フローベース生成モデルを利用したオフライン強化学習による動的環境下での移動ロボットナビゲーション

Mobile Robot Navigation in Dynamic Environments by Offline Reinforcement Learning using Flow-Based Generative Model

○正　松本耕平（九大）　　正　倉爪　亮（九大）

Kohei MATSUMOTO, Kyushu University, matsumoto@irvs.ait.kyushu-u.ac.jp
Ryo KURAZUME, Kyushu University

We propose a novel navigation method applying an offline reinforcement learning method based on Implicit Policy Constraint to mobile robot navigation in environments with pedestrians. The proposed method utilizes a flow-based generative model for the behavior policy, and the latent policy is trained using a method based on Advantage-Weighted Regression. The proposed method is evaluated in a simulation environment.

Key Words: Path Planning, Offline Reinforcement Learning, Pedestrian Avoidance

1　緒言

生活環境で活動するサービスロボットの実現には，歩行者が行き交う動的な環境での自律移動が不可欠である．このような環境での自律移動ロボットナビゲーションのために，これまでに深層強化学習を利用した手法が提案されている [1, 2, 3]．強化学習は学習時に，環境とのオンラインでのインタラクションが必要である．しかしながら，歩行者が行き交う環境での移動ロボットナビゲーションの問題設定では，実環境でのインタラクションを行うには危険が伴い，十分なデータを集めるためには多くの時間を要する．

一方，近年では，オフライン強化学習が活発に研究されている．オフライン強化学習は，探索によるデータ収集を行わず，静的なデータセットによって方策を最適化することを目的としており，環境とのインタラクションに課題を抱えるロボティクスへの応用が期待される．本研究では，歩行者が行き交う移動ロボットナビゲーションタスクに，オフライン強化学習を適用する．提案手法は，Implicit Policy Constraint [4] に基づく手法を採用し，行動方策としてフローベース生成モデルを用いる．また，潜在方策の学習手法として，Advantage-Weighted Regression [3, 5] をベースとした手法を利用する．

2　背景

本節では，本研究の背景となる手法について述べる．

2.1　オフライン強化学習

オフライン強化学習では，事前に収集された静的なデータセットのみを用いて学習を行う．通常の強化学習に必要となる探索が必要なくなるため，環境とのインタラクションを十分に行うことが難しいタスクには効果的である．オフライン強化学習を行う上で課題となるのは，分布外行動によって引き起こされる価値関数の外挿誤差である．データセットに存在していない遷移を引き起こす行動に対しては，価値関数は正常に価値を推定することができず，過大評価をしてしまう可能性がある．これを回避するために，方策に対して，データセット内の行動を生成するように制約を課す必要がある．方策に制約を課す方法は Explicit Policy Constraint (EPC) と Implicit Policy Constraint (IPC) の 2 つに分けられる [4]．EPC ではエージェントの方策の行動分布をデータセットの行動方策と明示的にマッチングさせる．一方，IPC では潜在行動空間を通じて，データセット内の行動を出力するように方策を暗黙的に制限する．

2.1.1　Implicit Policy Constraint

IPC の概念図を図 1 に示す．IPC に基づいてオフライン強化学習を行う場合，2 種類の方策，行動方策 (Behavior Policy) と潜在方策 (Implicit Policy) を 2 段階で学習する．先行研究 [4] では，行動方策には生成モデルの一種である Variational Autoencoder (VAE) が用いられ，データセットの行動を模倣するように学習される．一方，潜在方策は強化学習の方式に従って，行動方策が最適な行動を生成できる潜在変数を生成するように学習される．強化学習を行う過程で，行動方策外の行動を取らないため，エージェントがデータセットの分布外の行動を取ることを抑制できる．

