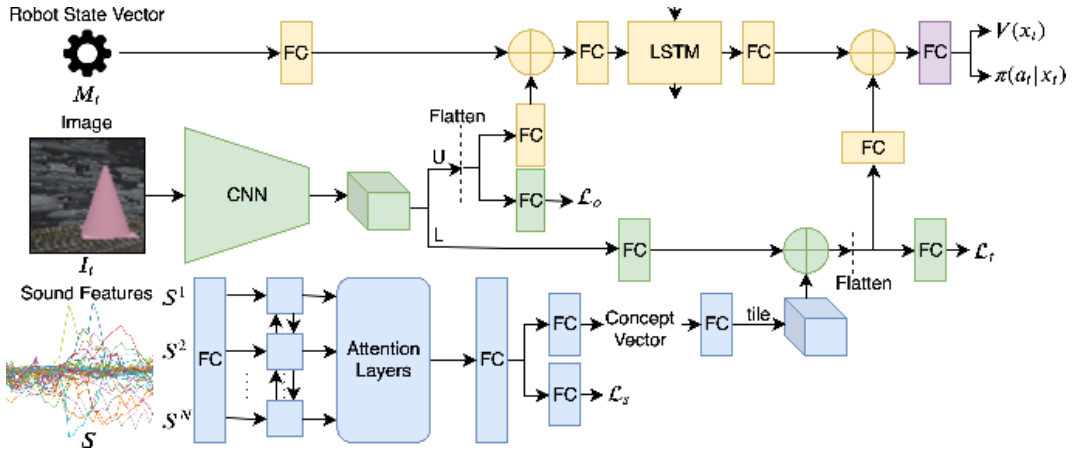

Figure 2: Network Architecture for Sound Interpretation Robot. Adapted from [24]

3.1 Chang's Speech Vision MMDRL (Sound Interpreter)

The first application of deep multi-modal RL on a robot navigation task investigated, is the work by Chang et al. [24] proposed an end-to-end deep neural network to incorporate auditory information into a visual navigation task.

3.1.1 Network Architecture

The architecture of their proposed system is an end-to-end architecture that receives sound features and images from each time step to output an action policy. It can be broken down into several main components, also shown in Figure 2.

The Sound Interpreter. Raw auditory signals are pre-processed to obtain Mel Frequency Cepstral Coefficients (MFCCs) as sound features. These sounds features are sent through a one-layer bi-directional long short-term memory (BiLSTM) network with several attentional layers to learn a concept vector C.

The Integrator. This component fuses visual and auditory information with the robot's current state belief. The current visual observation I_t is processed by a Convolutional Neural Network (CNN) which uses only 3x3 kernels for convolutions and 2x2 kernels for pooling [32] to extract features in a grid pattern [33]. The output is split into two branches, one fuses image features with the robot state vector M_t in a single layer LSTM and the other combines information from extracted image features with the concept vector C. The Integrator utilizes two task-specific loss functions, L_o for object recognition, and L_t for target identification and associating sound command to the target.

The Policy Learner. Reinforcement learning is used to train the policy and value function when given the fused sound-image and image-state vectors from The Integrator. Specifically, Proximal Policy Optimization (PPO) from OpenAI Baselines, which is a model-free policy gradient. The researches had tweaked the

original implementation to incorporate eight different observations/experiences the robot uses from the environment.

The Joint Architecture. The entire architecture is an end-to-end system that can be trained alone. The entire architecture uses a loss function L_{tot} which is a linear combination of all the other loss functions.

$$L_{tot} = w_{pg}L_{pg} + w_sL_s + w_oL_o + w_tL_t$$

where w_{pg}, w_s, w_o and w_t are scalar weights for each loss.

3.1.2 Environment Design

To determine if their model was robust in different environments, researchers designed two environments.

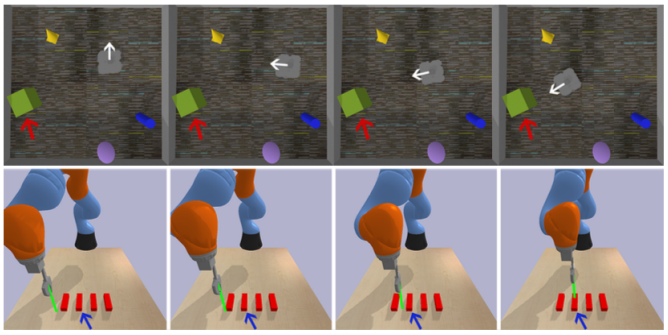

Figure 3: **Above:** Turtlebot agent simulation, **Below:** Kuka agent simulation. Adapted from [24].

The Turtlebot Environment, with a mobile TurtleBot3, is a 3D arena with four 3D objects (see Figure 3). Any action is defined as a change of the transitional velocity v_d, and a change of the angle ϕ_d in relation to the initial orientation. The reward that the robot is trained on is based on its distance and angle to the target, whether it collides with any object or wall. Furthermore, it gets an additional reward r_{goal} is +2 if the robot distance and angle to the target are within an acceptable range or +0.5 if only either the distance or angle is acceptable. Essentially, the agent's reward increase as it approaches the target while maintaining a safe distance.

