

修士論文

移動ロボットのための深層学習を用いた 歩行者の位置予測とナビゲーションへの応用

Pedestrian Position Prediction Using Deep Learning for Mobile Robots
and Its Application to Navigation

2024年12月30日 提出

指導教員 林原 靖男 教授

千葉工業大学 先進工学部 未来ロボティクス学科
23S1030 藤原柾

概要

移動ロボットのための深層学習を用いた 歩行者の位置予測とナビゲーションへの応用

歩行者と移動ロボットが混在する環境で、ロボットが人間の移動をある程度正確に予測し、安全な経路を動的に計画することは、安全性と効率性の両面から重要である。しかし、従来の研究の多くは主に歩行者同士の相互作用を考慮しており、移動ロボットの動きや、それが歩行者の行動に与える影響を包括的に取り入れた手法は限られていた。本研究では、この課題を解決するために、歩行者同士だけでなくロボットとの相互作用を同時にモデル化する軌道予測モデルを提案する。

本モデルを各種データセットで学習し、複数の評価指標でベースラインモデルと比較した結果、約9%の精度向上を達成した。また、実環境の実験では、ロボットの行動を考慮した予測が、考慮しない場合に比べて歩行者の軌道をより高精度に捉えることを確認している。さらに、予測結果をナビゲーションシステムに応用することで、動的な環境下でも安全性を高められることを実証した。一方、効率的な経路計画の面では今後さらなる検討が必要である。

本研究は、歩行者とロボットの双方向的な相互作用を反映した軌道予測モデルを開発・検証し、その有効性を示した。具体的には、グラフアテンションネットワークを用いて、歩行者とロボットの相互作用を同時にモデル化し、軌道予測の精度を向上させることを目指した。さらに、提案モデルを用いて、動的な環境下でのロボットのナビゲーションシステムの安全性を高めることを確認した。

キーワード：深層学習、自律移動ロボット、ナビゲーション

abstract

title

keywords:

目次

第 1 章	序論	1
1.1	背景	1
1.2	関連研究	3
1.3	目的	8
1.4	本論文の構成	8
第 2 章	要素技術	9
2.1	ナビゲーション	9
2.2	ニューラルネットワーク	10
2.3	再帰型ニューラルネットワーク	11
2.4	アテンション機構	12
2.5	グラフニューラルネットワーク	14
第 3 章	提案手法	16
3.1	本章の概要	16
3.2	問題設定	16
3.3	ネットワーク構造	17
3.3.1	エンコーダモジュール	18
3.3.2	グラフアテンションネットワーク	18
3.3.3	デコーダモジュール	19
3.4	ロボットの行動を考慮した軌道予測	19
3.5	学習方法	20

3.6	学習環境	21
3.7	ネットワークの予備実験	21
3.7.1	実験概要	21
3.7.2	学習条件	21
3.7.3	評価指標	22
3.7.4	結果と考察	23
第 4 章	ロボットの行動を考慮した歩行者の軌道予測の検証実験	25
4.1	本章の概要	25
4.2	実験概要	25
4.3	実験方法	25
4.4	結果と考察	27
第 5 章	予測結果のナビゲーションへの応用	30
5.1	本章の概要	30
5.2	システム概要	30
5.3	実ロボットにおける歩行者の位置計算	31
5.4	ナビゲーションにおける予測結果の利用方法	34
第 6 章	予測結果を利用したナビゲーションの実験	35
6.1	本章の概要	35
6.2	実験概要	35
6.3	実験方法	35
6.4	結果と考察	38
第 7 章	結論	41
7.1	まとめ	41
7.2	今後の展望	41
参考文献		42
付録		47

目次

vii

謝辞

48

図目次

1.1	Overview of Social-LSTM method.* ¹	3
1.2	Framework overview of proposed SR-LSTM.* ²	4
1.3	Illustration of the pedestrian-wise attention.* ²	4
1.4	Illustration of the pedestrian-wise attention.* ³	5
1.5	The Social-STGCNN Model.* ⁴	5
1.6	Encoder structure.* ⁵	6
2.1	Neural Network	10
2.2	Neural Network	11
2.3	Neural Network	13
2.4	Neural Network	15
3.1	Network Structure	18
3.2	Neural Network	20
3.3	ADE	22
3.4	FDE	22
4.1	Experimental environment	26
4.2	Tracking sensor setup	27
4.3	Infrared sensor setup	27
4.4	Predicted trajectory when the robot moves straight	28
4.5	Predicted trajectory when the robot moves right	29
4.6	Predicted trajectory when the robot moves left	29

5.1	Navigation System Overview	31
5.2	Pedestrian Position Calculation: (a) on Dataset, (b) on Real Robot	32
5.3	Global Costmap in This Study	34
5.4	Process each layer to generate a global costmap	34
6.1	Simulator Environment	36
6.2	Simulated Robot and Actor	36
6.3	Experiment Scenarios	37
6.4	Scenario1 result	39
6.5	Scenario2 result	39

表目次

3.1	Experimental Setup	21
3.2	Comparison of ADE/FDE for each model ^{*6}	23
3.3	Number of samples and density of each dataset ^{*7}	24
3.4	Number of parameters and inference time for each model ^{*6}	24
4.1	Comparison with and without taking into account the robot's behavior .	28

第1章

序論

1.1 背景

近年，ロボット技術の進歩に伴い，製造業，物流，サービス業など幅広い分野で自律移動ロボットの活用が進んでいる。人間とロボットが共存する環境では，ロボットが安全かつ効率的に動作するために，周囲の環境，特に人間の行動を正確に理解し，適切に対応することが求められる。なかでも歩行者は動的な障害物であり，その予測困難な動きはロボットのナビゲーションにとって大きな課題となっている。

この課題に対して，Social Force Model[1]などの物理法則に基づくルールベース手法が提案されてきた。これらの手法では，歩行者の動きを物理的な力の相互作用によってモデル化し，歩行者同士や障害物との関係を考慮することで，より現実的な動きを再現できるように試みている。しかし，複雑な環境や多様な人間の行動を完全に表現することは依然として困難である。

一方，近年では深層学習による歩行者の動きを予測する手法が注目されている。特に，リカレントニューラルネットワーク（RNN）[2, 3] やその発展形である長短期記憶（LSTM）[4] ネットワークを用いたアプローチが有望視されている。これらの手法は，歩行者の過去の軌跡を考慮して未来の動きを予測できる。また，グラフ畳み込みニューラルネットワーク（GCN）[5] やグラファテンションネットワーク（GAT）[6] を用いた手法も注目を集めている。これらの手法は，個々の歩行者をノード，相対距離などの関係性をエッジとみなしてグラフ化することで，歩行者間の相互作用を効率的に学習できる。

しかし、これらの深層学習ベースの手法を移動ロボットの文脈に直接適用した研究はまだ少ない。また、移動ロボットのナビゲーションに予測結果を応用した例の報告が少ない。さらに、移動ロボットを対象とする既存の研究でも、歩行者へのセンサ取り付けなど環境への介入が必要だったり、ロボットのナビゲーション性能を評価するための実験環境が十分に整備されていない場合が多い。

また、移動ロボットのナビゲーションに予測結果を応用する場合、歩行者同士の相互作用の他にロボットと人間との間にも相互作用が存在する。そのため、歩行者同士の相互作用だけではなく、ロボットと歩行者の相互作用も考慮する必要がある点も課題として挙げられる。つまり、ロボットの行動が歩行者の行動に影響を与えるため、歩行者のみならずロボットの行動も考慮した軌道予測が求められる。

そこで本研究では、上記2点の課題を克服するため、ロボットの行動を考慮した歩行者の軌道予測を行い、その予測結果を用いたナビゲーションシステムを提案する。そして、本システムによるナビゲーション性能を実験を通じて検証し、その有効性を示すことを目指す。

1.2 関連研究

近年，歩行者の複雑な動きを予測するために，深層学習を応用しようとする研究が注目を集めている。深層学習を歩行者の軌道予測に応用した例として，Alexandre ら [1] は，人間の動きを学習し，将来の軌跡を予測できる LSTM モデルを提案した。Social-LSTM は，歩行者の軌道予測に着目した最も初期の深層学習モデルの一つである。Social-LSTM は，Fig. 1.1 のように各歩行者の軌跡ごとに 1 つの LSTM を割り当て，プーリング機構を通じて複数の LSTM 間で情報を共有することで，歩行者間の相互作用を捉えることを可能にしている。このシステムにより，複数の個人にわたって相互作用から生じるさまざまな非線形行動をうまく予測することに成功した。

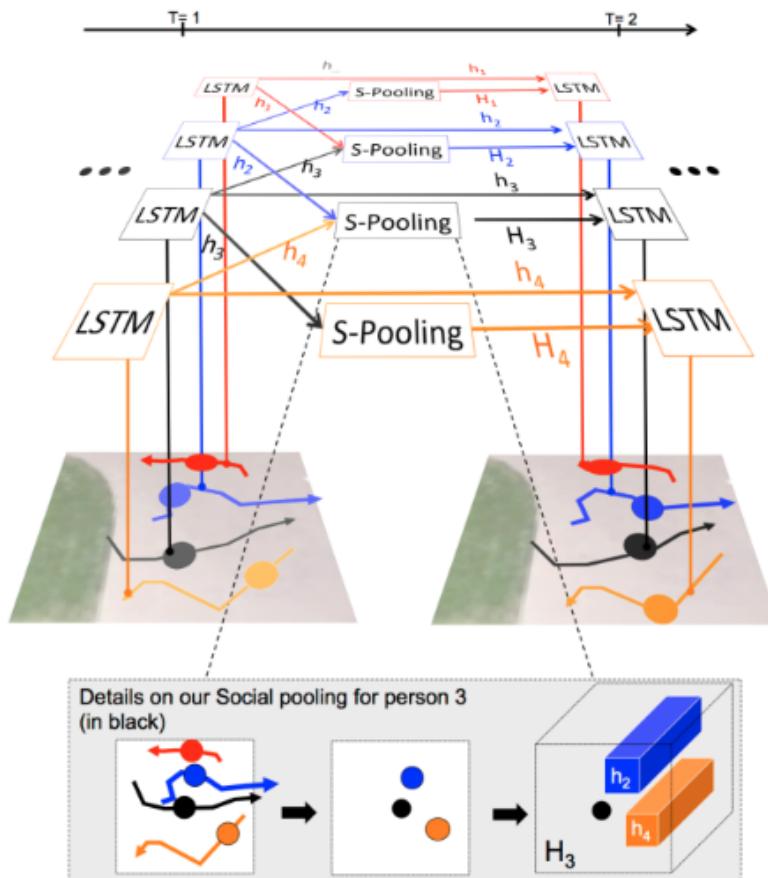


Fig. 1.1: Overview of Social-LSTM method.*¹

*¹ [1] より引用

Zhang ら [7] は、LSTM のシーケンス学習能力にエージェント間の相互作用をモデル化するため、Fig. 1.2 のような LSTM ネットワークの情報精密化モジュールを提案している。この研究では、Fig. 1.3 のように Attention 機構により、各歩行者の他者への影響を重み付けしている。

