FeUdal Networks for Hierarchical Reinforcement Learning

DeNA Co., Ltd.

AIシステム部 AI 研究開発グループ

甲野 佑

自己紹介

甲野 佑

所属：株式会社ディー・エヌ・エー　AI システム部 AI 研究開発グループ

出身：東京電機大学 (学部～博士) = おたく大学

研究：強化学習 ＋ 脳神経・認知 → ゲーム AI 開発 (DeNA)

- 認知や脳神経と機械学習を組み合わせて**純粋な理学研究**より**実用的な強化学習アルゴリズム**を構築したい
- 人間の柔軟性→不完全知覚(への対処)→限定合理性→汎用性＝階層性？

不完全知覚

環境の状態 s の欠損された観測 o (死角，記憶欠如，マルチエージェント性等が原因)＝ 不完全知覚問題 (Partially Observabl MDP, POMDP)

例：Montezuma’s Revenge

※ [Bellmare et al., 2012]

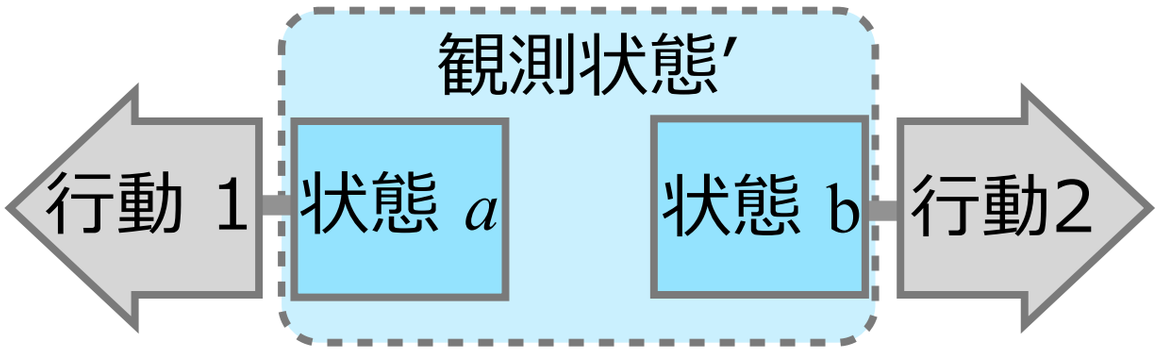
鍵を獲得した等 “他の部屋で何をやったか” を覚えていないと不完全知覚

→ 現実環境ではエージェントが得られる情報などたかが知れている

不完全知覚への問題

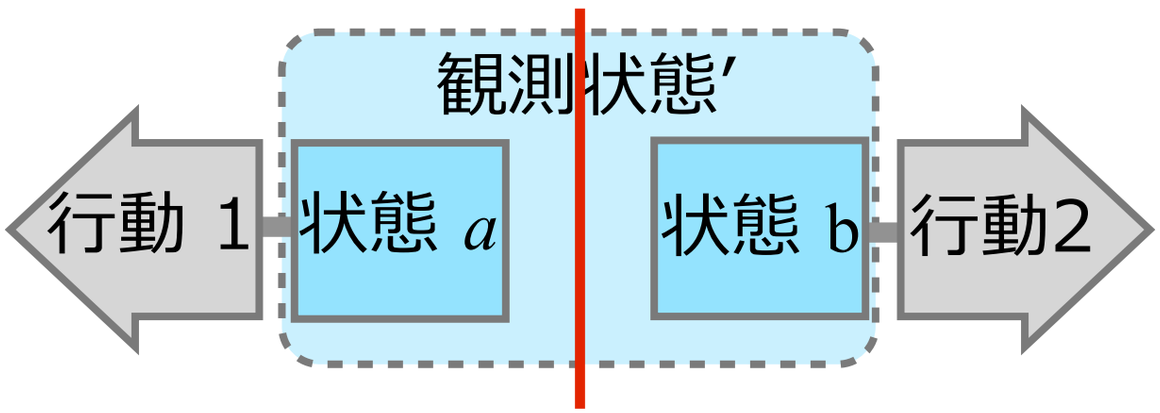
POMDP が困難な理由の一つは観測に対する真の状態の混同

→ 行動の選択確率 π は観測状態に対し**一意**しか持てない事



本来左に行くべき状態 a　　本来右に行くべき状態 b

→ **何らかの方法で分離してやれば良い**



不完全知覚への対処

完全記憶

機能：

- 初期状態から**全ての状態遷移を記憶**する事で“**本来異なる状態**”を分離

問題点：

- 全ての**履歴**を覚えているのは現実的に不可能
- 死角や他エージェントの内部状態はどちらにせよ問題
  - LSTM 等を使っても問題が解決するわけじゃない
- 真面目にやると状態数が**容易に爆発**する (かつ汎化されない)
  - 状態概念が複雑になり報酬などの獲得情報が環境全体に対してスパースになりやすい (**強化学習全般の問題**を加速させる)

不完全知覚への対処

階層型強化学習

機能：

- 意思決定を上位層と下位層に分割する (後述)
- 観測に対する MDP で事足りる部分 (サブタスク) を下位層が担当
- サブタスクの選択，遷移順を決める上位層が担当
- **手順，記憶**を上位層が**目的意識 (の遷移) として**扱い状態を分離
- 下位層の行動 (サブタスク適応) が使い回せるので**汎化性能**が高い

問題点：

- 自律的なサブタスク分割が**非常に困難**

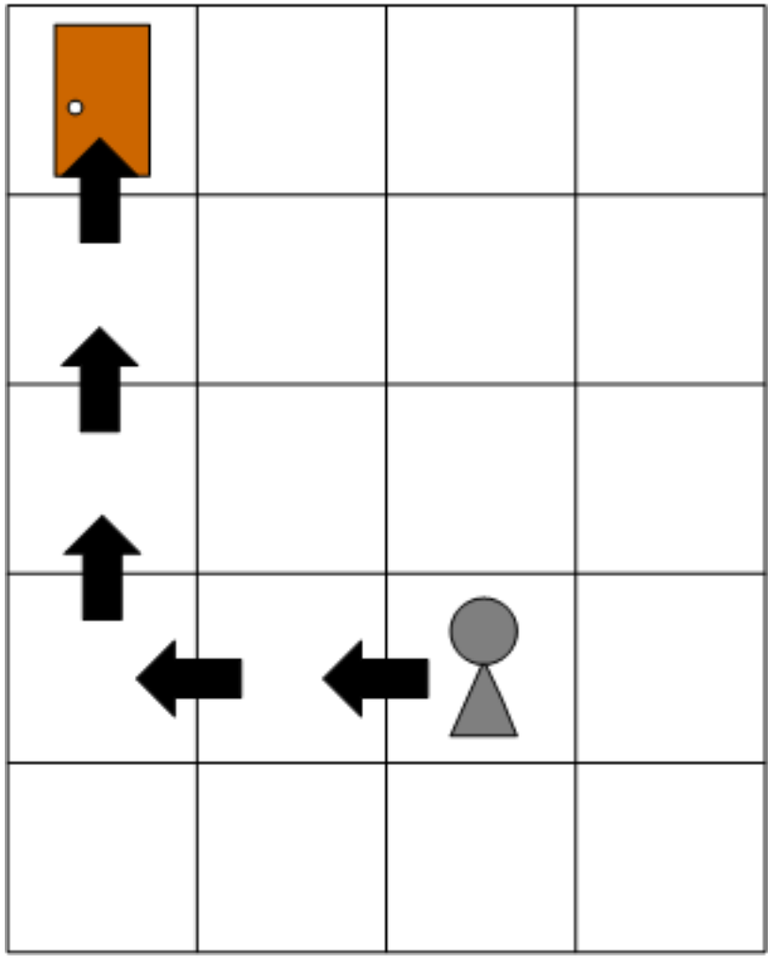
→ FeUdal Networks [Vezhnevets et al., 2017]というサブタスク分解を End-to-End で行うアーキテクチャを DeepMind が考案

