

2019 年度
卒業論文
複雑環境下でのロボット学習に向けた
深層状態空間モデルを用いた映像予測

2020 年 2 月
指導教員 松尾豊教授

東京大学工学部システム創成学科
知能社会システムコース
03-180961 近藤生也

概要

近年，深層ニューラルネットワーク (DNN) を使用した機械学習手法の発展を背景としてロボットの制御方策を自ら学習させるロボット学習の研究が進んでいる。ロボット学習において未来の観測，特に自らの行動に対するフィードバックを適切に予測することは，より良い制御方策の獲得に繋がる上，安全性の評価などに欠かせない。時間変化する環境の DNN を用いたモデル化手法として深層状態空間モデル (DSSM: Deep State Space Model) がありこれは昨今深層強化学習の分野で取り入れられているが，DSSM を用いて映像予測を学習することで良い環境の遷移モデルを獲得できることがわかっている。しかし現在一般的に使われている DSSM とその学習方法は各時刻の状態が低次元のベクトルで表現できるという仮定を暗に置いており，高次元な表現を仮定すると学習が進まず複雑な環境の予測問題にスケールさせる方法は自明でない。本研究ではまずこのような問題があることを実験的に指摘した上で，状態の表現に階層性を考えることで DSSM により高次元な状態ベクトルを使っても学習が進むよう改善し，さらに低次元な状態ベクトルを用いた場合よりも良い表現が獲得できることを示す。評価実験では様々な物体が置かれた机上でのロボットマニピュレーションを題材にしたデータセットを用いてロボットの行動で条件付けた映像予測を行う。本研究では単純な DSSM をベースラインに設定し，提案手法の有効性を示した上で，研究の展望と応用の可能性について述べる。

目次

概要	i
第 1 章 序論	1
1.1 本研究の背景	1
1.1.1 ロボット学習と行動条件付き映像予測	1
1.1.2 既存の行動条件付き映像予測手法の問題点	2
1.2 本研究の目的	3
1.3 本論文の構成	3
第 2 章 前提知識	4
2.1 変分自己符号化器 (VAE)	4
2.2 深層状態空間モデル (DSSM)	7
第 3 章 問題設定	11
3.1 データセット	11
3.2 問題設定	11
第 4 章 深層状態空間モデルの問題点	13
4.1 学習が失敗した例	14
4.2 学習が難しくなる理由の考察	15
第 5 章 状態表現の階層性を考慮することによる深層状態空間モデルの拡張	16
5.1 提案手法	16
5.1.1 状態表現の階層性	16
5.1.2 階層的な状態表現の遷移	17
5.1.3 確率モデル・最適化	18

5.2	類似手法との差分	19
第 6 章 実験		21
6.1	実験内容	21
6.1.1	ベースラインの実装	21
6.1.2	提案手法の実装	24
6.2	実験結果	24
6.2.1	定量評価(尤度)	24
6.2.2	定性評価	26
第 7 章 考察		30
7.1	本研究の貢献	30
7.2	課題	31
7.2.1	定性的な改善	31
7.2.2	学習が不安定になる問題	31
7.2.3	初期状態の推論	32
7.2.4	低階層の状態表現の再学習	32
7.3	展望	32
7.3.1	敵対的学习の導入	32
7.3.2	多視点からの映像予測	33
7.3.3	他の構造を持つ潜在表現との併用	33
7.3.4	メタ学習の導入	33
7.3.5	映像予測用データ収集の人による代替	34
7.4	応用	35
7.4.1	実機ロボットへの応用	35
7.4.2	物理シミュレーションの近似	35
7.4.3	微分可能な環境モデルとしての利用	36
第 8 章 結論		38
謝辞		39
参考文献		40

図目次

2.1	VAE のグラフィカルモデル	4
2.2	推論分布を導入した VAE のグラフィカルモデル	5
2.3	VAE を用いて生成された画像の例	6
2.4	DSSM のグラフィカルモデル	7
2.5	DSSM で生成される映像の例 (回転する数字画像)	10
2.6	DSSM で生成される映像の例 (強化学習エージェント)	10
3.1	データセットの模式図	12
3.2	問題設定の模式図	12
4.1	状態変数の次元を変えた時の DSSM の学習曲線	13
4.2	DSSM の学習中にサンプルされた映像の例	14
5.1	獲得される状態表現の包含関係	17
5.2	ベースラインの状態遷移の模式図	17
5.3	提案手法の状態遷移の模式図	18
5.4	提案手法のグラフィカルモデル	18
5.5	PGGAN の学習アルゴリズムの模式図	20
6.1	モデルアーキテクチャ	23
6.2	提案手法の学習曲線	25
6.3	映像予測結果 1	27
6.4	映像予測結果 2	28
6.5	30 フレームの予測結果の比較	29

表目次

6.1	手法ごとの定量評価指標(尤度)	25
-----	-----------------	----

第1章

序論

1.1 本研究の背景

1.1.1 ロボット学習と行動条件付き映像予測

多様な環境で様々なタスクが遂行可能な汎用的なロボット (generalist robots) の開発はロボット工学の最重要課題の一つである [1]. ロボットハードウェアの低価格化, 汎用的なロボットソフトウェアの普及に加え, 近年の急速な深層学習技術の発展を受けてロボットの制御方策を自ら学習させるロボット学習の研究が進んでおり, ロボットで遂行可能なタスクは着実に増えてきている.

ロボット学習において, 将来予測, 特に映像予測を明示的に学習することは,

- ロボット自身が映像予測を用いた行動計画をたてることが可能になる
- 映像予測結果を人が評価することでロボットの行動を予め評価できる

という大きく二つの点からで重要であると言える. 一点目の映像予測を用いた行動計画の例として, Hafner ら [2] は, 強化学習の問題設定において明示的に学習した映像予測モデルを用いることで行動系列をランダムにサンプリングして評価するような簡単なアルゴリズムで効率的なプランニングが可能であることを示した. 二点目の映像予測を人が評価する例として Ebert らによる研究 [3] では学習した映像予測モデルを用いて, ロボットの操作によって予想される物体の移動の軌跡を確率分布として出力することができ, これを用いて人がロボットの行動の正しさを予め判断することができる.

このようにロボット学習における映像予測は重要であるが, 映像予測だけを切り取って研究されることも多い. 映像予測の中でも, ロボットの行動の結果として観測される映像を予測す

る問題設定を行動条件付き映像予測と呼び [4][5][6]，様々な研究がなされてきている。近年高精度な行動条件付き映像予測手法がいくつか提案されており，ロボット学習研究で扱うタスクの高度化を背景にしてこれらの映像予測手法をより複雑な問題設定に適用していくことが考えられるが，いくつかの研究から既存の行動条件付き映像予測は上手く機能しない可能性がわかってきていている。ただしここでいう「より複雑な問題設定」とは具体的には環境中に複数の操作対象の物体が隣接し合って置かれている場合，操作対象が布などの非剛体物である場合など，観測の時間変化に多数のパラメータが関与していたり，複雑な物理法則がはたらいているような問題設定を想定している。

1.1.2 既存の行動条件付き映像予測手法の問題点

行動条件付き映像予測手法は大きく回帰型ニューラルネットワーク (**RNN**) ベースの手法と深層状態空間モデル (**DSSM**) ベースの手法に分けられる。Hafner ら [2] の研究など深層強化学習の問題で映像予測を明示的に行う場合は後者の DSSM ベースの手法が多く採用されるが，映像予測の問題では前者の RNN ベースの手法が多く使われている [7][8]。

RNN ベースの手法は予測した 1 ステップ先の画像を入力にして更に 1 ステップ先の画像を出力するというような，自らの出力を逐次入力する構造を持つ。RNN ベースの手法は DSSM と比較して高精度な映像を生成に長けている反面，近年提案されている RNN ベースの手法には (i) 誤差が蓄積しやすい (ii) 文脈を必要とする という問題点がある。

(i) について，RNN ベースのモデルを用いると常に直前のフレームを参照して次のフレームを予測するため短い期間の予測であれば精度は高くなるが，予測誤差が蓄積していくために長期の予測には向かないことが示されている [2]。

(ii) について，RNN はモデルの内部状態を十分に更新した後でないと適切に予測が行えず，予測を始める前に文脈としてそれより前の数フレームを与える必要があり，この文脈として与えられるフレーム数が少ないと予測が悪化することが知られている [8]。ロボット実機への応用を考えた場合，現在の状態から未来を予測する際に文脈を得るために先に数ステップ行動することは，予測してから行動するという目的意識に反しており実用的でない。このため RNN ベースの手法をロボット実機に応用する際には，文脈として現在の観測という 1 フレームのみ与れば十分機能するように改善する必要がある。このように，RNN ベースの手法は制約がありそもそも実ロボットへの応用に向いていない可能性がある。

一方，DSSM は各時刻の状態をベクトル（状態ベクトル）で表現し，毎時刻ロボットの行動によって状態ベクトルが遷移し，その時刻に観測される画像は状態ベクトルからの写像である

と考えて遷移モデルと写像のモデルを学習する。DSSM は強化学習の分野で長期の予測にも用いられているなど安定した未来の予測に長けているが RNN と比較して画像の生成時に直前の画像を用いないことから高精度な生成は難しく、また映像生成自体を目的にして DSSM を用いた研究は現状少ない。

1.2 本研究の目的

これらの研究背景を踏まえ、行動条件付き映像予測の問題をより複雑な問題設定にスケールさせることを目指し、本研究では特に実ロボットへの応用を重要視して DSSM ベースの行動条件付き映像予測に取り組む。まず DSSM では単純にモデルを大きくするだけでは映像予測の精度があがらずむしろ悪化することを実験的に示しその理由を簡単に考察する。その上で、モデルが獲得できる情報量を徐々に多くするような DSSM の拡張方法を提案し、大きなモデルでも学習が進むように改善・予測精度の向上を図る。さらに行動条件付き映像予測用のデータセットを用いて提案手法の有効性についての定性的・定量的な評価を行い、DSSM を使った際にもより高精度な映像予測を可能にすることを目指す。

1.3 本論文の構成

本論文の構成は以下の通りである。

第 2 章では、前提知識について説明する。

第 3 章では、本研究で扱う問題設定を整理する。

第 4 章では、深層状態空間モデルの問題点について述べる。

第 5 章では、状態表現の階層性を考慮することによる深層状態空間モデルの拡張を提案する。

第 6 章では、実験を行い提案手法の有効性を示す。

第 7 章では、前章までの議論を踏まえて考察し、応用の可能性について述べる。

最後に第 8 章で結論を述べる。

第 2 章

前提知識

本章では、まず深層状態空間モデル (Deep State Space Model, 以下 DSSM) のベースとなる変分自己符号化器 (Variational Auto Encoder, 以下 VAE) について説明し、続いて DSSM の説明を行う。

2.1 変分自己符号化器 (VAE)

変分自己符号化器 (VAE)[9] は深層生成モデルの一種である。VAE では、高次元のデータ $\mathbf{x} \in \mathbb{R}^n$ の背後に比較的低次元の潜在表現 $\mathbf{z} \in \mathbb{R}^m$ があると考え、図 2.1 のようなグラフィカルモデルに従ってデータの分布 $p(\mathbf{x})$ を次式で表す。

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \quad (2.1)$$

正しい $p(\mathbf{x}|\mathbf{z})$ のモデルが得られれば、 \mathbf{z} が与えられたときに $p(\mathbf{x}|\mathbf{z})$ のモデルを使ってそれに対応する \mathbf{x} を生成することができ、また正しい $p(\mathbf{z}|\mathbf{x})$ のモデルが得られると \mathbf{x} のコンパクトな表現としての \mathbf{z} が推論できる。VAE は画像のエンコードやデコードに優れており、深層



図 2.1 VAE のグラフィカルモデル

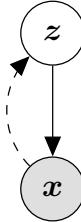


図 2.2 推論分布を導入した VAE のグラフィカルモデル. 実線は生成分布, 点線は推論分布を表す.

強化学習や異常検知など様々な分野に応用されている. [10][11]

ここからは VAE の理論的な説明を行う. まず VAE では, 以下の 2 つの仮定を置く.

$$p(z) = \mathcal{N}(z|0, \mathbf{I}) \quad (2.2)$$

$$p(z|x) = \mathcal{N}(z|\mu(x), \sigma(x)) \quad (2.3)$$

式 (2.2) は, 潜在表現の空間が標準正規分布に従うという仮定であり, 式 (2.3) は, x に条件づけられた潜在変数の分布も正規分布に従うという仮定となっている.