The Kuka Environment, consists of a table, a Kuka-IIWA robotic arm that moves within an XY-plane above the table, and four identical blocks of size 10 cm x 40 cm (see Figure 3). Every action is defined as a change of the gripper tip location in the x-axis and y-axis at time t. The network is trained based on the distance between its robot gripper and the target. The reward is highest when the gripper is at the center of the target block in the XY-plane.

3.1.3 Experimental Set-Up

The same network architecture is used to train agents in both environments. To train the networks for both environments, w_{pg} and w_s were set to 1. However, w_o and w_t were set to 0.5 in the Turtlebot environment respectively and 0 in the Kuka environment. The losses L_o and L_t were not relevant for the Kuka environment

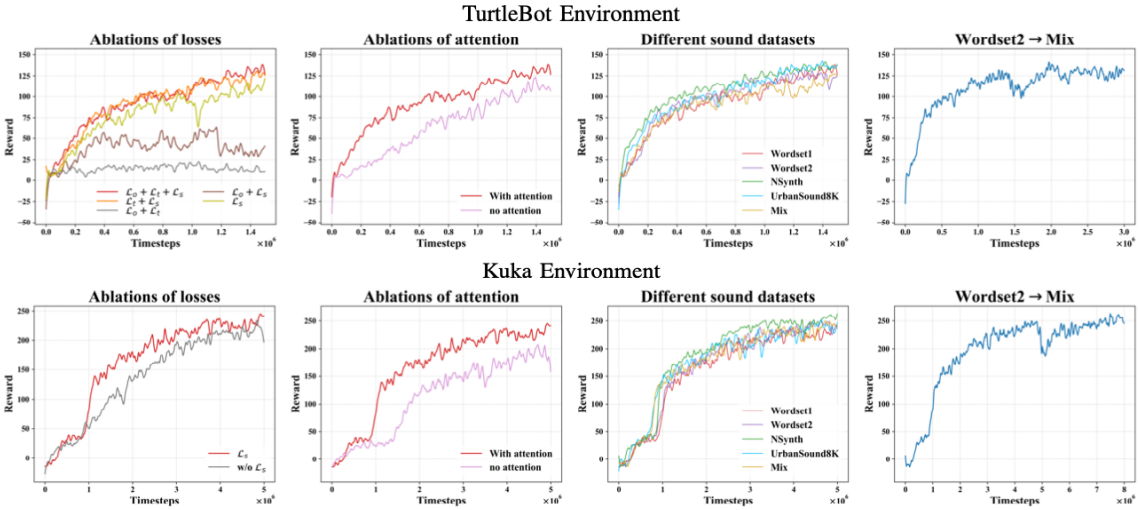

Figure 4: Training curves of our network in TurtleBot environment (top) and Kuka environment (bottom). Left: Loss Function analyzes. Middle Left: Attentional Layer analyzes. Middle Right: analyzes of different sound datasets. Right: Learning process for the experiment "Wordset2 → Mix. Adapted from [24]."

because the blocks are identical and are always in view.

Researchers were interested to find out if their model was robust against various types of sounds. As such, they experimented with single-word speech signals, environmental sounds, and single-tone signals. Two sets of words were selected from the Speech Commands Dataset [34] for (1) single-word speech signals. The UrbanSound8K Dataset [35] was used for (2) environmental sounds. Similarly, single tone samples were taken from the NSynth Dataset [36].

Also, the researchers created a Mix Dataset to determine if their network would be able to map different sounds to a single object, i.e., the utterance 'Dog' and the sound of a dog bark to one object. This dataset combines word utterances with environmental sounds. With that, researchers conducted an experiment they called 'WordSet2 → Mix' to show that sound interpretation can evolve dynamically in the network.

3.1.4 Network Performance and Evaluation

Sound Interpretation Researchers observed that their network was able to interpret types of audio signals, including non-speech sounds. This can be observed in the graph of 'Different Sound Datasets in Figure 4, the network performs similarly across all sounds types. Also, researchers observe that the training curve and rewards of the model on the various sound datasets, including Mix dataset, are similar. From there, researchers conclude that the model can associate the utterance "dog" and the sounds of a dog barking in the same concept. Moreover, based on their 'Wordset2 → Mix' experiment, they could conclude that interpretations

of the model of sound signals evolved dynamically. From Figure 4 in the graph of 'Wordset2 → Mix', we see that the reward dips slightly, corresponding to the addition of two new sound signals, but it quickly increases and achieves a similar reward and success rate as on the other sound datasets. Furthermore, using additional time steps to learn new sounds was significantly more efficient than training the network to learn the new sounds from scratch.

Loss Functions analyzes, researchers removed one loss function at a time to identify its importance in the model. They found that L_s influences the feature extraction of the sound interpreter component. Researchers attributed this to guidance for sound interpretation, which facilitates policy learning. On the other hand, we did not see a big impact of L_o or $L_t + L_o$ during training. However, omitting these reduced the success rate at training time. The researchers attributed this to the guidance L_o provides in new situations. Additionally, without $L_t + L_o$, the agent was not able to choose the right target and often went directly for the first object it saw. As such, the researchers deemed it necessary to have all three auxiliary losses for any agent in the Turtlebot environment.

Real World Experiments, researchers stated that agents in the Turtlebot environment were easily transferred into a real TurtleBot3 robot without additional training. Researchers found that sometimes when the agent is unable to recognize an object as the target, it would continue searching and fail to approach any object. However, the success rate was above 90% across 10 sound commands.