本題：FeUdal Networks (FuN) の前に

階層型強化学習のイメージを少し

通常(非階層型)の強化学習イメージ

方策：$\pi(a;s)$



細かな意思決定

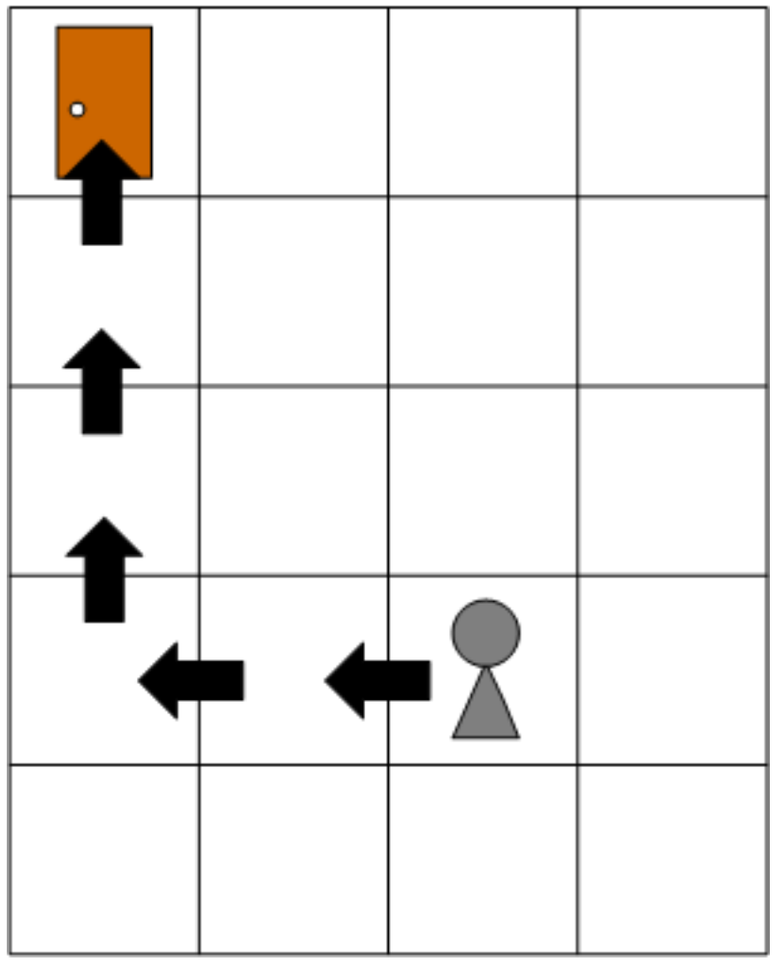
プリミティブな行動：a

階層型強化学習イメージ

下位層方策：$\pi(a;s,g)$

× 複数

上位層方策：$\pi_{goal}(g;s)$

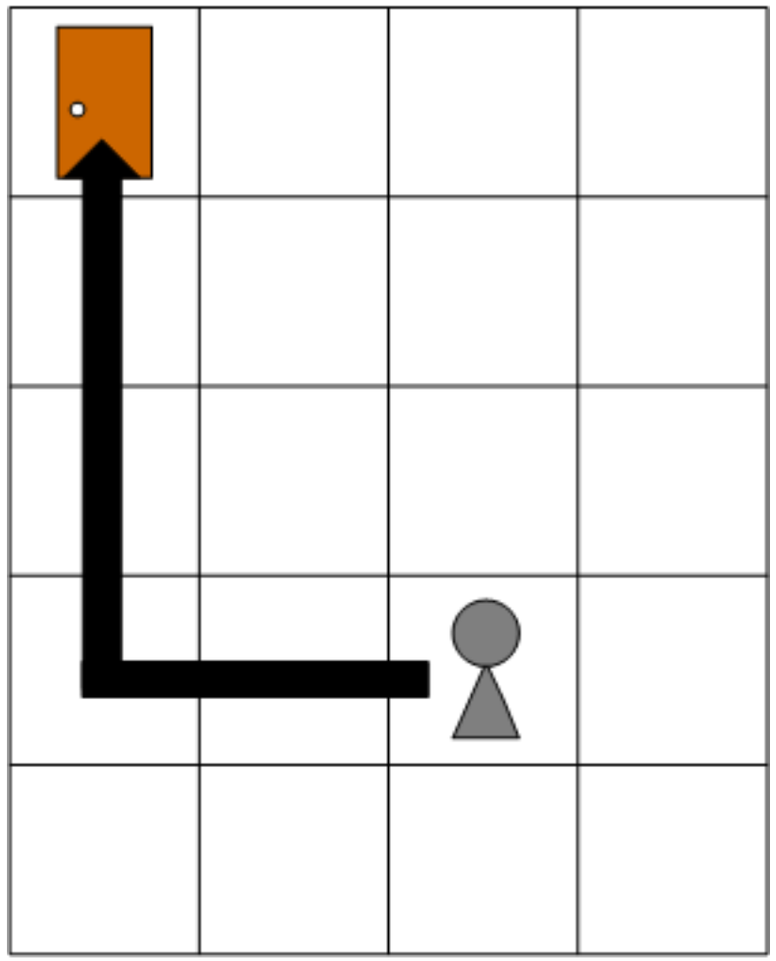


細かな意思決定

プリミティブな行動：a

下位層方策を選択（g として）

試行錯誤から上位層方策を生成？



大まかな意思決定

目的指向：g

不完全知覚への対処 (再掲)

階層型強化学習

機能：

- 意思決定を上位層と下位層に分割する (後述)
- 観測に対する MDP で事足りる部分 (サブタスク) を下位層が担当
- サブタスクの選択，遷移順を決める上位層が担当
- **手順，記憶**を上位層が**目的意識 (の遷移) として**扱い状態を分離
- 下位層の行動 (サブタスク適応) が使い回せるので**汎化性能**が高い

問題点：

- 自律的なサブタスク分割が**非常に困難**

→ FeUdal Networks [Vezhnevets et al., 2017] というサブタスク分解を End-to-End で行うアーキテクチャを DeepMind が考案

FeUdal Networks - 背景概念

Feudal reinforcement learning, FRL [Dayan and Hinton, 1993]