VAE ではさらに x が与えられたときの z の条件付き確率 $p(z|x)$ を近似する $q(z|x)$ を導入する. この $q(z|x)$ は推論分布, 近似分布などと呼ばれ, グラフィカルモデル中に記述すると図 2.2 となる. ただし $p(z|x)$ を直接考えるのは, $p(x|z)$ を表すニューラルネットのパラメータで解析的に正しい $p(z|x)$ を表すことが難しいためである. この z の推論モデル $q(z|x)$ と x の生成モデル $p(x|z)$ を適当なニューラルネットワークを使ってモデル化すると, それらのパラメータは x のデータ集合が与えられた際に最尤推定によって求めることができる.

最尤推定の際には式 (2.2) の対数尤度をとって導出される, 以下の変分下限を用いる.

$$\log p(x) = \log \int p(x|z)p(z)dz \quad (2.4)$$

$$= \log \int q(z|x) \frac{p(x|z)p(z)}{q(z|x)} dz \quad (2.5)$$

$$\geq \int q(z|x) \log \frac{p(x|z)p(z)}{q(z|x)} dz \quad (2.6)$$

$$= \int q(z|x) \log p(x|z) dz - \int q(z|x) \log \frac{q(z|x)}{p(z)} dz \\ = \mathbb{E}_{z \sim q(z|x)} [\log p(x|z)] - D_{KL}(q(z|x) \| p(z)) \quad (2.7)$$

式 (2.6) にはイエンセンの不等式を用いている. 式 (2.5) では, 数値計算をする都合上式

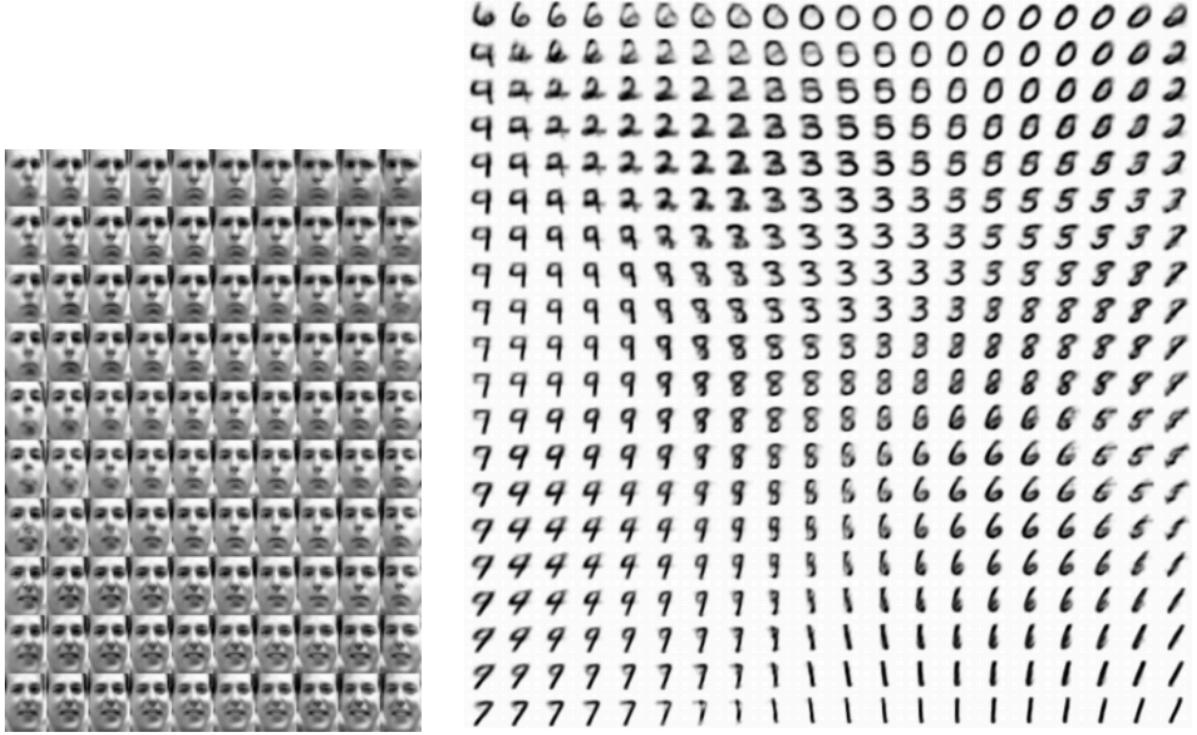


図 2.3 VAE を用いて生成された画像の例 ([9] より引用)

の z の周辺化が難しいため、先述した $q(z|x)$ を導入し、更に式 (2.7) 第一項で $q(z|x)$ からサンプリングされる L 個の z を用いて $\frac{1}{L} \sum_l \log p(x|z)$ でモンテカルロ近似することによって、周辺化を排除している。ただし通常 $L = 1$ で計算される。式 (2.7) の第 2 項の D_{KL} (カルバックリーブラー距離) は、いま $p(z)$, $q(z|x)$ 共にガウス分布を仮定しているため解析的に計算することができ、また式 (2.7) の第 1 項は、 $p(x|z)$ の分布に正規分布やベルヌーイ分布を仮定することで二乗誤差やクロスエントロピー誤差として計算できる。この変分下限を目的関数にすることで生成モデル $p(x|z)$ と推論モデル $q(z|x)$ を同時に学習することができる。

以上が VAE の概要である。このように学習された VAE は、事前分布 $p(z)$ から z をサンプリングし、デコーダを通すことで図 2.3 のような新たなデータを生成することができる。左は顔画像のデータセット Frey Face を用いて学習した VAE によって新たに生成された画像、右は MNIST で学習した VAE によって新たに生成された画像である。この 2 つの図は、それぞれ 2 次元の潜在変数をおいた VAE で学習したのち z の値を少しずつずらしながら新しい画像を生成したもので、なめらかに変化した画像が生成できていることから画像空間という高次元の空間上のデータを上手く低次元の空間に埋め込んでいることがわかる。

2.2 深層状態空間モデル (DSSM)

深層状態空間モデル (Deep State Space Model: DSSM, Deep Karman Filters, Deep Markov Model, 単に SSM などとも呼ばれる)[12][13]について説明する。VAE がデータ一つ一つの潜在表現を考えるのに対し, DSSM は時間変化があるデータの各時刻の潜在表現を考えて更にその潜在表現の時間変化をモデル化したものであり, VAE を時系列方向に拡張したモデルとみなすこともできる。DSSM ではこの潜在表現のことを状態表現と呼ぶ。

DSSM は特に深層強化学習の分野で用いられており, エージェントがある環境中で行動を起こした結果として視覚フィードバックや報酬フィードバックなどが得られるときに, DSSM で将来の観測を正しく予測できるよう学習することで良い環境の状態表現を獲得することができ, さらにその状態表現を用いることで良い行動方策が得られる [14].

本論文では図 2.4 のようなグラフィカルモデルで表される DSSM のモデルを考えるが, 行動系列が与えられない場合や, 強化学習のように環境の状態に応じて報酬が与えられる問題においても以下の説明は同様にして考えることができる。

ここから DSSM の理論的な説明を行う。DSSM は式 (2.8) 式が示すように, 初期状態 s_0 と行動系列 $a_{1:T}$ で条件付けられた観測 $o_{1:T}$ を予測するモデルである。ただし T は観測の系列長である。DSSM では各時刻の観測 o_t は各時刻の状態表現 s_t から生成されると考え, 各時刻の状態表現 s_t を明示的に推論する。初期状態 s_0 の定め方は任意だが, 多くの場合ゼロベク

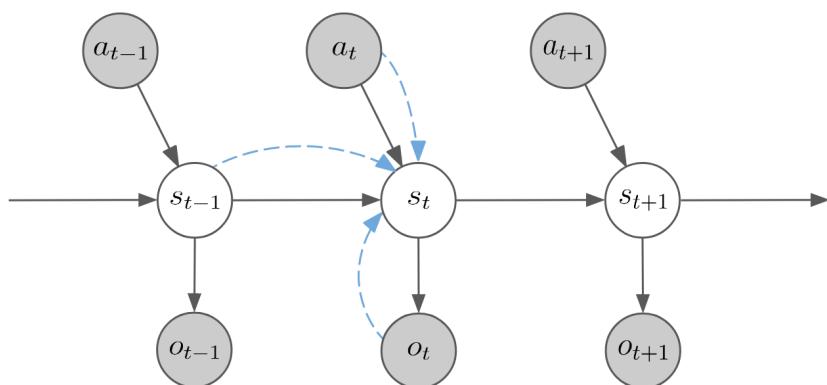


図 2.4 DSSM のグラフィカルモデル。実線は生成分布, 点線は推論分布を表す。

トルが用いられる。以降、簡単のため条件 s_0 は省略する。

$$\begin{aligned} p(o_{1:T}|a_{1:T}, s_0) &= \prod_{t=1}^T p(o_t|a_{1:t}, s_0) \\ &= \prod_{t=1}^T \int p(o_t|s_t)p(s_t|s_{t-1}, a_t)ds_t \end{aligned} \quad (2.8)$$

DSSM では以下の仮定をおく。式 (2.9) は各時刻の状態ベクトル s_t は正規分布に従うという仮定で、式 (2.10) は o_t で条件付けられた各時刻の状態ベクトル s_t も正規分布に従うという仮定である。

$$p(s_t|s_{t-1}, a_t) = \mathcal{N}(s_t|\mu(s_{t-1}, a_t), \sigma(s_{t-1}, a_t)) \quad (2.9)$$

$$p(s_t|s_{t-1}, a_t, o_t) = \mathcal{N}(s_t|\mu(s_{t-1}, a_t, o_t), \sigma(s_{t-1}, a_t, o_t)) \quad (2.10)$$

さらに DSSM では VAE と同様に s_t の推論モデル $q(s_t|s_{t-1}, a_t, o_t)$ を導入し、生成過程と推論モデルをニューラルネットによってモデル化し、最尤推定によってそれらのパラメータを求める。

以下の変分下限で最尤推定を行う。

$$\begin{aligned}
\log p(o_{1:T}|a_{1:T}) &= \log \prod_{t=1}^T p(o_t|a_{1:t}) \\
&= \log \prod_{t=1}^T \int p(o_t|s_t) p(s_t|s_{t-1}, a_t) ds_t \\
&= \sum_{t=1}^T \log \int p(o_t|s_t) p(s_t|s_{t-1}, a_t) ds_t \\
&= \sum_{t=1}^T \log \int q(s_t|s_{t-1}, a_t, o_t) \frac{p(o_t|s_t)p(s_t|s_{t-1}, a_t)}{q(s_t|s_{t-1}, a_t, o_t)} ds_t
\end{aligned} \tag{2.11}$$

$$\geq \sum_{t=1}^T \int q(s_t|s_{t-1}, a_t, o_t) \log \frac{p(o_t|s_t)p(s_t|s_{t-1}, a_t)}{q(s_t|s_{t-1}, a_t, o_t)} ds_t \tag{2.12}$$

$$\begin{aligned}
&= \sum_{t=1}^T \left(\int q(s_t|s_{t-1}, a_t, o_t) \log p(o_t|s_t) ds_t \right. \\
&\quad \left. - \int q(s_t|s_{t-1}, a_t, o_t) \log \frac{p(s_t|s_{t-1}, a_t)}{q(s_t|s_{t-1}, a_t, o_t)} ds_t \right) \\
&= \sum_{t=1}^T \left(\mathbb{E}_{s_t \sim q(s_t|s_{t-1}, a_t, o_t)} [\log p(o_t|s_t)] \right. \\
&\quad \left. - \mathbb{E}_{s_{t-1} \sim q(s_{t-1}|s_{t-2}, a_{t-1}, o_{t-1})} [\text{D}_{\text{KL}}(q(s_t|s_{t-1}, a_t, o_t) \| p(s_t|s_{t-1}, a_t))] \right)
\end{aligned} \tag{2.13}$$