3.2 Chen's Speech Vision MMDRL (SoundSpaces)

In the second application, audio-visual robot navigation for complex, acoustically, and visually realistic 3D environments [37] is discussed. The major focus of Chen et al.'s [23] research was on how audio cues can be detected and followed to efficiently navigate in an indoor environment. Indoor environments with sound-emitting targets are tricky to navigate when solely processing visual information. Audio signals partially reveal the geometry of the space, detect obstacles or even detect the materials of major surfaces, thereby complementing the visual streams. Audio cues also provide additional information and are very helpful when the visual cues are unavailable due to lighting. Chen et al. propose a network to mitigate this issue by taking advantage of audio signals. The researchers aimed to address the embodied navigation using audio-visual cues and is the first work to show this in a realistic 3D environment with an end-to-end approach.

3.2.1 Network Architecture

Input Pre-Processing. For processing the audio and visual content together, a multi-modal network architecture is designed as shown in Figure 5. Audio signals are first pre-processed into spectrograms through the application of Short-Time Fourier Transform. Visual signals are given as RGB(D) images if the depth is

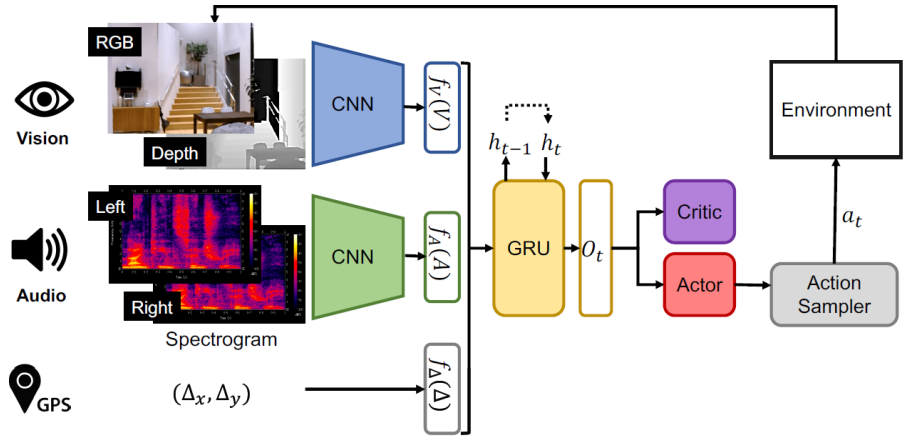

Figure 5: Audio Visual Navigation Network for SoundSpaces. Adapted from [37].

available. Additionally, a GPS provides a relative displacement vector of the agent to the goal in 2D space. These 3 pieces of information are given as input into the network which outputs a probability distribution over which actions to take next, i.e., an action policy.

Audio and Visual Processing. Both audio and visual inputs, A and V, are processed using CNNs f_A and f_V. Although both CNNs have separate weights, they have the same architecture of 3 convolutional layers ((8 x 8), (4 x 4), (3 x 3)) and a linear layer with ReLU activations on each layer. The outputs, f_A(A) and f_V(V), are concatenated together and with the relative displacement vector Δ [23]. See Figure 5.

Multi-modal fusion The concatenation of the f_A(A), f_V(V) and Δ is given as input to GRU which has input and hidden layers of size 512 and a single recurrent layer. The GRU operates on the current step's input as well as the accumulated history of states h_{t-1} The GRU updates the history to h_t and outputs the representation of the agent's state o_t.

Reinforcement Learning. Finally, through Proximal Policy Optimization, the value of the state $V_\theta(o_t, h_{t-1}$ and the policy distribution $\pi_\theta(a_t|o_t, h_{t-1}$ are estimated using the critic and actor heads of the model, which are both linear layers. The reward function is defined as follows:

- +10 if action Stop is executed at the goal location and -0.01 otherwise, per time step
- +1 for reducing the geodesic distance to goal and -1 otherwise

Essentially, reward increases as the agent approach the goal and is highest when the agent stops at the goal. Geodesic distance as opposed to Euclidean distance refers to the distance between two points, taking into consideration the shape of

the environment. The entropy maximization term is added to the cumulative reward optimization for better space exploration. The rewards are discounted with a decay of 0.99.

3.2.2 Environment Design

Researchers highlight two variants of the navigation task to identify how well an agent can perform with only audio cues. For these tasks, their proposed architecture which is a multi-modal deep reinforcement learning approach is used to combine the audio-visual information and train the navigation policies.

1. PointGoal: The target is identified by a GPS cue. The agent only receives a directional hint in terms of a point vector in a visual space.

2. AudioGoal: The target is identified by the sound it emits, even when it is not in the visual space. The agent only receives an audio hint but is able to explore in an audio-visual space. The audio hint is a function of the location of the agent, the goal, and the material and structure of the room.

3. AudioPointGoal: The target is identified by the sound it emits and a GPS cue. The agent gets both audio and directional hints.

Sound reflections in the room are modeled using room acoustic modeling and a bidirectional path tracing algorithm [38]. Room impulse responses (RIR), the transfer function between a sound source and microphone, are pre-computed to simulate the acoustics of the environment.

The action space has 4 actions: MoveForward, TurnLeft, TurnRight, and Stop. The sensory inputs it receives are RGB images, depth, and binaural sound or/and GPS, depending on the task. An episode consists of a scene, agent start location, agent start rotation, goal location, and source audio waveform. An episode is successful if the action Stop is executed exactly at the target location.