- 意思決定を上位層と下位層に分割する
  - 下位意思決定者 Sub-Maneer, 上位意思決定者 Manager という概念を提供
- FuNs (本題) のアイディア元
- Sutton の強化学習本よりも前なので概念的なものに近い？

FeUdal Networks - 背景概念

Option [Sutton et al., 1999]

- 長期的な行動方策 Option (通常の方策概念に goal 状態 = termination condition を加えたもの) とその Option そのものを選択する policy-over-option を導入
- Semi-MDP への対処が主眼 (意思決定が疎)
- 個別の Option の自律的な学習は不十分な側面がある
- 比較的**古典的な話**

Option-Critic [Bacon et al., 2017]

- Option を Deep RL 化したもの (Goal 概念が薄い？)
- End-to-End で学習可能
- 上位方策と下位方策の役割が**未分化**する問題点が存在

FeUdal Networks - 概要

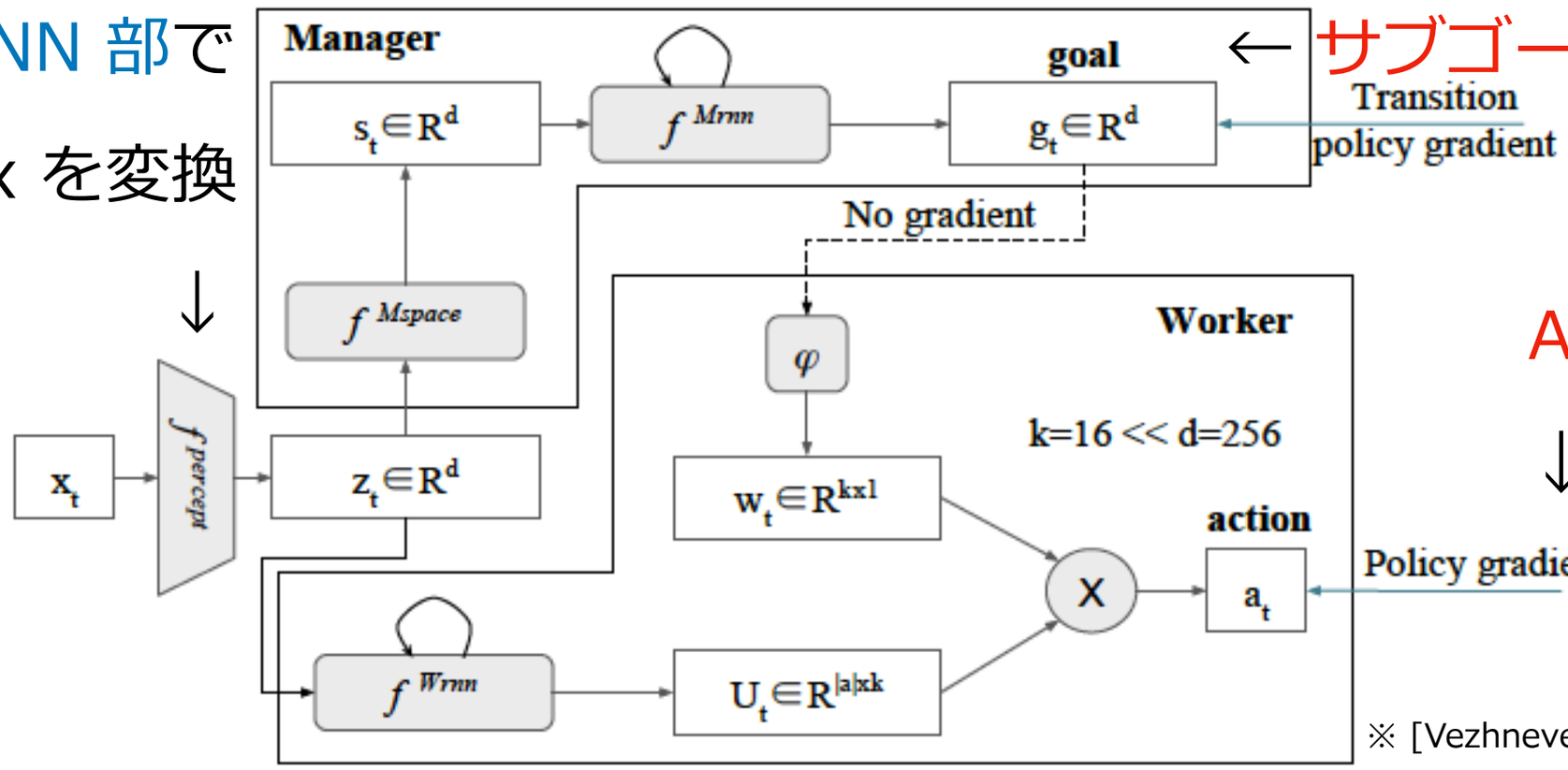
FRL の意思決定を上位層と下位層に分割が根幹アイディア

- 下位意思決定者 **Worker** (FRL での sub-maneer)
- 上位意思決定者 **Manager**

↓ 上位の行動 sub-policy を選択する Manager

CNN 部で

観測 x を変換



A3C で学習

↓

※ [Vezhnevets et al., 2017]

↑ 下位の行動 primitive-policy を選択する Worker

FeUdal Networks - 概要

FuN はアーキテクチャ全域で微分可能 ＝ Manager によるサブタスク分割を End-to-End で学習可能

類似概念 Option と FuN の大きな違いは Goal 概念があるか否か

- Option は Goal に到達したら上位方策によって次の Goal を決定
- FuN は定数時間 c step によって意思決定タイミングを区切る
  - この**定数時間 c step** の**割り切り**が秘訣の一つ？
  - 論文中の結果は $c = 10$