式 (2.12) にはイエンセンの不等式を用いている。式 (2.11) では、 s_t の周辺化を排除するために先述した近似分布 $q(s_t|s_{t-1}, a_t, o_t)$ を導入し、さらに式 (2.13) で s_t, s_{t-1} をサンプリングしてモンテカルロ近似を行う。式 (2.13) の第 2 項のカルバックライブラ一距離は、いま $p(s_t|s_{t-1}, a_t)$, $q(s_t|s_{t-1}, a_t, o_t)$ 共にガウス分布を仮定しているため、解析的に計算することができ、また式 (2.7) の第 1 項は、 $p(o_t|s_t)$ の分布に正規分布やベルヌーイ分布を仮定することで尤度が計算できる。

この変分下限を目的関数にすることで各モデルを学習することができる。また学習時には 10 フレーム程度先までの予測を行うのが一般的であるが、学習時の予測の長さとモデルの評価時の予測の長さを揃える必要はなく、評価時にはより長期の予測も可能である。

最後に s_0 の求め方について、初期状態が必ず一定となる問題設定ではゼロベクトルなど一定の値に決め打ちすることができるが、様々な初期状態が考えられる場合は何らかの方法で推論する必要がある。本研究では慣例に習って s_0 を $s_0 \sim q(s_0|\vec{0}, \vec{0}, o_0)$ によって定め、(2.13) の第 2 項の s_0 のサンプリングは ($o_{t=-1}$ や $a_{t=-1}$ が与えられない前提を考えるため) これで

置き換える。 s_0 の推論方法については議論の余地があると考えられ、これについては考察「初期状態の推論」で触れる。

以上が DSSM の概要である。DSSM を用いると、図 2.5、図 2.6 のように数フレーム先を予測することができる。図 2.5 は、行動系列として回転情報が与えられており、回転する画像の予測を行う。図 2.6 は四足歩行するシミュレータ上のエージェントの第三者視点からの観測を予測しており、行動系列としてエージェントの各関節に加えられる力が与えられている。DSSM は長期の予測をした際にも生成自体が安定している点で自己回帰モデルと比較して優れている。



図 2.5 DSSM で生成される映像の例 (回転する数字画像). [12] より引用

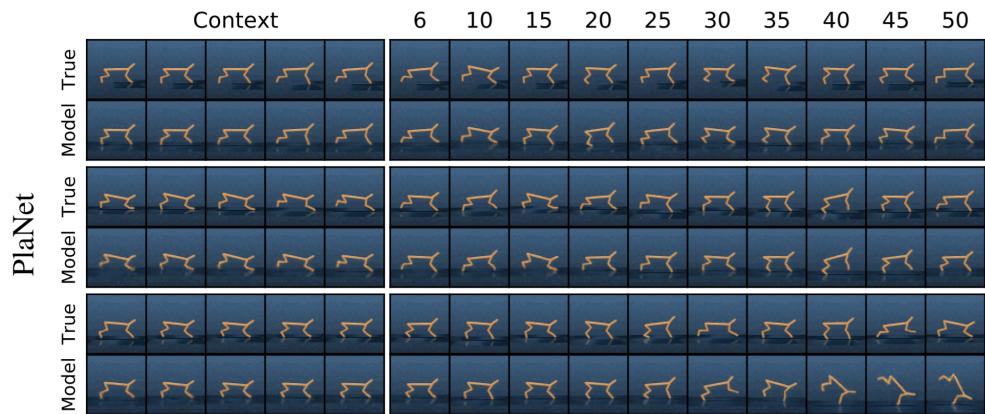


図 2.6 DSSM で生成される映像の例 (強化学習エージェント). [2] より引用

第3章

問題設定

第3章では、まず本研究で扱うデータセットについて説明し、続いて問題設定について述べる。

3.1 データセット

本研究では、映像予測の研究分野で一般的に用いられる BAIR Push Dataset[15] を用いて各手法を評価する。データセットの模式図を図3.1に示す。BAIR Push Datasetは行動条件付き映像予測と行動条件をつけない映像予測のどちらの研究でも用いられるデータセットであり、カリフォルニア大学バークレー校によって制作・公開されている。このデータセットは、雑多な物体がおかれた机の上をロボットアームがランダムに搔き乱すようにして様々なデータが記録されており、今回はその中から行動系列 \vec{a} と固定視点から観測された画像系列 \vec{o} を用いる。今回用いる行動系列 \vec{a} は、具体的にはロボットのエンドエフェクタの位置姿勢の命令値で、4次元のベクトルになっている。観測画像は 64x64 サイズの RGB 画像で、これらのデータは 10hz で撮られている。また訓練時と評価時に使われる物体は揃えられている。

3.2 問題設定

本研究では行動条件付き映像予測の問題を解く。問題設定の模式図を図3.2に示す。具体的には、初期観測 o_0 とある行動主体が実行した行動系列 $a_{1:10}$ が与えられたときに $o_{1:10}$ を生成し、その生成される映像の尤度を高めることを目指す。ただし訓練時には行動系列と観測系列の組 $\{\vec{a}, \vec{o}\}$ の訓練用のデータセットを用いることができ、評価時には、訓練データには含まれないが訓練データと同じ条件で収集された評価用のデータセットを用いる。本研究の目的は、

この行動条件付き映像予測問題においてベースラインに DSSM を設定し, DSSM が BAIR Push Dataset のような複雑な映像の予測問題にスケールするよう拡張することである.

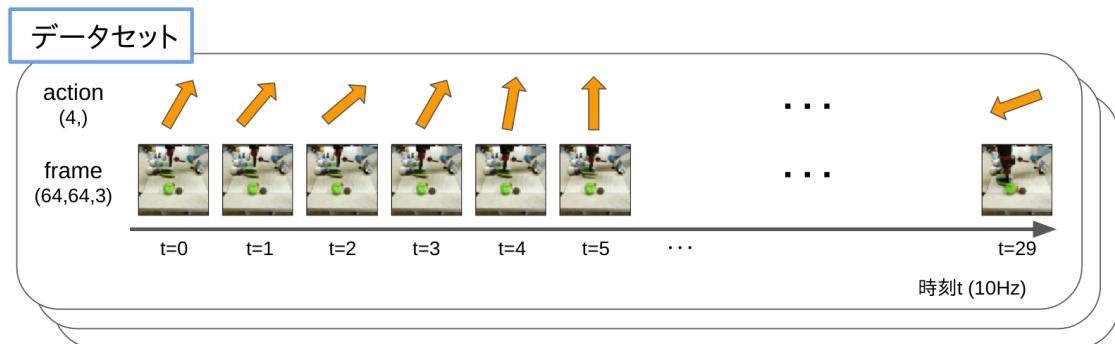


図 3.1 データセットの模式図. 括弧内の数字はデータの次元を示す.

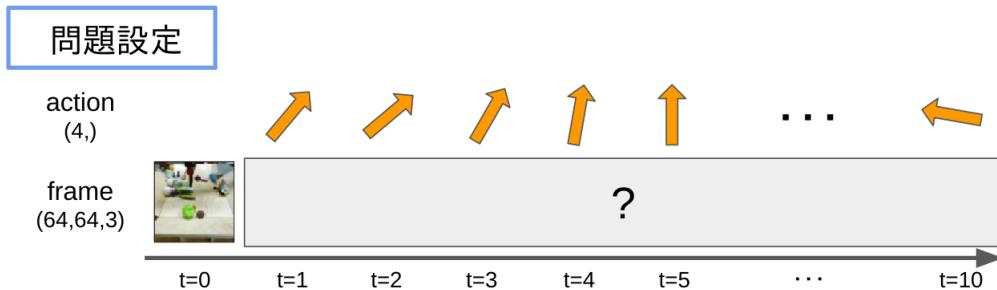


図 3.2 問題設定の模式図. 初期観測と行動系列が与えられたときに将来の観測を生成する.

第 4 章

深層状態空間モデルの問題点

深層状態空間モデルを用いてより複雑な環境の映像予測を考える場合、潜在的な情報が多くなるはずであるためより大きな状態表現を扱えるようモデルを大きくする必要がある。素朴には状態変数の次元を大きくすることがよいと考えられるが、予備実験の中で深層状態空間モデルは状態変数の次元を大きくすると学習が困難になる、または学習が安定しにくくなることがわかった。まずこのことを示した上で、この理由を考察する。

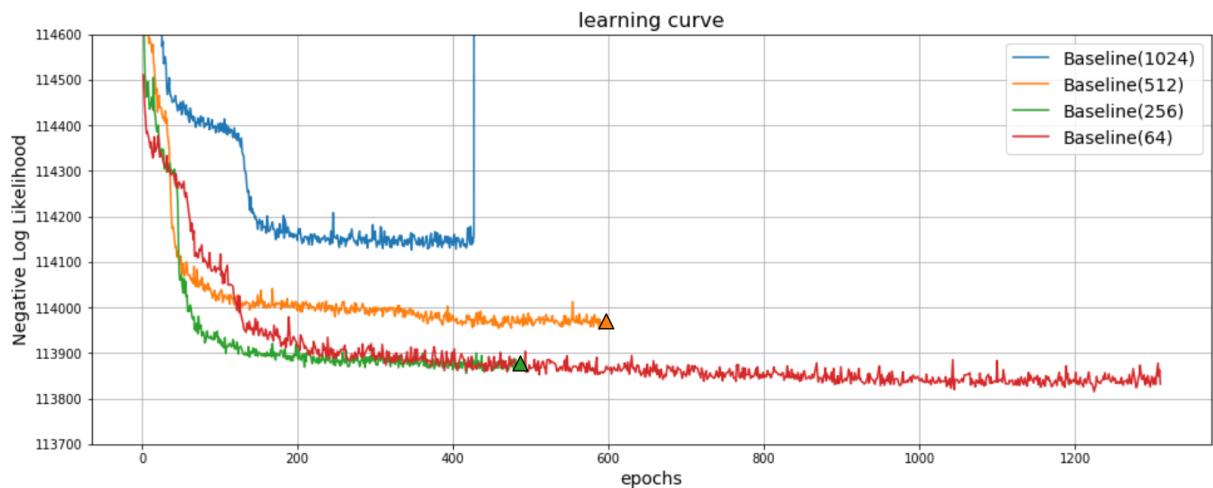


図 4.1 状態変数の次元を変えた時の DSSM の学習曲線。横軸が epoch 数で縦軸が目的関数の値である。グラフ中に三角で示されるのは目的関数の値に NaN が output されてしまったことを示す。Baseline(1024) は 400epoch を過ぎたあたりで目的関数の値が発散し異常な値をとっている。

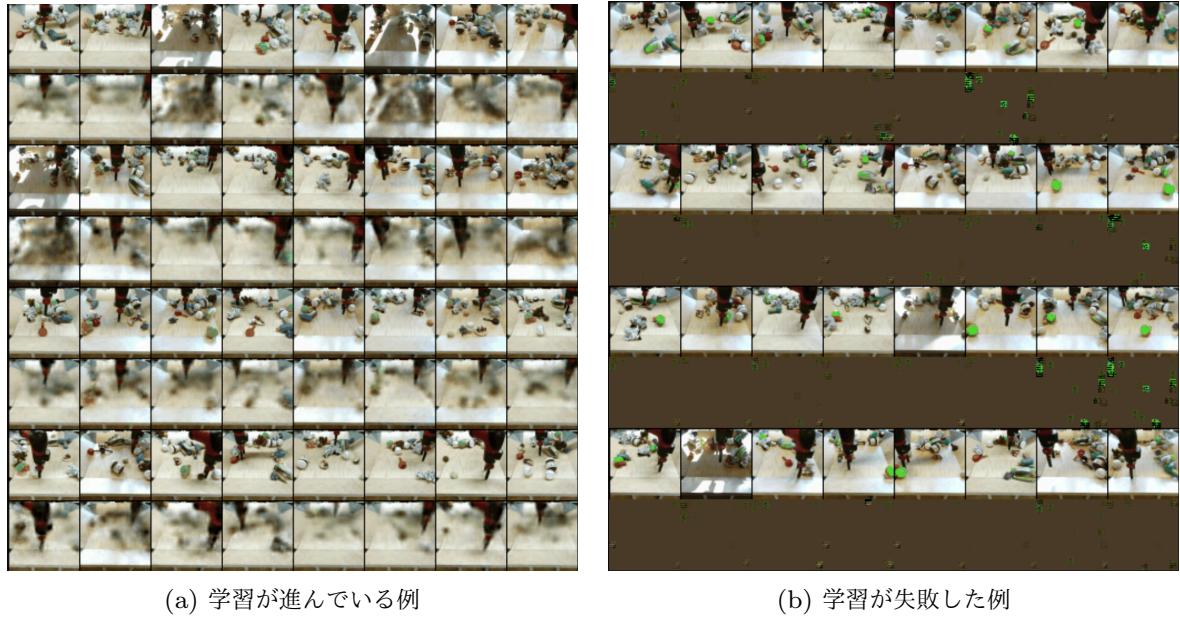


図 4.2 DSSM の学習中にサンプルされた映像の例. 上から奇数段目が正解映像で, 上から偶数段目が一段上の映像の行動条件付き映像予測結果である. 10 フレームの映像予測を行っているが, 図ではある一時刻の観測のみが示されている

