

# RSSM の実世界画像予測への応用

## Application of RSSM to Real World Image Prediction

楠井俊朗<sup>\*1</sup> 大谷真也<sup>\*2</sup> 高野剛志<sup>\*3</sup> 福田健人<sup>\*4</sup> 本田純也<sup>\*5</sup>  
Kusui Toshiro Otani Shinya Takano Tsuyoshi Fukuda Kento Honda Junya

<sup>\*1</sup>東京大学 The University of Tokyo <sup>\*2</sup>神戸大学 Kobe University <sup>\*3</sup>千葉大学大学院融合理工学部 Graduate School of Science and Engineering, Chiba University  
<sup>\*4</sup>放送大学 The Open University of Japan <sup>\*5</sup>熊本高等専門学校 National Institute of Technology (KOSEN), Kumamoto College

Most of the conventional RSSM research has used series images on a simulator as training data. Examples of training based on real-world videos include DayDreamer’s XArm task, but they assume a fixed target. In this study, we evaluate the performance of RSSM-based models in handling real-world objects with periodic behavior when the length of the period is changed. The task is to learn how to catch a toy fish with its mouth open on a periodically rotating disk using a simple arm that moves in the vertical direction. The model is designed to predict the position of a fish captured by a camera using RSSM, and to reward the user for catching the fish.

### 1. 研究背景・目的

PlaNet[Hafner 19] や Dreamer[Hafner 20] などの世界モデルでは、Recurrent State-Space Model (RSSM) という時系列深層生成モデルが用いられている。これらの研究では、世界モデルをシミュレータ上の系列画像データに基づいて学習しているが、実世界データに対してどの程度活用できるかについては十分に明らかにされていない。そこで本研究では、実世界からのカメラ画像データを訓練集合として、報酬を幾つか設計し、RSSM など世界モデルで用いられる時系列深層生成モデルを学習した結果について調査した。

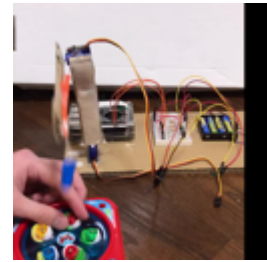


図 1: 実験装置

### 2. 関連研究

#### 2.1 RSSM の研究

深層マルコフモデル (RNN) は、マルコフ性を仮定したもつで時系列の状態を予測するモデルであり、一つ前の状態から学習する次の状態とこれまでの状態全てから学習する次の状態が等しくなるように学習をおこなっていく。RSSM は、RNN 及びガウス過程を組み合わせてモデル化することで、予測を高精度化したものである。

#### 2.2 実ロボットへの応用を考えた研究

DayDreamer[Wu 22] は、DreamerV2[Hafner 21] を実ロボットへ応用した研究であり、Xarm を用いたタスクは画像を入力として、DreamerV2 を用いたタスクの学習を行っている点で、本研究と共通点がある。Xarm のタスクは、報酬が疎であることから学習には長時間かかる。

### 3. 実験手法

本節では、実験装置と RSSM の実装内容について述べる。

#### 3.1 実験装置

周期的に回転する円盤上で口が開いている魚のおもちゃを、上下方向で動く簡易なアームを用いて釣り上げるというタスクを設計する。

連絡先: 氏名, 所属, 住所, 電話番号, Fax 番号, 電子メールアドレスなど

表 1 実験装置の詳細

実験装置	詳細
カメラ	iPhone 7
魚	ぐる〜んぐる〜ん魚釣り+100 円魚釣り
釣竿アーム	Raspberry Pi 4+マイクロサーボSG-90

魚の実験装置では、市販されている「ぐる〜んぐる〜ん魚釣り」というおもちゃを使用している。その魚の口の中に 100 円魚釣りに付属していたマグネットを入れている。「ぐる〜んぐる〜ん魚釣り」では魚の池が回転する速度を調節できる。今回、撮影した動画では池の回転速度は低速にして行った。また、魚の色については緑色と黄色の 2 色を使用している。これは、画像処理によって魚を検出しやすくするためにこの 2 色を採用している。

釣竿アームではサーボモータ 1 個を Raspberry Pi 4 Model B で制御している。制御では、1 自由度のアームが周期的に上下運動するようにしている。また、アーム自体は簡易的にダンボールで作成し、アーム部分にマグネットつきのオレンジ釣竿を取り付けている。そして、アーム先端の部分にオレンジ色の補色となる青色の目印を取り付けている。これは、魚の口の中に目印となる点を打ってそのずれ等を計測するためである。

