

# 強化学習を用いた ボース・アインシュタイン凝縮体の制御

電気通信大学 斎藤弘樹

# はじめに (自己紹介、この研究を始めた動機など)

もともとの専門は冷却原子気体、量子流体

最近、機械学習の量子多体問題への応用に興味を持つ

Carleo&Troyer, Science **355**, 602 (2017)

〔 本セミナー第6回 吉岡信行さん  
第8回 野村悠祐さん 〕

HS, JPSJ **86**, 093001 (2017)

HS and M. Kato, JPSJ **87**, 014001 (2018)

HS, JPSJ **87**, 074002 (2018)

N. Irikura and HS, Phys. Rev. Res. **2**, 013284 (2020)

強化学習にも興味 (AlphaGo, ...)

冷却原子気体の分野に応用できないか？

➡ HS, JPSJ **89**, 074006 (2020)

“Creation and manipulation of quantized vortices in Bose-Einstein condensates using reinforcement learning”

# 内容

- 強化学習とは
- ボース・アインシュタイン凝縮体と量子渦
- 強化学習を用いた量子渦生成

# 強化学習の応用



## ARTICLE

doi:10.1038/nature16961

### Mastering the game of Go with deep neural networks and tree search

David Silver<sup>1\*</sup>, Aja Huang<sup>1\*</sup>, Chris J. Maddison<sup>1</sup>, Arthur Guez<sup>1</sup>, Laurent Sifre<sup>1</sup>, George van den Driessche<sup>1</sup>, Julian Schrittwieser<sup>1</sup>, Ioannis Antonoglou<sup>1</sup>, Veda Panneershelvam<sup>1</sup>, Marc Lanctot<sup>1</sup>, Sander Dieleman<sup>1</sup>, Dominik Grewe<sup>1</sup>, John Nham<sup>2</sup>, Nal Kalchbrenner<sup>1</sup>, Ilya Sutskever<sup>2</sup>, Timothy Lillicrap<sup>1</sup>, Madeleine Leach<sup>1</sup>, Koray Kavukcuoglu<sup>1</sup>, Thore Graepel<sup>1</sup> & Demis Hassabis<sup>1</sup>

Nature **529**, 484 (2016)

## ARTICLE

doi:10.1038/nature24270

### Mastering the game of Go without human knowledge

David Silver<sup>1\*</sup>, Julian Schrittwieser<sup>1\*</sup>, Karen Simonyan<sup>1\*</sup>, Ioannis Antonoglou<sup>1</sup>, Aja Huang<sup>1</sup>, Arthur Guez<sup>1</sup>, Thomas Hubert<sup>1</sup>, Lucas Baker<sup>1</sup>, Matthew Lai<sup>1</sup>, Adrian Bolton<sup>1</sup>, Yutian Chen<sup>1</sup>, Timothy Lillicrap<sup>1</sup>, Fan Hui<sup>1</sup>, Laurent Sifre<sup>1</sup>, George van den Driessche<sup>1</sup>, Thore Graepel<sup>1</sup> & Demis Hassabis<sup>1</sup>

Nature **550**, 354 (2017)

# 強化学習の応用

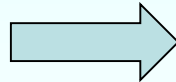
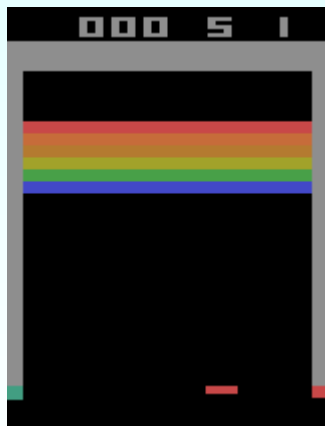
## LETTER

doi:10.1038/nature14236

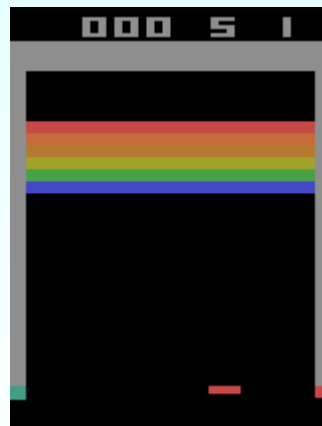
### Human-level control through deep reinforcement learning

Volodymyr Mnih<sup>1\*</sup>, Koray Kavukcuoglu<sup>1\*</sup>, David Silver<sup>1\*</sup>, Andrei A. Rusu<sup>1</sup>, Joel Veness<sup>1</sup>, Marc G. Bellemare<sup>1</sup>, Alex Graves<sup>1</sup>, Martin Riedmiller<sup>1</sup>, Andreas K. Fidjeland<sup>1</sup>, Georg Ostrovski<sup>1</sup>, Stig Petersen<sup>1</sup>, Charles Beattie<sup>1</sup>, Amir Sadik<sup>1</sup>, Ioannis Antonoglou<sup>1</sup>, Helen King<sup>1</sup>, Dharshan Kumaran<sup>1</sup>, Daan Wierstra<sup>1</sup>, Shane Legg<sup>1</sup> & Demis Hassabis<sup>1</sup>

Nature **518**, 529 (2015)