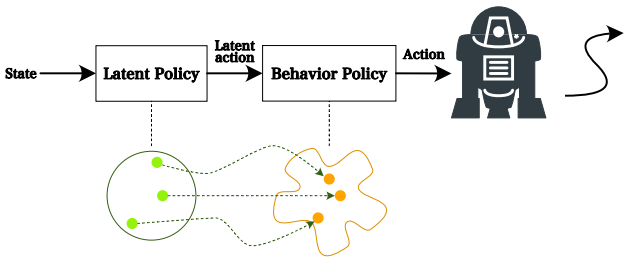

Fig.1 Conceptual diagram of Implicit Policy Constraint

2.2　フローベース生成モデル

フローベース生成モデルは正規分布から生成された潜在変数に対し，可逆な変換を重ねることで目標分布内のサンプルに変換する生成モデルである．他の生成モデルに比べて正確な尤度が計算できることや，可逆変換を用いることから，データから潜在変数への正確な変換が可能である特徴がある．図 2 にフローベース生成モデルの概念図を示す．　この図のように，フローベース生成

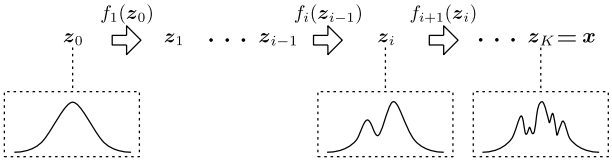

Fig.2 Conceptual diagram of flow-based generative model

モデルは潜在変数 $\boldsymbol{z}$ に可逆変換 f を繰り替えし適用することで，目標のサンプル $\boldsymbol{x}$ を生成する．目標のサンプル $\boldsymbol{x}$ に対して，以下の式で対数尤度を求めることができ，負の対数尤度を最小化することによって学習が行われる．

$$\log p(\boldsymbol{x}) = \log p(\boldsymbol{z}_0) - \sum_{i=1}^{K} \log \left| \det \frac{df_i}{d\boldsymbol{z}_{i-1}} \right| \qquad (1)$$

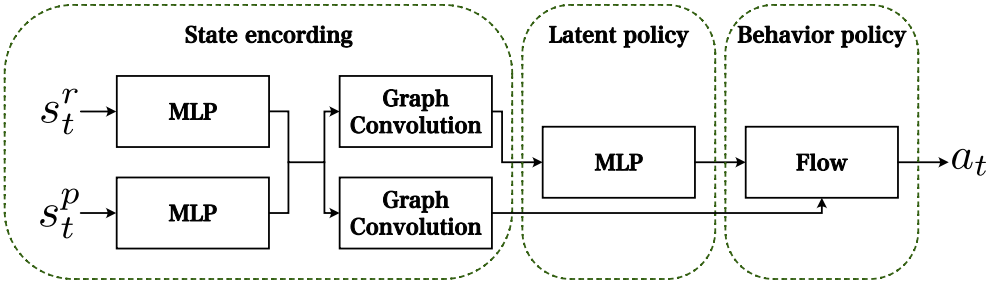

Fig.3 Architecture of policy in the proposed method

3 提案手法

3.1 状態，行動及び報酬の設定

本研究では，オフライン強化学習における，状態及び行動を以下のように設定する．

- 状態：歩行者とロボットの位置と速度データを状態として扱う．各歩行者，ロボットの観測はベクトル $(p_x^i, p_y^i, v_x^i, v_y^i)$ であり，i 番目の歩行者またはロボットにおいて，(p_x^i, p_y^i) は位置，(v_x^i, v_y^i) は速度を表す．
- 行動：本研究では，ホロノミックな全方位移動ロボットを想定し，2 次元空間におけるロボットの x 軸方向の入力速度 v_x と y 軸方向の入力速度 v_y からなる 2 次元ベクトル (v_x, v_y) を用いる．

また，報酬には式 (2) を用いる．d_t はロボットと周囲の歩行者間の最小距離を表し，p_t^r は時刻 t におけるロボットの位置，p_g はロボットの目標位置を示す．

$$R(s_t) = \begin{cases} -0.25 & \text{if } d_t < 0 \\ -0.1 + d_t/2 & \text{else if } d_t < 0.2 \\ 1 & \text{else if } p_t^r = p_g \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