3.2.3 Experimental Set-Up

With regard to the datasets used, pre-computed audio renderings SoundSpaces hosted within Habitat for Matterport3D and Replica are introduced.

1. Matterport3D: 87 Matterport3D environments are used. These are the real-world homes and other indoor environments with 3D meshes and image scans, with 517 m2 floor space.

2. Replica: It is a dataset of 18 apartments, hotels, offices, and room scenes with 3D meshes.

3. Habitat: It is an open-source platform for fast 3D simulation with an API that supports RGB, depth, and semantic rendering.

To evaluate their model, different episodes are generated. Episodes are generated by choosing a scene and a random start and goal location. Simple episodes where the geodesic distance is less than 4 steps or the task can be done by the agent moving in a straight line are ignored. SPL (success rate normalized by inverse path length) which is standard for navigation tasks is used as the metric for evaluating navigation. An episode is marked as successful only if the action Stop was executed at the goal.
Three non-learning baselines are adopted from previous works. (1) 'Random' always chooses a random action, (2) 'MoveForward' always chooses to move forward and will turn right before moving forward if it encounters an obstacle, and (3) 'GoalFollower' which orients itself towards the coal and then moves forward. All three baselines will stop when it reaches the goal.

3.2.4 Network Performance and Evaluation

During the evaluation, the authors try to address three main objectives - (1) if the audio helps with navigation tasks, (2) if the audio can replace GPS when a target is a sound-emitting object, and (3) how the various sound sources affect network performance.

Researchers found that audio indeed helped with navigation, as audio improved the accuracy significantly. In both the Replica and the Matterport3D environments, agents performed better in the AudioPointGoal as compared to the PointGoal. It suggests that the addition of sound allows an agent to capture spatial information such as depth more easily than an agent that only processes vision.

Concerning whether audio can completely replace GPS cues for a sound-emitting target, researchers looked at the differences in agent performance in the AudioGoal task and the PointGoal task. Researchers found that AudioGoal agents did not suffer due to noisy GPS readings, but the PointGoal and AudioPointGoal agents did. This may provide evidence that audio gives good spatial cues and could possibly supplant GPS, the researchers also found that the learned audio features naturally encode spatial information such as the distance and angle to the goal. Furthermore, the influence of audio and visual input depends on the environmental context and goal placement. Researchers found that agents tend to draw dynamically from both modalities to decide which actions to take.

To understand the influence of different sound sources, 102 sound clips were split into training, validation, and test sets with 73, 11, and 18 clips respectively. The generalization becomes more difficult as the scenario changes from a single heard sound, to multiple heard sounds, to multiple unheard sounds. AudioPointGoal outperforms the PointGoal agent and performs similarly on heard and unheard sounds. AudioGoal's accuracy declines with varied heard and unheard sounds. Researchers found this logical as the task of following an unfamiliar sound is more difficult, but they expect that a larger training set with more variety will resolve

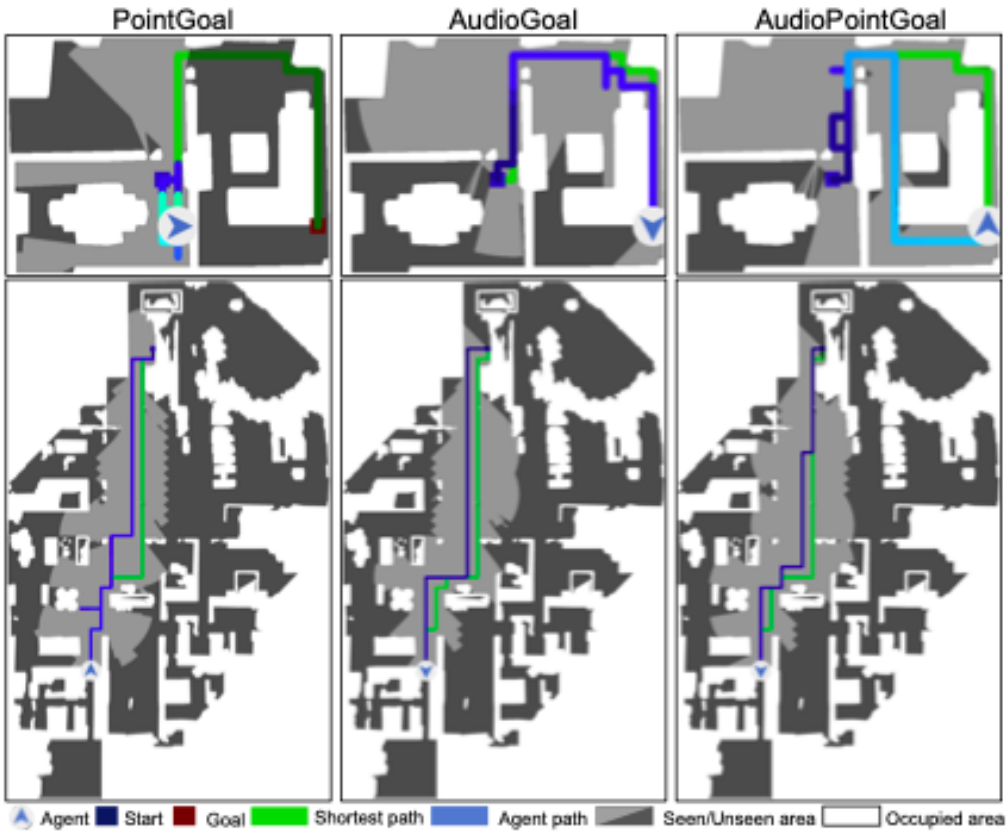

Figure 6: Top Row: Replica Environment - The PointGoal agent performs worst. In contrast, the AudioGoal and AudioPointGoal agents better sense the target: the sound travels through the door and the agent leaves the starting room immediately. Bottom Row: Matterport Environment - the AudioGoal agent performs best. Adapted from [37].

this.