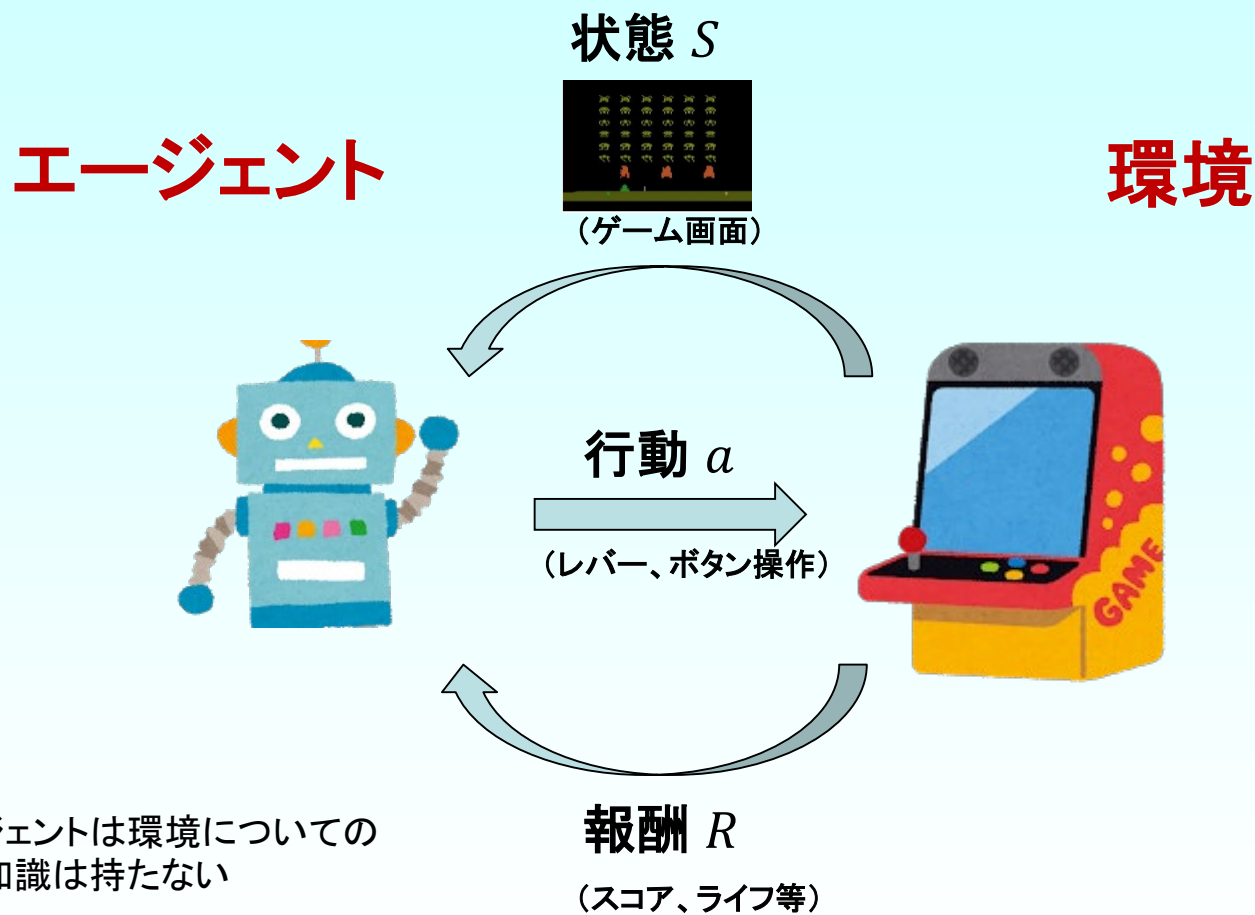
学習



49 games of Atari 2600



# 強化学習とは



エージェントは環境についての  
予備知識は持たない

経験を通じて最適な行動を学習していく

# 最適な行動

直近の報酬を最大にすることが最適とは限らない。  
将来の報酬も含めて最適化したい。

$Q(s, a)$  状態  $s$  のとき行動  $a$  を取ることの「価値」  
これが最大となるような行動  $a$  を取ればよい

$$Q(s, a) = \sum_{n=0}^{\infty} \gamma^n \times (n \text{ ステップ後の報酬})$$

割引付き合計報酬

割引率  $\gamma$  は 0 から 1 の実数

Bellman 最適方程式

$$Q(s, a) = R_{s,a} + \gamma \max_{a'} Q(s_{\text{next}}, a')$$

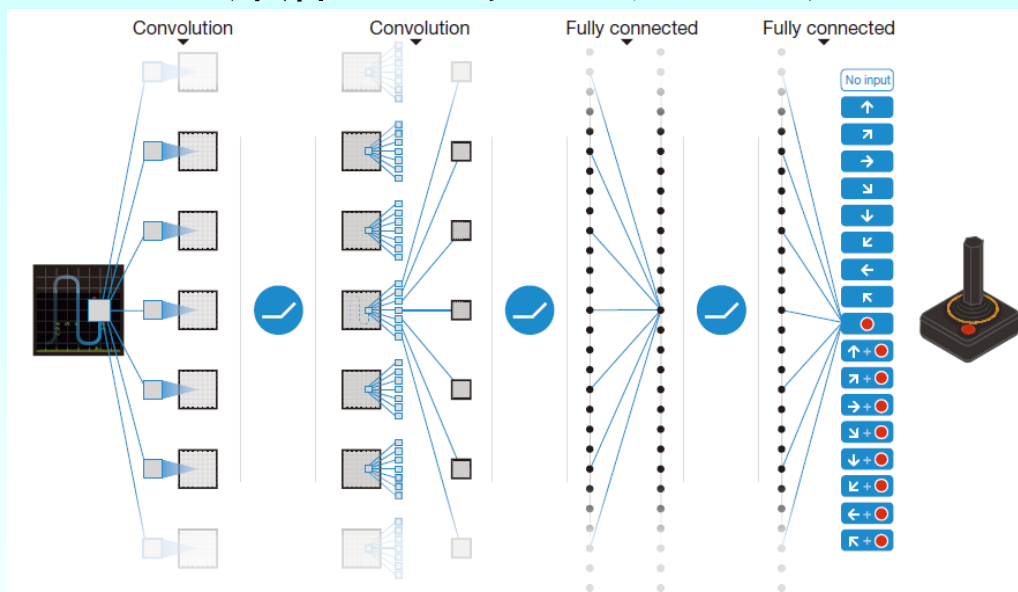
↑  
状態  $s$  で行動  $a$   
を取ったとき  
の報酬

↑  
状態  $s$  で行動  $a$   
を取ったとき  
の次の状態

Bellman方程式の解は将来の(割引付き)合計報酬を最大化する

# Deep-Q Learning

## 深層ニューラルネットワーク



入力はゲーム画面

出力は  $Q(s, a)$

ゲーム画面

どのボタンを押すか

$Q(s, a) \rightarrow R_{s,a} + \gamma \max_{a'} Q(s_{\text{next}}, a')$  となるようにネットワークを学習