4.1 学習が失敗した例

図 4.1 は, 通常の DSSM モデルを状態変数の次元を何通りかに変えてモデルを構築し, 学習時の訓練用データでの目的関数の値の増減をプロットしたものである. 状態変数を比較的低次元の 64 次元に設定すると順調に学習が進み, 図 4.1 の左図に示すように少しづつ近い映像が出力されるようになる. しかし状態変数の次元を 512 次元に設定すると明らかに精度が悪化し, 1024 次元にするとさらに悪化している. 図 4.1 の右図は 1024 次元での実験で目的関数の値が発散した後の生成映像の様子で, 他の次元で学習が失敗した場合も同じような映像が生成される. 256 次元や 512 次元での実験で途中から目的関数の値に NaN が output されてしまうのは, 第 6 章の「実験の安定化」で述べるように, カルバックライブラリ距離を小さくすることに失敗しゼロ除算が発生したためである. 1024 次元で値が発散している理由は定かでないが, 256 次元, 512 次元と同様の理由であると考えられる.

ELBO が NaN をとったり発散したりすることは数値計算の丸め誤差などの理由も関わるので一旦考えないとしても, 明らかに状態変数の次元を大きくすることで精度が下がっていることが図 4.1 からわかる.

4.2 学習が難しくなる理由の考察

VAE の学習では前節で述べたような問題は起こらないことを踏まえると、DSSM で学習が難しいのは状態変数の遷移の部分であると考えられる。さらに、状態変数に高次元を仮定したとき、状態変数の遷移の部分では状態変数の生成モデル・推論モデルともに高次元ベクトルから高次元ベクトルへの写像を学習する必要が生じてまずこの部分で学習に時間がかかりやすくなっている、さらに推論モデルが十分に学習されないまま状態変数の事前分布と事後分布のカルバックライブラ一距離の最小化が図られてしまうので、学習が安定しにくくなっていると考えられる。

のことから、単に状態変数を高次元にすることは DSSM の性質上適切ではなく、DSSM をより複雑な環境を扱う問題にスケールさせるためには他の方法を考える必要がある。以上の予備実験を受け、次章ではシンプルな DSSM の拡張方法を提案する。

第 5 章

状態表現の階層性を考慮することによる深層状態空間モデルの拡張

第 4 章の問題を受け、本章ではシンプルな DSSM の拡張方法を提案する。まず提案手法について述べ、続いて既存の類似手法との差分について述べる。

5.1 提案手法

第 4 章で述べたようにベースラインの DSSM では潜在変数の次元を大きくすると学習がうまくいかなかった。しかし予備実験で得た、低次元の潜在変数を用いたときには部分的に学習が進んだという事実をヒントにし、状態変数の次元を大きくしていく方向性はそのままで状態表現の階層性を考えることにより、より複雑な問題設定においても学習可能な DSSM の拡張方法を提案する。

5.1.1 状態表現の階層性

はじめに、ベースラインの DSSM において潜在変数の次元を変えた時に獲得される情報について考察する。低次元の状態変数で獲得できる情報は高次元の状態変数を用いた場合にも当然獲得できると考えた場合、図 5.1 のように高次元の状態変数が持つ情報は低次元の状態変数が持つ情報をほぼ内包していると考えることができる。ここで状態変数の次元をより大きくしたときに精度がむしろ悪化することが問題であったが、これは第 4 章で述べたとおり、状態変数の次元が大きくなったときに高次元ベクトルから高次元ベクトルへの写像を学習する必要が生じるべき写像先がなかなか定まらないことが原因であると考えられ、何らかの方法で遷移モデルの学習を補助することで図のようにより多くの情報を獲得できる可能性がある。

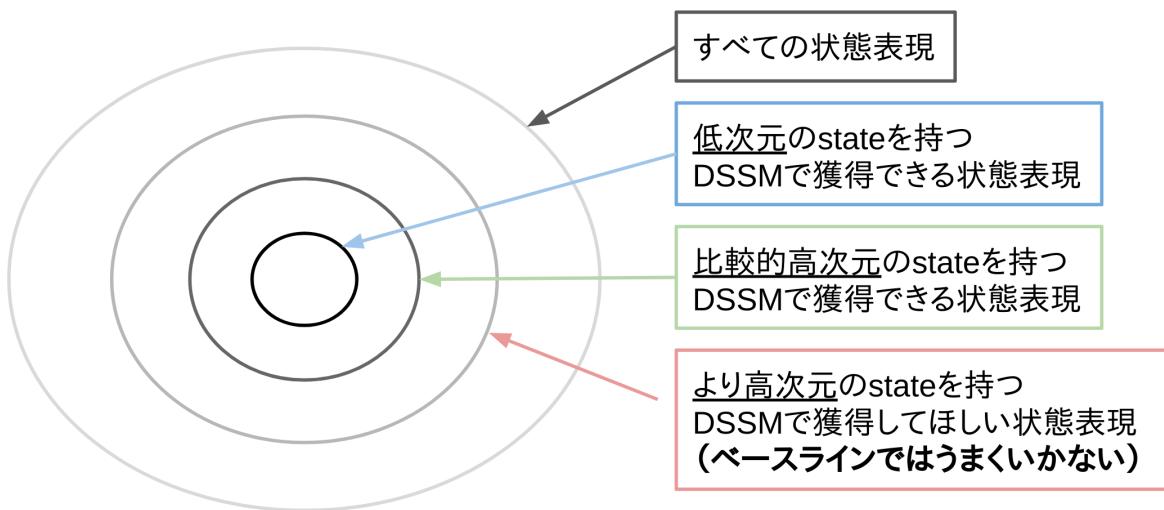


図 5.1 獲得される状態表現の包含関係

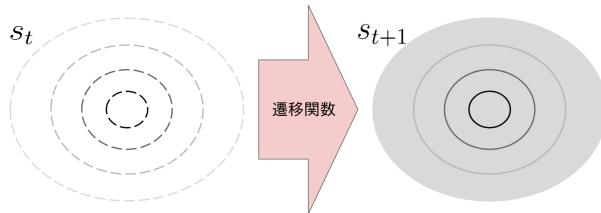


図 5.2 ベースラインの状態遷移の模式図。白色が遷移前、灰色が遷移後の状態表現を表す。

5.1.2 階層的な状態表現の遷移

ベースラインの状態表現の遷移は図 5.2 が示すように状態変数が持つすべての情報を一度に変換することを考えているが、直感的に一括で変換することは学習が難しいと思われる。状態表現を一括で変換する代わりに、前節のような階層性の概念を導入することで図 5.3 のように簡単に遷移が学習できる部分から順に遷移させていくような方法を考えることができる。

このような階層的な状態表現の遷移を考えると、はじめから高次元の状態表現の遷移を考えずに学習の習熟度に合わせて徐々に高次元の状態ベクトルの遷移を学習することができ、学習がスムーズに進みやすくなると考えられる。

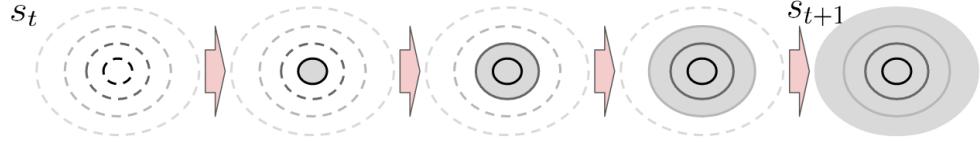


図 5.3 提案手法の状態遷移の模式図. 白色が遷移前, 灰色が遷移後の状態表現を表す.

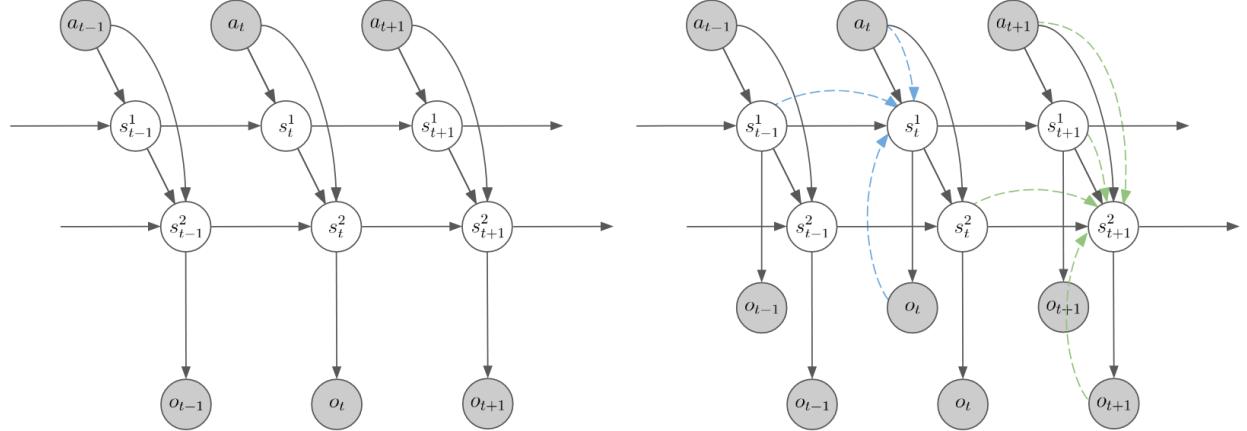


図 5.4 提案手法のグラフィカルモデル. 点線の s^1, s^2 の推論分布は簡単のため時刻 $t, t-1$ でのみ記載している. 左側の図が評価時の生成過程を示しており, 右側の図が訓練時のデータの流れを推論分布を含めて示している. 右図について, 2つずつ記載されている o_t は同じデータを示すが. 異なる s から独立に生成されることを明示している.

5.1.3 確率モデル・最適化

ここまでで状態変数の階層性とその遷移を考えたが, この階層性の仮定は DSSM の性能の向上に十分寄与しうると考え, 状態変数の階層性の解釈を DSSM に取り入れ以下のようなモデルとその最適化アルゴリズムを提案する.