FeUdal Networks - 特徴

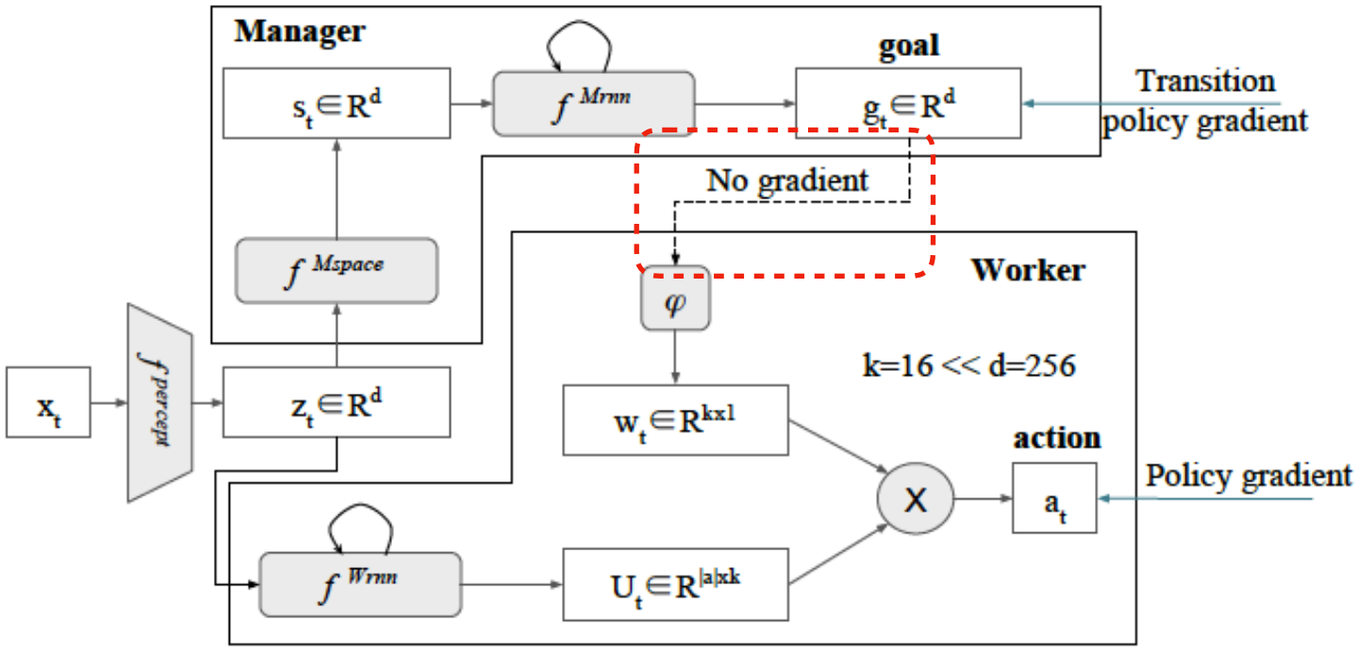
FuNs の特徴

1. Transition policy gradient for training the Manager
    - Manager 訓練のための遷移方策勾配 (遷移方策＝上位層方策)
2. Relative rather than absolute goals
    - 絶対的ではなく相対的な定義によるサブゴール形成
        - 獲得された潜在状態ベクトル s に対する差分を g として学習
3. Lower temporal resolution for Manager
    - Manager の扱う時間間隔の低解像度化
    - 意思決定タイミングと Dilated LSTM
4. Intrinsic motivation for the Worker
    - 内部報酬(といっても比較的弱め)

Transition policy gradient for the Manager

Manager (上位層)訓練のための遷移方策勾配：

Worker (下位層)とは**異なる勾配**を定義



※ [Vezhnevets et al., 2017]

$$\nabla g_t = A_t^M \nabla_\theta d_{\cos}(s_{t+c} - s_t, g_t(\theta))$$

$$※\ A_t^M = R_t - V_t^M(x_t, \theta)$$

$$\nabla \pi_t = A_t^D \nabla_\theta \log \pi(a_t | x_t; \theta)$$

$$※\ A_t^D = (R_t + \alpha R_t^I - V_t^D(x_t; \theta))$$

↑

Manager から与えられる仮想的な収益

※ Option-Critic は下位層の意思決定者からの勾配を使って学習

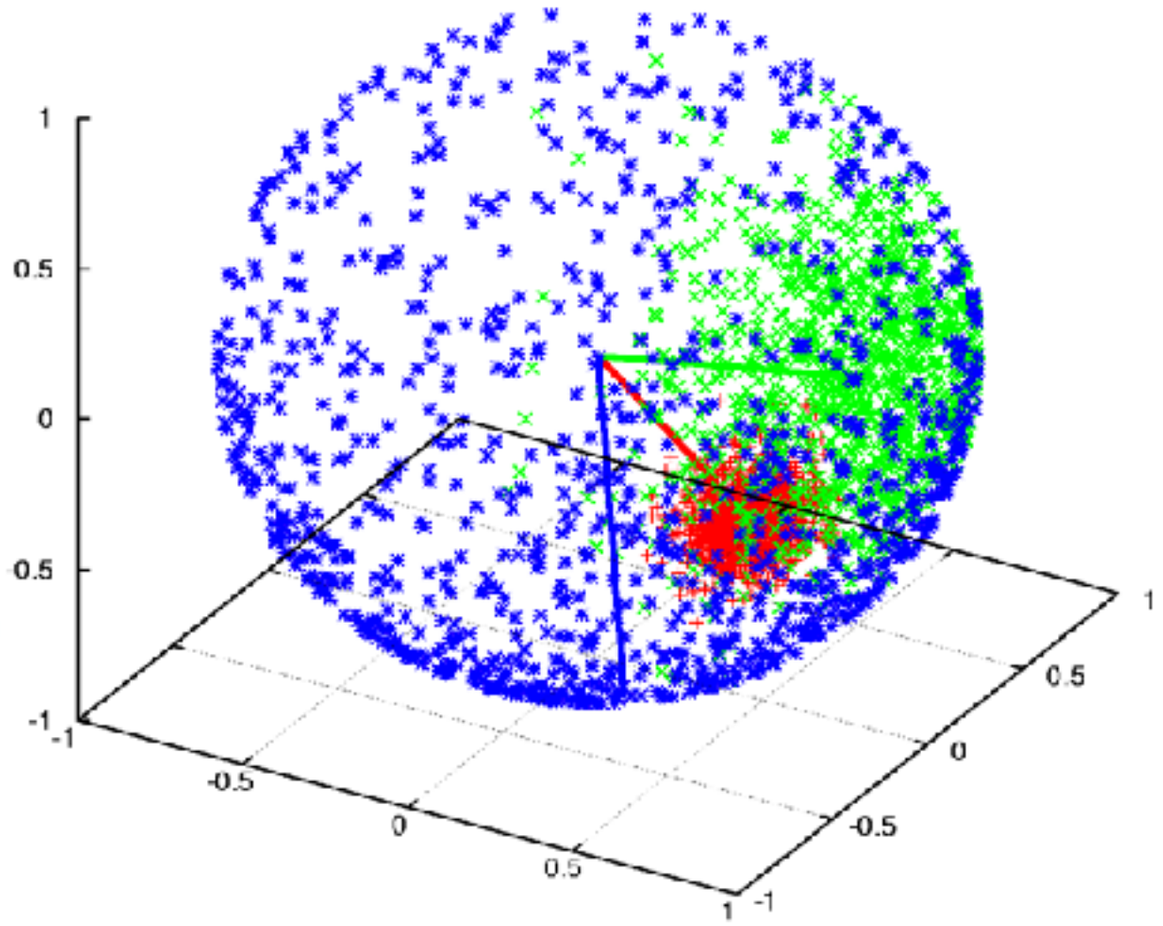
Relative rather than absolute goals

絶対ではなく相対的なサブゴール： g_t $s_{t+c} - s_t$

Von Mises–Fisher 分布 (d 次元の**方向**の正規分布) を近似

Cos 類似度との比例関係：

$$p(s_{t+c}|s_t, o_t) \propto e^{d_{\cos}(s_{t+c}-s_t, g_t)}$$



※ https://en.wikipedia.org/wiki/Von_Mises-Fisher_distribution

実際の相対的ゴールの更新式：

$$\nabla g_t = A_l^M \nabla_\theta d_{\cos}(s_{t+c} - s_t, g_t(\theta))$$

↓

意味的には：

$$\nabla_\theta \pi_t^{TP} = \mathbb{E}\left[(R_l - V(s_l)) \nabla_\theta \log p(s_{l+c}|s_l, \mu(s_l, \theta))\right]$$