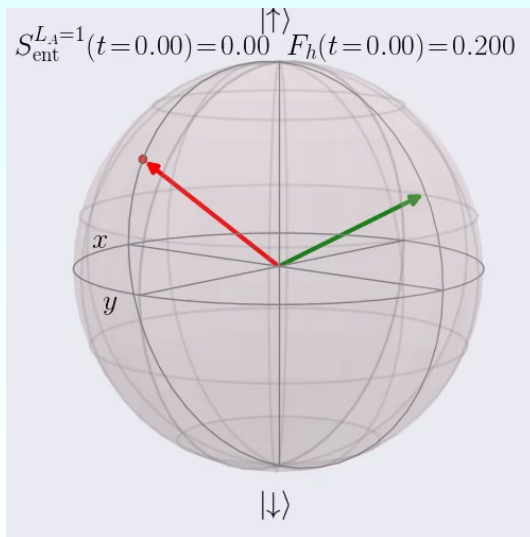


# 量子系への応用

## スピン系の制御

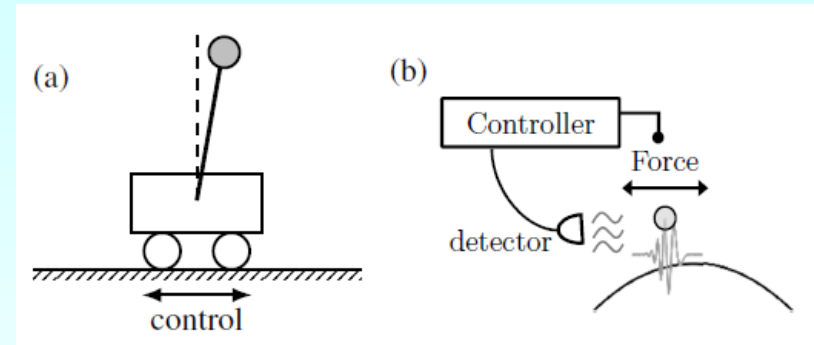
$$H[h_x(t)] = -S^z - h_x(t)S^x,$$

$$H[h_x(t)] = -\sum_{j=1}^L [S_{j+1}^z S_j^z + g S_j^z + h_x(t) S_j^x].$$



M. Bukov et al., PRX **8**, 031086 (2018)

## 量子カートポールの制御

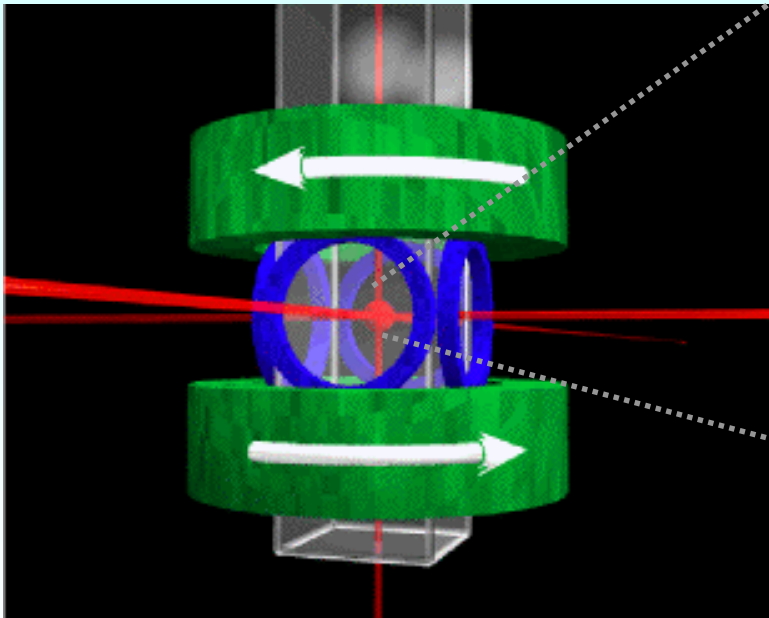


Z. T. Wang, Y. Ashida, M. Ueda, arXiv:1910.09200  
(to be published in PRL)

# 内容

- 強化学習とは
- ボース・アインシュタイン凝縮体と量子渦
- 強化学習を用いた量子渦生成

# 原子気体のボース・アインシュタイン凝縮 (BEC)



M. H. Anderson et al., Science **269**, 198 (1995)



## The Nobel Prize in Physics 2001

"for the achievement of Bose-Einstein condensation in dilute gases of alkali atoms, and for early fundamental studies of the properties of the condensates"



Eric A. Cornell



Wolfgang Ketterle



Carl E. Wieman

$T_c \sim 100 \text{ nK}$   
# of atoms  $10^4 \sim 6$

# 平均場近似

多体ハミルトニアン

$$\hat{H} = \int \hat{\psi}^\dagger \left( -\frac{\hbar^2}{2m} \nabla^2 + V \right) \hat{\psi} d\mathbf{r} + g \int \hat{\psi}^\dagger \hat{\psi}^\dagger \hat{\psi} \hat{\psi} d\mathbf{r}$$

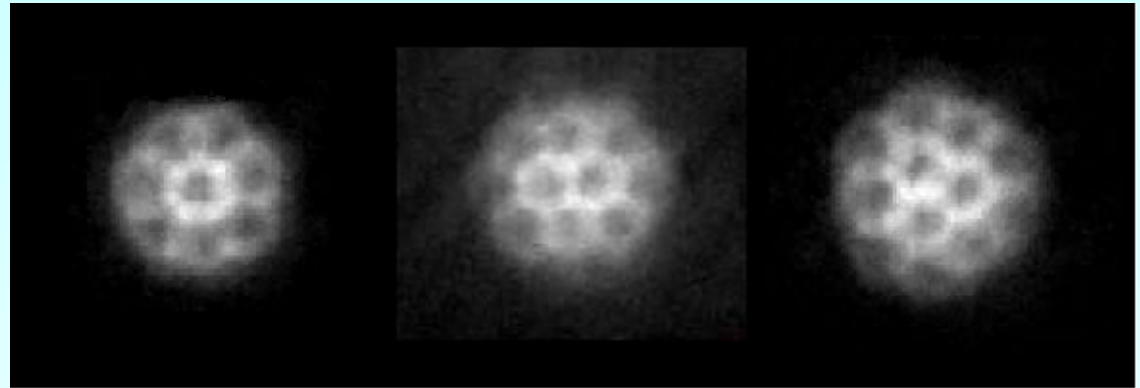
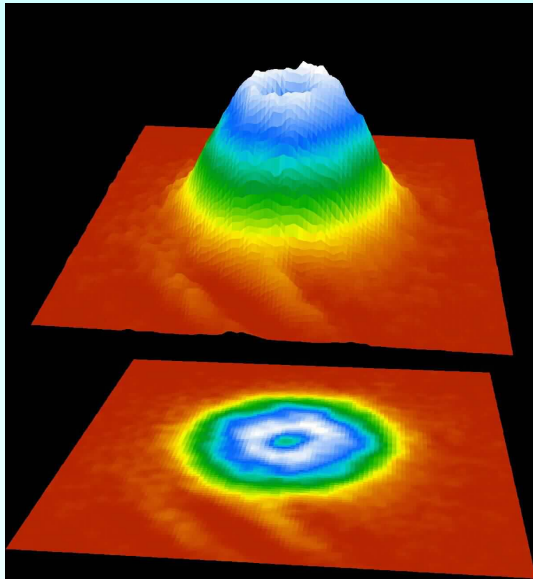
すべての原子が同じ波動関数を占有していると近似する