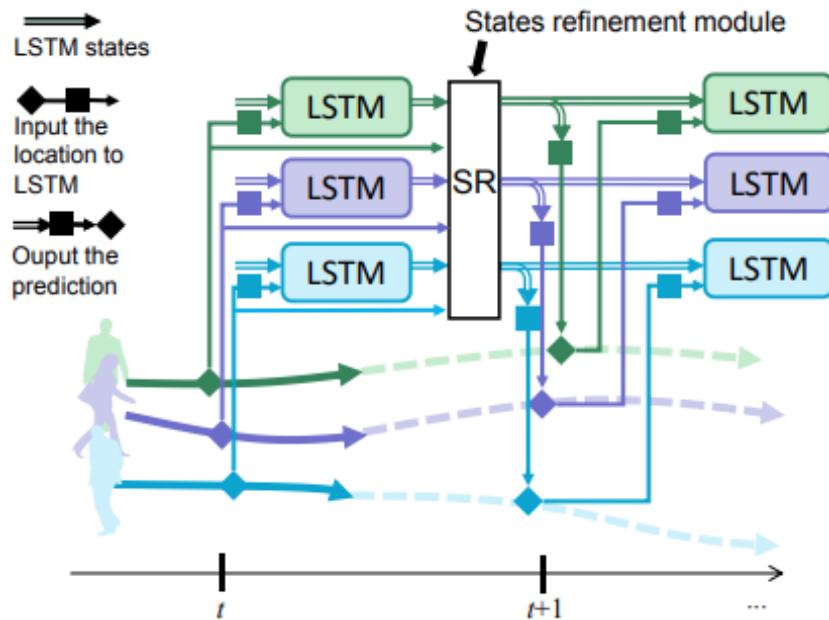


Fig. 1.2: Framework overview of proposed SR-LSTM.*²

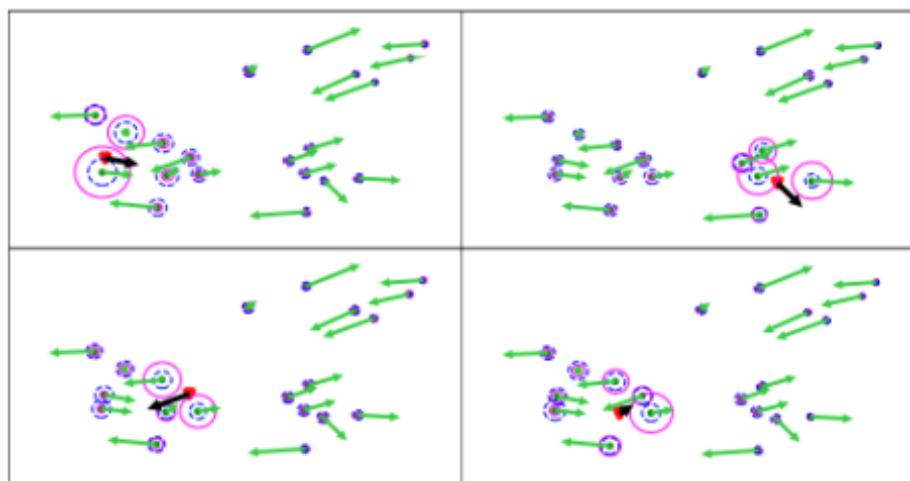


Fig. 1.3: Illustration of the pedestrian-wise attention.*²

*² [7] より引用

同様に，歩行者の注意に着目した研究を Kosaraju ら [8] も行っている。この研究は，LSTM を用いて各歩行者の軌跡をモデル化し，グラファテンションネットワーク（GAT）を用いて，現実的な歩行者の軌道予測を実現している。この研究では，グラフは Fig. 1.4 のように歩行者をノード，距離をエッジとしてモデル化される。このグラフ構造を用いることで，歩行者間の相互作用を効果的に捉え，より正確な軌道予測が可能となる。

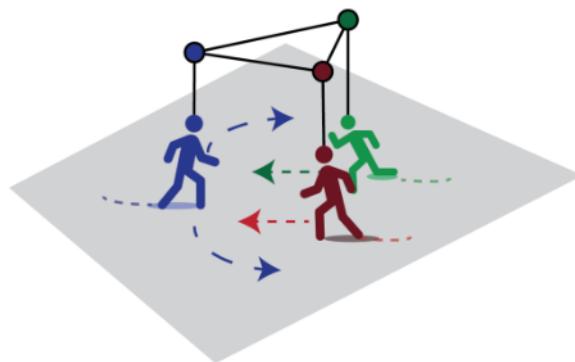


Fig. 1.4: Illustration of the pedestrian-wise attention.*³

また，Mohanmed ら [9] は軌跡をグラフとしてモデル化し，歩行者間のユークリッド距離で重み付けされたエッジは歩行者間の相互作用を表している。Fig. 1.5 のように，モデルはグラフ畳み込みニューラルネットワークと時間畳み込みネットワーク（TCN）を用いて，時空間グラフ上で動作し，一度にシーケンス全体を予測できるようにしている。

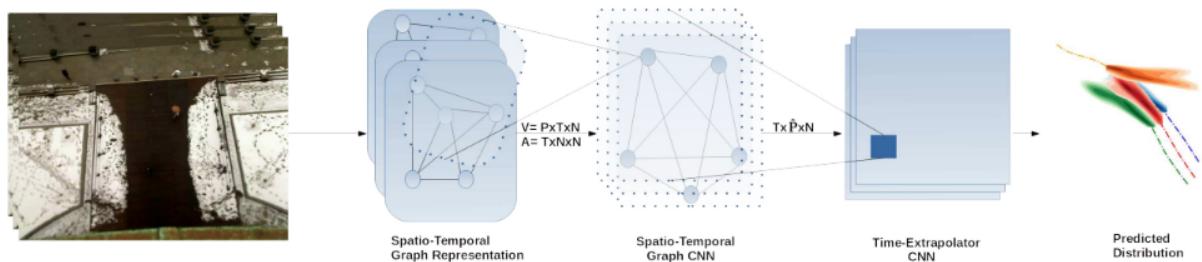


Fig. 1.5: The Social-STGCNN Model.*⁴

*³ [8] より引用

*⁴ [9] より引用

移動ロボットで歩行者の軌道予測を行う場合，歩行者同士の相互作用の他にロボットと人間にも相互作用が存在する．そのため，歩行者だけでなくロボットの行動も考慮した軌道予測手法が求められる．

丹野ら [10] は，歩行者の軌道予測に過去の動きだけでなく，Fig. 2 に示すようにロボットが選択する将来の動きを考慮して予測をする手法を提案している．実験では，ロボットの将来の行動を考慮しない場合と比較して，ロボットの動きに応じて変化する歩行者の動きを予測できることが確認されている．しかし，この手法では学習器の事前学習時に，データセット内の歩行者をロボットに置き換えて学習している．すなわち，歩行者の行動がロボットの行動と近似できるという前提のもとでモデルを構築していることになる．この前提に基づけば，モデル構造におけるロボットのエンコード部分を省略し，人間とロボットでネットワークを共有化できる可能性が考えられる．これは，ネットワークの肥大化を防ぐ上で有効である．本研究では，4 章においてこの論文 [10] を参考に実験を行う．

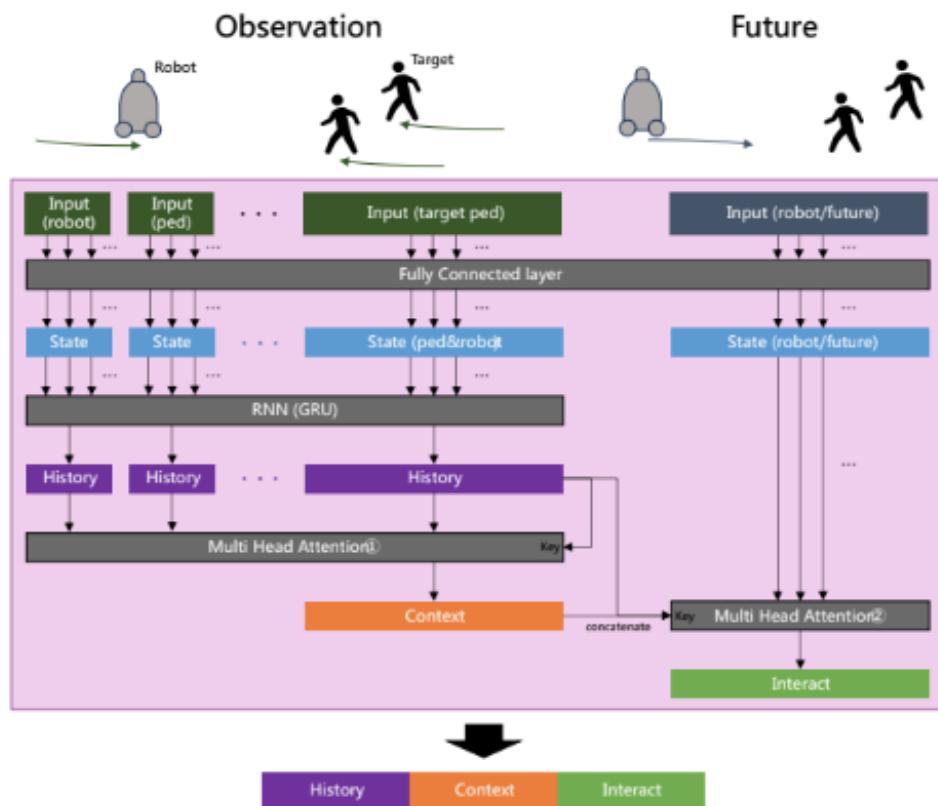


Fig. 1.6: Encoder structure.*⁵

*⁵ [10] より引用

このように、深層学習を歩行者の軌道予測タスクに応用する例は少なくない。しかし、その多くは俯瞰図形式のデータセットを用いて学習機を訓練しており、実際の移動ロボットを対象とした研究は少ない。また、移動ロボットを扱っている場合でも、予測対象の歩行者にトラッキングセンサを取り付けるなど環境に介入している例や、シミュレータから真値を取得しているなど、移動ロボットが収集するセンサデータのみで完結する手法はあまり報告されていない。そのため、深層学習用いて予測した歩行者の軌道を移動ロボットのナビゲーションに直接活用している研究は、現時点では非常に限られている。

1.3 目的

1.2 節で紹介した深層学習によるアプローチは、歩行者の複雑な動きを予測する上で有用である。しかし、環境への介入なしに予測結果を移動ロボットのナビゲーションへ応用している研究事例は少ない。また、歩行者の軌道予測を移動ロボットで活用するにあたっては、ロボットと歩行者間の相互作用も考慮すべきである。そこで本論文では、環境への介入を要しないかたちでロボットの行動を考慮した歩行者の軌道予測手法を提案する。さらに、本手法を移動ロボットのナビゲーションに応用し、ロボットが安全かつ効率的に移動できるか検討する。

1.4 本論文の構成

本論文は、全7章で構成されている。第1章では、研究の背景や関連研究、目的を述べた。第2章では、本研究に用いた要素技術について述べる。第3章では、本研究で提案する手法について具体的に説明し、設計したネットワークの基本性能を確認した。第4章では、提案手法の有効性を検証する実験について述べる。第5章では、予測結果を用いたナビゲーションシステムの構築について述べる。第6章では、実験で予測結果を利用したナビゲーションの有効性を検証する。第7章では、本研究の成果をまとめ、今後の展望について述べる。

第2章

要素技術

2.1 ナビゲーション

LiDAR やオドメトリなどのセンサ情報をを利用して自律走行する移動ロボットの多くは，ROS Navigation stack[11, 12] を採用している。例えば，つくば市内の遊歩道を移動ロボットが自律走行する技術チャレンジ「つくばチャレンジ」において，原らの技術調査 [13] によると，少なくとも参加 76 チーム中 22 チームが自律走行で ROS Navigation stack 使用していたと報告されている。これは，オープンソースのソフトウェアにとしては最多の利用数であった。ROS Navigation stack には，以下の主要な機能が含まれている。

- 自己位置推定 (Localization)

AMCL (Adaptive Monte Carlo Localization) アルゴリズムを用いて，ロボットの現在位置を推定する。

- 地図生成 (Mapping)

SLAM (Simultaneous Localization and Mapping) 技術を活用し，環境のマッピングを行う。

- 経路計画 (Path Planning)

Dijkstra 法 [14] や A*アルゴリズム [15] を用いて，最適経路を計画する。

- 障害物回避 (Obstacle Avoidance)

センサデータに基づいて障害物を検知し，動的に回避しながら移動する。

2.2 ニューラルネットワーク

ニューラルネットワーク (Neural Network) は、人間の脳の神経回路に着想を得た計算モデルであり、機械学習や人工知能の分野で重要な役割を果たしている。大量のデータから学習し、パターン認識や予測などの複雑なタスクを遂行できる点が特徴である。

ニューラルネットワークは、多数のノード（ニューロン）が層状に接続された構造を持つ。各ノードは入力信号を受け取り、重み付けを行い、活性化関数を適用することで出力信号を生成する。これらのノードは以下の3層に大別される。

- **入力層 (Input Layer)**

入力データを受け取る層であり、各ノードが入力データの各特徴量に対応する。

- **隠れ層 (Hidden Layer)**

入力層と出力層の間に位置する層である。複数の隠れ層を設けることで、複雑な非線形関係をモデル化でき、より高度な特徴表現の獲得が可能となる。

- **出力層 (Output Layer)**

ネットワークの最終的な出力を生成する層である。

各層のノードは次の層のノードと相互に接続され、接続には重みが割り当てられている。学習の過程では、誤差逆伝播などの最適化手法を用いてこれらの重みを調整し、ネットワークが目的のタスクを精度良く実行できるように最適化される。多層構造を持つことで複雑なパターンや特徴を学習し、従来では困難であった高精度な予測を実現した学習手法を深層学習と呼ぶ。深層学習は、画像認識や自然言語処理、音声認識など幅広い分野で活用されており、YOLO (You Only Look Once) [16] や ChatGPT[17, 18, 19] といった成功例が多数存在し、高い性能が評価されている。



Fig. 2.1: Neural Network

2.3 再帰型ニューラルネットワーク

再帰型ニューラルネットワーク (Recurrent Neural Network, RNN) [2, 3] は、時系列データや系列データを扱うために設計されたニューラルネットワークの一種である。RNN は隠れ層の出力を次の時間ステップの入力として再利用し、データの時間的依存性をモデル化できる点に特徴がある。この仕組みによって、従来のニューラルネットワークでは困難だった順序情報の学習が可能となる。