3.2 モデルアーキテクチャ

提案手法は Actor-Critic の方式をとり，価値関数と方策関数をそれぞれ学習する．価値関数には Multilayer Perceptron (MLP) を用い，方策関数は潜在方策と行動方策の 2 種類に分けて学習する．図 3 に提案手法の方策のアーキテクチャを示す．ロボットの状態 s_t^r と歩行者の状態 s_t^p はグラフ畳み込みを用いて統合され，潜在方策と行動方策に入力される．潜在方策には MLP を利用する．グラフ畳み込みに用いる隣接行列 A_t のカーネルは，式 (3) に示すように，各位置の差分の L_2 ノルムの逆数に基づく [6].

$$a_t^{ij} = \begin{cases} 1/\left\|p_t^i - p_t^j\right\|_2 & \text{if } \left\|p_t^i - p_t^j\right\|_2 \neq 0 \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

さらに，式 (4) に示す正規化処理を行い $\tilde{A}_t$ を実際の計算に用いる．

$$\tilde{A}_t = \Lambda_t^{-\frac{1}{2}} \hat{A}_t \Lambda_t^{-\frac{1}{2}} \tag{4}$$

ここで，$\hat{A}_t = A_t + I$ であり，Λ_t は次数行列である．

行動方策には Affine Coupling Layer [7] を持つフローベース生成モデルを用いる．Affine Coupling Layer は入力ベクトル x を $x_{1:d}, x_{d+1:D}$ に分割し，$x_{1:d}$ を条件として，$x_{d+1:D}$ にスケーリングと定数オフセットの加算からなる変換を適用し，出力 y を得る．提案手法では，グラフ畳み込み後のロボットの特徴量よって条件付けを行うため，Affine Coupling Layer の順伝搬は式 (5) で表され，逆伝搬は式 (6) で表される．q^r はグラフ畳み込み後のロボットの特徴量を表す．また，変換内の関数 s と t には MLP を用いる．

$$\begin{cases} y_{1:d} & = x_{1:d} \\ y_{d+1:D} & = x_{d+1:D} \odot \exp(s(x_{1:d}, q^r)) + t(x_{1:d}, q^r) \end{cases} \tag{5}$$

$$\begin{cases} x_{1:d} & = y_{1:d} \\ x_{d+1:D} & = (y_{d+1:D} - t(y_{1:d}, q^r)) \odot \exp(-s(y_{1:d}, q^r))) \end{cases} \tag{6}$$

3.3 学習手法

この方策は，模倣学習ステップと強化学習ステップの 2 段階で行われる．模倣学習では，行動方策及び価値関数を学習する．模倣学習の手順をアルゴリズム 1 に示す．$\log p(\boldsymbol{a}_t|\boldsymbol{s}_t)$ は式 (1) を基に，式 (5) を $f_{1,\ldots,K}$ として用いることで計算される．

Algorithm 1: Imitation learning

Initialize the behavior policy f_p^b and the critic f_v
for $i = 1$ **to** E^{im} **do**
  Sample a minibatch $\mathcal{B} = \{(s_t, a_t, r_t^d, s_{t+1})_{i=1,\ldots,T}\}$ from dataset $\mathcal{D}$
  Update f_p^b by minimizing $L_{\text{flow}} = -\frac{1}{T}\log p(\boldsymbol{a}_t|\boldsymbol{s}_t)$
  Update f_v by minimizing $L_{\text{value}} = \text{MSE}(f_v(\boldsymbol{s}_t), \boldsymbol{r}_t^d)$
end

強化学習は Advantage-Weighted Regression [3, 5] を，提案するモデル向けに拡張した手法で行う．この学習手法をアルゴリズム 2 に示す．このアルゴリズムでは，フローベース生成モデル