Gross-Pitaevskii (GP) 方程式

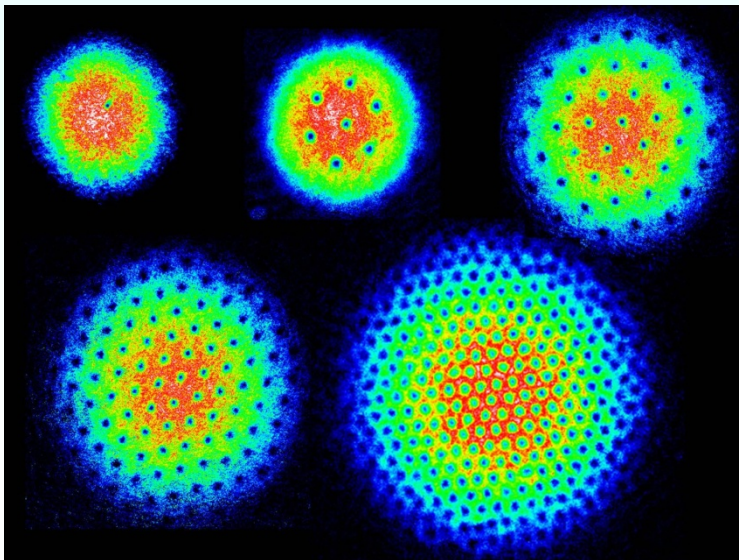
$$i\hbar \frac{\partial \psi}{\partial t} = \left( -\frac{\hbar^2}{2m} \nabla^2 + V + \underline{g|\psi|^2} \right) \psi$$

原子間相互作用を表す非線形項

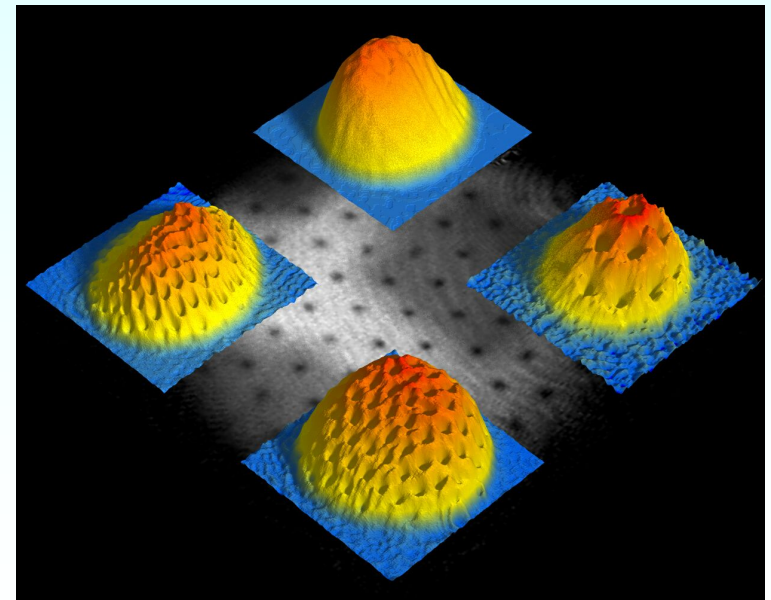
# 回転するBEC



ENS



JILA



MIT

# 渦の量子化

$$\begin{aligned}\mathbf{v} &= \frac{\hbar}{2mi|\psi|^2}(\psi^*\nabla\psi - \psi\nabla\psi^*) \\ &= \frac{\hbar}{m}\nabla\phi\end{aligned}$$

$$\psi(\mathbf{r}) = A(\mathbf{r})e^{i\phi(\mathbf{r})}$$

## 循環の量子化

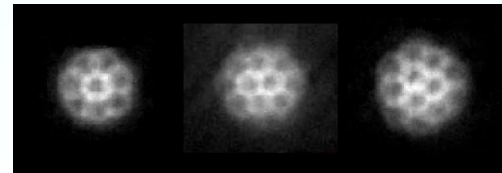
$$\oint_C \mathbf{v} \cdot d\mathbf{r} = \frac{\hbar}{m} \oint_C \nabla\phi \cdot d\mathbf{r} = \frac{h}{m}\ell$$

$\ell = \text{整数}$

循環は  $h/m$  の整数倍でなければならない  
(Onsager, Feynman)

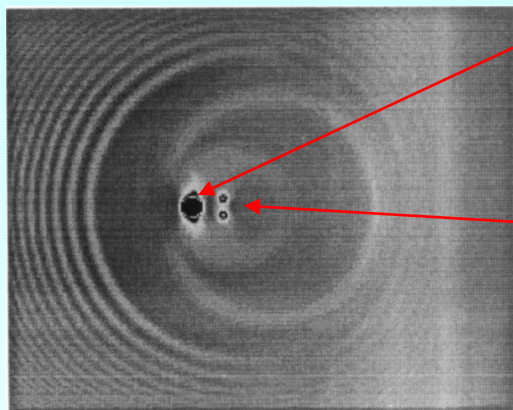
$$\mathbf{v} = \frac{\hbar}{m} \frac{1}{r}$$

遠心力で渦周辺の  
原子密度が下がる



# 量子渦の生成

GP方程式の数値計算

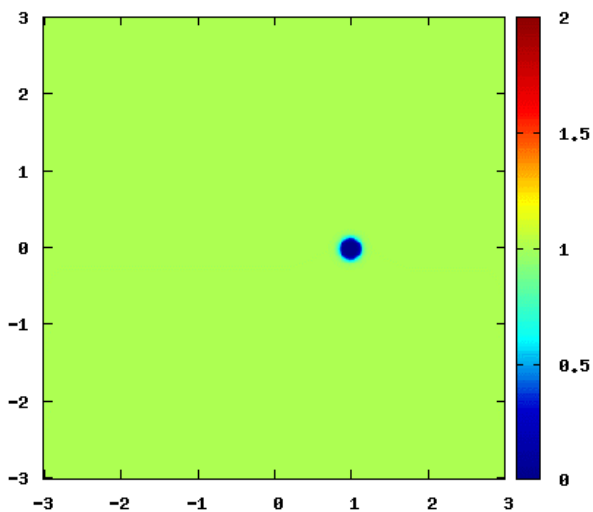


ポテンシャル  
(左へ動いている)