提案手法のグラフィカルモデルを図 5.4 に示す. 提案手法は, DSSM の状態変数を N 層に階層化したモデルである. 図 5.4 は 2 階層の提案モデルを表している. 図 5.4 の上側の状態変数から一階層の状態変数 (s^1)・二階層の状態変数 (s^2) と呼ぶことになると一階層の状態変数が低次元ベクトル, 二階層の状態変数が高次元ベクトルになっており, 高次元の状態変数の生成・推論時に低次元の状態変数を用いるようなモデルとなっている. 高次元の状態変数の遷移時に低階層の状態変数を用いて写像先に関する情報を補助的に与えることで, 学習を安定化させる効果が期待される. またモデルの評価時には, 最高層の観測の生成モデル $p(o_t|s_t^N)$ を

Algorithm 1 N 階層 DSSM の学習アルゴリズム

Require: 階層数 N **for** $i = 1$ to N **do** **while** i 階層のネットワークの学習が収束していない **do** $1 \sim i - 1$ 階層のネットワークのパラメータを固定し, i 階層のネットワークを以下の目的関数 L で学習する

$$L(a_{1:T}, o_{1:T}) =$$

$$\begin{cases} \sum_{t=1}^T (\mathbb{E}_{s_t^i} [\log p(o_t | s_t^i)] - \mathbb{E}_{s_{t-1}^i} [\text{D}_{\text{KL}}(q(s_t^i | s_{t-1}^i, a_t, o_t) \| p(s_t^i | s_{t-1}^i, a_t))]) & (i = 1) \\ \sum_{t=1}^T (\mathbb{E}_{s_t^i} [\log p(o_t | s_t^i)] - \mathbb{E}_{s_{t-1}^i} [\text{D}_{\text{KL}}(q(s_t^i | s_{t-1}^i, a_t, o_t, s_t^{i-1}) \| p(s_t^i | s_{t-1}^i, a_t, s_t^{i-1}))]) & (i > 1) \end{cases}$$

end while**end for**

用いる。

次に提案手法の学習アルゴリズムをアルゴリズム 1 に示す。この学習アルゴリズムは前節「階層的な状態表現の遷移」で述べた、習熟度に合わせて徐々に高次元の状態ベクトルの遷移を学習するという考え方に基づいており、これにより安定した学習が見込める。今回簡単のために高階層の潜在表現の学習時にはそれより低階層の状態表現の学習を止めているが他の方法も考えられ、これについては考察「低階層の状態表現の再学習」で述べる。

5.2 類似手法との差分

提案手法と類似手法の差分について整理する。DSSM 自体を用いた映像予測の研究はこれまであまりされていないが、これは第 1 章でも述べたとおり DSSM は自己回帰モデルと比べて高精度な生成には向いていないためだと考えられる。そのため本節では階層性を考慮した画像生成と映像生成の既存研究について取り上げる。

まず、モデルに階層性を取り入れた画像生成の先行研究として DRAW[16], PGGAN[17] がある。DRAW は VAE ベースの深層生成モデルであり、潜在変数を複数用意し階層的にすることでデータの潜在表現に単純な正規分布を仮定しないより複雑な表現を可能にしている。DRAW は一度にすべての潜在表現を学習するが、時系列データを扱う本提案手法では毎時刻

の状態表現の遷移を学習することが難しいため潜在表現を順に学習するアルゴリズムを採用している。VAE ベースで潜在表現の階層性を取り入れた研究としては他にも [18][19][20] がある。PGGAN は図 5.5 に示すような敵対的生成モデル [21] を用いた画像生成手法で、低解像度の画像の生成からはじめ、学習が進むにつれてニューラルネットワークモデルを徐々に多層にすることで高解像度の画像生成を行う。低次元の隠れ変数を変数から学習していくという点で提案手法と似ているが、PGGAN は何も条件付けしない生成を行っており、過去の状態や行動で条件付けして画像を生成する行動条件付き映像予測の問題設定に用いる方法は自明ではなかった。

次に、映像予測で階層的なモデルを扱う手法とし FutureGAN[22]、て VRNN[23] がある。FutureGAN は先述の PGGAN を映像予測に応用した手法で、過去の 6 フレーム程度の映像を入力とし未来の 25 フレーム程度の映像を予測することができる。VRNN は直前のフレームを用いて次のフレームを予測する自己回帰モデルになっており、潜在変数から画像を出力する過程で得られるいくつかの隠れ変数を次のフレームの予測にも用いるようにして潜在変数の階層性を考えたモデルになっている。FutureGAN や VRNN は予測する際に過去数フレームの映像を用意する必要があるが、ベースラインの DSSM や提案手法では初期状態の推論用に過去 1 フレームのみ使うことを前提としたモデルとなっている。

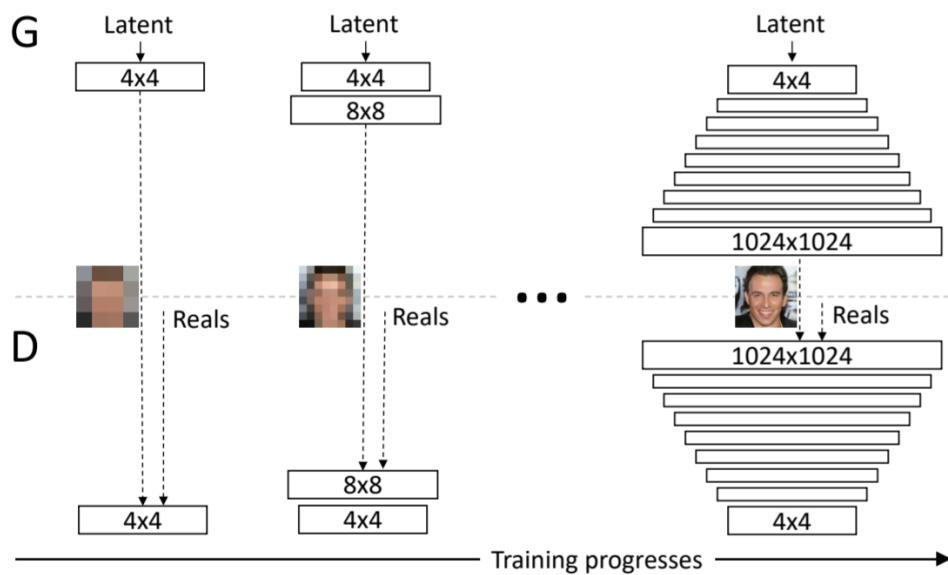


図 5.5 PGGAN の学習アルゴリズムの模式図 ([17] より引用)

第 6 章

実験

第 5 章で述べた提案手法の有効性を検証するために, BAIR Push Dataset[15] を用いて評価実験を行った. 本章では実験の内容について説明した後に実験結果について述べる.

6.1 実験内容

第 5 章で述べた提案手法とベースラインの比較を行う. ベースラインは第 2 章で述べたシンプルな DSSM とし, 状態ベクトルの次元を 64, 256, 512, 1024 の 4 通りに変えて実験を行う. 提案手法は 64 次元と 512 次元の二階層の状態ベクトルを持つモデルと, 64 次元と 512 次元と 1024 次元の三階層の状態ベクトルを持つモデルとした. ベースラインと提案手法の実装の差は必要最小限にとどめ, どちらも学習時には 10 フレーム先までの予測を行った. またパラメータの最適化にはそれぞれ確率的勾配降下法アルゴリズム Adam[24] を用いた. 評価の際には, 定量評価として予測誤差 (負の対数尤度) を測り, 合わせて定性評価も行う.

6.1.1 ベースラインの実装

深層学習フレームワーク Pytorch[25] と, 深層生成モデルライブラリ Pixyz[26] を主に用いた. 学習用データの読み込みとその最適化には深層学習フレームワーク TensorFlow[27] を用いた. また実装では, DSSM を用いた強化学習手法である PlaNet の公開実装 [28] を参考にした. まずベースラインの実装を説明した後, 提案手法の実装について述べる.

モデルアーキテクチャ

ベースラインは 4 つの部分モデルから構成される.

- デコーダー $p(s_t|x_t)$
- エンコーダー $q(x_t|h_t)$
- 遷移モデル (事前分布) $p(s_t|s_{t-1}, a_t)$
- 遷移モデル (事後分布) $q(s_t|s_{t-1}, a_t, h_t)$

遷移モデル (事前分布) とデコーダーが生成モデルに相当し, 提案手法のグラフィカルモデルの実線部分を表す. 遷移モデル (事後分布) とエンコーダーが推論モデルに相当し, 提案手法のグラフィカルモデルの点線部分を表している.

デコーダー・エンコーダー

図 6.1a にアーキテクチャの模式図を示す. PlaNet に倣い, デコーダー/エンコーダー モデルには WorldModel[10] の論文中に記載されているモデルを採用した. デコーダーは各時刻の状態ベクトル s_t を全結合層で 1024 次元の隠れ変数に変換したあと, 4 層の逆畳み込み層で観測画像と同じサイズである 64x64 サイズの RGB 画像に変換する. 出力は各ピクセルごとに正規分布をおくが, 本研究では簡単のためその分散はそれぞれ 1 に固定している. エンコーダーは各時刻の観測を 4 層の畳み込み層と 1 層の全結合層で 1024 次元の隠れ変数 h_t に変換する.

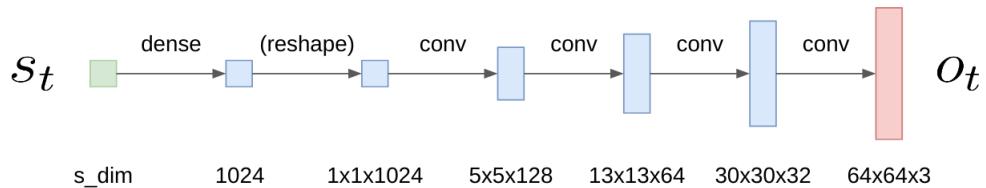
本研究では, 状態変数の次元を変えた際にも隠れ層のパラメータ数などは一切変えなかった.

遷移モデル

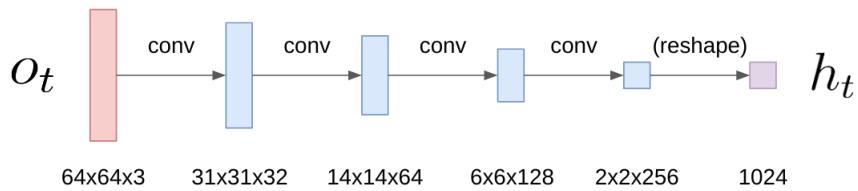
図 6.1b にアーキテクチャの模式図を示す. 遷移モデルには, ある時刻の状態ベクトル s_t を生成過程に従って生成する事前分布モデルと, ある時刻の観測 o_t も与えられたときに s_t を推論する事後分布モデルの 2 種類を用意する.

この 2 つのモデルはアーキテクチャはほとんど共通で入力とするデータのみが異なる. 事前分布モデルでは一つ前の時刻の状態 s_{t-1} と行動 a_t を入力とし, 事後分布ではそれに加えその時刻の観測 o_t をエンコードして得たデータ h_t を入力とする. 出力はどちらのモデルも次の時刻の状態ベクトルの分布の母数 (平均と標準偏差) である. アーキテクチャは, 入力情報をすべて連結して一度全結合層で変換したものを, 続けて平均を求める全結合層と標準偏差を求める全結合層のそれぞれで変換し, 求めたい分布の平均と標準偏差を出力する.

デコーダーモデル

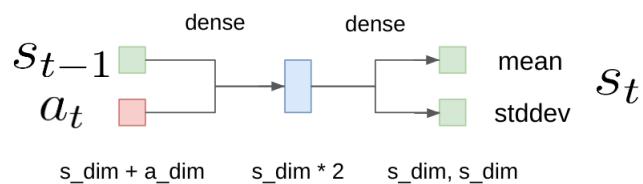


エンコーダーモデル

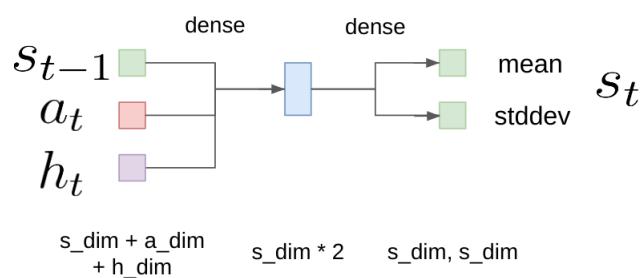


(a) デコーダーとエンコーダーのアーキテクチャ

遷移モデル(事前分布)



遷移モデル(事前分布)



(b) 遷移モデルのアーキテクチャ

図 6.1 モデルアーキテクチャ。「dense」は全結合層、「conv」は畠み込み層、「reshape」は変数の変形を意味する。

学習の安定化

潜在変数の次元を大きくした際、学習の初期段階でカルバックライブラー距離の計算値が発散することがよく起こる。これは遷移モデルが次の時刻の状態ベクトルの分布の標準偏差として0に非常に近い値を出力してしまった結果（丸め誤差が発生して）カルバックライブラー距離の計算時にゼロ除算が発生してしまうためである。この問題は遷移モデルが输出する標準偏差に下限を設けることで解決することができる。今回の実験では潜在変数の次元を1024次元にした際にこのカルバックライブラー距離が学習開始後直ちに発散する問題が発生したので、標準偏差の下限値を 10^{-7} とすることで学習をある程度継続することができた。ただし標準偏差に下限を設けることは経験的に学習を難しくすることが予備実験でわかっていたので、他の条件での実験の際には適用しなかった。