↑ Von Mises–Fisher 分布

Lower temporal resolution for Manager

Manager が扱う時間間隔の低解像度化

Dilated LSTM :

$$\hat{h}_t^{t\%r}, g_t = LSTM(s_t, \hat{h}_{t-1}^{t\%r}; \theta^{LSTM})$$

r (= c ; 論文中) 系列数の LSTM を用意し，r サイクル毎に新たな状態を保存していく

→ Manager レベルでは長期の記憶を保持 (r step 間隔)

Intrinsic motivation for the Worker

内部報酬：

$$r_t^I = 1/c \sum_{i=1}^{c} d_{\cos}(s_t - s_{t-i}, g_{t-i})$$

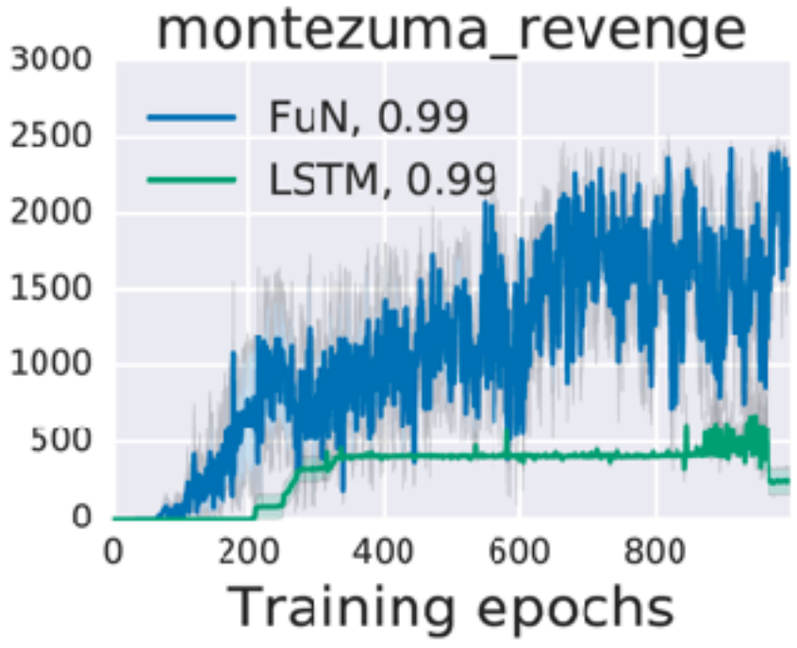
↑ 予測されたゴールとの類似度

※ $0 \sim c$ step の間に goal に近づけば OK！

流行りの好奇心 (intrinsic reword) とはまた異なる

環境探索効果は少ないので好奇心との組み合わせは可能

結果 - Montezuma’s Revenge

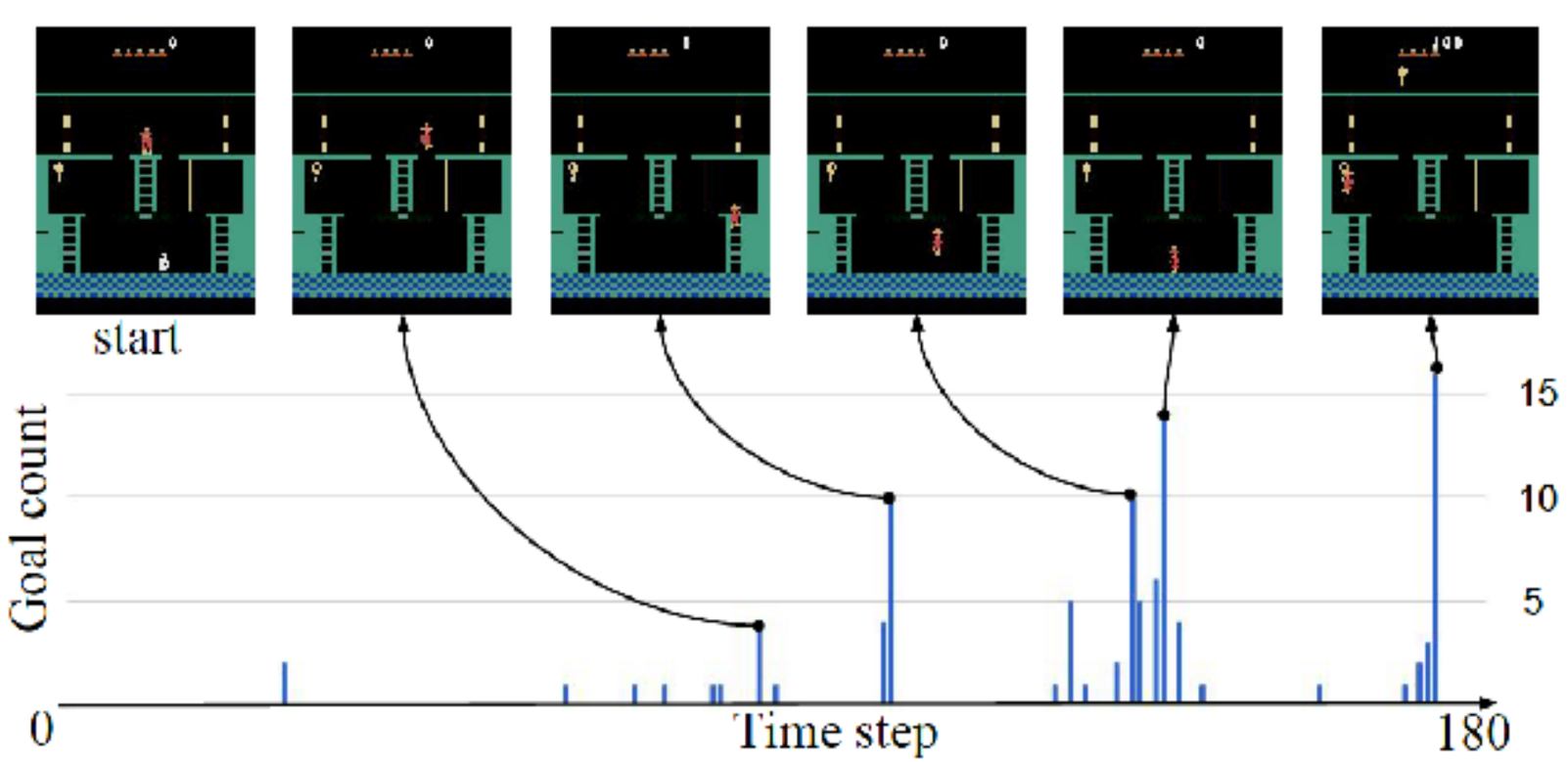


← 素の LSTM より良い

200 epoch 未満で最初の部屋を突破

(1 epoch = 100万 step)