生成された量子渦対

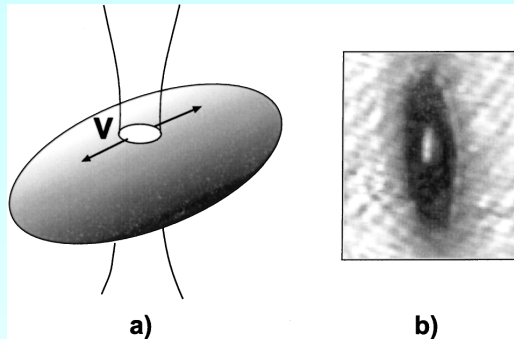
臨界速度を超えると  
量子渦対が生成される

T. Frisch et al., PRL **69**, 1644 (1992)



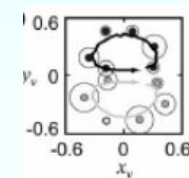
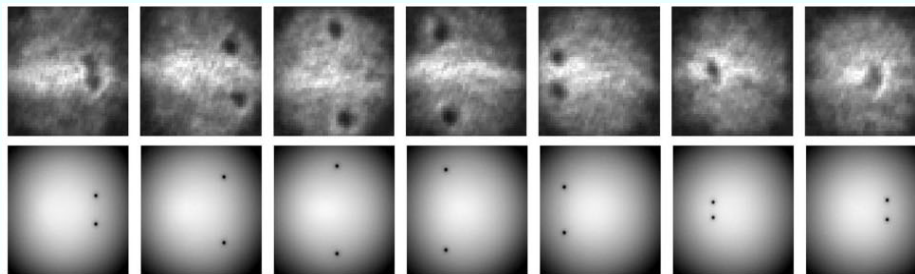
# 量子渦の生成(実験)

レーザーによるポテンシャル



S. Inouye *et al.*, PRL **87**, 080402 (2001)

渦対生成の観測



(ポテンシャルは渦生成後消されている)

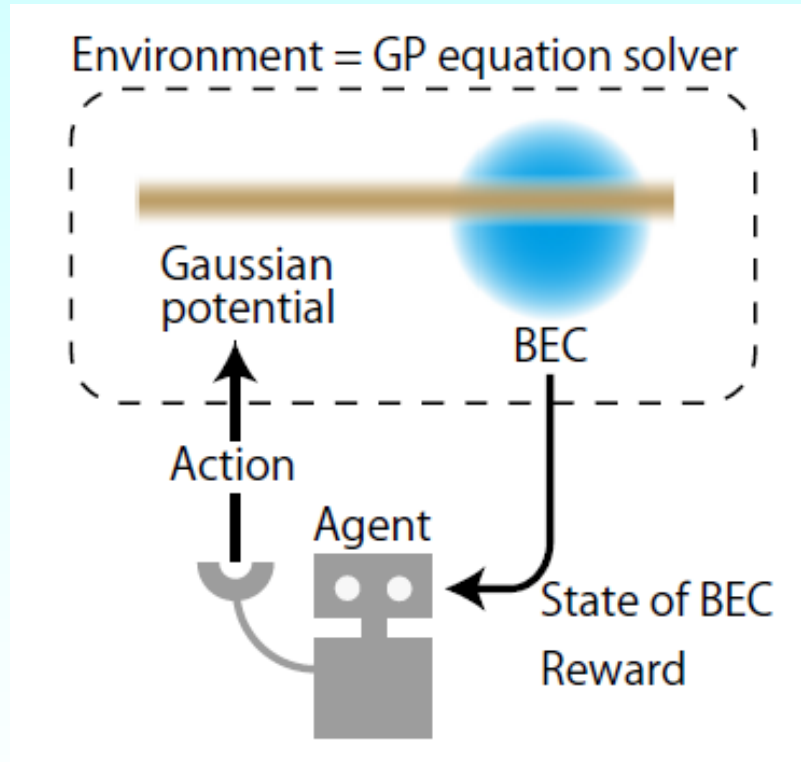
T. W. Neely *et al.*, PRL **104**, 160401 (2010)



# 内容

- 強化学習とは
- ボース・アインシュタイン凝縮体と量子渦
- 強化学習を用いた量子渦生成

# 強化学習によるBECの制御



「環境」はBEC＋外部ポテンシャル(レーザー)

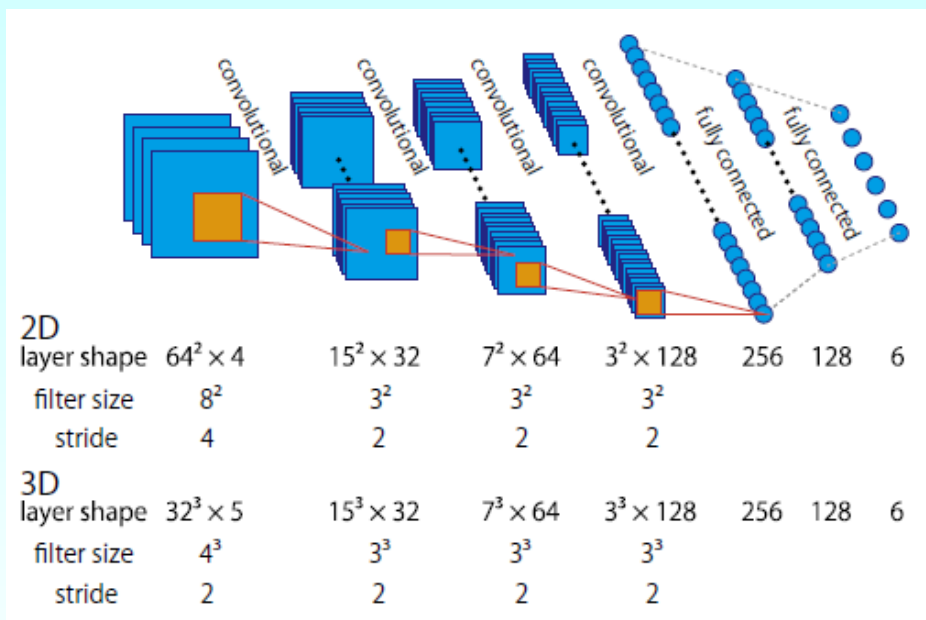
「環境」はGP方程式に従い時間発展し、エージェントは密度分布や速度分布などの情報を得られる

$$i\hbar \frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m} \nabla^2 \psi + V_{\text{trap}}(\mathbf{r})\psi + V_G(\mathbf{r}, t)\psi + g|\psi|^2\psi,$$

外部ポテンシャルの位置や形を制御してBECを望みの状態にすることが目的

# 方法

## 畳み込みニューラルネットワーク(CNN)



### Algorithm 1

```

Initialize deep-Q network  $Q$ 
Initialize target network as  $\hat{Q} = Q$ 
for episode = 1,  $N_{\text{episode}}$  do
  Initialize environment and get initial observation  $s_0$ 
  for  $t = 0, T_{\text{end}}$  do
    Select action  $a_t = \arg\max_a Q(s_t, a)$  with  $\epsilon$ -greedy exploration
    Execute action  $a_t$  on environment and get reward  $r_t$  and next state  $s_{t+\Delta t}$ 
    Store  $(s_t, a_t, r_t, s_{t+\Delta t})$  in replay memory
    Sample minibatch of  $(s_t, a_t, r_t, s_{t+\Delta t})$  from replay memory
    Set  $y_t = r_t$  if  $t = T_{\text{end}}$ , otherwise  $y_t = r_t + \gamma \hat{Q}(s_{t+\Delta t}, a')$ , where  $a' = \arg\max_a Q(s_{t+\Delta t}, a)$ 
    Train network using gradient of  $L(y_t - Q(s_t, a_t))$ 
    Copy  $\hat{Q} = Q$  every  $C$  steps
  end for
end for
  