カメラは iPhone7 を使用して撮影を行った。撮影する画面に「魚」と「釣竿アーム」の両方が入るように調整した。また、背景には白いダンボールを使用して背景の影響で結果に影響が生じないように設定した。また、撮影時の照明も白色で撮影す

るように設定した。これらは魚と釣竿アーム以外の要因で結果に影響を及ぼさないようにする工夫である。

### 3.2 RSSM の実装

RSSM で将来の魚釣りアームの軌道及び、魚の位置を予測する。未来予測をした想像上の軌道上で、Dreamer の Actor-Critic を実行し、魚釣りアームの行動モデルを更新する。

#### 3.2.1 学習

魚釣りタスクにおける観測は、魚釣りアームの軌道と回転する魚の動画となる。魚釣りアームの軌道は、上下移動のみで 1 次元ベクトルで表される。魚釣りアームの軌道は行動  $a_{t-1}$ 、行動を受けた魚釣りアームの軌道と回転する魚の動画は観測値  $o_t$  として、それぞれ RSSM に入力される。RSSM は、現在までの観測値  $o_{1:t}$  と行動  $a_{1:t-1}$  から決定論的な RNN の内部状態  $h_t$  を獲得し、 $h_t$  で条件される確率分布  $p(s_t|h_t)$  (prior) と、 $h_t$  と観測値  $o_t$  で条件される確率分布  $p(s_t|h_t, o_{t-1})$  (posterior) を得る。RSSM の誤差は 3 種類ある。一つ目は prior と posterior の KL ダイバージェンス、二つ目は  $p(s_t|h_t)$  から再構成される観測値  $o'_t$  と実際の観測値  $o_t$  の誤差、三つ目は現在の状態から推測した報酬と実際の報酬の誤差である。これら三つの誤差を最小化することで、RSSM は将来の魚釣りアームの軌道及び魚の位置を予測する。Dreamer の Actor-Critic による学習は、RSSM で予測した将来の状態を用いて行われる。この将来の状態は、魚釣りアームの軌道と回転する魚の動画である。想像上の将来状態で魚釣りアームの行動を行動モデルから取得する。その行動から得られる想像上の報酬を最大化するように行動モデルを学習する。

#### 3.2.2 実世界の予測

学習によって得られたモデルが実世界をどのように予測しているのか可視化することを試みた。

ここでは、open-loop による未来予測を行う。すなわち、予測の起点として実世界で収集したデータを与え、モデルに世界の次の状態を予測させ、以降は実世界の観測に基づくフィードバックを与えず、自身の予測結果のみを利用して、繰り返して次の状態を予測させる。この際、状態の予測を意図していたため、報酬に行動がよらないように報酬は状態によらずランダムに与えた。

可視化の実装は以下の通りである。

1. 実験装置で撮影して得られた動画を隠れ状態に順次蓄積する
2. 蓄積された隠れ状態を元にして次の状態を予測する
3. 予測した状態を入力値（動画の 1 フレーム）として復元して可視化する
4. 予測結果を隠れ状態に蓄積し、再び次の状態を予測する（2 に戻る）

今回は 300 フレームの未来予測を行わせた。その 1 フレームを図 2 に示す。

予測した状態を連続して表示すると、「ぐる～んぐる～ん魚釣り」の回転する池とその周辺や、実験装置のアームがときおり不規則に動いている様子が観察できる。このことから、本モデルは世界の変化しうる部分および変化の様相を一定程度学習できていると考えられる。

一方で、魚の回転する移動パターンは十分に予測できておらず、さらなる予測精度向上の余地があることも確認できた。



図 2: open-loop による予測結果

#### 3.2.3 報酬を設計したもとの実世界の予測

魚釣りタスクにおいて、アームによって魚が釣り上がった際に報酬を与えるようにすることを考える。本研究では、以下の 3 つの報酬を設計し、未来予測を行った。3 つの設計は以下のとおりである。

1. アームの先端と魚の口の  $\ell_2$  距離を報酬にする
2. 魚が持ち上がり、池の上から画像中央部にいるときに報酬を与える
3. 1 と 2 を組み合わせた報酬を与える