RNN の基本構造は、入力層・隠れ層・出力層から構成される。時系列データを用いる際、系列データの各時刻に対応する入力が順次 RNN に与えられ、隠れ層は現在の入力と直前の隠れ状態を組み合わせて次の隠れ状態を生成する。過去の情報を保持するこの隠れ層が、長期依存関係を学習する上で重要な役割を担う。

数式で表すと、時刻 t における隠れ状態 h_t は式 (2.1) のように計算される。

$$h_t = f(W_{ih}x_t + W_{hh}h_{t-1} + b_n) \quad (2.1)$$

ここで、 x_t は時刻 t の入力、 h_{t-1} は時刻 $t - 1$ の隠れ状態、 W_{ih} と W_{hh} はそれぞれ入力重みと隠れ層の重み行列、 b_n はバイアス項、そして f は非線形活性化関数である。

RNN には、勾配が時間を経るにつれて減少または増大し続けることで学習が困難になる勾配消失問題 [20, 21, 22] や勾配爆発問題が存在する。これらの問題は、長い系列データを扱う場合に十分な長期依存性を学習できない原因となる。そこで、LSTM (Long Short Term Memory) [4] や GRU (Gated Recurrent Unit) [23] などの改良モデルが提案されている。これらのモデルは、ゲート機構を用いて重要な情報を選択的に保持または忘却することで、長期依存性を効率よく学習するように設計されている。



Fig. 2.2: Neural Network

2.4 アテンション機構

アテンション機構 [24] は、ニューラルネットワークの文脈で導入された手法の一つであり、特に自然言語処理や画像認識などにおいて顕著な成果を示している。その主な目的は、入力データに含まれる重要な情報に焦点を当て、それ以外の情報を適度に省くことである。この仕組みにより、モデルはより効率的かつ効果的に学習を進めることが可能となる。

アテンション機構の基本的な考え方とは、入力を構成する各要素がどの程度重要かをスコア付けし、そのスコアをもとに加重平均を行うことである。これにより、モデルはタスクにとって重要な部分に「注意」を向けることが可能となる。このスコア付けの過程では、キー (Key)、クエリ (Query)、バリュー (Value) の 3 つの成分が用いられる。それぞれの役割を以下に示す。

- クエリ (Query)

特定の出力を生成する際、入力シーケンス全体の中で「どこに注目すべきか」を判断するためのベクトルである。

- キー (Key)

入力データ各部分の重要度を示す指標となるベクトルであり、クエリとの比較によって異なる部分の関連性が評価される。

- バリュー (Value)

実際に計算される情報の内容を保持するベクトルである。注意スコアに基づいて加重平均が適用され、最終的な出力が得られる。

アテンション機構の計算では、まずクエリとキーの内積によって各要素間の関連性を評価し、ソフトマックス関数による正規化によって注意スコアを算出する。その後、このスコアを用いてバリューに対する加重平均を行い、最終的な結果を出力する。

この手法は、特に Transformer アーキテクチャにおいて重要な役割を果たしている。Self-Attention として知られる拡張では、入力シーケンスの各要素が他のすべての要素とどのように関連しているかを同時に評価する。これにより、従来のリカレントニューラルネットワーク (RNN) や畳み込みニューラルネットワーク (CNN) [25, 26] が抱えていた長距離依存性の問

題を効果的に緩和できる。

さらにアテンション機構は、自然言語処理のみならず、画像認識や音声処理など幅広い領域に応用されている。例えば、画像セグメンテーションタスクでは、画像内の特定領域に焦点を当てることで高精度な結果が得られる。このようにアテンション機構は汎用的かつ強力な手法として、多くの応用領域でその有用性が示されている。



Fig. 2.3: Neural Network

2.5 グラフニューラルネットワーク

グラフニューラルネットワーク (Graph Neural Network ; GNN) は、ノード (頂点) とエッジ (辺) で構成されるグラフ構造のデータを処理するために設計されたニューラルネットワークの一種である。GNN では、各ノードの特徴ベクトルを学習すると同時に、ノード間の関係性を考慮した上で予測や分類を行う。

GNN の基本的なアイデアは、各ノードの特徴ベクトルをその隣接ノードの特徴ベクトルと組み合わせて反復的に更新することである。この一連の更新は、メッセージパッシングと呼ばれ、複数の層を通じて繰り返し行われる。各層では、ノードの特徴ベクトルが隣接ノードからの情報を集約して更新されるため、最終的にグラフ全体にわたる依存関係を捉えることが可能となる。代表的な GNN のアーキテクチャとしては、以下のようなものが挙げられる。

- **Graph Convolutional Networks (GCN)**

グラフ畳み込みネットワークは、畳み込みに相当する操作をグラフに適用し、隣接ノードの情報を考慮して各ノードの特徴を更新する。

- **Graph Attention Networks (GAT)**

グラフアテンションネットワークは、隣接するノード間に異なる重みを付与することで、重要度の高いノードからの情報を強調する。アテンション機構を取り入れることで、より柔軟な学習が可能となる。

- **GraphSAGE**

ノードサンプリングと情報集約の手法を組み合わせることで、大規模なグラフを効率的に扱う。各ステップでサブグラフを取り出し、必要最小限の隣接ノードのみを考慮して学習を行うため、計算コストを抑制できる。

GNN は、ソーシャルネットワーク分析、化学分子の特性予測、知識グラフの補完など、さまざまな分野で応用されている。グラフ構造データの持つ複雑なパターンを学習し、従来の手法では困難とされてきた問題を解決可能である点が GNN の大きな利点である。



Fig. 2.4: Neural Network

第3章

提案手法

3.1 本章の概要

本章では、ロボットの行動を考慮した歩行者の軌道予測手法を提案し、その詳細を述べる。まず、問題設定として歩行者のグラフ表現と軌道予測タスクの定義を行う。次に、提案手法で用いるエンコーダ・デコーダ構造のネットワークについて説明する。さらに、ロボットの行動を考慮した軌道予測の方法を示し、学習方法と学習環境についても述べる。最後に、ネットワークの基本性能を確認する。

3.2 問題設定

本論文における歩行者のグラフ表現を説明する。まず、時刻 t におけるシーン内の歩行者の位置を表す空間グラフを

$$G_t = (V_t, E_t) \quad (3.1)$$

と定義する。グラフ G_t のノードの集合を

$$V_t = \{v_t^i \mid i = 1, \dots, N\} \quad (3.2)$$

と表す。各ノード v_t^i は歩行者 i の位置 (x_t^i, y_t^i) を属性として持つ。一方、グラフ G_t のエッジの集合を

$$E_t = \{e_t^{ij} \mid i, j = 1, \dots, N\} \quad (3.3)$$

と表す。ノード v_t^i と v_t^j が接続されている場合に $e_t^{ij} = 1$ となり、そうでない場合 $e_t^{ij} = 0$ となる。

本論文における軌道予測タスクとは、過去の観測情報をもとに、将来の歩行者の位置 (x, y 座標) を予測ステップ分出力することである。歩行者 i の各時刻の位置を

$$V_t = \{v_t^i = (x_t^i, y_t^i) \in \mathbb{R}^2 \mid i = 1, \dots, N\} \quad (3.4)$$

と表す。また、観測時間 t_{obs} において得られた全ての歩行者の位置データを

$$V_{obs} = \{V_t \mid t = 1, \dots, t_{obs}\} \quad (3.5)$$

とする。ここで、時刻 t の歩行者 i の位置の確率分布を

$$\hat{V}_t = \{\hat{p}_t^i = (\hat{x}_t^i, \hat{y}_t^i) \mid i = 1, \dots, N\} \quad (3.6)$$

と表す。そうすると、予測時間 t_{pred} にわたって予測される全ての歩行者の位置データを

$$V_{pred} = \{\hat{V}_t \mid t = t_{obs+1}, \dots, t_{pred}\} \quad (3.7)$$

と表すことができる。また、 p_t^i が 2 次元正規分布 $\mathcal{N}(\mu_t^i, \sigma_t^i, \rho_t^i)$ に従うと仮定する。すなわち、

$$\hat{p}_t^i \sim \mathcal{N}(\hat{\mu}_t^i, \hat{\sigma}_t^i, \hat{\rho}_t^i) \quad (3.8)$$

を満たすものとする。ここで、平均は $\hat{\mu}_t^i = (\hat{\mu}_{x,t}^i, \hat{\mu}_{y,t}^i)$ 、分散は $\hat{\sigma}_t^i = (\hat{\sigma}_{x,t}^i, \hat{\sigma}_{y,t}^i)$ 、相関係数は $\hat{\rho}_t^i$ である。これらを後述するネットワークの出力とする。

3.3 ネットワーク構造

本手法で用いるネットワークは、エンコーダ・デコーダ構造で構成されている。ネットワークは、主にエンコーダモジュール、グラフアテンションネットワークモジュール、デコーダモジュールの 3 つから構成される。エンコーダモジュールでは入力データから特徴量を抽出し、グラフアテンションネットワークを用いてノード間の関係性を学習する。その後、得られた潜在表現をもとにデコーダモジュールで所定のシーケンス長にわたる予測を行う。ネットワークの概要を Fig. 3.1 に示す。

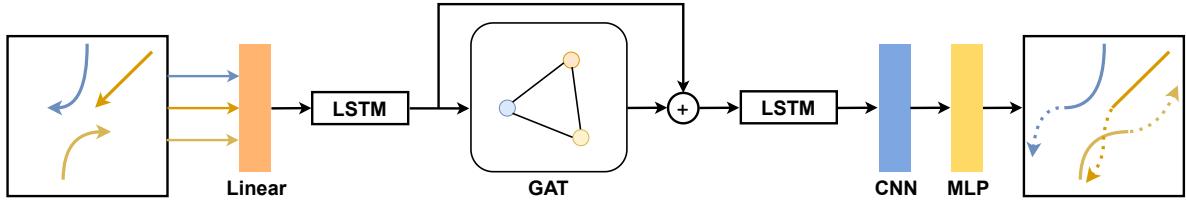


Fig. 3.1: Network Structure

3.3.1 エンコーダモジュール

エンコーダでは、まず各時刻における歩行者の過去位置を線形変換層 ϕ_{emb} によって高次元空間へ埋め込む。この埋め込み表現を LSTM 層に入力し、歩行者の過去の軌跡を時間的に処理した隠れ状態を出力する。ここで得られた隠れ状態は、歩行者の過去の動きを要約した表現となり、次に続くグラファテンションネットワークモジュールに渡される。エンコーダにおける処理を式 (3.9), (3.10) に示す。なお、 v_t^i と v_i^t は同義とする。

$$e_i^t = \phi_{emb}(v_i^t; W_{emb}) \quad (3.9)$$

$$h_i = \text{LSTM}_{en}(e_i^t, h_i^{t-1}; W_{en}) \quad (3.10)$$

3.3.2 グラファテンションネットワーク

本ネットワークの中核を担うグラファテンションネットワーク (GAT) [6] は、マルチヘッドアテンション機構を用いる複数の層で構成される。各 GAT 層では、近傍ノードの特徴を集約して各ノードの特徴ベクトルを更新し、ノード間の関係強度に基づいて重み付けを行う。そのため、関連性の高いノードからの情報が強調される。複数の GAT 層を積み重ねることで、グラフ構造における高次の依存関係を捉えることが可能となる。また、各タイムステップで対応する隣接行列を用いてグラフ畳み込みが実行され、動的なグラフ構造の変化にも対応できる。

歩行者 i の隠れ状態 h_i が与えられたとき、全歩行者に対して複数の GAT 層を適用する。各層は以下のように適用される。式 (3.11) の e_{ij} はノード i に対するノード j の特徴の重要度を表している。 W は重み行列、 a は共有アテンション機構である。また、活性化関数 σ として本手法ではネットワーク全体にわたり PReLU[27] を採用した。

$$e_{ij} = a(Wh_i, Wh_j) \quad (3.11)$$

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) \quad (3.12)$$

$$h'_i = \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} Wh_j \right) \quad (3.13)$$

3.3.3 デコーダモジュール

GAT 層からの出力は、デコーダモジュールによって将来の歩行者の位置に変換される。まず、2 つ目の LSTM 層を用いて GAT 層により生成された時空間特徴表現をさらに時間方向に処理する。次に、1 次元畳み込み層 (CNN) を適用し、入力シーケンス長から予測シーケンス長への変換を行う。この層は、予測時間に合わせて特徴表現の次元を調整する役割を果たす。最後に、多層パーセプトロン (MLP) を用いて予測値を生成する。式 (3.17) に示すように、本ネットワークの最終出力は 5 次元である。なお、 \hat{P}_i^t と \hat{P}_t^i は同義である。

$$h''_i = \text{LSTM}_{dec}(h'_i, h_{deci}; W_{dec}) \quad (3.14)$$

$$c_i^t = \text{CNN}(h''_i; W_{cnn}) \quad (3.15)$$

$$\hat{P}_i^t = \text{MLP}_d(c_i^t; W_d) \quad (3.16)$$

$$\hat{P}_t^i = [\hat{\mu}_{x,t}^i, \hat{\mu}_{y,t}^i, \hat{\sigma}_{x,t}^i, \hat{\sigma}_{y,t}^i, \hat{\rho}_t^i] \quad (3.17)$$