6.1.2 提案手法の実装

提案手法はベースラインの節で説明した部分モデルをほぼそのまま用いる。二階層目以上の第*i*層では、各遷移モデルの入力として一時刻前の状態変数 s_{t-1}^i と行動 a_t だけでなく、一つ低次元の層の同じ時刻の状態ベクトル s_t^{i-1} も入力とする。また、提案手法では、パラメータを固定している層の状態ベクトルのサンプリングには、モデルの評価時の設定に合わせて常に事前分布を用いている。これは、パラメータを固定している層はそれ以上学習されないために、事前分布より良い表現が下の階層に渡されることではなく、むしろ事前分布で足りない表現を積極的に下の階層の学習で獲得できるようにするためにある。その他はベースラインの実装と変わっていない。

6.2 実験結果

簡単のため、ここから状態変数の次元を64にしたベースラインモデルを「ベースライン(64)」、状態変数の次元を64と512にした提案手法モデルを「提案手法(64 + 512)」などと呼ぶ。

6.2.1 定量評価(尤度)

図6.2は、ベースラインと提案手法の学習時の評価用データでの負の対数尤度の値の推移をプロットしたもの(学習曲線)である。ただし提案手法(64 + 512)は一階層部分に1000

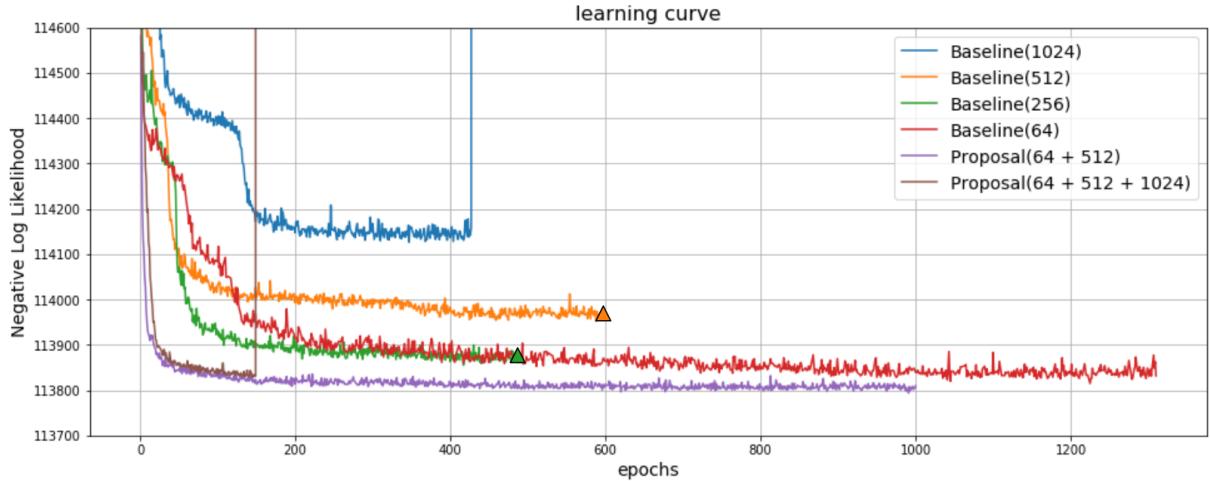


図 6.2 ベースラインと提案手法の学習曲線

表 6.1 手法ごとの定量評価指標 (尤度)

手法	負の対数尤度
ベースライン (64)	1.1384×10^5
ベースライン (256)	1.1387×10^5
ベースライン (512)	1.1397×10^5
ベースライン (1024)	1.1416×10^5
提案手法 (64 + 512)	1.1381×10^5
提案手法 (64 + 512 + 1024)	1.1383×10^5

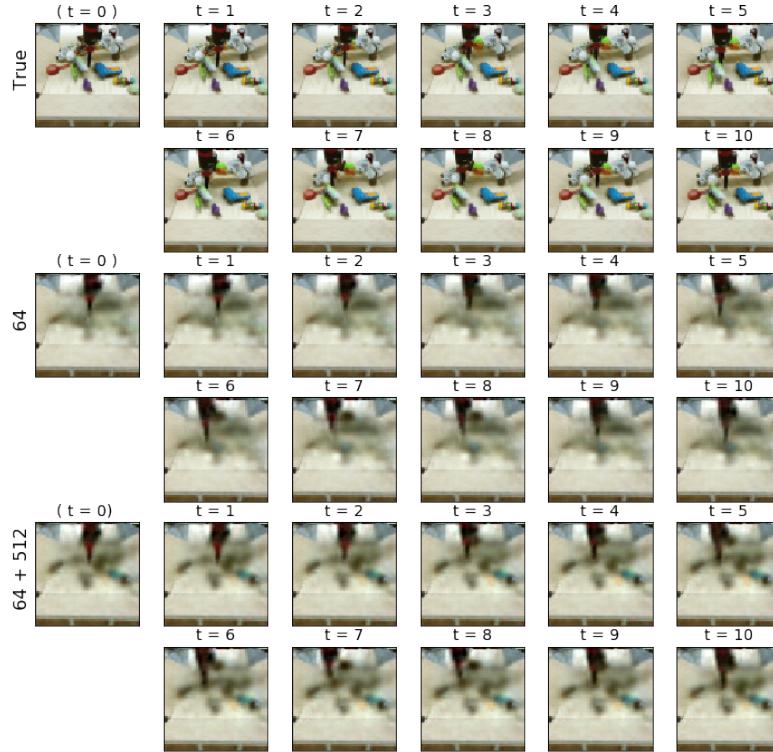
epoch 学習したベースライン (64) を用いており、学習曲線としてプロットされているのは二階層部分の学習時の負の対数尤度である。同様に提案手法 (64 + 512 + 1024) は一階層部分に 1000 epoch 学習したベースライン (64) を、二階層部分に 1000 epoch 学習した提案手法 (64 + 512) を用いており、学習曲線としてプロットされているのは三階層部分の学習時の負の対数尤度である。また表 (6.1) に最終的な負の対数尤度を示す。

まず図 6.2 について、ベースライン (512) と提案手法 (64 + 512) との比較から、二階層にすることで潜在変数の次元を 512 次元と大きくした際にもうまく学習が進むようになったことがわかる。これは、高次元の状態変数の学習時に学習済みの低次元の状態変数の情報を用いることで狙い通り学習が安定したためだと考えられる。次にベースライン (64) と提案手法 (64 + 512) の比較から、高次元の潜在変数で学習させたことによってより高い尤度の映像を生成できるようになったことがわかる。これは潜在変数を高次元にした方が多くの情報を保持

しやすく、適切に学習が進みさえすれば予測性能が向上しやすくなるためだと考えられる。最後に三階層の提案手法 ($64 + 512 + 1024$) について見ると、ベースライン (1024) と比較し明らかに学習が進んだものの、ベースライン ($64 + 512$) と比較して精度の改善は見られず、潜在変数の次元を大きくすることだけでは精度の向上には限界があるということが伺える。

6.2.2 定性評価

図 6.3、図 6.4 は評価用データで 10 フレームの行動条件付き映像予測により生成された映像のサンプルで、正解映像とベースライン (64) による予測映像、提案手法 ($64 + 512$) による予測映像を比較している。どちらのモデルもロボットアームの位置はほぼ正しく予測ができるが、環境中の物体の見た目には差が見られた。図 6.3a、図 6.3b を見ると、ベースラインでは環境中の物体の輪郭がぼやけて灰色がかっているのに対し、提案手法では特に物体が密集していない場合に比較的物体一つ一つの生成が鮮明になっていることが見て取れる。これは潜在変数の次元を大きくし保持できる情報を増やすことに成功したためだと言える。しかしフレームごとの見た目には改善が見られたものの、物理的な操作についての予測はあまり改善が見られなかった。図 6.4a では、正解映像ではロボットアームの移動によって中央の緑の物体の姿勢を変えているが、そもそもベースラインではその物体を生成できておらず、提案手法でも僅かに黒っぽい点として表されている程度で姿勢の変化を読み取ることはできない。図 6.4b は、正解映像では滑らせるようにして 2 つの物体を近づけており、提案手法のでは 2 つの物体の輪郭が見られるが、机の色やロボットアームの色と途中から同化してその動きを追うことは困難である。このように物理的な操作の予測を改善するにはまず生成される映像の 1 フレーム 1 フレームの質を上げる必要があると考えられ、今回は物理的な操作の予測の改善はあまり見られなかった。最後に図 6.5 は、各モデルで学習時よりも長い 30 フレームの映像予測をした際の生成映像である。ベースライン・提案手法ともにロボットアームの動きはほぼ正解映像と一致しているが、ベースラインでは右端に映る物体が映像を予測するにつれて青色から緑色に変わってしまっているのに対し、提案手法は背景や周りの物体について一貫性が保たれていた。これは、潜在変数の次元が小さいまま性能をあげようすると潜在表現に含まれる情報が密になりすぎて遷移時に遷移とは関係のない情報も変化させてしまいやすいという問題がある可能性があり、一方提案手法は高次元の潜在変数を使うためにそのような問題は生じなかつたと考えられる。



(a)

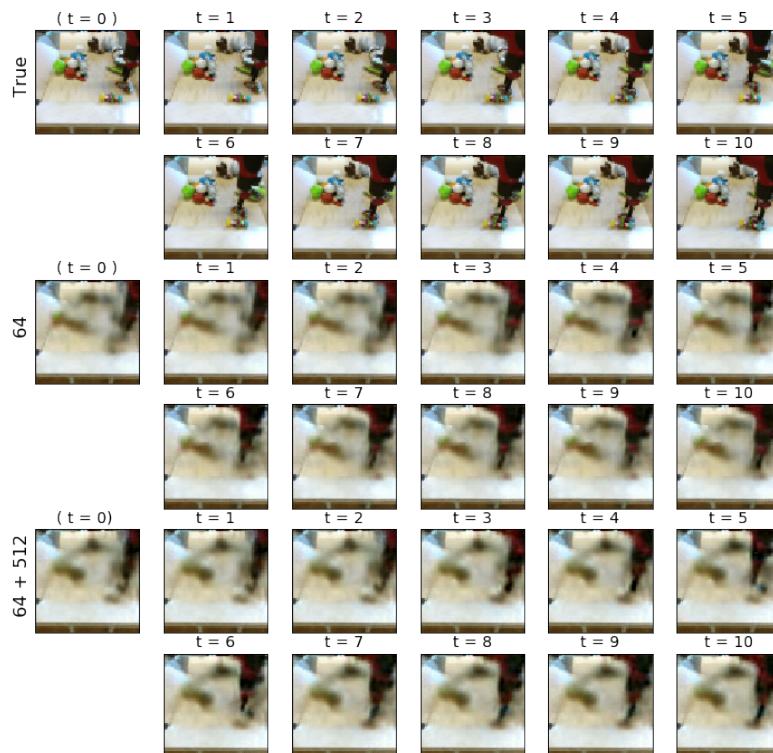


(b)

図 6.3 映像予測結果 1. 「True」が正解映像、「64」がベースライン（64）での予測結果、「64 + 512」が提案手法（64 + 512）での予測結果を示す。また $t = 0$ は初期状態の推論時に与えられるフレームを示す。



(a)



(b)

図 6.4 映像予測結果 2. 「True」が正解映像、「64」がベースライン（64）での予測結果、「64 + 512」が提案手法（64 + 512）での予測結果を示す。また $t = 0$ は初期状態の推論時に与えられるフレームを示す。