1～3 についても 3.2.2 節と同様に、300 フレームの未来予測を行わせた。その結果を、順に図 3,4,5 に示す。



図 3: 報酬 1 による予測結果

図 3 においては、図 2 に比べてアームが大きく下がっていることが分かる。これは、アームが低い位置にある方がアームと魚の位置を近づき報酬が大きくなるためだと考えられる。図 4 においては、魚が持ち上がった状態で報酬を付与しているた

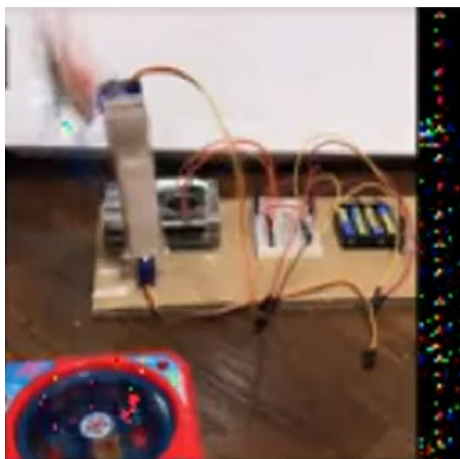


図 4: 報酬 2 による予測結果

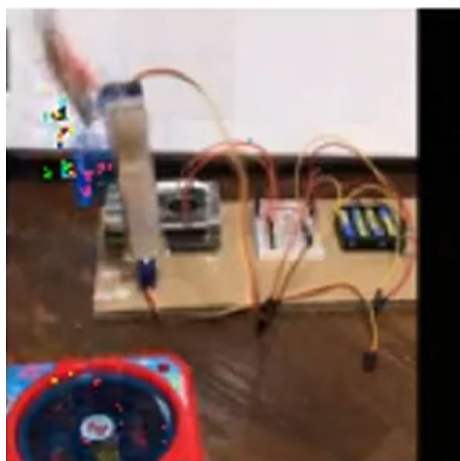


図 5: 報酬 3 による予測結果

め池にいる魚の数が減っていることが分かる．図 5 においては，アームの先端に魚がついているように見え，他の手法に比べ一番魚を釣れているように思われる．このことから 1 と 2 を組み合わせた報酬を用いた手法で魚釣りタスクの学習を行うと，他の手法に比べてより魚を釣り上げやすくなるのではないかと考えられる．

## 4. まとめ

### 4.1 実験装置

本プロジェクトでは，魚釣りタスクにおいて RSSM を実装した．RSSM の実世界画像予測を用いて制御まで行うことが理想だったが，魚釣りの制御の実装までは至らなかった．しかし，関連研究にある XArm タスクにおいて魚釣りという周期性のある新たなタスクを設定し，実世界画像予測まで行うことができた．この部分が本プロジェクトで先行研究から進歩した点である．

本プロジェクトではサーボモータ 1 個を使用した 1 自由度のタスクを設定したが，拡張としてサーボモータを増やして 2 自由度での制御が実現できると考えている．2 自由度にすることで上下運動に加えて左右運動も制御することができ，より釣竿アームを柔軟に動作させることができる．

実験装置では，魚の色や釣竿アームの色が RSSM を実装する上で障壁となった．本プロジェクトでは，簡易的に魚と釣竿アームに調整を行い対処したが，理想的には魚の色を全部統一し，釣竿アームもプラスチックや木などの丈夫な材料を使用することが望ましい．

### 4.2 RSSM の実装

RSSM において学習する画像の解像度を  $128 \times 128$  にしたのもあり，比較的高画質の予測が可能となり，小さな魚を対象とした学習ができた．画像の予測タスクにおいてどのような報酬設計によって魚が釣れやすくなりそうかが分かったため，今後は魚釣りタスクの学習において本研究で設計した報酬を適用することで魚がより釣れやすくなるかの検証を行いたい．

## 参考文献

- [Hafner 19] Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J.: Learning latent dynamics for planning from pixels, ICML, Vol. 97, pp. 2555–2565 (2019)
- [Hafner 20] Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M.: Dream to control: Learning behaviors by latent imagination, ICLR (2020)
- [Hafner 21] Hafner, D., Lillicrap, T., Norouzi, M., and Ba, J.: Mastering atari with discrete world models, ICLR (2021)
- [Wu 22] Wu, P., Escontrela, A., Hafner, D., Goldberg, K., and Abbeel, P.: DayDreamer: World models for physical robot learning, CoRL (2022)