Algorithm 2: Value-Weighted Regression with flow-based generative model

Initialize the latent policy f_p^l
for $i = 1$ **to** E^{rl} **do**
  Sample a minibatch $\mathcal{B} = \{(s_t, a_t, r_t^d, s_{t+1})_{i=1,\ldots,T}\}$ from dataset $\mathcal{D}$
  Generate latent actions using the latent policy $\boldsymbol{\psi}_t = f_p^l(\boldsymbol{s}_t)$
  Generate behavior actions using the behavior policy $\boldsymbol{a}_t^p = f_p^b(\boldsymbol{\psi}_t)$
  Generate target behavior actions using the behavior policy with the base distribution $\hat{\boldsymbol{a}}_t^p = f_p^b(\boldsymbol{\psi}_t^b)$
  Obtain immediate rewards $\boldsymbol{r}_t^{im}$ and next state $\hat{\boldsymbol{s}}_{t+1}$ of transitions with $\hat{\boldsymbol{a}}_t^p$
  Calculate weights for regression $W = Q(\boldsymbol{s}_t, \hat{\boldsymbol{a}}_t^p) = \boldsymbol{r}_t^{im} + \gamma f_v(\hat{\boldsymbol{s}}_{t+1})$
  Update f_p^l by minimizing $L_{\text{awr}} = \frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{a}_t^p - \hat{\boldsymbol{a}}_t^p)^2 W$
  Update f_v by minimizing $L_{\text{value}} = \text{MSE}(f_v(\boldsymbol{s}_t), \boldsymbol{r}_t^d)$
end

によって生成される行動に対して，価値関数によって価値を推定し，この価値を重みとして重み付き回帰学習を行うことで，潜在方策を学習する．γ は割引率であり，ψ_t^b は，正規分布よりサンプルされる．

4　シミュレーション実験

4.1　シミュレーション環境

シミュレーション実験では，CrowdNav 環境の circle crossing シナリオを利用する [1, 2]．このシナリオでは，ロボットは初期位置 $(x, y) = (0, -4)$ からゴール地点 $(x, y) = (0, 4)$ を目指して進む．歩行者は ORCA に従って行動し，初期化時に半径 4m の円上にランダムに配置される．評価は 500 パターンのテストケースを用いて行う．また，環境内の歩行者は 5 人に設定した．比較手法としては， Behavior Cloning (BC), フローベース生成モデルによる模倣学習 (Flow), TD3-BC [8] を用いる．データセットには，シミュレーション環境において，ロボットを ORCA に基づいて動作させ収集した，2000 シナリオ分の遷移情報を用いる．

4.2　実験結果

提案手法による結果の軌跡の一部を図 4 に，時間経過による動作の様子を図 5 に示す．また，提案手法と比較手法の成功率，衝突率，実行時間，平均収益を表 1 に示す．

Table 1 Numerical comparison of the proposed method and other methods

Method	Success [%]	Collision [%]	Exec. time [s]	Avg. return
BC	48.4	51.6	11.99	0.170
Flow	42.2	57.8	12.24	0.118
TD3-BC [8]	26.8	68.4	10.9	0.017
Proposed	57.0	42.8	12.19	0.241

この結果より，提案手法が最も成功率が高いことがわかる．BC と TD3-BC は実行時間において，提案手法を上回っているが，衝突回数が多いため，平均収益は提案手法を大きく下回る．また，Flow と提案手法を比較した場合，提案手法の方が成功率も高く，実行時間も短くなっていることがわかる．これによって，提案手法の強化学習ステップによって，模倣学習によって学習されたフローベースの方策の性能を改善できることが示された．

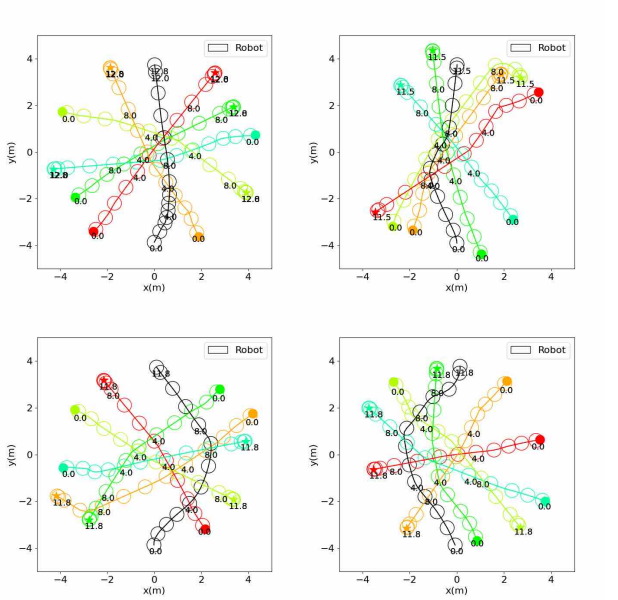

Fig.4 Sample trajectories of the results of the proposed method. The black trajectories describe the robot, and the colored trajectories describe pedestrians.