さらに、本ネットワークでは Fig. 3.1 に示すように、GAT 層と LSTM 層の出力に対してスキップ接続 [28] を導入することで、勾配消失問題 [20, 21, 22] を軽減し、学習を安定化させる。本アーキテクチャにより、時系列データの動的な変化とグラフ構造におけるノード間の複雑な依存関係を効果的に捉え、高精度な予測を実現する。

3.4 口ボットの行動を考慮した軌道予測

本研究では、歩行者が頻繁に行き交うような環境において、低速走行する移動口ボットの行動が歩行者の行動と近似可能であると仮定する。すなわち、口ボットと歩行者を区別せずに予測を行う方針を採用する。口ボットの行動を考慮した軌道予測を行うために、口ボットの経路情報を追加の入力としてネットワークに与える。具体的には、口ボットの位置情報を歩行者の

位置情報と同様にグラフのノードとして扱い、エンコーダモジュールで処理する。これにより、ロボットと歩行者の相互作用を考慮した予測が可能となる。

ロボットの位置を表すノード r_t を追加し、グラフ G_t を拡張する。拡張されたグラフ $\tilde{G}_t = (\tilde{V}_t, \tilde{E}_t)$ は、 $\tilde{V}_t = V_t \cup \{r_t\}$ と定義される。エッジ集合 \tilde{E}_t も同様に拡張され、ロボットと歩行者間のエッジが追加されるが、ネットワーク構造自体を変更する必要はない。

拡張グラフ \tilde{G}_t は既存のエンコーダモジュールにそのまま入力可能であり、ロボットの位置 r_t も他のノードと同様に埋め込み表現 e_r^t に変換され、LSTM 層で処理される。グラフアテンションネットワークでは、ロボットと歩行者間の相互作用を考慮したアテンション重みが計算される。デコーダモジュールでは、ロボットの位置情報を含む潜在表現を基づき、将来の歩行者の位置を予測する。以上により、ロボットの行動を考慮した軌道予測が可能となる。4 章では、実験で提案手法の有効性を評価する。

3.5 学習方法

本研究で提案するネットワークの学習方法の詳細について述べる。本ネットワークは、以下の負の対数尤度を最小化するように学習される。式 (3.18) の損失関数を用いる。

$$L^i = - \sum_{t=t_{obs}+1}^{t_{pred}} \log (P(\hat{p}_t^i | \hat{\mu}_t^i, \hat{\sigma}_t^i, \hat{\rho}_t^i)) \quad (3.18)$$

本ネットワークの学習には、2 つの歩行者の軌跡データを含むデータセットを用いる。ETH データセット [29] と UCY データセット [30] である。この 2 つのデータセットは、図のような 5 つのシーン (ETH, HOTEL, UNIV, ZARA1, ZARA2) から構成され、合計 1536 人の歩行者のデータが含まれている。データセットの軌跡は 0.4 秒ごとにサンプリングされたものである。



Fig. 3.2: Neural Network

3.6 学習環境

本研究で行う学習は全て以下の Table 3.1 に示す環境で実施する .

Table 3.1: Experimental Setup

OS	Ubuntu 20.04.6 LTS
CPU	Intel Core i7-10700F
GPU	NVIDIA GeForce RTX 3060
Memory	32GB
Language	Python 3.8.10
Framework	PyTorch 2.4.1

3.7 ネットワークの予備実験

ロボットの行動を考慮した軌道予測を実験で確認する前に，まず提案したネットワークの基本的な性能を確認するため，予備実験を行った .

3.7.1 実験概要

本実験では，Mohamed ら [9] の実験方法に倣って，事前に学習したモデルを後述する 2 種類の指標で評価する . その結果を複数のベースラインモデルと比較し，提案ネットワークの性能を確認する .

3.7.2 学習条件

ネットワークの学習には，先行研究 [1, 9] と同様にリープワンアウト (Leave One Out) 方式を採用することで，データセットを最大限に活用することができる . 具体的には，5 つのシーンのうち 1 つをテストデータとして除外し，残り 4 つのシーンを用いてモデルの訓練と検証をする . テスト時には，8 ステップにあたる 3.2 秒間観測し，次の 12 ステップにあたる 4.8 秒間の歩行者の将来位置を予測する . 学習パラメータとしては，バッチサイズを 128，オ

プティマイザに Adam[31] を用い、学習率は $1e^{-3}$ に設定した。最大エポック数は 250 とし、この条件下でネットワークを学習した。

3.7.3 評価指標

モデルの評価には、以下の 2 つの指標を用いた。

- ADE (Average Displacement Error) [29]:

$$\text{ADE} = \frac{1}{TN} \sum_{t=1}^T \sum_{i=1}^N \|p_t^i - \hat{p}_t^i\|_2 \quad (3.19)$$

ここで、 p_t^i は時刻 t における歩行者 i の実際の位置、 \hat{p}_t^i は予測された位置、 N は歩行者の総数、 T は予測の総時間ステップ数である。本研究では $T = 12$ である。Fig. 3.3 に示すように、ADE は実際の位置と予測位置の平均的な距離誤差を示す。

- FDE (Final Displacement Error) [1]:

$$\text{FDE} = \frac{1}{N} \sum_{i=1}^N \|p_{t_{pred}}^i - \hat{p}_{t_{pred}}^i\|_2, \quad t = t_{pred} \quad (3.20)$$

ここで、 p_t^i は最終時刻 t_{pred} における歩行者 i の実際の位置、 \hat{p}_t^i は予測された位置である。Fig. 3.4 に示すように、FDE は予測の最終時刻における実際の位置と予測位置の距離誤差を示す。

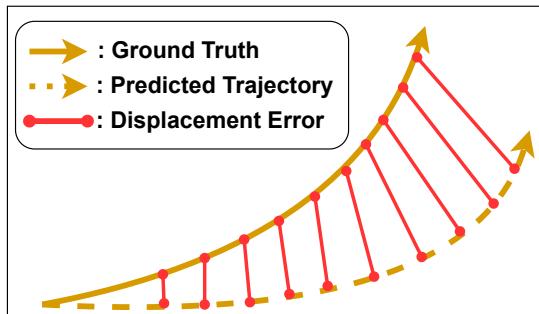


Fig. 3.3: ADE

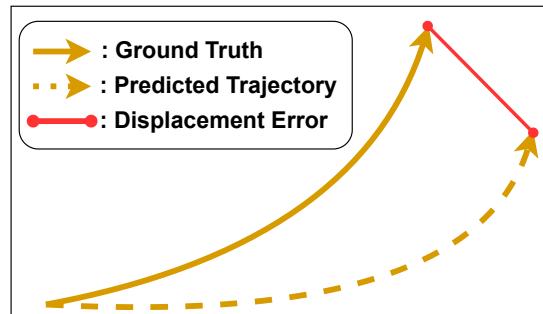


Fig. 3.4: FDE

3.7.4 結果と考察

Table 3.2: Comparison of ADE/FDE for each model^{*6}

Model	ETH	HOTEL	UNIV	ZARA1	ZARA2	Average
Linear [1]	1.33/2.94	0.39/0.72	0.82/1.59	0.62/1.21	0.77/1.42	0.79/1.59
SR-LSTM-2 [7]	0.63/1.25	0.37/0.74	0.51/1.10	0.41/0.90	0.32/0.70	0.45/0.94
S-LSTM [1]	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54
S-GAN-P [32]	0.87/1.62	0.67/1.52	0.76/1.52	0.35/0.68	0.42/0.84	0.61/1.21
SoPhie [33]	0.70/1.43	0.76/1.67	0.54/1.24	0.30/0.63	0.38/0.78	0.54/1.15
CGNS [34]	0.62 /1.40	0.70/0.93	0.48/1.20	0.32/0.59	0.35/0.71	0.49/0.97
PIF [35]	0.73/1.65	0.30 /0.59	0.60/1.27	0.38/0.81	0.31/0.68	0.46/1.00
STSGN [36]	0.75/1.63	0.63/1.40	0.48/1.08	0.30/0.65	0.26 /0.57	0.48/0.99
GAT [8]	0.68/1.29	0.68/1.40	0.57/1.29	0.29 /0.60	0.37/0.75	0.52/1.07
Social-BIGAT [8]	0.69/1.29	0.49/1.01	0.55/1.32	0.30/0.62	0.36/0.75	0.48/1.00
Social-STGCNN[9]	0.64/ 1.11	0.49/0.85	0.44/0.79	0.34/0.53	0.30/0.48	0.44/0.75
ours	0.69/1.24	0.33/0.52	0.42/0.78	0.29/0.49	0.26/0.45	0.40/0.69

Table 3.2 に示すように、本手法のモデルは 2 つの指標において他のベースラインモデルを上回る結果を示した。平均 ADE は 0.40 であり、最良のベースラインモデルと比較して約 9 % の改善を示した。また、FDE においても約 8 % の誤差減少が確認できた。Table 3.3 に示すとおり、5 つのシーンのうち HOTEL, ZARA1, ZARA2 は歩行者密度が低く、歩行者同士の複雑な相互作用が比較的少ないため、ADE と FDE の値は他のシーンよりも小さくなると考えられる。一方、ETH と UNIV は歩行者密度が高く、複雑な相互作用が多いいため、ADE と FDE の値が相対的に大きくなる傾向が見られた。

—— ここに具体的なネットワークの構造について言及している考察を追加 ——

^{*6} [9] のデータを基に作成

Table 3.4 は、各モデルのサイズと推論時間を本手法のモデルと比較したものである。Social-STGCNN[9] を除き、他のベースラインモデルと比較して、本手法のモデルはパラメータ数および推論時間の両方において効率的であることが分かる。しかし、本手法のモデルのパラメータ数は 28.6K と、Social-STGCNN の約 3 倍に増加しており、推論時間も約 20 倍ほど遅くなっている。移動ロボットのナビゲーションにおいては実時間性が重要となるが、本研究のタスクでは 0.4 秒以内に予測が完了すれば十分なため、この推論時間の増加は実用上大きな問題にはならないと言える。

Table 3.3: Number of samples and density of each dataset^{*7}

Scene	Number of pedestrians	density [person/m ²]
ETH	358	8
HOTEL	389	5
UCY	434	8
ZARA1	148	4
ZARA2	204	5

Table 3.4: Number of parameters and inference time for each model^{*6}

	Parameters count	Inference time
S-LSTM [1]	264K (9.2x)	1.1789 (27.4x)
SR-LSTM-2 [7]	64.9K (2.3x)	0.1578 (3.7x)
S-GAN-P [32]	46.3K (1.6x)	0.0968 (2.3x)
PIF [35]	360.3K (12.6x)	0.1145 (2.7x)
Social-STGCNN [9]	7.6K (0.3x)	0.0020 (0.05x)
ours	28.6K	0.043

^{*7} [37] のデータを基に作成^{*6} [9] のデータを基に作成

第4章

ロボットの行動を考慮した歩行者の軌道予測の検証実験

4.1 本章の概要

本章では、ロボットと歩行者の相互作用を観察し、提案手法を用いて歩行者の軌道を予測する実験を行った結果について述べる。まず、実験の概要を4.2節で説明し、次に実験方法を4.3節で詳述する。最後に、実験結果とその考察を4.4節で示す。

4.2 実験概要

本研究の提案手法の有効性を検証するため、実環境においてロボットと歩行者がどのように相互作用するかを観察し、提案手法を用いて行動を予測する実験を行った。実験には20代の男性4名が参加し、そのうち無作為に選ばれた2名のデータを予測用として利用した。

4.3 実験方法

本実験は、丹野らの研究[10]を参考に実施した。実験はFig. 4.1に示す室内環境で行い、Fig. 4.2aに示す移動ロボット(ORNE-box2[38])を使用した。実験参加者は、移動ロボットの正面から歩行を開始し、事前に指定された目的地(A/B/C)のいずれかまでロボットを避けながら歩行する。各参加者には、各目的地への歩行を3回ずつ行ってもらった。ロボットの

行動が歩行者の将来の軌道にどのように影響を与えるかを観察するため、ロボットは 0.3m/s の一定速度で直進した後、以下の 3 つの挙動のいずれかを事前に知らせることなく無作為に選択して実行する。