In general, the results show that when the visual information is augmented with audio, it enhances the directional cues as well as provides spatial information about the environment, which are beneficial for better navigation of a robot agent, see Figure 6 .

3.3 Wang's Speech Vision MMDRL

We look at the Vision-Language Navigation task in this approach, which was proposed by [25]. The robot navigation task is challenging when it comes to the robot following the instructions provided. The reasoning involved is not restricted to just one modal of understanding, but a combination of different modalities. This task becomes even more difficult when it comes to the robot being in a new environment. In this paper, they try to address three different issues in such a robot navigation system in a 3D environment: (1) Cross-Model Grounding, (2) Ill-posed feedback, and (3) Generalization. For resolving each issue, different approaches

are proposed and tested. We describe these approaches and further look at how well they resolve these issues.

3.3.1 Network Architecture

The researchers proposed to use a **Reinforced Cross-modal Matching(RCM)** model for this task. This model comprises two modules: 1) a reasoning navigator π_θ, 2) a matching critic V_β. An overview can be seen in Fig. 8. The reinforcement learning model is used by the robot to learn indoor navigation based on given instructions and an observed panoramic view. Additionally, two reward functions are used: (1) an extrinsic reward that is provided by the environment and measures the success signal and the navigation error of each action, and (2) an intrinsic reward that comes from our matching critic and measures the alignment between the language instruction X and the navigator's trajectory τ. The researchers developed the two modules using LSTM, as summarized below:

The Reasoning Navigator π_θ module is responsible for interpreting the text instructions into a sequence of actions by incorporating both text instructions with the panoramic view and a historical context. A language encoder LSTM is used to encode the language instruction into a set of textual features. Similarly, an attention-based trajectory encoder LSTM is used to encode the trajectory history. Additionally, a CNN is used to extract a visual appearance feature vector. The reasoning navigator takes as input the trajectory history, textual and visual context to predict an output. For policy learning, the extrinsic reward function is used.

Matching Critic V_β: The intrinsic reward function from the matching critic is introduced. It computes how well the actions generated from the reasoning navigator matches the textual instructions. The intrinsic reward is taken as the probability of reconstructing the instruction, and the researchers used an attention-based sequence-to-sequence language model to compute this. The intrinsic reward is helpful if the robot had followed a different path to reach the correct destination, hence penalizing the deviation from the path.

Cross-modal grounding is hence done using these models and a good policy is learned by employing extrinsic and intrinsic reward functions, which also helps for the ill-posed feedback issue.

For learning policy, the agent is warm started by using demonstration actions to perform supervised learning with maximum likelihood estimation, which ensures good policy on the seen environments. When it comes to unseen environments and to make the policy more generalizable, extrinsic and intrinsic reward functions are used to refine this policy. The extrinsic reward function, R_{extr}, is calculated through relative navigation distance and also by checking if the agent has reached a point that is within a threshold distance set from the target. The intrinsic reward function, R_{intr}, is computed as described previously, by encouraging the path

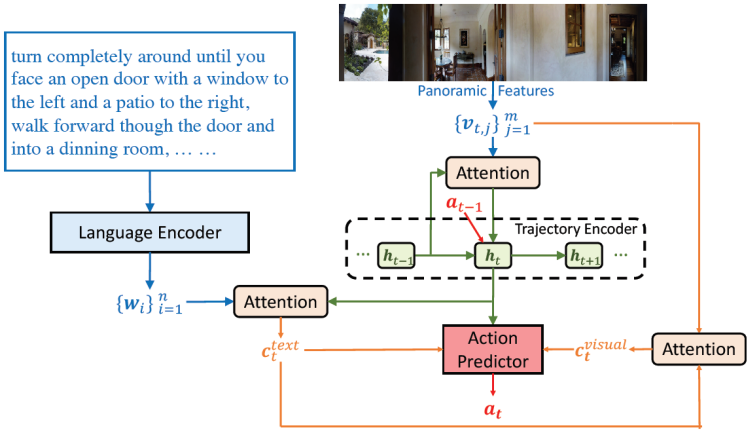

Figure 7: Cross-modal reasoning navigator at step t. Adapted from [25].

given via instructions and penalizing if the agent deviates from the path. The RL loss is computed using both these rewards as:

$$L_{rl} = -\mathbb{E}_{a_t \sim \pi_\theta}[A_t]$$

where $A_t = R_{extr} + \delta R_{intr}$. δ is the hyperparameter that weighs the intrinsic reward, a_t is a sequence of actions that is mapped with respect to the reasoning navigator π_θ, a policy-based agent.