```

入力は

- ・ 密度分布  $|\psi|^2$
- ・ 流束分布  $\hbar(\psi^* \nabla \psi - \psi \nabla \psi^*) / (2mi)$
- ・ ポテンシャル分布  $V_G$

出力は6つ

6種類の行動(後述)についてのQ値

リプレイメモリ (50000)

$\epsilon$ -greedy方策

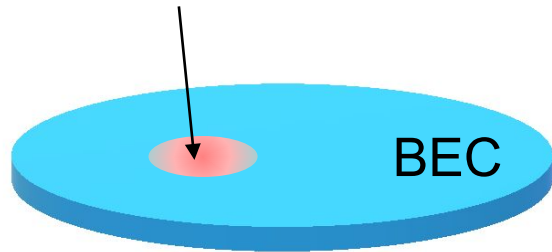
double-Q learning

Huber loss

# 擬二次元系

z方向に強く閉じ込められ  
二次元的に振る舞う系

ガウシアンポテンシャル



トラップポテンシャル

$$V_{\text{trap}} = m[\omega_{\perp}^2(x^2 + y^2) + \omega_z^2 z^2]/2$$

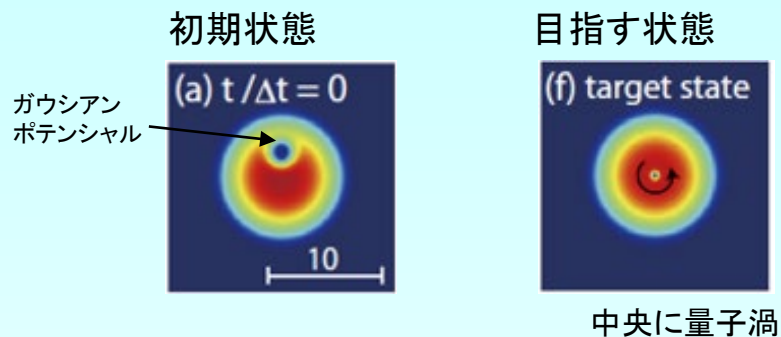
ガウシアンポテンシャル

$$V_G(x, y, t) = A_{\perp}(t) \exp \left\{ \frac{[x - \xi(t)]^2}{d^2} + \frac{[y - \eta(t)]^2}{d^2} \right\},$$

6種類の行動

1.  $\xi$  を少し増加
2.  $\xi$  を少し減少
3.  $\eta$  を少し増加
4.  $\eta$  を少し減少
5.  $A$  を少し増加
6.  $A$  を少し減少

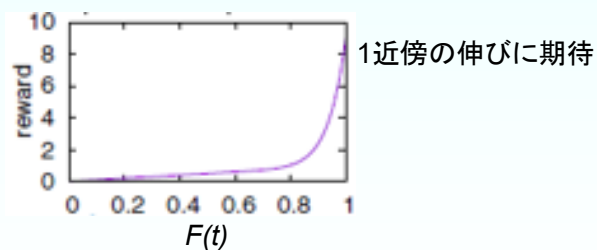
# 結果：擬二次元系



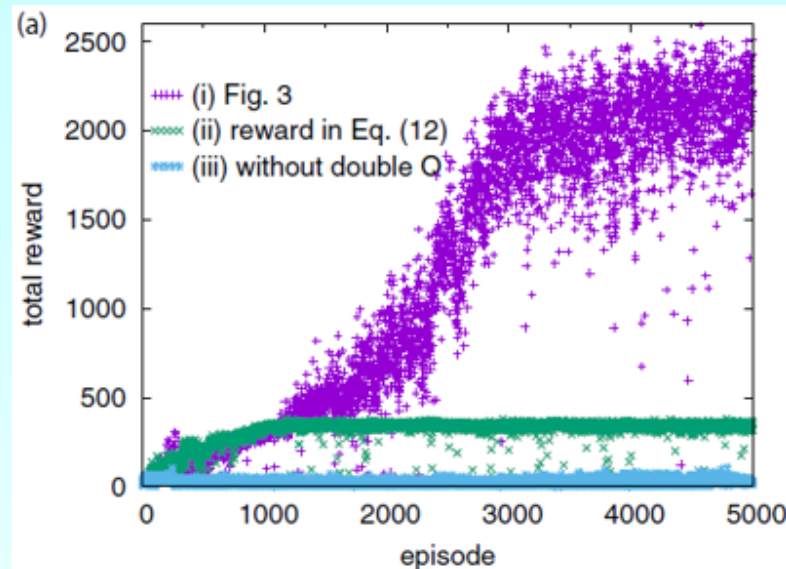
報酬  $r_t$

$$F(t) = \left| \int \psi_{\text{target}}^*(\mathbf{r}) \psi(\mathbf{r}, t) d\mathbf{r} \right|^2$$