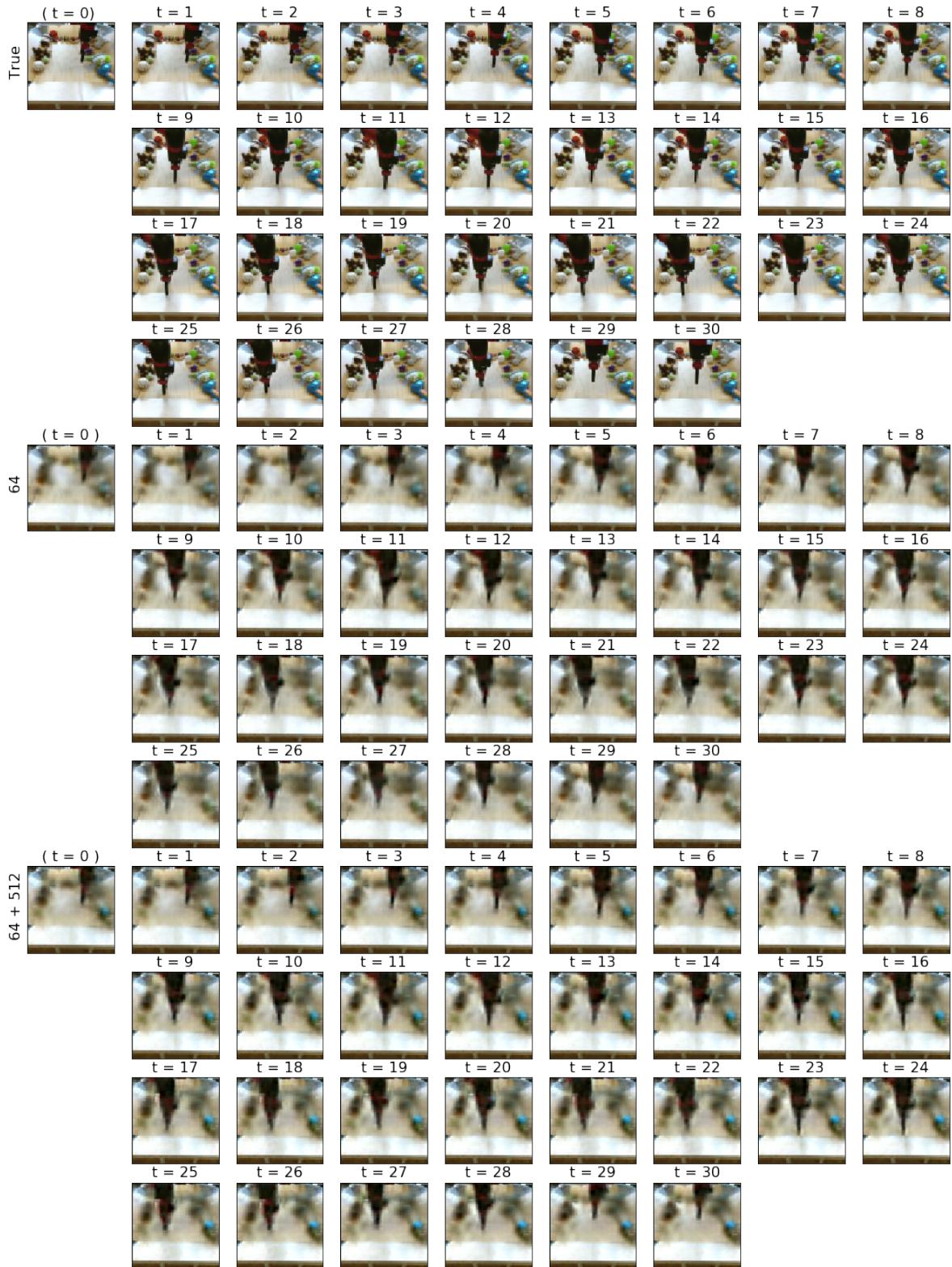


図 6.5 30 フレームの予測結果の比較。「True」が正解映像、「64」がベースライン（64）での予測結果、「64 + 512」が提案手法（64 + 512）での予測結果を示す。また $t = 0$ は初期状態の推論時に与えられるフレームを示す。

第 7 章

考察

実験結果を受けて、第 7 章ではより高精度な生成や発展的な問題設定に向けて考察を行う。

7.1 本研究の貢献

本研究では DSSM の複雑環境下での応用を見据えて DSSM の問題点を指摘し、その拡張を行った。

まず第 4 章の予備実験によって、DSSM は状態変数の次元を大きくすると精度が悪化する問題があることを示した。そこで、状態変数を多階層にして低次元の状態変数から順に学習を行う提案手法により、これまで精度が悪化していたような高次元の状態変数を用いても学習を進めることができ、さらにベースラインである一階層モデルよりも高い精度が出ることを示した（図 6.2）。二階層 DSSM モデルの実験では、生成される映像の色彩や物体の輪郭などに改善が見られ、これは状態変数の次元を大きくしたことによってより多くの情報が獲得しやすくなつたためだと考えられる（図 6.3）。また二階層モデルで学習時より長期の映像予測を行った場合にも、ベースライン同様ロボットアームの位置は正確にとらえられ、さらにベースラインと比較して背景の一貫性の改善が見られた（図 6.5）。しかし、環境内の物体の物理的な操作の予測については、ベースライン・提案手法ともに生成された予測映像から読み取ることは難しく（図 6.4），これはデコーダー・エンコーダーモデルより表現力の高いモデルに変えることで改善する可能性はあるが、さらなる研究の必要がある。最後に、三階層の DSSM の実験では、1024 次元という高次元な潜在変数を使った場合にも学習を進めさせることができ、1024 次元の一階層モデルと比較して精度が大きく改善し 64 次元の一階層モデルと比較しても同等以上の性能まで改善したが、学習が途中から不安定になる問題が見られ（図 6.2）これについてもさらなる調査が必要である。

以上をまとめると、本研究では DSSM が獲得できる情報を増やすべく、従来よりも状態変数の次元を大きくして学習が進められる方法を示し、さらに映像予測が改善することを示した。映像予測の既存研究では広く自己回帰ベースの手法が使われているが、これはロボット実機への応用上制約が多いいため自己回帰に依らない映像予測手法の研究が重要になると考えられ、この研究課題のもとで本研究は一定の貢献になると考える。

7.2 課題

7.2.1 定性的な改善

今回の研究では、尤度上は DSSM の映像予測を改善できたものの特に物理的な操作の予測の面であまり定性的には改善が確認できなかった。これはそもそもデコーダー・エンコーダーが貧弱で物体一つ一つをはっきりと潜在表現として獲得できていないことが問題であった可能性がある。物体の生成がぼやけてしまうのはモデルの表現力がそもそも弱かったことが原因として考えられるが、物体の移動もあまりうまく捉えられていない理由はとして次の 3 つを考えた。

- 生成結果がぼやけたままだと物体の移動などを捉えることが難しい
- 条件付けする行動系列が直接的には関与しない物体の移動は、DSSM での学習が難しい
- 画像をデコードしやすい表現と遷移を計算しやすい表現は必ずしも一致しないため、DSSM はデコードと遷移のどちらかを優先して学習してしまう問題を持っている

2 が真の理由である場合より根本的な DSSM の改善を考える必要があるが、素朴には遷移モデルの中間層を増やすことや、補助的な行動系列を生成するなどのアプローチが考えられる。3 が原因であれば、潜在変数を遷移に特化させる変数とデコードに特化させる変数に分けるなどのアプローチが考えられる。また 1 が原因であった場合には表現力の高いデコーダーを用意するなどするなどで改善が期待できる。原因の解明とそれを踏まえた改善は今後の課題である。

7.2.2 学習が不安定になる問題

潜在変数を高次元にした際、ベースラインの実験や提案手法の三階層モデルの実験で学習時に急激に目的関数の値が発散する問題があった。目的関数の値の発散と同時に状態変数の値も急激に変化し状態変数のノルムや分散の値も急激に大きくなってしまうことが現状わかっているが、問題発生の詳細なメカニズムはまだ確認できていない。深層状態空間モデルをより正し

く理解し適切な拡張を考えていくためにもこの問題の原因をより詳細に調べていきたい。

7.2.3 初期状態の推論

今回は初期状態 s_0 の推論には、経験的な手法として状態変数の事後分布モデルに初期観測とゼロベクトルを渡することで生成される状態ベクトルを用いたが、これが最適なのかは定かではない。例えば自己回帰モデルを用いる強化学習 [29] では、初期状態の推論時に、学習に使う観測系列が始まる直前までの十分な長さの観測系列を使って、ゼロベクトルで表される仮の初期状態を十分な回数更新して(これを burn-in と呼ぶ) 真の初期状態として用いることが良いとし、さらに初期状態の推論に用いる観測系列が長いほど精度が向上すると報告した研究がある [29]。本研究ではではロボットの実応用を想定しており、初期状態から予め十分先の将来を予想をした上で行動を開始して欲しいという目的意識があるためこの burn-in は直接適用できないが、同じ初期観測を用いてでも複数回状態ベクトルを更新することは有効である可能性があるなど、研究の余地がある。

7.2.4 低階層の状態表現の再学習

今回提案手法では、複数階層の DSSM モデルを学習させる際には低階層のモデルのパラメータを固定している。しかし学習を簡単にする上ではこれでよいものの、全体としての性能の向上を考えたときにモデルのパラメータを固定することが良いとは必ずしも言えない。今回低階層のモデルは低次元の状態表現で将来の観測の再構成ができるように学習するが、高階層のモデルの学習時には低階層のモデルは再構成しやすい表現を持つ必要はなく、むしろ高階層の状態表現を使った将来予測が簡単になるよう、例えば高階層の状態遷移を補助しやすくなるようなより中心的な状態表現とその遷移を獲得することが求められるはずである。したがって、学習のどこかの段階でモデルの低階層部分を、低階層の状態表現からのデコードをせずに再学習することでより全体の性能が高まる可能性が考えられる。

7.3 展望

7.3.1 敵対的学習の導入

本研究で扱った DSSM では損失関数に二乗誤差を利用しているため、画像空間上の小さな特徴量、例えば小さい物体や背景に近い色をした物体などは無視されやすい。第 2 章で取り上げた VAE や DSSM など最尤推定を用いる深層生成モデルは目的関数に生成誤差を取るため

に一般的に共通してこのような問題を持つ。この問題は、近年高品質な深層生成モデルとして研究が盛んな敵対的生成ネットワーク、GAN(Generative Adversarial Networks)[21]を補助的に用いることによる解決などが考えられる。

7.3.2 多視点からの映像予測

本研究では1視点からの映像予測を行っているが、ロボット実機に映像予測を応用する際にはロボットの視点が動くような問題設定や、より高性能な予測を行うために多視点からの観測が使える前提で映像予測を行うというような問題設定なども考えられる。深層学習研究で複数視点からの観測を扱う問題の先行研究として Generation Query Network(GQN)[30]があり、複数視点からの観測とその視点位置が与えられたときに別の視点からの観測を予測して生成する問題を解いている。この研究では視点不变な共通の潜在変数の獲得を行い、クエリとして視点情報が渡されたときにその視点での観測を返すモデルを考えている。DSSM に GQN のアプローチを組み合わせることは、より良い表現の獲得と実用性の向上につながるため重要であり、実際 Temporal GQN[31]などこの方向性の研究はすでに進められているが、このような問題設定においても本研究の提案手法は有効であると考えられるので検証を行いたい。

7.3.3 他の構造を持つ潜在表現との併用

今回の提案手法では潜在変数の階層性のみを仮定してモデルを構築したが、近年の映像予測モデルのほとんどは潜在変数に平面的な構造を仮定し、状態表現をベクトルではなく行列として扱うことで性能の向上を実現している[7][8][23]。状態表現に階層性をもたせることの有効性は実験で示したとおりであるが、他にも環境内でより立体的な行動を考える際には潜在変数に立体性を仮定したり、さらに例えば操作対象の物体が1つで構造が既知の場合には操作対象のメッシュ情報やグラフ構造を状態表現の構造として用いることも可能であると考えられる。このようなベクトル以外の構造を持つ状態表現を学習したい場合においても、低次元のベクトル状の潜在表現を用いて階層的なモデルを考えることで精度を高めたり学習をスムースに進められるようになる可能性があり、本研究の提案アルゴリズムが活かせると考える。

7.3.4 メタ学習の導入

機械学習研究において、似たタスク集合が与えられたときにそれらのタスク全てに汎化できるようなモデルの獲得を目指すメタ学習という研究分野がある。行動条件付き映像予測では与