Self Supervised Imitation Learning (SIL):
The robots may perform well when they learn from an already seen environment, but it is challenging for them to perform well in unseen environments. The paper proposes a new approach, SIL, which can explore and adapt by itself in unseen environments. It takes advantage of the trajectories that the matching critic had evaluated. The process is shown in Fig. 8, where given an instruction, the navigator produces a set of possible trajectories. The matching critic determines the best trajectory, $\hat{\tau}$, and that is stored in the Replay Buffer. The agent optimizes itself by self-supervision by reusing these good trajectories from the replay buffer. This in turn improves the generalizability of the system. The supervised learning loss of this step with $\hat{\tau}$ as the ground truth and action $\hat{a}_t$ stored in replay buffer at state s_t is represented as:

$$L_{sil} = -\mathbb{E}[log(\pi_\theta(\hat{a}_t|s_t))]$$

3.3.2 Experimental Set-Up

The agent is trained in seen environments and then tested on previously unseen environments. The agent learns through trial and error in unseen environments.

Dataset: Room-to-Room (R2R) dataset, which consists of panoramic view images of rooms, is used and split into training (14,025 instructions), seen validation (1,020), unseen validation (2,349), and test (4,173) sets.

Evaluation Metrics: 5 evaluation metrics are used: Path Length (PL), Navigation Error (NE), Oracle Success Rate (OSR): the success rate at the closest point to the goal that the agent has visited along the trajectory, Success Rate (SR), Success rate weighted by inverse Path Length (SPL), which is used as the primary metric for evaluation.

Features of images are extracted using ResNet-152 CNN. The pre-trained GloVe word embeddings are used for the initialization of textual features. The maximal length of the action path and instruction set is set to 10 and 80 respectively. The policy is learned by warm starting via supervised learning loss and later with RL training. Further, self-supervised imitation learning is executed to improve the policy. The weight δ of the intrinsic reward is 2.

3.3.3 Network Performance and Evaluation

The RCM outperforms the existing methods and SIL was able to imitate the RCM agent's behaviors and generate efficient policies. SPL of the combination of RCM and SIL were reported as the final results which were the best among all the prior work. Moreover, RCM is improved by SIL on SR and SPL metrics.

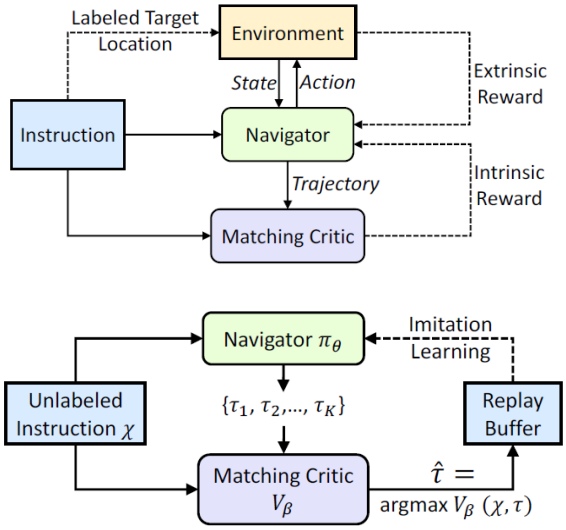

Figure 8: Above: RCM overview, Below: SIL Overview. Adapted from [25].

Furthermore, an ablation study was performed by the researchers by removing intrinsic reward, extrinsic reward, and cross-modal reasoning to understand their effects. The Success Rate for unseen environments dropped when the intrinsic reward was removed, indicating its importance for exploration of such environment. Similar results were observed when extrinsic rewards were not used and only supervised learning was used for validation, which indicates that reinforcement learning brings performance gain. It was also observed that using cross-modal reasoning, the navigator had improved as compared to the baseline on historical context, visually-conditioned textual context, and textually-conditioned visual context for decision making.

The results also showed better results as compared to other approaches on unseen environments, thereby making this system generalizable. However, the researchers also reported that the system suffers from error accumulation issues in some test

case scenarios.

4 Discussion

The generalization of the models is an important characteristic when the agent tries to move around an indoor environment. The indoor environments are dynamic and very different from one place to another. The robots should be designed in a way that they can localize themselves with time and perform well. Reinforcement learning has helped with this aspect, as it enables the robot to learn continuously. We saw how sounds can be used to navigate in an unknown environment in 3.2, by following the sound source and determining the path using visual cues. We also saw how generalizability can be enhanced through self-supervised imitation learning, described in section 3.3.1. Additionally, we see that in 3.1 and 3.2, the proposed models were able to handle different types of sound signals, beyond speech. This generalized sound processing would allow an indoor robot to be responsive not only to humans but also to other sound-emitting objects in the environment, as well as map multiple sounds to the same source.

Most of the approaches had architectures that were easy to understand because they could be broken down into smaller components. Although there are differences in each approach, we observed in most of the approaches that there are always components that handle each modality, and they are connected downstream to a reinforcement learning component which is responsible for deciding which action to take next. The difference between them lies in which type of networks should be used to process information in each modality. One approach used the same CNN for both audio and visual signals, another used an LSTM for audio and a CNN for vision, and the third one used LSTM for textual and CNN for vision. These decisions depend on what the researchers wish the network to learn, some may want to extract features while others may want to create a mapping between input and a concept.

Similarly, various different reinforcement learning algorithms can be used in the reinforcement learning component. Additionally, there are different ways to pass the information downstream. Most architectures concatenate the outputs from the different modal-specific components before passing it on to the reinforcement learning component. However, some may utilize a mechanism such as a Gated Recurrent Unit to accumulate information from several time steps before passing the information downstream. The reinforcement learning component generally takes information from the different modalities and maps it into an action policy.