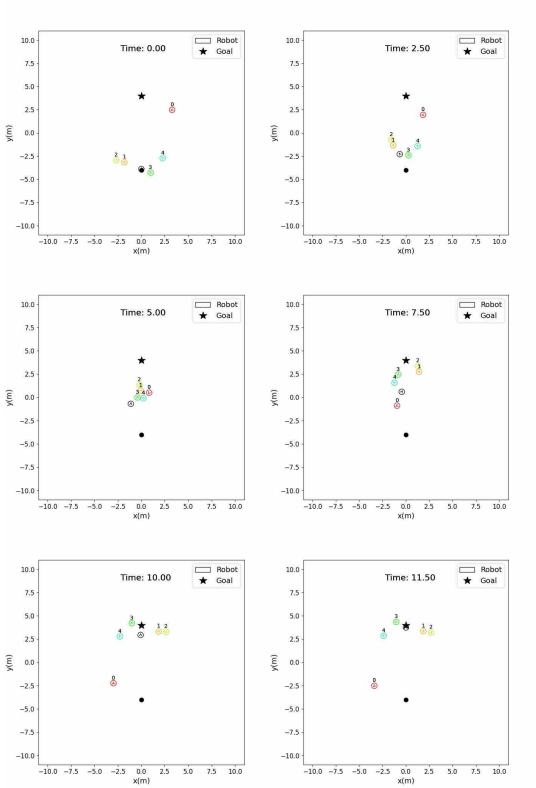

Fig.5 Movement of the robot over time with the proposed method. The black circles describe the robot, and the colored circles describe pedestrians.

5　まとめと今後の展望

本研究では，Implicit Policy Constraint に基づくオフライン強化学習を，歩行者が行き交う環境での移動ロボットナビゲーションに適用した．提案手法は，グラフ畳み込みを用いてロボットと歩行者の状態の情報を統合し，フローベース生成モデルによる行動方策と，Advantage-Weighted Regression をベースにした手法により学習された潜在方策によって行動生成を行う．また，シミュレーション環境によって実験を行い，提案手法の有効性を確認した．

本稿の結果において，提案手法は 6 割弱程度の成功率であったが，これは十分な性能ではない．今後は，より高い成功率を達成できるように，モデルや学習方法の改善に取り組んで行く．加えて，より多くのオフライン強化学習手法との比較や，ベンチマークタスクでの提案手法の評価，実環境での実験にも取り組んで行く．

謝辞

本研究の一部は，JSPS 科研費 JP20H00230 の助成を受けたものである．

参考文献

[1] Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6015–6022, 2019.

[2] Changan Chen, Sha Hu, Payam Nikdel, Greg Mori, and Manolis Savva. Relational graph learning for crowd navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10007–10013, 2020.

[3] Xueyou Zhang, Wei Xi, Xian Guo, Yongchun Fang, Bin Wang, Wulong Liu, and Jianye Hao. Relational Navigation Learning in Continuous Action Space among Crowds. In *Proceedings of the*

IEEE International Conference on Robotics and Automation (ICRA), pp. 3175–3181, 2021.

[4] Wenxuan Zhou, Sujay Bajracharya, and David Held. PLAS: Latent Action Space for Offline Reinforcement Learning. In *Proceedings of the Annual Conference on Robot Learning (CoRL)*, pp. 1719–1735, 2020.

[5] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. *CoRR*, 2019.

[6] Abduallah 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 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14412–14420, 2020.

[7] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[8] Scott Fujimoto and Shixiang Shane Gu. A Minimalist Approach to Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 20132–20145, 2021.