- 速度を変えず直進
- 速度を変えず右に避ける
- 速度を変えず左に避ける

Fig. 4.2 に示すように、参加者とロボットの位置は Oculus Rift の専用コントローラを用いて計測する。また、正確なトラッキングを行うため、Fig. 4.1 の斜線部に 2 基の赤外線センサを Fig. 4.3 のように配置する。取得データは、ETH[29]、UCY[30] データセットのフォーマットに準拠する。

予測の評価手順は、3 章で行った予備実験と同様とした。ロボットの行動を考慮した軌道予測手法の有効性を評価するため、ロボットの行動を考慮する場合と考慮しない場合の両方で予測を行い、それぞれの結果を比較する。

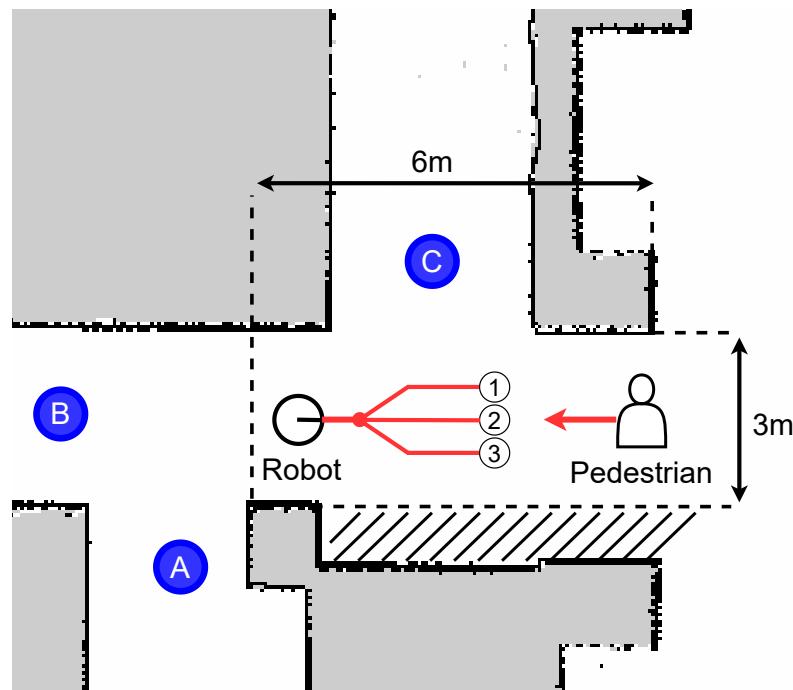


Fig. 4.1: Experimental environment

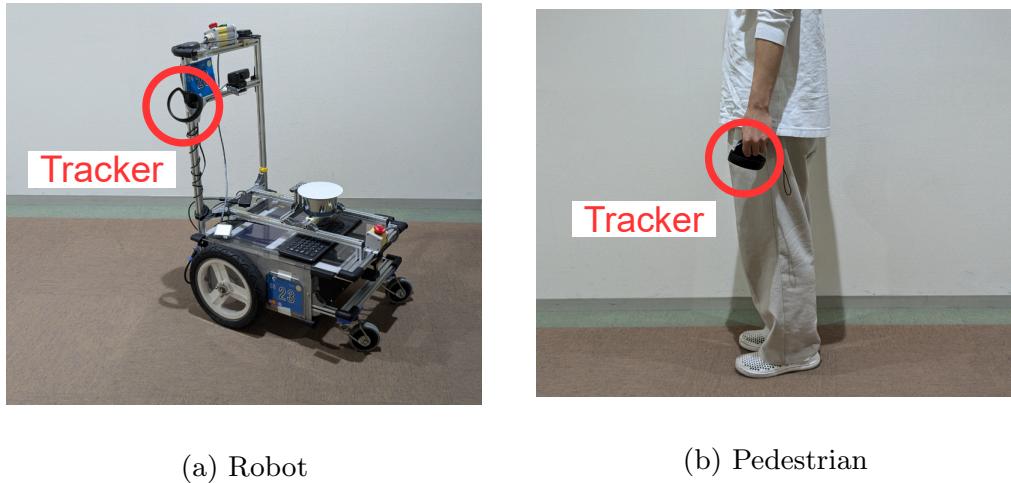


Fig. 4.2: Tracking sensor setup

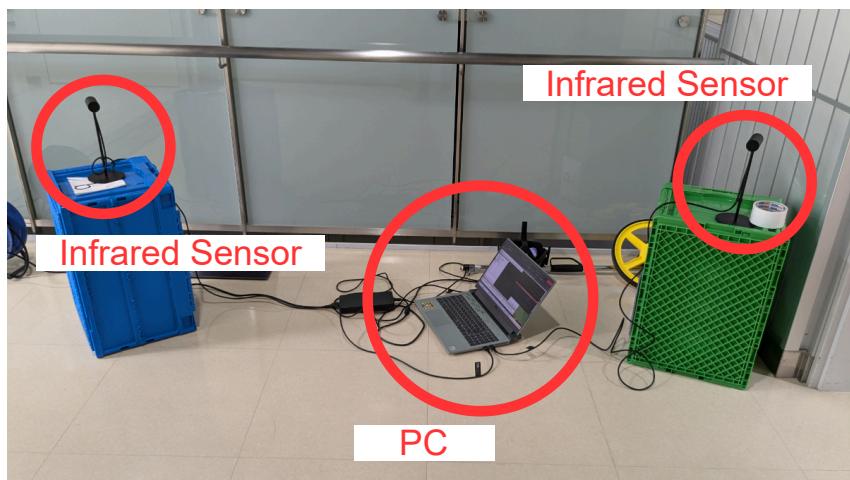


Fig. 4.3: Infrared sensor setup

4.4 結果と考察

Table 4.1 に、ロボットの行動を考慮する場合と考慮しない場合における各評価指標の値を示す。ロボットの行動を考慮しない場合と比較して、ADE は 7.5 %、FDE は 15.9 % 誤差低減が確認できる。この結果から、ロボットの行動を考慮することで、歩行者の軌道予測の精度が向上することが分かる。特に FDE の改善が顕著であり、ロボットの動きが歩行者の最終的な位置に大きな影響を与えていていることを示唆している可能性がある。一方で、ADE の改善幅は比較的小さく、Fig. 4.1 に示したように、3箇所の目的地のうち 2箇所で角を曲がる必要があるため、途中の軌道予測が困難になり、評価指標の値の向上幅が抑えられたと考えられる。

Table 4.1: Comparison with and without taking into account the robot's behavior

	ADE	FDE
Normal	0.40	0.69
Considering robot behavior	0.37	0.58

Fig. 4.4 , Fig. 4.5 , Fig. 4.6 は，実験で取得した目的地 A への歩行データに対する予測例である．これらの例では，ロボットが行動を取り始める時刻を最終観測時刻とした．ロボットの行動を考慮しない場合は，いずれのロボット拳動に対しても回避行動を伴う予測が見られなかった．一方，ロボットの行動を考慮した予測では，より真値に近い軌道を推定している様子が確認できる．観測時間内においては，歩行者の行動は一貫して直進していたため，その後のロボットを回避する動きを予測することは本来困難である．しかし，Fig. 4.4 , Fig. 4.5 ではロボットを回避するような予測が確認できる．このことから，ロボットの行動を考慮する軌道予測が歩行者とロボット間の相互作用を適切に捉えていると考えられる．

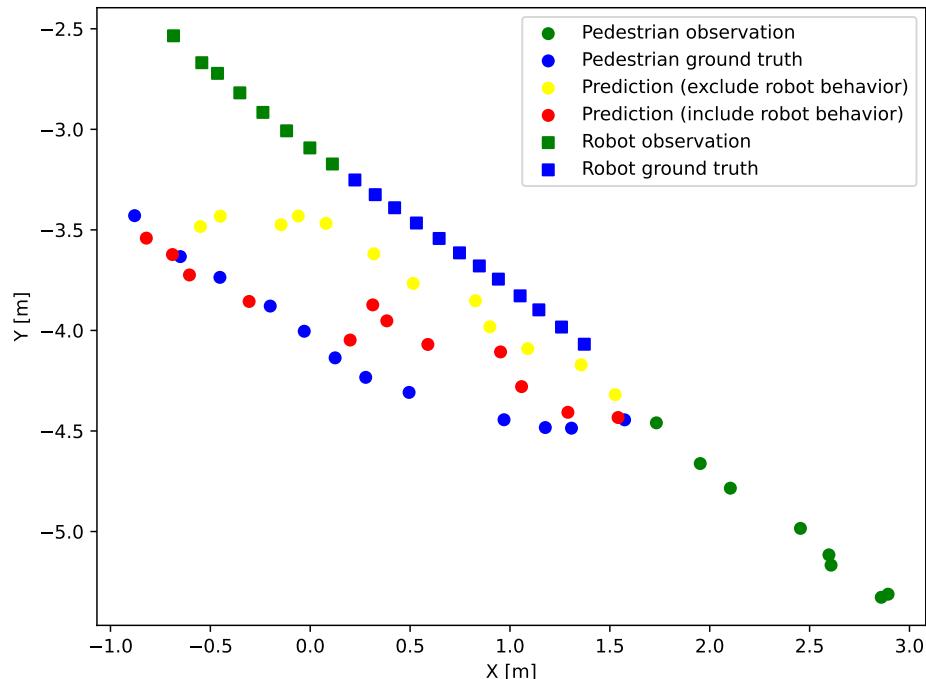


Fig. 4.4: Predicted trajectory when the robot moves straight

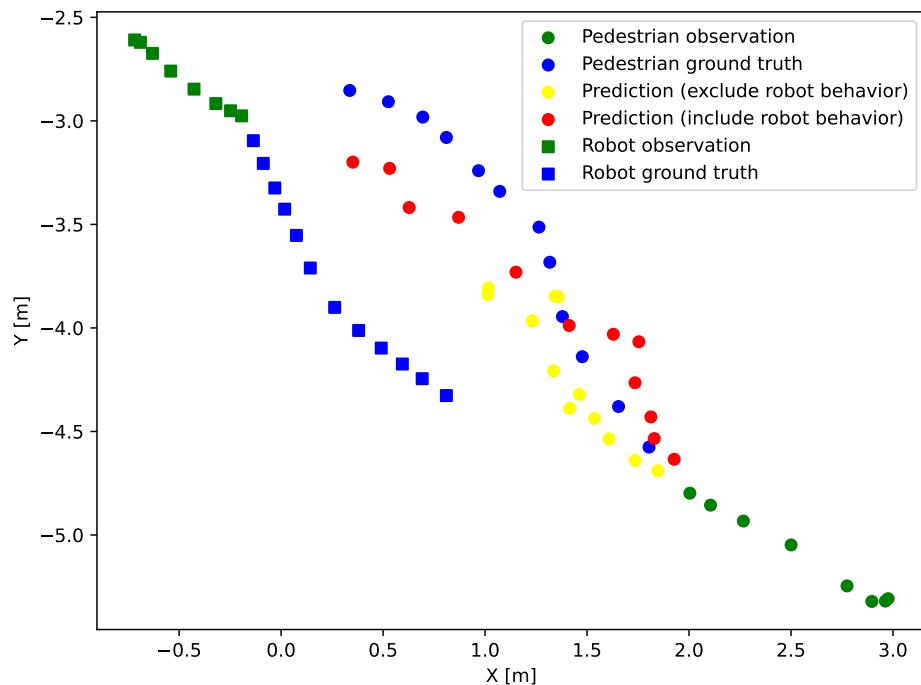


Fig. 4.5: Predicted trajectory when the robot moves right

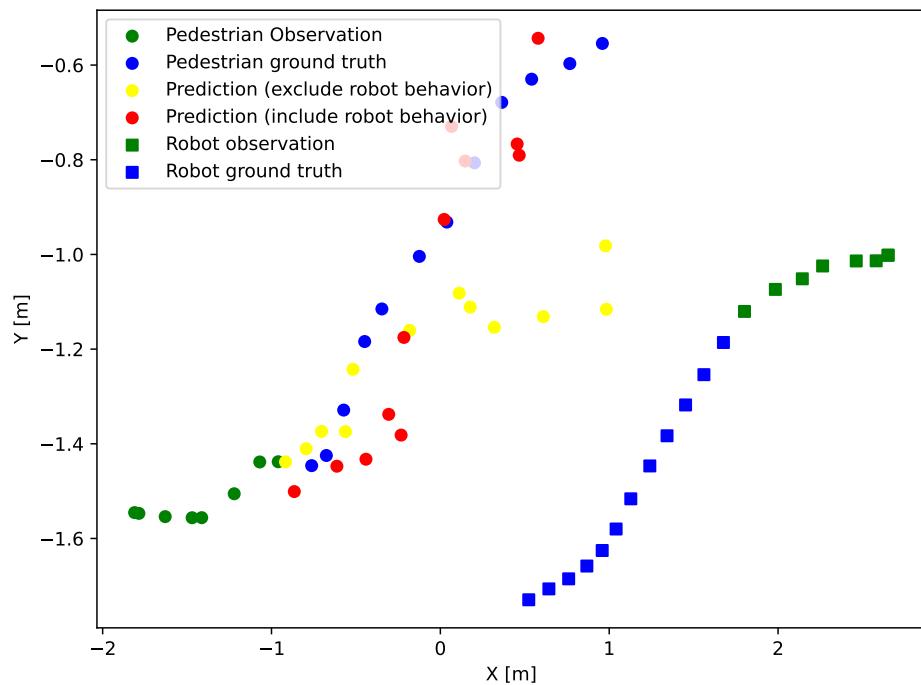


Fig. 4.6: Predicted trajectory when the robot moves left

第5章

予測結果のナビゲーションへの応用

5.1 本章の概要

本章では、ナビゲーションシステムの概要、実ロボットにおける歩行者の位置計算の詳細、およびナビゲーションにおける予測結果の利用方法について述べる。まず、5.2節でシステムの概要を示し、次に5.3節で実ロボットにおける歩行者の位置計算の詳細を説明する。最後に、5.4節でナビゲーションにおける予測結果の利用方法について述べる。