As such, an end-to-end network, such as the ones described above, provide a means to overcome one of the most obvious challenges in multi-modality - the fusion of data from different modality. This is otherwise a huge challenge because of how the information is typically represented [39]. The reinforcement learning components

essentially act as the integrator to address this problem. As long as the outputs of each modal-specific component can be represented in the same way, i.e., in a tensor, the MMDRL architecture can be trained as a whole.

4.1 Advantages of MMDRL

Upon reviewing the various approaches, the advantages of MMDRL networks for domestic robotics are quite clear. Multi-modality allows the robot to infer spatial information from several types of information and sensors. For example, in 3.2 we see that audio signals boost the robot's spatial awareness even when GPS signals are noisy. Multi-modality allows the robot to use cross-modal grounding, in which the robot can use multiple types of information for reference [40], and correct errors from noisy sensors.

Furthermore, such an architecture could be robust to different types of sounds, whereas a rule-based system or even an automatic speech recognition system would fail to capture non-speech mappings. It would no longer be necessary to draw on domain knowledge or to purchase the most advanced speech recognition system. The MMDRL network would be able to extract relevant features from any type of sound signals without any architectural changes to the network, which is useful in a dynamic environment such as a household.

Beyond that, MMDRL can be trained on simulations and then be transferred into a real-world application when the parameters have been tuned. This way, less testing needs to be done on the robot, preventing wear and tear in which the robot's hardware gets damaged from repetitive use. Additionally, as was mentioned before, such an architecture can be trained end-to-end and the network can identify which features are relevant for its task from raw signals without additional rules or handwritten features.

4.2 Challenges of MMDRL

Multi-modal learning is in itself challenging due to the complex nature of integrating different modalities, i.e., the reinforcement learning module which maps inputs from multiple modalities into an action policy. There are several approaches to perform this integration, but finding the right approach for the right kind of task becomes a challenge.

Moreover, the generalizability of the models needs higher focus when it comes to the task of indoor navigation of robots because indoor environments are highly dynamic and training robots in every new environment is not feasible. Although reinforcement learning tries to address this challenge, the error accumulation in the model described in Section 3.3 is still present and that leads to bad decisions by the robot agent in complex tasks [25].

Furthermore, the robot designs are very particular to their tasks, whereas the need for the future is a multitasking robot, which is only navigating in an indoor space in this case. However, doing that is expensive as the robot would have to learn multiple action policies which may require attention to different features from each modality and multi-modal learning is already computationally expensive because there are many computations that are to be carried out in each time step.

5 Future Work

With regard to recommendations for future work in MMDRL, researchers should stick to the overall architecture of MMDRL - with components for each modality that is fused together in the reinforcement learning component. This way, more thought can be placed into selecting the best sub-architectures for each component, how these components should be connected, loss functions, and reward functions.

It would be best if loss functions are used in every component in the network, i.e., any component responsible for a modality or any subtask should have its own loss function. This would facilitate the learning in each component.

Furthermore, the reward functions should be prioritized and assigned based on their relevance to the agent's progress, as was seen in the approach defined in Section 3.3 which improved the model's generalizability. Agents who are aware when making sub-optimal decisions and can compensate for it will eliminate error accumulation in complex situations that may occur in real-life settings.

However, the researcher would also have to take into consideration the aim of the robot. This is because there is a trade-off between the amount of computation and generalizability. A general MMDRL agent that can learn various tasks would be beneficial in a household, however, more action policies would be necessary for the robot to learn, and it is more difficult to identify relevant features of raw signal across all tasks. On the other hand, if a researcher has only one single task in mind when designing the agent, it would be possible to reduce the computations by feeding only task-relevant features into the MMDRL network. In this regard, MMDRL networks are great because they can be used to meet different aims without significant changes to the architecture.

References

[1] Agostino G Bruzzone et al. "Introducing intelligence and autonomy into industrial robots to address operations into dangerous area". In: *International conference on modelling and simulation for autonomous systesm*. Springer. 2018, pp. 433–444.

[2] Matt Knudson and Kagan Tumer. "Adaptive navigation for autonomous robots". In: *Robotics and Autonomous Systems* 59.6 (2011), pp. 410–420.

[3] Philippe Moutarlier and Raja Chatila. "An experimental system for incremental environment modelling by an autonomous mobile robot". In: *Experimental Robotics I*. Springer. 1990, pp. 327–346.

[4] Moshe Kam, Xiaoxun Zhu, and Paul Kalata. "Sensor fusion for mobile robot navigation". In: *Proceedings of the IEEE* 85.1 (1997), pp. 108–119.

[5] Alain Berthoz and Isabelle Viaud-Delmon. "Multisensory integration in spatial orientation". In: *Current Opinion in Neurobiology* 9.6 (1999), pp. 708–712. ISSN: 0959-4388. DOI: https://doi.org/10.1016/S0959-4388(99)00041-0. URL: https://www.sciencedirect.com/science/article/pii/S0959438899000410.

[6] Casey O'Callaghan. "Perception and Multimodality". In: 2012.

[7] Arda Senocak et al. "On Learning Association of Sound Source and Visual Scenes". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2018.

[8] Yuke Zhu et al. "Target-driven visual navigation in indoor scenes using deep reinforcement learning". In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 3357–3364.