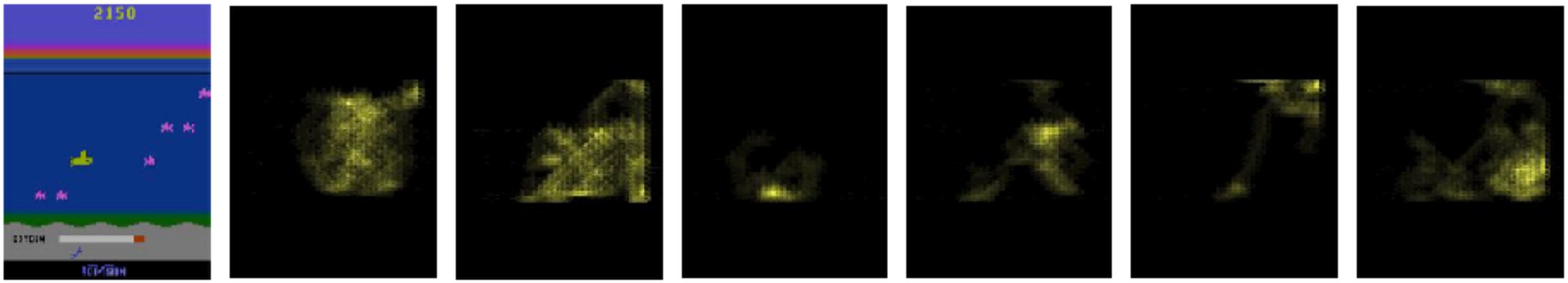
↓ 以前の時系列出力にとって goal になってる数



※ [Vezhnevets et al., 2017]

結果 - サブ方策の確認

ゴール固定時の動き：



Example frame　LSTM　Full FuN　sub-policy 1　sub-policy 2　sub-policy 3　sub-policy 4

Manager がゲーム中経験したサブゴール g を記憶しておき，あえて固定して行動させる事でサブ方策の動きを確認できる

上記グラデーションマップは Agent の空間滞在比率を平均化したもの

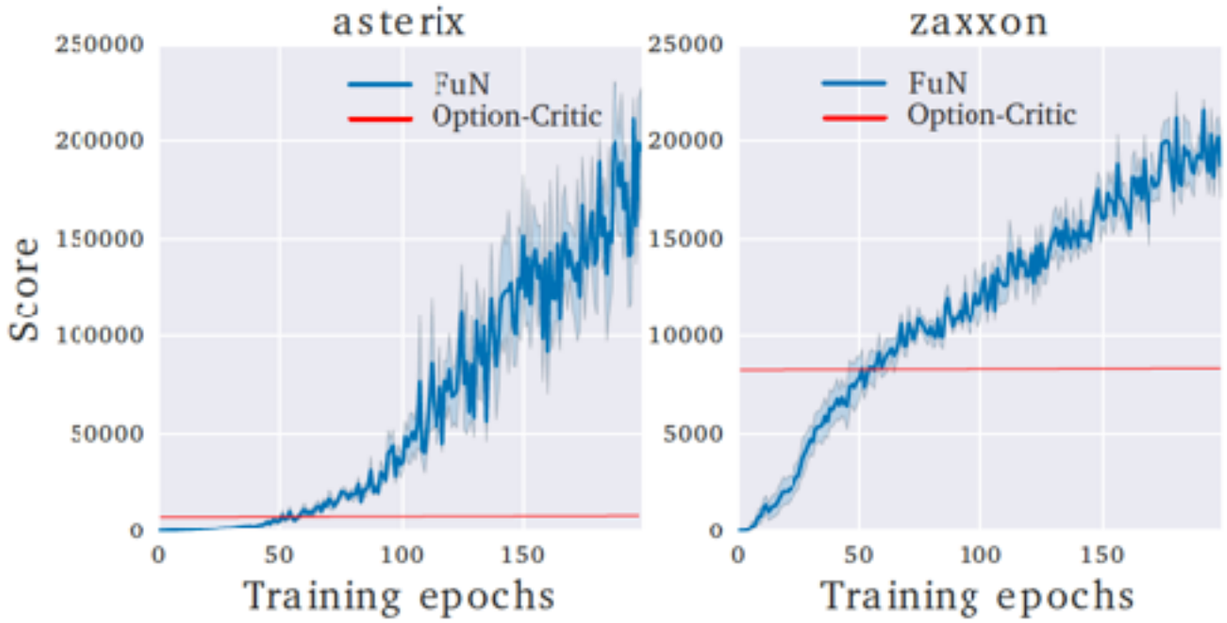
→ 異なる動きが確認できる

例：sub-policy 3 は空気を補充している

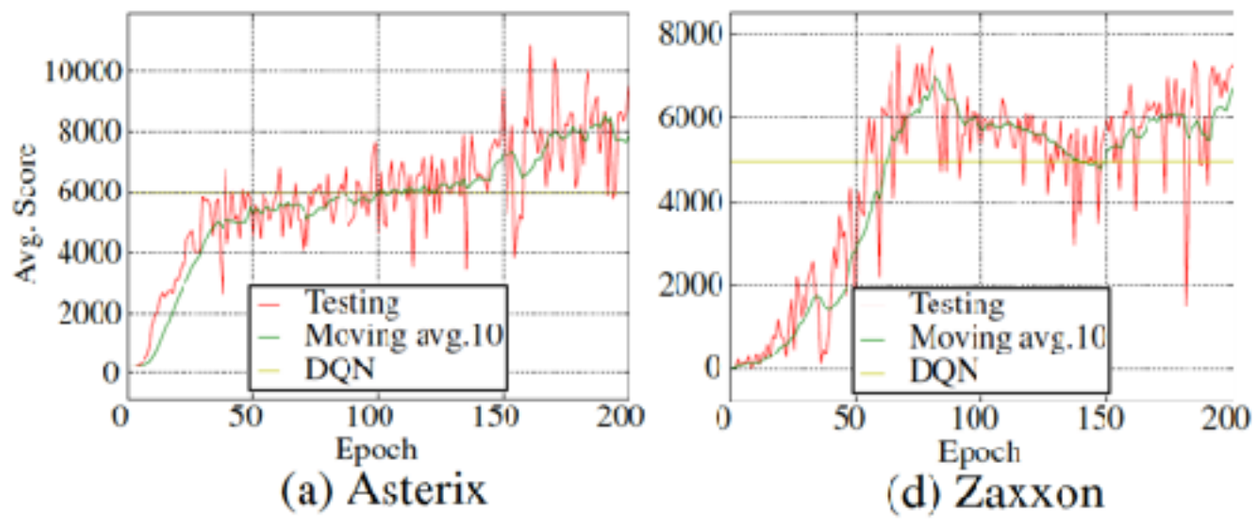
※ [Vezhnevets et al., 2017]

結果 - Option-Critic との比較

FuNs :



Option-Critic :



同じ End-to-End な階層型強化学習 Option-Critic と比較して良い成績

停滞気味の Option-Critic に比べて，FuN は更に上がり続けている

※ [Vezhnevets et al., 2017]

結果 - アイディアの正しさ

4種の欠損型 FuN との比較:

Non feudal FuN：方策勾配で訓練，内部報酬も使わない (Option-Criticに近い)