5.2 システム概要

Fig. 5.1に、本研究で構築したナビゲーションシステムの概要図を示す。本システムは以下に示す3つの主要モジュールで構成される。

- 制御モジュール

制御モジュールは、ROS Navigation Stack の主要コンポーネントである move_base により構成される。センサデータと目標位置を受け取り、適切な制御指令をロボットに送信する。

- 認識モジュール

認識モジュールは、YOLO を用いて歩行者の検出・追跡し、観測時間分のデータをまとめて時系列データとして出力する。

- 予測モジュール

予測モジュールは、認識モジュールから受け取ったデータをもとに、3章で述べたネットワークを用いて歩行者の位置を予測する。その後、制御モジュールの global_costmap の独自レイヤで予測結果をコストマップに反映させる。

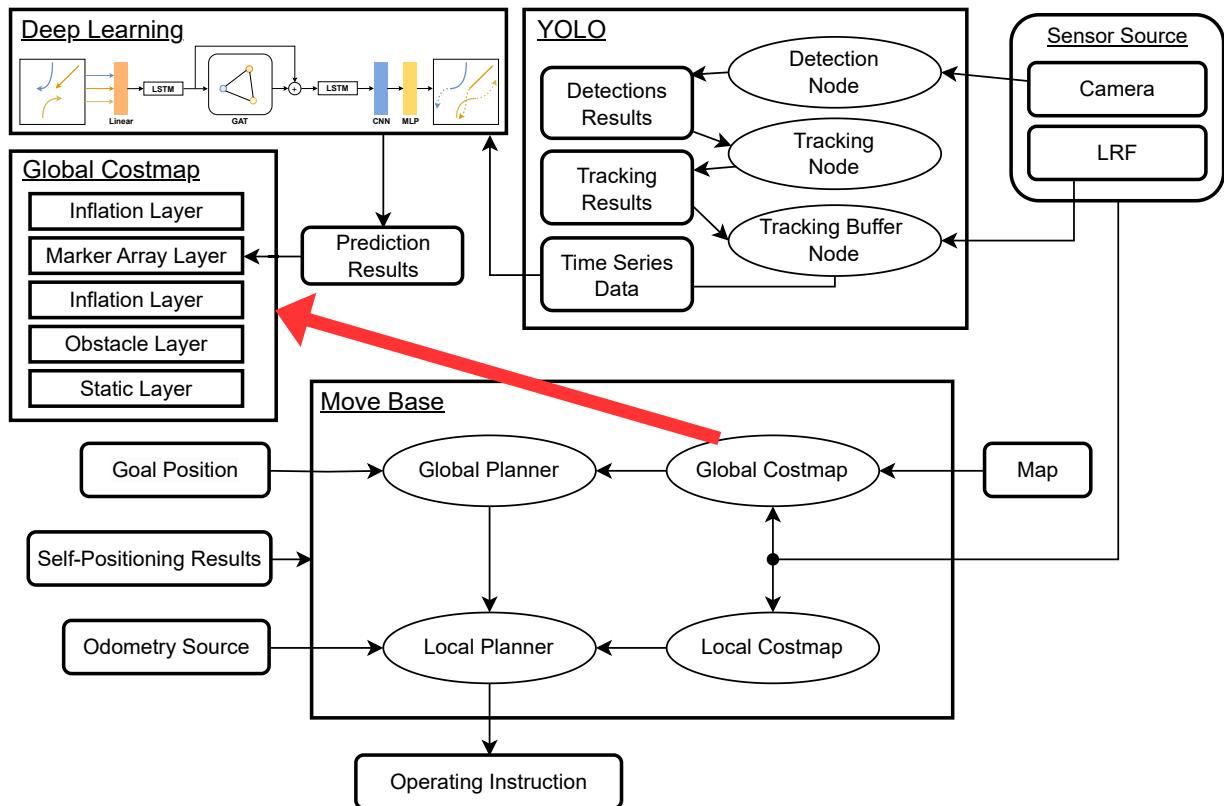
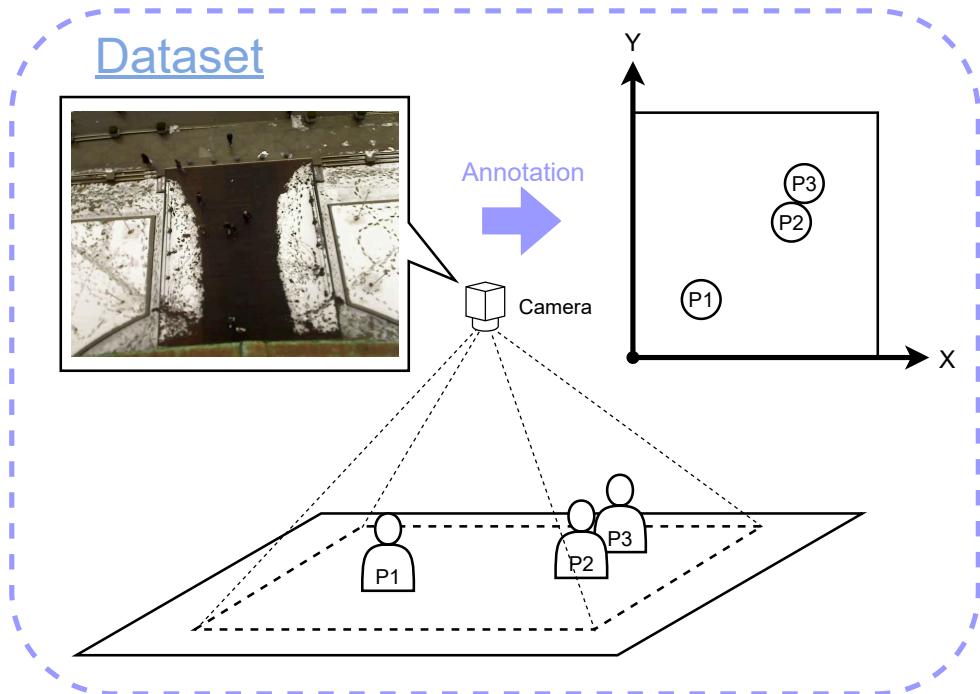


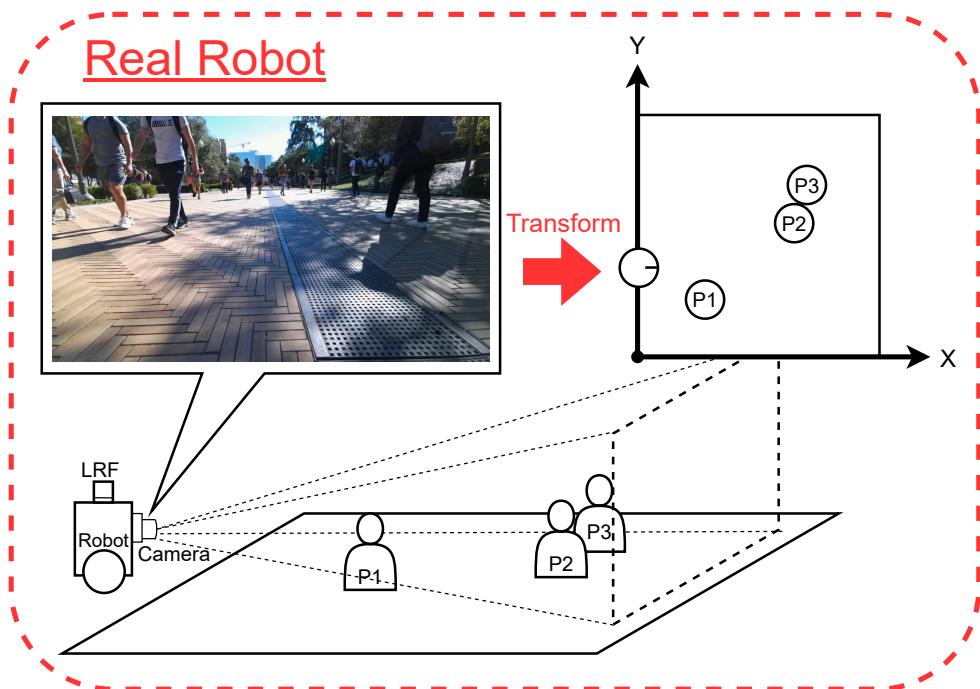
Fig. 5.1: Navigation System Overview

5.3 実ロボットにおける歩行者の位置計算

Fig. 5.2 に、俯瞰視点のデータセットと実ロボット上での歩行者位置推定のイメージを示す。ここで、データセットとは ETH[29] や UCY[30] データセットのように、俯瞰視点のデータセットのことを指す。Fig. 5.2a のように、データセットは俯瞰視点の映像と各時刻における歩行者のワールド座標系上の位置が含まれている。しかし、実ロボットで歩行者位置を推定する際は、ロボットがセンサを通じて取得した情報のみを用いてワールド座標系の位置を算出しなければならない。Fig. 5.2b の例のように、1人称視点で撮影した画像中の歩行者の位置を座標変換することで、ワールド座標系上の位置を得る。これにより、データセットと同じ形式の歩行者の位置データが得られ、データセットで学習したモデルを再学習することなく実ロボット上に適用できる。



(a) Pedestrian Position Calculation on Dataset



(b) Pedestrian Position Calculation on Real Robot

Fig. 5.2: Pedestrian Position Calculation: (a) on Dataset, (b) on Real Robot

本研究での実ロボットにおける歩行者の位置計算の流れと詳細を以下に示す。

1. RGB カメラ画像と LRF データを取得

ロボット前方に取り付けた RGB カメラから歩行者の画像を取得し、同時に LRF (Laser Range Finder) から周囲の距離データを得る。いずれも歩行者の位置推定に必要となる情報を提供する。

2. YOLO で人間検出

YOLO[16] は、高速・高精度な物体検出アルゴリズムであり、RGB カメラ画像から歩行者を検出する。本研究では、YOLOv9[39] の学習済みモデル (yolov9c.pt) を使用し、画像上で歩行者の位置を特定する。

3. 検出した個体の画像上での角度を計算

検出した歩行者の画像上での位置をもとに、式 (5.1) に従いロボットカメラの視野内における角度 θ_{camera} を求める。

$$\theta_{camera} = -\frac{(x_{center} - \frac{w_{img}}{2}) \cdot fov_{horizontal}}{W_{img}} \quad (5.1)$$

ここで、 w_{img} は画像の幅、 $fov_{horizontal}$ はカメラの水平視野角、 x_{center} が検出バウンディングボックスの中心の x 座標である。計算後、 θ_{camera} を $-\pi$ から π の範囲に正規化する。

4. 計算した角度と LRF データからロボットとの相対位置を計算

画像上で得られた角度情報に対応する LRF データの要素を取り出す。そして、そのデータを用いて歩行者とロボットとの相対位置を計算する。LRF はロボットからの距離情報を提供し、歩行者の正確な位置を特定するために使用される。

5. 相対位置をワールド座標へ座標変換

最後に、得られた相対位置をワールド座標系に変換する。この座標変換は、ロボットのナビゲーションシステムの map フレームに基づいて行われる。ワールド座標系での位置が特定されることで、ロボットは歩行者の位置を正確に把握し、適切なナビゲーションを行うことができる。