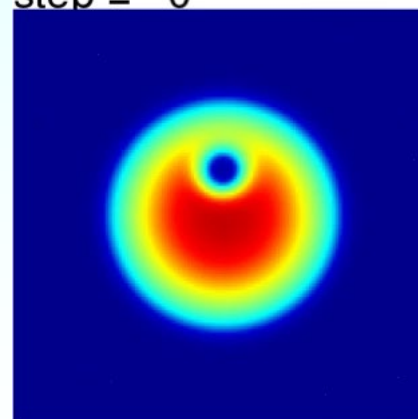
$$r_t = F(t) + 8[F(t)]^{16}$$



学習の過程

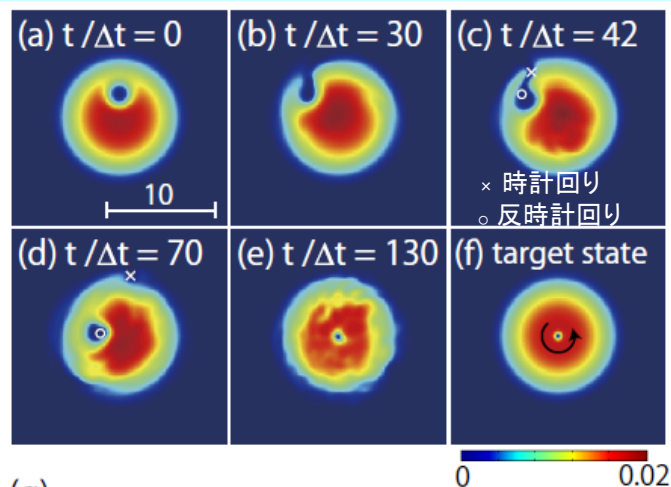


step = 0

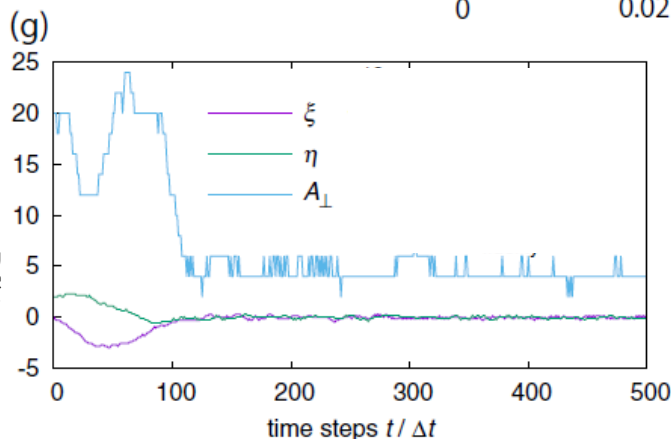


ベストな結果

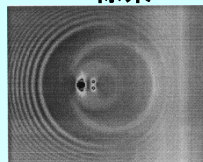
# 結果：擬二次元系



最適なポテンシャル  
の動き



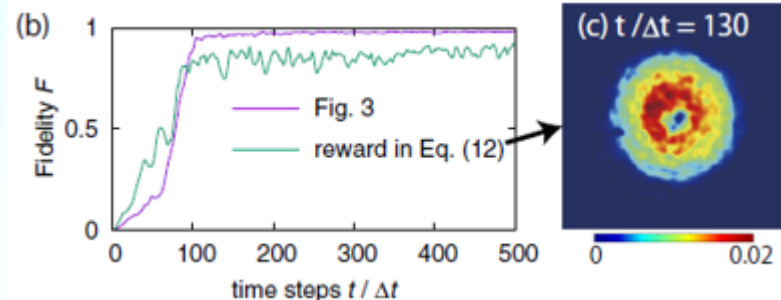
一様系



BECの縁に行き、時計回り量子渦だけ外に逃がす (b), (c)

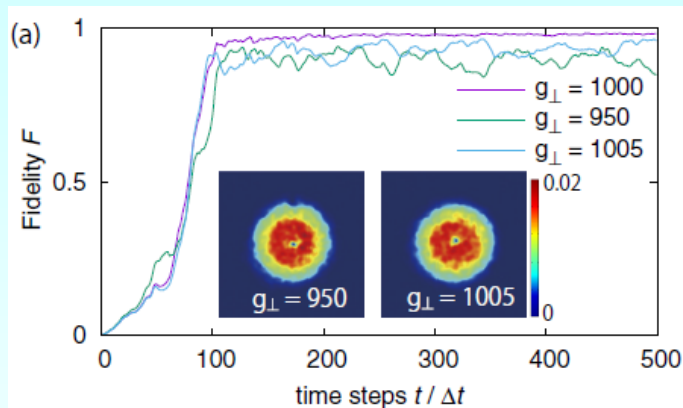
量子渦を中心まで運んだあとポテンシャル消滅 (d), (e)

$$\text{報酬} = F(t) = \left| \int \psi_{\text{target}}^*(r) \psi(r, t) dr \right|^2 \text{ のとき}$$

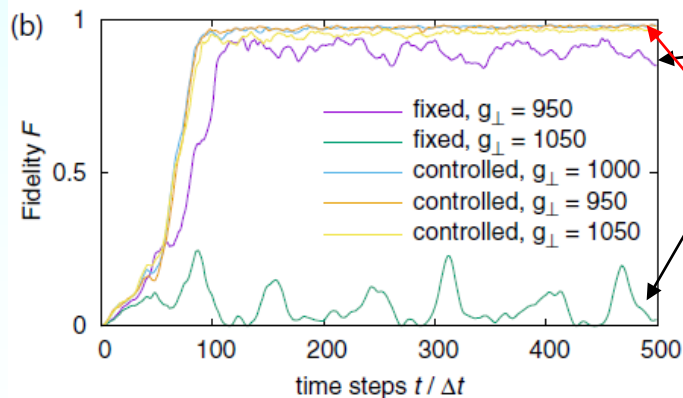


# 相互作用係数 $g$ がずれた場合

実験条件がずれた場合、例えば原子数  $N \propto g$  が想定と違った場合



$g = 1000$  で最適化した  
ポテンシャルの動きを  
別の  $g$  に適用



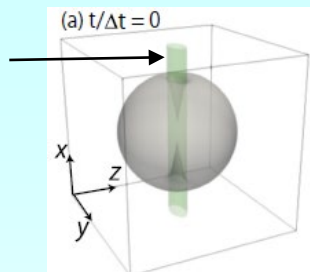
$g = 1000$  で最適化した  
ポテンシャルの動きを  
別の  $g$  に適用

$g = 1000$  で最適化した  
エージェントを別の  $g$  に適用

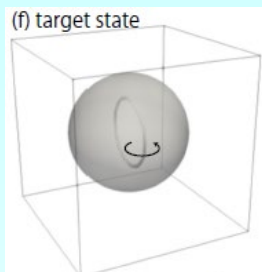
訓練されたエージェントは  
高い適応能力を持つ

# 三次元系

ガウシアン  
ポテンシャル



初期状態



目指す状態

定常的な量子渦輪



## 6種類の行動

1.  $\zeta$  を少し増加
2.  $\zeta$  を少し減少
3.  $d_y$  を少し増加
4.  $d_y$  を少し減少
5.  $A$  を少し増加
6.  $A$  を少し減少