Pure feudal FuN：Worker に内部報酬を使わない

Manager via PG FuN： Manager を方策勾配で訓練

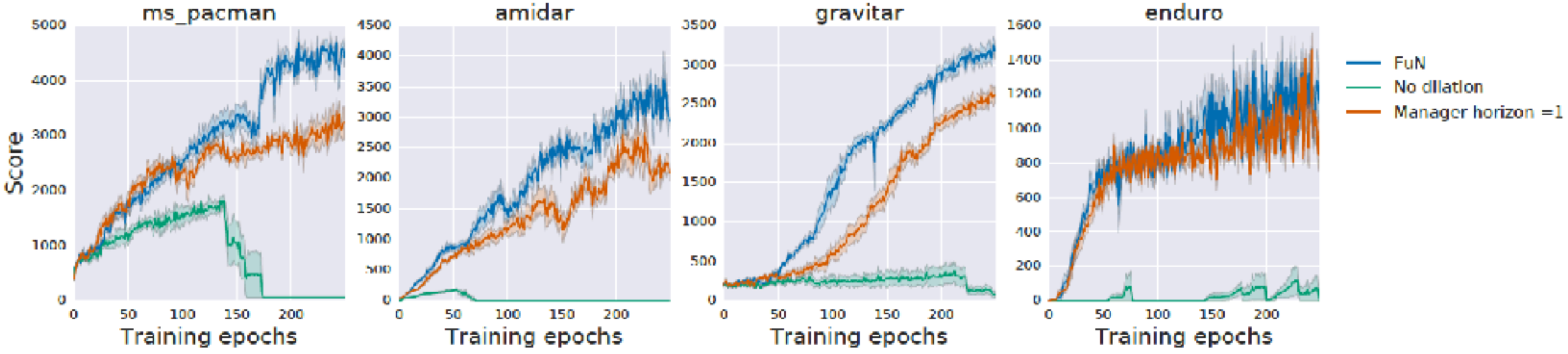
Absolute goals Fun：絶対ゴールを使用 (具体的な定義は読みきれなかった)

→ 全てにおいて FuN が勝利 = 3つのアイディアの有効性

※ [Vezhnevets et al., 2017]

結果 - アイディアの正しさ

Dilated LSTM に関する比較:



No dilation：Manager に通常 LSTM を使用

Manager horizon = 1： $c = 1$ で Manager と Dilated LSTM を運用

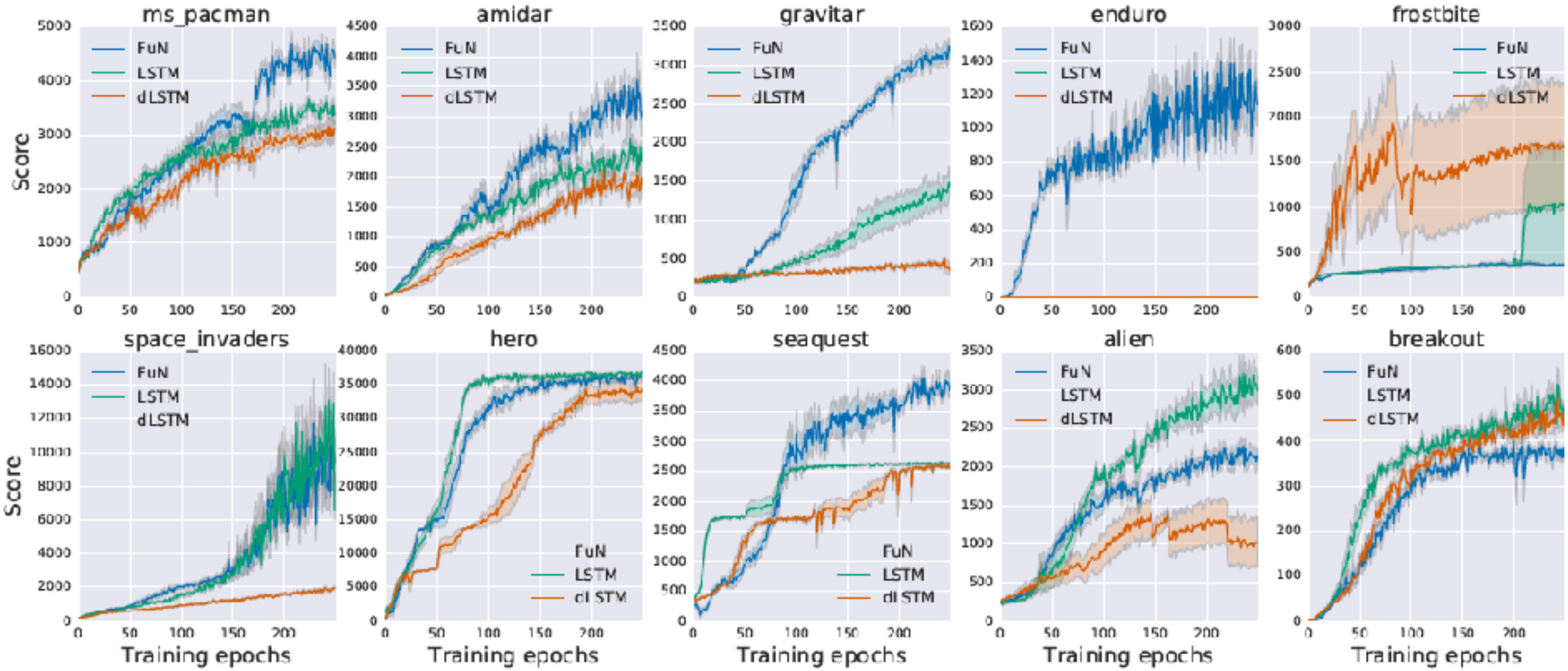
→ 全てにおいて FuN が勝利 = Dilated LSTM の有効性

$c = 1$ でも上がってはいる

※ [Vezhnevets et al., 2017]

結果 - アイディアの正しさ

Dilated LSTM に関する比較:



dLSTM：FuN ではなく通常の A3C にDilated LSTMのみを使用

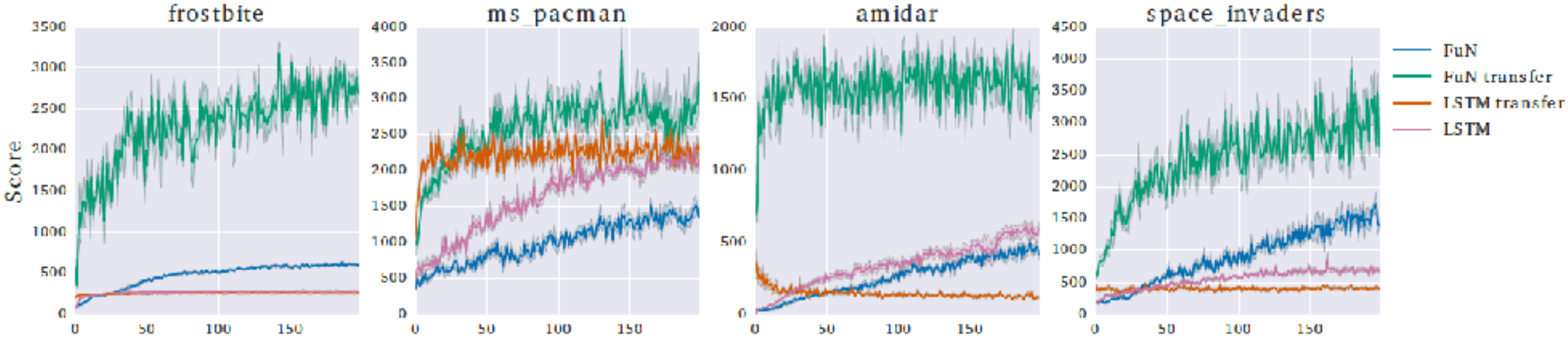
→ 基本的には FuN と 通常LSTM が勝利

= Dilated LSTM は Manager レベルだから有効

※ [Vezhnevets et al., 2017]