えられたデータから環境の遷移を学習するが、遷移自体が同じで観測が異なるような場合、例えば本研究で扱った BAIR Push Dataset をベースに考えると机や操作対象の物体、そしてロボットの操作方策は同じままロボットの機種だけが異なるような場合にも適切に映像予測がしたいというような問題設定が考えられるが、このような設定ではメタ学習手法を用いることで様々なロボットに対応できる映像予測が可能になる。この場合具体的には、状態表現にロボットの機種情報が含まれないよう状態表現とロボットの機種クラスの相互情報量を最小化する制約を置いて遷移を学習することでロボットの機種に依存しない遷移モデルを獲得することができ、代わりにデコーダーにロボットの機種クラス情報を追加で渡すことで適切に映像予測が可能になると予想できる。

7.3.5 映像予測用データ収集の人による代替

映像予測を行うにはまずデータを収集する必要があるが、映像予測の実ロボットでの応用を考えた場合、BAIR Push Dataset のようにランダムにロボットを動かしてデータを集めただけでは本来不十分である。理想的には、実ロボットが環境中である程度期待する動きをすることがわかっている場合にも、期待する行動系列をまんべんなく試行してデータを集めたい。しかしそもそもロボットが環境を適切に操作するための制御方策を獲得すること自体難しいことが多い、適切に映像予測用のデータが集められるような良い制御方策が獲得できていればわざわざ映像を予測する必要はないかもしれない。そこで、考えられるのがデータの収集を人間が行う方法で、人間であれば環境内で期待する様々な行動系列実行することは容易い。ここで以下の二つの問題が発生するがこれらは近年の研究や技術で簡単に解決することが見込める。

- ロボットとは異なり、人間の行動系列情報を取得することが難しい
- データセットでは人間が、実応用ではロボットが操作を行うため、特にデータセットに操作主体である人間が写り込んでいる場合、適切に映像予測を学習できても実応用に利用できない可能性がある

まず一点目については、これは操作主体の人をカメラや工学センサー、磁気センサーなどを使ったモーショントラッキングやモーションキャプチャの技術によって人の行動系列（この場合人の運動）を計測することができる。二点目については、例えば TCN[32] で提案されているような手法で、同じ操作を行う人間の行動系列データとロボットの行動系列データの対応関係を予め学習した上で、データセットとロボットの実観測から人やロボットが写り込んでいる部分を削除してこれまで通り遷移モデルを学習・利用するアプローチが考えられる。

このようにしてロボットの代わりに人がデータを集めることで、非剛体物の操作などより複雑な映像予測の問題設定を扱うことが可能になると考える。

7.4 応用

本研究や前節の展望で述べたことを踏まえて考えられる映像予測手法の応用の可能性について、実機ロボットへの応用、物理シミュレーションの近似、微分可能なモデルとしての利用、の3つの観点から述べる。

7.4.1 実機ロボットへの応用

第1章でも触れたように、映像予測手法はロボットが自身の制御方策を学習するとき、そしてロボットを使う人がロボットの行動を予め評価したいような場合に用いられると考える。ロボットの制御方策の獲得への応用としては、映像予測を学習することによって獲得された環境の状態表現を用いた強化学習や、ランダムにロボットが行動した際の結果を予測し良い結果になる行動を採用して動くようなサンプリングベースの研究が既に進んでおり、実機応用が期待される。しかし深層学習を用いたロボットの制御は人がルールベースで記述したプログラムによる制御とは異なり、ロボットの行動内容や行動結果をロボットのユーザーが予測することは難しい。そのため、特に工場や小さい子供のいる環境など注意が必要な現場では映像予測手法によってロボット自身や環境の変化を予測し、ロボットのユーザーに予測結果を伝えて行動の安全性を確認したあとに行動するような手続きが求められると考えられ、このような場面で映像予測手法が重要になると考えられる。

7.4.2 物理シミュレーションの近似

本研究で扱った深層学習ベースの行動条件付き映像予測では物理現象の結果として観測されるデータの予測を行うが、これは物理現象の近似という意味で現行の物理シミュレーションソフトウェアと共に目的を持っていると解釈ができる。現在、物理シミュレーションの手法としては、既知の様々な物理法則を記述し微小時間・微小空間単位で逐次的に各領域の状態を計算して全体の結果を予測することが一般的である。しかしこのアプローチは、物理法則が既知である必要があり、また複雑な物理法則に対しては予測に膨大な時間がかかりリアルタイムに予測ができないという問題がある。例えば[33]は、蕎麦にオイスターソースをかけて混ぜる物理シミュレーションを扱っているが、1フレームごとの見た目はとてもリアルなもののが現

象としては依然不自然な部分があり、さらに 30fps で 1 秒の予測をするのに 29 時間かかると報告している。現行の物理演算ベースの映像予測に対し深層学習ベースの映像予測は、物理法則の正しさや多視点から見た際の一貫性を保証することが難しいなど機能として制限は多いものの、必ずしも物理現象が既知である必要はなく、また一度学習すればリアルタイムで予測を行うことができる。さらに、必要なデータを集めることで機能の制限を解消することもできるはずである。このようなことから、映像予測手法は複雑な物体の物理シミュレーションの機能を部分的に代替できる可能性があり、これによってシミュレートできる物体や現象の幅が広げられると考える。

7.4.3 微分可能な環境モデルとしての利用

深層学習を用いてロボットの制御方策を学習するロボット学習において、基本的にロボットは実環境かシミュレーション環境で行動を起こすまでは結果を観測することができない。これは、ロボットの行動を入力、その結果の観測を出力と考えると、入力と出力の間に環境という未知の関数が存在し、得られた出力（観測）から入力（行動）を最適化する際に直接勾配法を用いることができないことを意味する。深層強化学習ではこの問題を様々なアプローチで解決しているが、それでも環境中で多くの試行を重ねる必要があり学習に時間がかかる上、エージェントの経験の偏りによって獲得される方策が変わってくるため性能を安定して上げることは難しい。近年の深層強化学習研究では環境との相互作用からエージェントが自ら環境のモデル自身を明示的に学習し、この学習された環境（世界モデル／内部モデル）[10] を代わりに使うことで実環境での試行回数を減らし学習効率を高めるアプローチが精力的に研究されている。この世界モデルと同じような発想で、実機ロボットにおいて行動と結果の間に環境という未知の関数があるならば、環境という関数自体を微分可能なモデルで近似できれば、ロボットの行動方策を直接勾配法で最適化できるはずであるとする考えがある [34]。具体的には、近似した環境モデル env と最適化したい制御方策 π を用いてロボットがある初期状態 s_{start} からある目標状態 s_{goal} までの行動をプランニングしたときに、環境モデル env によって予測される結果状態 \hat{s}_{goal} との誤差を小さくするように勾配法で直接方策を学習する。

$$env(\pi(s_{start})) = \hat{s}_{goal} \Leftrightarrow s_{goal}$$

ここで、本研究のようなディープラーニングベースのモデルは微分可能であるため、環境の近似モデルが得られれば勾配法で直接方策を最適化することができる。特に環境が既知で、予

め他のエキスパートロボットやあるいは人間によって多くの試行を重ねてデータの収集ができる、環境を高い精度で近似するモデルが得られる場合には、この近似モデルを使った方策の学習は非常に有効であると考えられる。このように映像予測手法が発展することによって全く新しいアプローチのロボット学習の研究が可能になり、さらに環境のモデルをロボットユーザー間で共有する仕組みを作ることで、ロボットの応用の可能性がさらに広がると考える。

第 8 章

結論

本論文では、実機ロボットへの応用を見据えて深層状態空間モデルを用いた映像予測について取り上げた。まず深層強化学習などで用いられているシンプルな深層状態空間モデルでは複雑な環境を扱う問題設定に上手くスケールしない問題を示した。その上で深層状態空間モデルの状態表現の階層性を明示的にモデル化した提案手法によってより高次元の状態表現を扱えるようにし、さらに映像予測の性能が向上することを示した。実験では、定性的な大きな優位性は示せなかったものの、高次元の状態変数を用いた学習を可能にしたことは深層状態空間モデルの大きな問題を克服したと言える。これにより、これまで映像予測の分野では実機ロボットへの応用上の制約が多いにも関わらず自己回帰的なモデルの研究が主流であったが、状態空間モデルベースの手法が見直されるきっかけになるかもしれない。

第 7 章では展望として深層状態空間モデルの研究の方向性を複数上げたが、これらの研究をすすめることによってより高性能な予測が可能になり、また実機への応用の可能性も高められると考える。さらに応用の例として、映像予測の実機応用に加え、新たな物理シミュレーションの近似アプローチと新たなロボット学習のあり方の可能性について述べた。この二つの応用例は現段階では可能性の話に過ぎず実現可能かは定かでないがどちらも実現すれば社会的な価値は大きいと考えられるので、今後も慎重に研究を継続していきたい。

謝辞

本論文を作成するにあたり、多くの方々にご協力をいただきました。

松尾豊教授には、私が取り組みたい研究テーマについて興味を持っていただき、研究アイディアの方向性や可能性について多くのアドバイスをいただきました。当初私がやると言った研究内容の多くは展望や応用の節で触れるに留まってしまいましたが、今後また時間をとって研究し形にしたいと考えています。

岩澤有祐特任助教には具体的な研究内容を決める段階から執筆までサポートしていただき、岩澤特任助教の的確な研究方針のアドバイスがあって辛うじて卒業論文をまとめることができました。先行研究や提案手法の実装にあたっては、今回使わせて頂いた深層生成モデルのライブラリを開発している特任研究員の鈴木雅大さん、博士課程の金子貴輝さん、学科後輩の原田憲旺くんにたいへんお世話になりました。また実験がうまく行かなかつた際には博士課程の阿久澤圭さん、修士課程の谷口尚平さんに実装上のアドバイスをいただき、研究を進めることができました。修士課程の松嶋達也さんには毎日のように面倒を見ていただき、研究の仕方から研究生活についてまで非常に多くのことを学ばせていただきました。特任研究員の河野慎さんは私に学会参加を勧めて下さり、発表を目標に本研究を大きく加速させることができました。研究室のサーバーチームの皆様には日頃からお世話になり、お陰様で不自由なく実験を進めることができました。皆様本当にありがとうございました。

最後に、研究室配属から一年の間、あらゆる面でサポートをしてくださった松尾研究室の皆様に御礼申しあげて、謝辞とさせて頂きます。

東京大学工学部システム創成学科

知能社会システムコース

松尾研究室学部 4 年

平成 31 年 2 月 近藤生也

参考文献

- [1] Diego Escudero-Rodrigo and Renato Alquézar Mancho. Distance-based kernels for dynamical movement primitives. 2015.
- [2] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565, 2019.
- [3] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control, 2018.
- [4] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games, 2015.
- [5] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction, 2016.
- [6] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction, 2017.
- [7] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior, 2018.
- [8] Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V. Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks, 2019.
- [9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.
- [10] David Ha and Jürgen Schmidhuber. World models (2018), 1803.
- [11] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey, 2019.
- [12] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters.(2015). *arXiv*

- preprint arXiv:1511.05121*, 2015.
- [13] Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
 - [14] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2019.
 - [15] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections, 2017.
 - [16] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation, 2015.
 - [17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2017.
 - [18] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders, 2016.
 - [19] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from generative models, 2017.
 - [20] Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling, 2019.
 - [21] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
 - [22] S. Aigner and M. Körner. Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing gans. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLII-2/W16, p. 3–11, Sep 2019.
 - [23] Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction, 2019.
 - [24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.
 - [25] Pytorch. <https://pytorch.org/>.
 - [26] Pixyz. <https://github.com/masa-su/pixyz/>.

- [27] Tensorflow. <https://www.tensorflow.org/>.
- [28] Deep planning network. <https://github.com/google-research/planet>.
- [29] Steven Kapturowski, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. 2018.
- [30] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, Vol. 360, No. 6394, pp. 1204–1210, 2018.
- [31] Gautam Singh, Jaesik Yoon, Youngsung Son, and Sungjin Ahn. Sequential neural processes, 2019.
- [32] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.
- [33] Yun Fei, Christopher Batty, Eitan Grinspun, and Changxi Zheng. A multi-scale model for coupling strands with shear-dependent liquid. *ACM Transactions on Graphics (TOG)*, Vol. 38, No. 6, pp. 1–20, 2019.
- [34] Jürgen Schmidhuber. Making the world differentiable: On using fully recurrent self-supervised neural networks for dynamic reinforcement learning and planning in non-stationary environments. *Institut für Informatik, Technische Universität München. Technical Report FKI-126*, Vol. 90, , 1990.