5.4 ナビゲーションにおける予測結果の利用方法

Fig. 5.3 に示すように、グローバルコストマップに独自のレイヤとして Marker Array Layer を追加した。このレイヤでは、0.4 秒ごとに受け取る予測結果をコストマップへ反映する。3.3.3 節で述べたとおり、予測結果のデータ形式はガウス分布を構成する 5 次元の要素である。本研究では、そのガウス分布からサンプリングした軌道を予測結果としてコストマップに反映する。Fig. 5.4 のように、各レイヤでの処理を積み重ねて、最終的なグローバルコストマップを生成する。グローバルコストマップを拡張したのは、2.1 項で述べたように、多くのロボットで使用されている Navigation Stack で予測結果を容易に利用できるようにするためである。

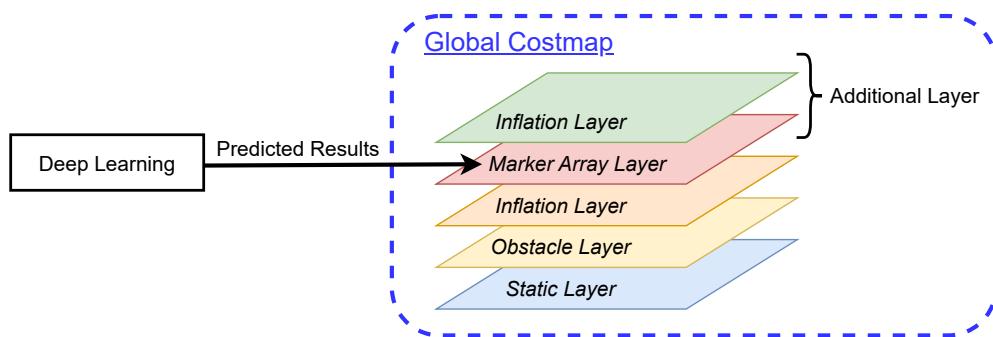


Fig. 5.3: Global Costmap in This Study

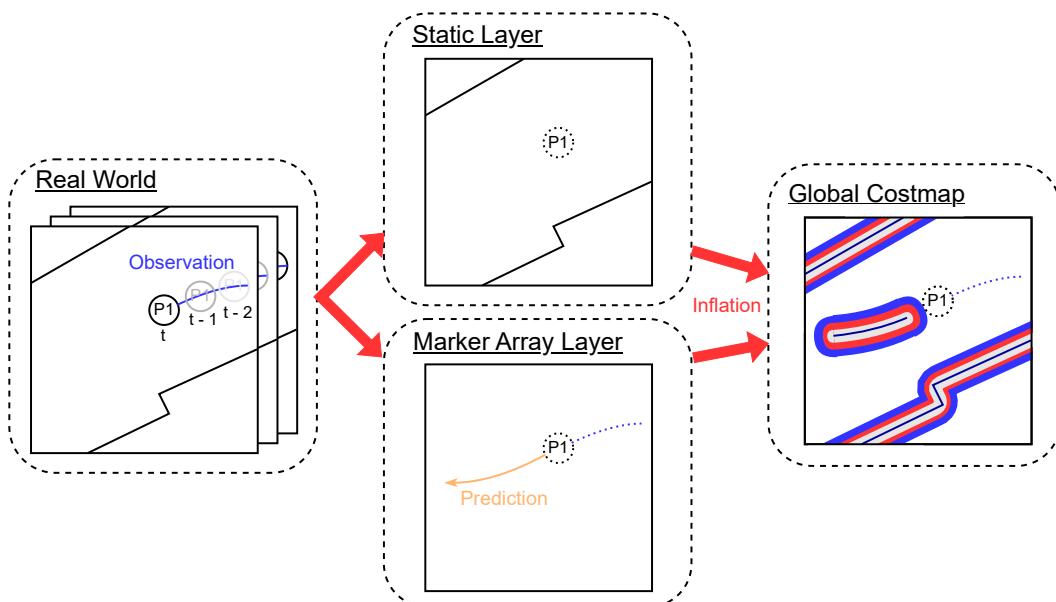


Fig. 5.4: Process each layer to generate a global costmap

第6章

予測結果を利用したナビゲーション の実験

6.1 本章の概要

本章では、移動ロボットのナビゲーションで予測結果を利用する実験を行い、その結果を評価する。まず、実験の概要と方法について説明し、次に実験シナリオと使用したシミュレータ環境について述べる。最後に、実験結果を基に考察を行う。

6.2 実験概要

予測結果を応用したナビゲーションシステムによって、移動ロボットの挙動がどのように変化するかを観察・評価するための実験を行う。実験では、2つのナビゲーションシナリオで移動ロボットを自律走行させ、各シナリオにつき10回ずつ走行を実施してデータを取得した。

6.3 実験方法

本実験の予測には、3章で提案したネットワークを用いた。ただし、使用したモデルはホールドアウト(Hold Out)方式で学習を行った。それ以外は同様の学習条件である。また、5章で述べたシステムを用いて、ナビゲーションに予測結果を適用した。

実験は Fig. 6.1 に示すように、Gazebo[40] のシミュレータ環境上で行う。また、Fig. 6.2a

に示すように，シミュレータで再現した移動ロボット（ORNE-box2[38]）を用い，歩行者は Fig. 6.2b に示す Gazebo のプラグイン [41] を使用してシミュレータ内に配置した。

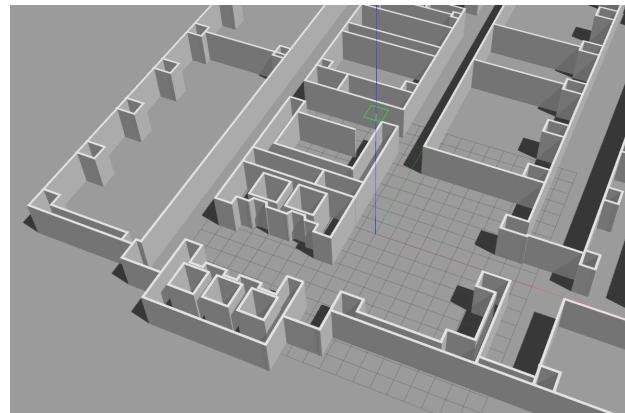


Fig. 6.1: Simulator Environment

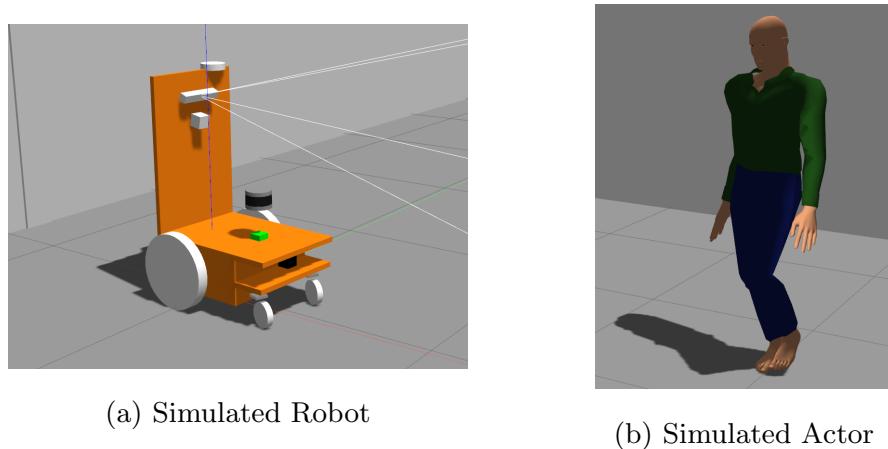
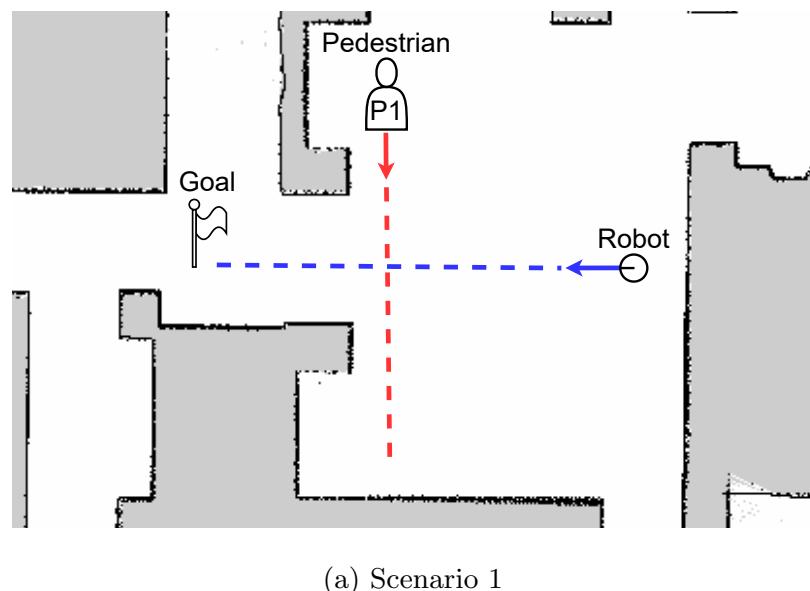


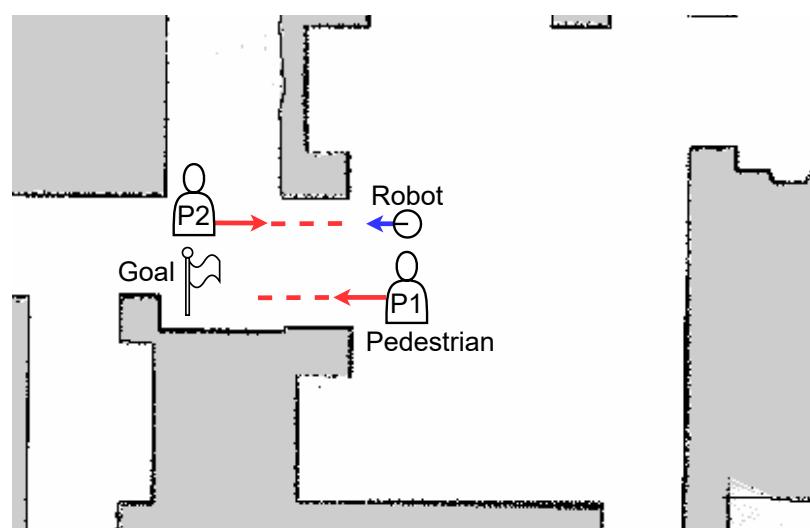
Fig. 6.2: Simulated Robot and Actor

本研究で用いた移動ロボットの自律走行ソフトウェアおよびパラメータ設定は，つくばチャレンジ 2024EX @イーアスつくば [42]において，千葉工業大学 未来ロボティクス学科 box2 チームが完走した際の構成を参考にしている（Github[43] で公開）。なお，後述のシナリオには狭い通路を走行する箇所が含まれているため，yaw 方向の角速度を小さくなるように調整している。

実験は Fig. 6.3 に示す 2 種類のシナリオで行った。Fig. 6.3a のシナリオ 1 では、ロボットが直線経路を進む途中に歩行者が横断する状況を設定した。Fig. 6.3b のシナリオ 2 では、ロボットが狭い通路を走行する際に 2 名の歩行者がすれ違う状況を設定し、より複雑な環境下でのナビゲーションを想定している。



(a) Scenario 1



(b) Scenario 2

Fig. 6.3: Experiment Scenarios

評価は以下の4項目に基づいて行った。

- 走行時間
- 走行距離
- 歩行者とロボット間の最小距離
- 最小の Time to Collision (TTC)

ナビゲーションの効率性は走行時間・距離が小さいほど高く、安全性は最小距離・最小 TTC が大きいほど高いと考えられる。TTC は、移動物体同士が現在の速度と進行方向を維持した場合の衝突するまでの時間を示し、以下の式で計算される。

$$\mathbf{v}_{robot} = \begin{bmatrix} v_{robot,x} \\ v_{robot,y} \end{bmatrix}, \quad \mathbf{v}_{ped} = \begin{bmatrix} v_{ped,x} \\ v_{ped,y} \end{bmatrix} \quad (6.1)$$

$$\mathbf{v}_{relative} = \mathbf{v}_{robot} - \mathbf{v}_{ped} \quad (6.2)$$

$$TTC = \frac{d}{\|\mathbf{v}_{relative}\|} \quad (6.3)$$

ここで、 d はロボットと歩行者の距離、 \mathbf{v}_{robot} はロボットの速度ベクトル、 \mathbf{v}_{ped} は歩行者の速度ベクトルである。