等方的調和振動子ポテンシャル +  
ガウシアンポテンシャル

$$V_G(\mathbf{r}, t) = A(t) \exp \left\{ \frac{y^2}{d_y^2(t)} + \frac{[z - \zeta(t)]^2}{d_z^2} \right\},$$

x方向に沿ったレーザービームに対応

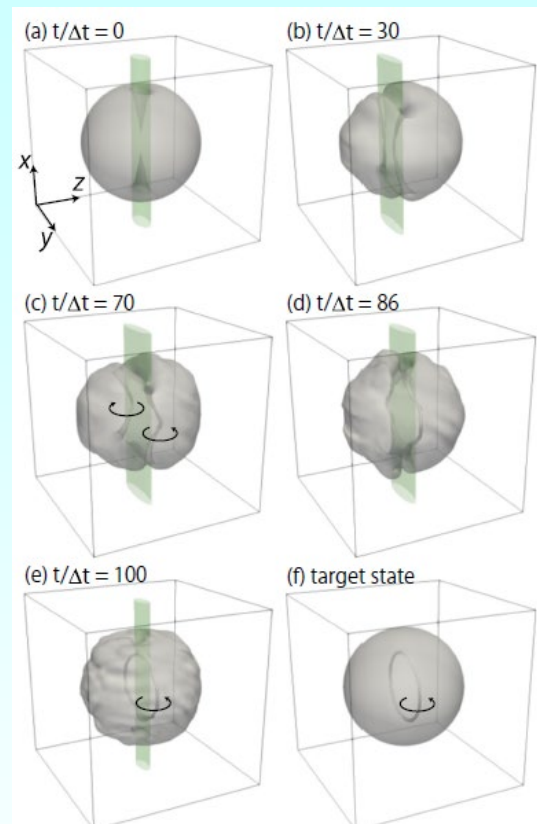
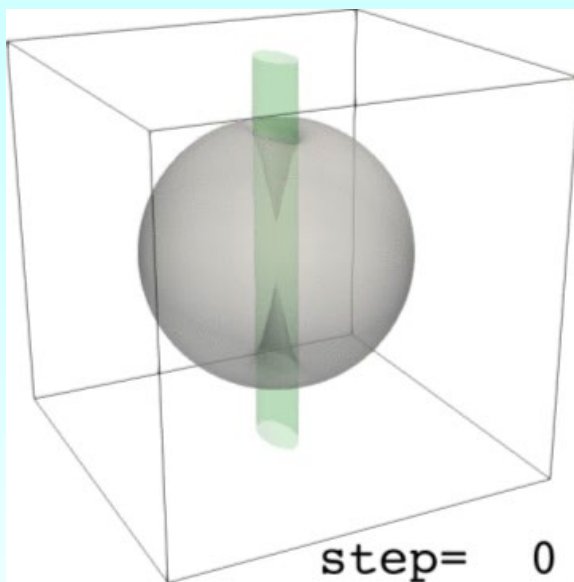
報酬は先ほどと同じ

$$F(t) = \left| \int \psi_{\text{target}}^*(\mathbf{r}) \psi(\mathbf{r}, t) d\mathbf{r} \right|^2$$

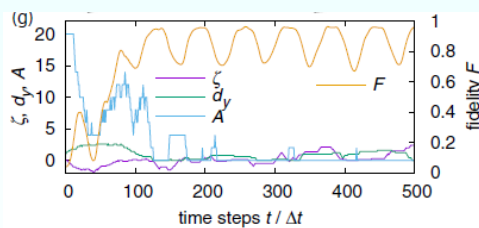
$$r_t = F(t) + 8[F(t)]^{16}$$



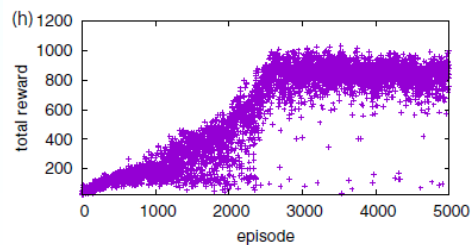
# 結果：三次元系



最適なポテンシャル  
の動き



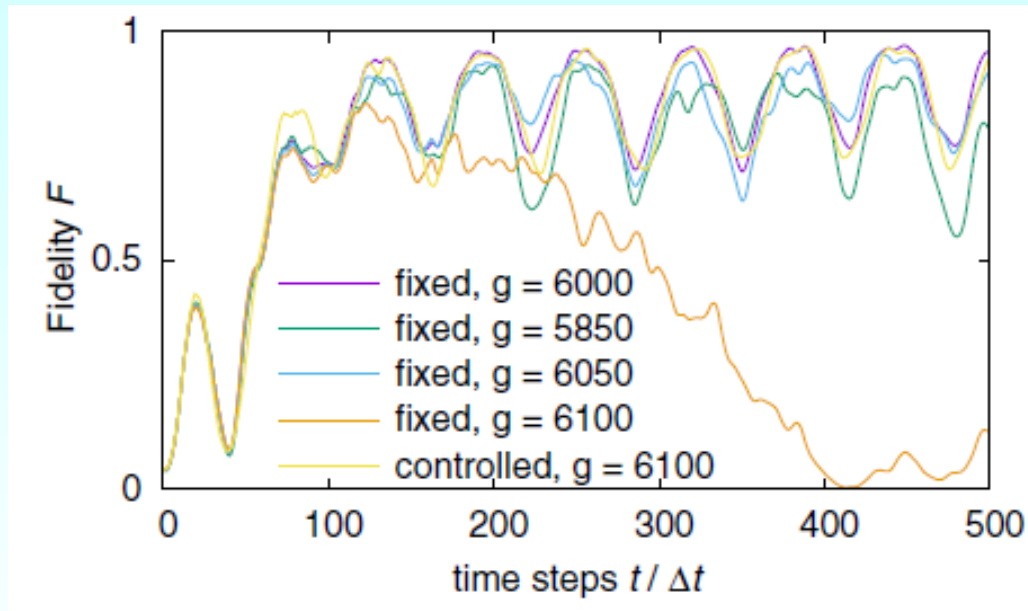
学習の過程



ポテンシャルを  $-z$  方向に動かして量子渦対を作る (b),(c)

ポテンシャルを平たくして微調整すると2本の渦線が閉じて (d)  
定常的な渦輪の状態になる (e)

# 相互作用係数 $g$ がずれた場合



黄色以外

$g = 6000$  で最適化した  
ポテンシャルの動きを  
別の  $g$  に適用

⇒  $g = 5850-6050$  なら  
ある程度良い結果  
 $g = 6100$  はだめ

黄色

$g = 6000$  で最適化した  
エージェントを  $g = 6100$  に適用

⇒  $g = 6100$  でも良い結果。  
訓練されたエージェントの  
高い適応能力

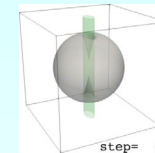
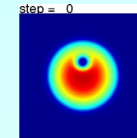
# まとめ

H. Saito, JPSJ **89**, 074006 (2020)

強化学習(Deep-Q learning)をボース・アインシュタイン凝縮体の制御に応用

障害物ポテンシャルを制御してかき回すことで、単一量子渦状態や

量子渦輪状態を生成できることがわかった。



# 展望

- ・ 実際の実験系の制御(測定ベースで)
- ・ より面白い状態の生成(多成分BECにおけるSkyrmionなど)