[9] Dimitris Milioris. "Efficient indoor localization via reinforcement learning". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 8350–8354.

[10] Hee Rak Beom and Hyung Suck Cho. "A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning". In: *IEEE transactions on Systems, Man, and Cybernetics* 25.3 (1995), pp. 464–477.

[11] Theo Van Leeuwen. "Multimodality". In: *The Routledge handbook of applied linguistics*. Routledge, 2011, pp. 688–702.

[12] Jiquan Ngiam et al. "Multimodal deep learning". In: *ICML*. 2011.

[13] Raghuveer M Rao and Manoj K Arora. "Overview of image processing". In: *Advanced image processing techniques for remotely sensed hyperspectral data*. Springer, 2004, pp. 51–85.

[14] Andreas Spanias, Ted Painter, and Venkatraman Atti. *Audio signal processing and coding*. John Wiley & Sons, 2006.

[15] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[16] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. "Reinforcement Learning: A Survey". In: *CoRR* cs.AI/9605103 (1996). URL: https://arxiv.org/abs/cs/9605103.

[17] *What Is Reinforcement Learning?* https://nl.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html. Accessed: 2021-07-02.

[18] Haoran Wang, Thaleia Zariphopoulou, and Xun Yu Zhou. "Reinforcement Learning in Continuous Time and Space: A Stochastic Control Approach". In: *Journal of Machine Learning Research* 21.198 (2020), pp. 1–34. URL: http://jmlr.org/papers/v21/19-144.html.

[19] Vijay Konda and John Tsitsiklis. "Actor-Critic Algorithms". In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press, 2000. URL: https://proceedings.neurips.cc/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[20] Swati Aggarwal, Kushagra Sharma, and Manisha Priyadarshini. "Robot navigation: Review of techniques and research challenges". In: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. 2016, pp. 3660–3665.

[21] Aleksandra Faust et al. *PRM-RL: Long-range Robotic Navigation Tasks by Combining Reinforcement Learning and Sampling-based Planning*. 2018. arXiv: 1710.03937 [cs.AI].

[22] Pararth Shah et al. "FollowNet: Robot Navigation by Following Natural Language Directions with Deep Reinforcement Learning". In: *CoRR* abs/1805.06150 (2018). arXiv: 1805.06150. URL: http://arxiv.org/abs/1805.06150.

[23] Changan Chen et al. "Soundspaces: Audio-visual navigation in 3d environments". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2020.

[24] Peixin Chang et al. "Robot Sound Interpretation: Combining Sight and Sound in Learning-Based Control". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 5580–5587. DOI: 10.1109/IROS45743.2020.9341196.

[25] Xin Wang et al. "Vision-Language Navigation Policy Learning and Adaptation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1. DOI: 10.1109/TPAMI.2020.2972281.

[26] Juan Li, Xiang He, and Jia Li. "2D LiDAR and camera fusion in 3D modeling of indoor environment". In: *2015 National Aerospace and Electronics Conference (NAECON)*. IEEE. 2015, pp. 379–383.

[27] Jean-Bernard Hayet, Frédéric Lerasle, and Michel Devy. "A visual landmark framework for mobile robot navigation". In: *Image and Vision Computing* 25.8 (2007), pp. 1341–1351.

[28] Abhijith R. Puthussery et al. "A deep vision landmark framework for robot navigation". In: *2017 12th System of Systems Engineering Conference (SoSE)*. 2017, pp. 1–6. DOI: 10.1109/SYSOSE.2017.7994976.

[29] Chaoqun Wang et al. "Autonomous mobile robot navigation in uneven and unstructured indoor environments". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 109–116. DOI: 10.1109/IROS.2017.8202145.

[30] Michelle A Lee et al. "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8943–8950.

[31] Guan-Horng Liu et al. "Learning end-to-end multimodal sensor policies for autonomous navigation". In: *Conference on Robot Learning*. PMLR. 2017, pp. 249–261.

[32] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[33] Rikiya Yamashita et al. "Convolutional neural networks: an overview and application in radiology". In: *Insights into imaging* 9.4 (2018), pp. 611–629.

[34] Pete Warden. "Speech commands: A dataset for limited-vocabulary speech recognition". In: *arXiv preprint arXiv:1804.03209* (2018).

[35] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. "A dataset and taxonomy for urban sound research". In: *Proceedings of the 22nd ACM international conference on Multimedia*. 2014, pp. 1041–1044.

[36] Jesse Engel et al. "Neural audio synthesis of musical notes with wavenet autoencoders". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1068–1077.

[37] Changan Chen et al. "SoundSpaces: Audio-Visual Navigation in 3D Environments". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Cham: Springer International Publishing, 2020, pp. 17–36. ISBN: 978-3-030-58539-6.

[38] Chunxiao Cao et al. "Interactive sound propagation with bidirectional path tracing". In: *ACM Transactions on Graphics (TOG)* 35.6 (2016), pp. 1–11.

[39] Dana Lahat, Tülay Adal, and Christian Jutten. "Challenges in multimodal data fusion". In: *2014 22nd European Signal Processing Conference (EUSIPCO)*. IEEE. 2014, pp. 101–105.

[40] Douwe Kiela and Stephen Clark. "Multi-and cross-modal semantics beyond vision: Grounding in auditory perception". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 2461–2470.