6.4 結果と考察

Fig. 6.4, Fig. 6.5 は、予測結果を用いないナビゲーションとロボットの行動を考慮した予測結果を用いるナビゲーションの比較結果である。棒グラフは平均値、エラーバーは標準偏差を表す。まず、Fig. 6.4 に示すシナリオ 1 の結果を比較すると、走行時間が約 1.8 倍、走行距離が約 1.01 倍と悪化した一方で、最小距離が約 13 倍、最小 TTC が約 17 倍と大幅に改善した。次に、Fig. 6.5 に示すシナリオ 2 の結果では、走行時間が約 1.5 倍、走行距離が約 1.01 倍、歩行者 2 の最小距離が 0.9 倍と悪化したが、歩行者 1 の最小距離が 1.84 倍、歩行者 1 の最小 TTC が約 1.3 倍、歩行者 2 の最小 TTC が約 1.1 倍と改善した。これらの結果から、予測結果を用いることで、ロボットのナビゲーションがより安全になる可能性があることが示唆される。一方で、効率性の面では課題が残ることも確認された。

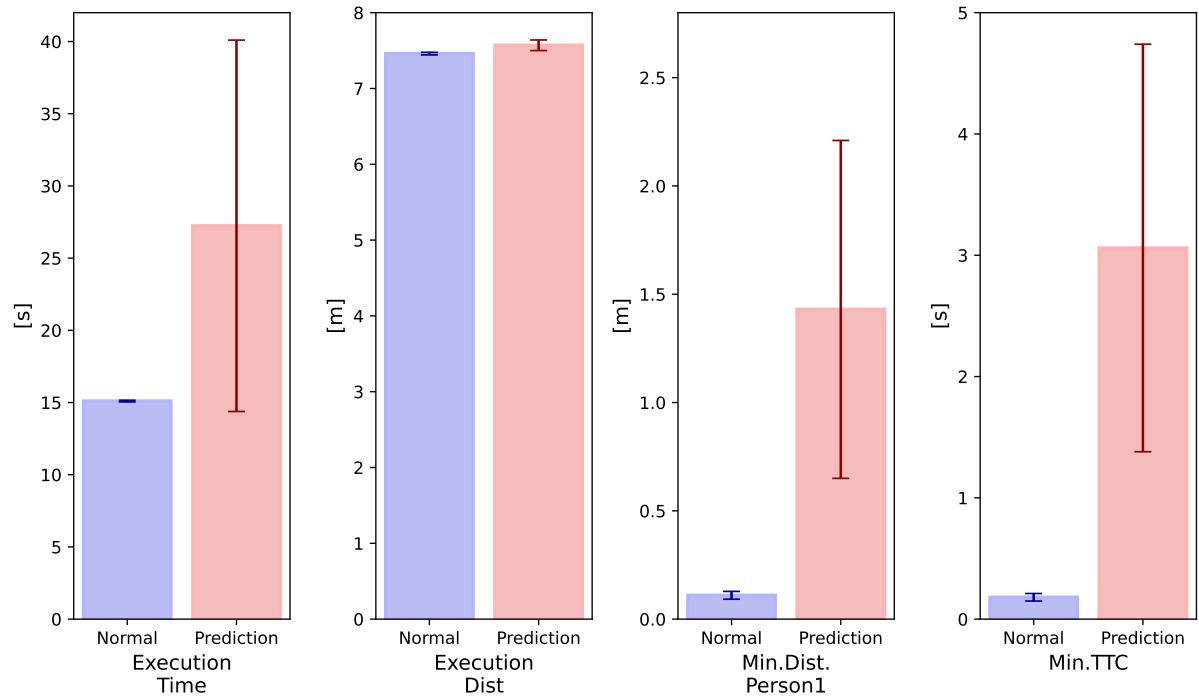


Fig. 6.4: Scenario1 result

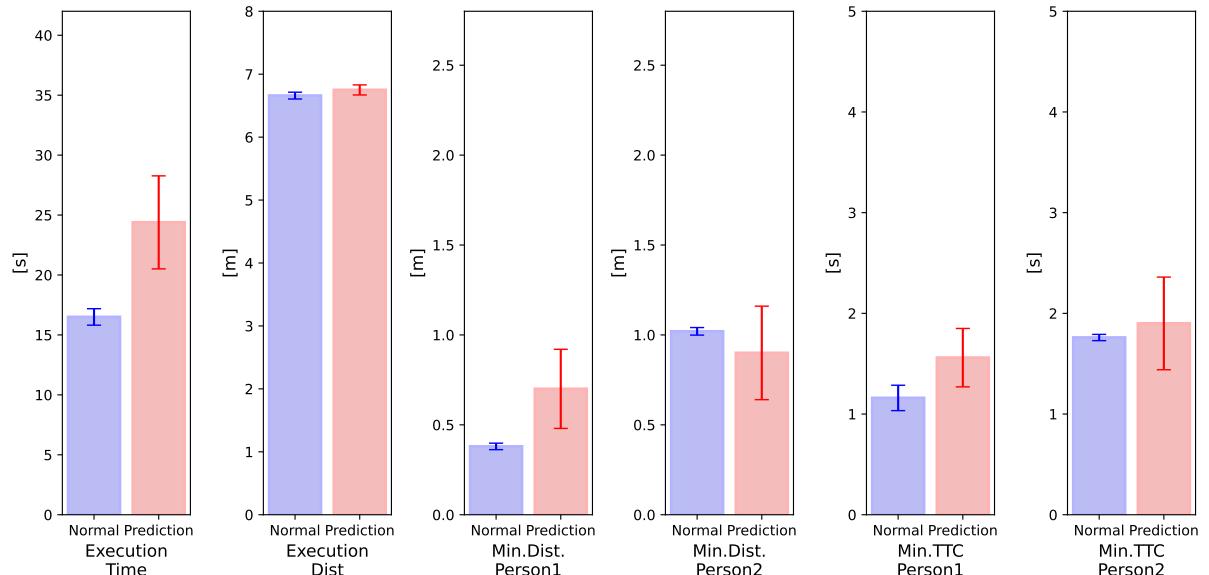


Fig. 6.5: Scenario2 result

シナリオ1では、歩行者との最小距離と最小TTCが大幅に改善されており、予測結果を活用することで早期に安全に停止を行い、歩行者との接触リスクを低減できる一方、回避行動に伴う走行時間・距離の増加が生じたと考えられる。つまり、安全性が向上する一方で、効率性の面では課題が残るような結果となった。

シナリオ2では、歩行者1に対する最小距離と最小TTCが改善されているが、歩行者2との最小距離が悪化している。これは、狭い通路でのすれ違い時にロボットが歩行者2に対してより接近する状況が発生したことを意味している。この結果から、狭い空間でのナビゲーションにおいては、予測結果の精度には改善の余地があると考えられる。また、本研究の学習に用いたデータセットはいずれも屋外の広い空間のデータであり、狭い環境での学習サンプルが不足していたことも一因と考えられる。

本実験で用いたシミュレータの歩行者は、一定の線形軌道を常に一定速度で歩行しており、現実の歩行者の動きとは大きく異なっている。つまり、歩行者同士や歩行者とロボット間など、全ての移動体の間に相互作用が存在しない。その結果、相互作用を重視して学習したモデルと実験環境の性質が乖離し、予測性能が低下する可能性がある。さらに、予測結果を用いる場合の標準偏差が大きい要因として、以下の点が考えられる。

- 正規分布からサンプリングされた軌道が実際の軌道と大きく異なる場合がある
- 0.4秒ごとに反映される予測結果がロボットの経路計画にチャタリングを引き起こす
- 前後の予測結果（例えば、 $t - 1$ と t ）に一貫性がない

これらの課題を解決するためには、狭い空間など多様な環境、相互作用を含むデータセットの活用や、提案したナビゲーションシステムの予測結果の扱い方の改善などが今後の検討課題となる。

第7章

結論

7.1 まとめ

本研究では、移動ロボットのための深層学習を用いた歩行者の位置予測と、そのナビゲーションへの応用を検討した。具体的には、ロボットの行動を考慮した歩行者の軌道予測手法を提案し、その有効性をシミュレーションおよび実環境での実験を通じて評価した。

提案手法では、グラフアテンションネットワークを用いて歩行者間およびロボットとの相互作用をモデル化し、将来の歩行者の位置を高精度に予測できることを確認した。シミュレーション実験において、予測結果をナビゲーションに組み込むことで、ロボットの安全性が向上することが確認された。一方で、走行時間や走行距離といった効率性の面では課題が残ることが明らかになった。

実環境での実験では、ロボットの行動を考慮することで、歩行者の軌道予測の精度が向上することが示された。特に、ロボットの動きが歩行者の最終的な位置に大きな影響を与えることを示す結果が得られた。以上の結果から、提案手法は移動ロボットのナビゲーションにおいて有用である可能性があると考えられるが、さらなる効率性の向上に向けた手法の改良が課題として残る。

7.2 今後の展望

参考文献

- [1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.
- [2] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, Vol. 323, No. 6088, pp. 533–536, 1986.
- [3] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, Vol. 71, No. 599-607, p. 6, 1986.
- [4] S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.
- [5] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [6] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, Vol. 1050, No. 20, pp. 10–48550, 2017.
- [7] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12085–12094, 2019.
- [8] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting

- using bicycle-gan and graph attention networks. *Advances in neural information processing systems*, Vol. 32, , 2019.
- [9] Abdulla Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14424–14432, 2020.
- [10] 丹野壮一郎, 田村雄介, 平田泰久. 移動ロボットが選択する将来の軌道が歩行者に与える影響を考慮した歩行者軌道予測. 第 24 回計測自動制御学会システムインテグレーション部門講演会, pp. 2423–2425, 2023.
- [11] ros planning. Github - ros-planning/navigation: Ros navigation stack. code for finding where the robot is and how it can get somewhere else. <https://github.com/ros-planning/navigation>. (Accessed on 2024-12-16).
- [12] ros navigation. Github - ros-navigation/navigation2: Ros 2 navigation framework and system. <https://github.com/ros-navigation/navigation2>. [Online; accessed 2024-12-16].
- [13] 原祥堯, 萬礼応, 富沢哲雄, 伊達央, 大川一也, 大矢晃久. つくばチャレンジ 2023 全チームの技術動向調査. ロボティクス・メカトロニクス講演会 2024 講演論文集, pp. 1P1-N02(1)–1P1-N02(4), 2024.
- [14] Edsger W Dijkstra. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: his life, work, and legacy*, pp. 287–290. 2022.
- [15] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, pp. 100–107, 1968.
- [16] J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [17] Alec Radford. Improving language understanding by generative pre-training. 2018.
- [18] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, Vol. 1, No. 8, p. 9, 2019.

- [19] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, Vol. 33, pp. 1877–1901, 2020.
- [20] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.
- [21] Thomas Weinlein. Diplomarbeit im fach informatik.
- [22] J Schmidhuber. Deep learning in neural networks: An overview, 2015.
- [23] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [24] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
- [25] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, Vol. 36, No. 4, pp. 193–202, 1980.
- [26] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, Vol. 1, No. 4, pp. 541–551, 1989.
- [27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.
- [28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [29] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pp. 261–268. IEEE, 2009.
- [30] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In

- Computer graphics forum*, Vol. 26, pp. 655–664. Wiley Online Library, 2007.
- [31] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [32] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2255–2264, 2018.
- [33] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1349–1358, 2019.
- [34] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Conditional generative neural system for probabilistic trajectory prediction. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6150–6156. IEEE, 2019.
- [35] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5725–5734, 2019.
- [36] Lidan Zhang, Qi She, and Ping Guo. Stochastic trajectory prediction with social graph network. *arXiv preprint arXiv:1907.10233*, 2019.
- [37] 箕浦大晃, 平川翼, 山下隆義, 藤吉弘亘. Deep learning を用いた移動物体間のインタラクションを考慮した経路予測の研究動向. 電子情報通信学会論文誌 D, Vol. 105, No. 5, pp. 372–404, 2022.
- [38] 井口颯人, 樋高聖人, 野村駿斗, 村林孝太郎, 上田隆一, 林原靖男. 屋外自律移動ロボットプラットフォーム orne-box の開発 orne-box の検証・改良 . ロボティクス・メカトロニクス講演会講演概要集 2023, pp. 1P1–I06. 一般社団法人 日本機械学会, 2023.
- [39] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. In *European Conference on Computer Vision*, pp. 1–21. Springer, 2025.

- [40] Gazebo. <https://gazebosim.org/home>. [Online; accessed 2024-12-28].
- [41] Actors gazebo ionic documentation. <https://gazebosim.org/docs/latest/actors/>. [Online; accessed 2024-12-28].
- [42] つくばチャレンジ EX — つくばチャレンジ 2024. <https://tsukubachallenge.jp/2024/about/exra>. [Online; accessed 2024-12-28].
- [43] open-rdc/orne-box at TC_2024_EX. https://github.com/open-rdc/orne-box/tree/TC_2024_EX. [Online; accessed 2024-12-28].
- [44] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, Vol. 51, No. 5, p. 4282, 1995.
- [45] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, Vol. 65, No. 6, p. 386, 1958.

付録

謝辞

本研究を進めるにあたり，1年に渡り，熱心にご指導を頂いた林原靖男教授に深く感謝いたします。