結果 - 転移への試み

Action repeat of 4 から Action repeat 無しへの重み転



一定フレーム数 (論文中では 4 frame) の間同じ行動をする**通常のやり方で学習した重み**を，1 frame ごとに行動を意思決定するタスクに転用

その学習しないでの成績 (流石に各時間関係パラメータは 4 倍にする)

→ FuN の高い成績は Manager で学習した上位方策の有用性を意味する

→ 同一タスクだと Maneger の**汎用性の高さ**の証明には**ならない気も？**

※ [Vezhnevets et al., 2017]

感想

End-to-End なサブゴール形成

- 絶対的ではなく相対的なゴール定義 (ある種の**未来方向**予測) というアイディアで成したのは興味深い
  - 固定長時間 c step を可変長にできるとなお良い
    - 長い時間長でゴールを定義したい場合への対処を考えて
  - 目的論的には Option の方がサブゴールと言える

※ [Vezhnevets et al., 2017]

感想

LSTM への依存性

- 不完全情報への対処は所詮 LSTM まかせ
  - 成績で Option-Critic に優っているのは A3C や LSTM のおかげ
  - 固定長 c step サイクルの Dilated LSTM が有効だった？
    - FuNs 自体が Dilated LSTM を使うための構造とも捉えられる
    - 長期的な Dilated LSTM といっても限界が存在するはず

→ より圧縮(記号化)された記憶表現を End-to-End で学習すべき？

引用文献 (スライド中)

[Vezhnevets et al., 2017] Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. *FeUdal Networks for Hierarchical Reinforcement Learning*. ArXiv. Retrieved from http://arxiv.org/abs/1703.01161 (2017).

[Bellmare et al., 2012] Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. *The arcade learning environment: An evaluation platform for general agents*. Journal of Artificial Intelligence Research. (2012).

[Dayan and Hinton, 1993] Dayan, P., and Hinton, G. E. *Feudal reinforcement learning*. In NIPS . Morgan Kaufmann Publishers. (1993).

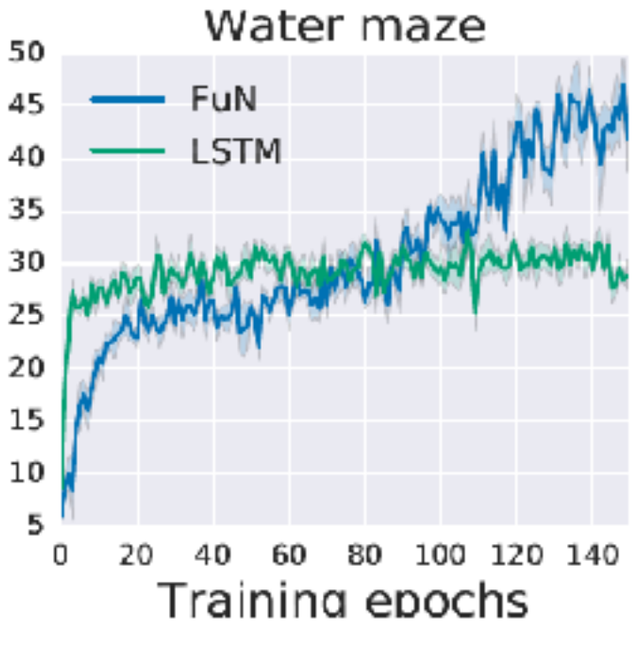
[Sutton et al., 1999] Sutton, R. S., Precup, D., and Singh, S. *Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning*. Artificial intelligence. (1999).

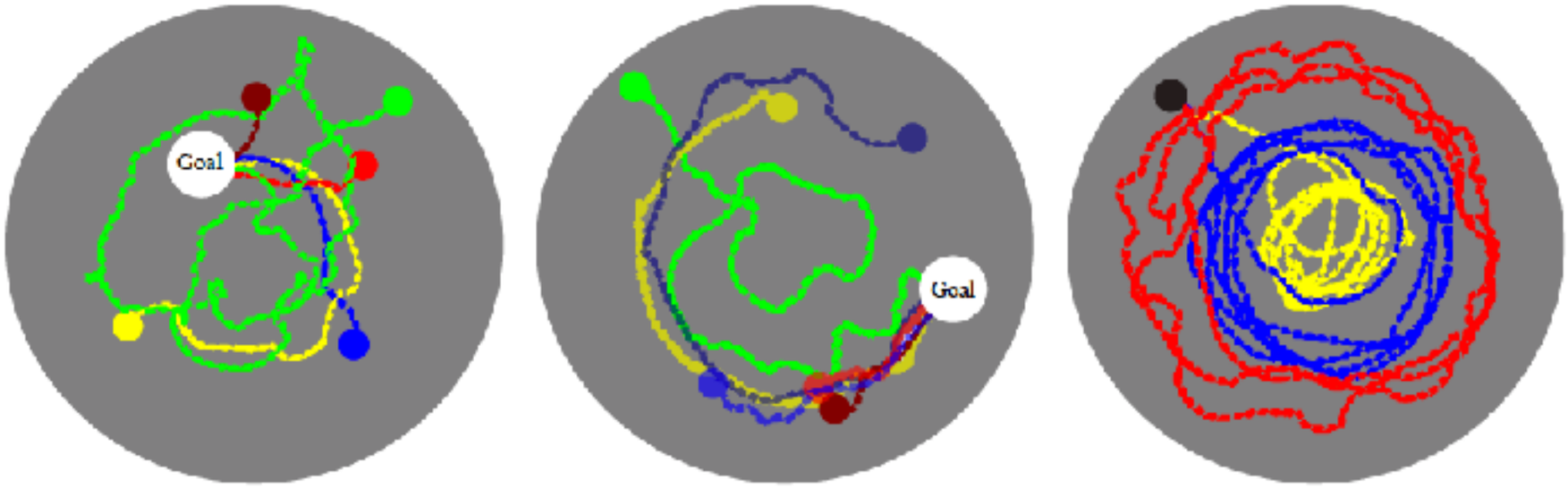
[Bacon et al., 2017] Bacon, P. L., Precup, D., and Harb, J. *The option-critic architecture*. In AAAI. (2017).

[Von Mises–Fisher distribution] https://en.wikipedia.org/wiki/Von_Mises-Fisher_distribution

結果 - オマケ

Water maze で実際に獲得された行動：





Staet (緑)位置からの行動以外，同じ半径で回転して探索するサブ方策が学習される

右端は goal をランダムに設定，200step 固定して学習して得られたサブ方策ケース

※ [Vezhnevets et al., 2017]