人工知能学会 人工知能基本問題研究会

深層生成モデルと世界モデル

東京大学工学系研究科　技術経営戦略学専攻　特任助教

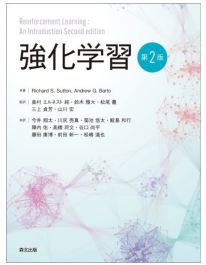
鈴木雅大

2023/03/17

自己紹介

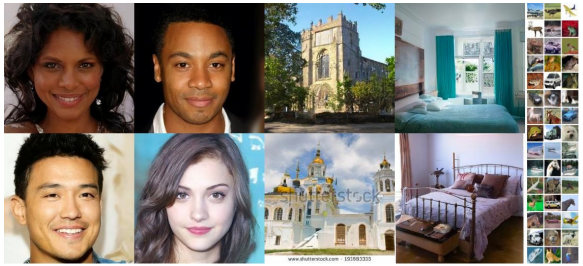
鈴木雅大（東京大学松尾研究室　特任助教）

- 経歴
  - 2015年3月 北海道大学情報科学研究科修了
  - 2018年3月 東京大学工学系研究科修了
  - 2018年4月～2020年7月 東京大学工学系研究科 特任研究員
  - 2020年8月～ 東京大学工学系研究科 特任助教
- 研究分野：
  - 深層生成モデル・マルチモーダル学習・転移学習（ゼロショット学習）
- 活動など：
  - Deep Learning基礎講座・サマースクール「深層生成モデル」・「世界モデルと知能」などの講義担当
  - 「深層学習（Goodfellow著）」「強化学習第2版（Sutton著）」の監訳・分担翻訳



深層生成モデル

- 深層学習の研究分野では，深層生成モデルの研究が進んでいる．
  - 生成系（画像や文書）の他に，異常検知，半教師あり学習，表現学習，メタ学習など



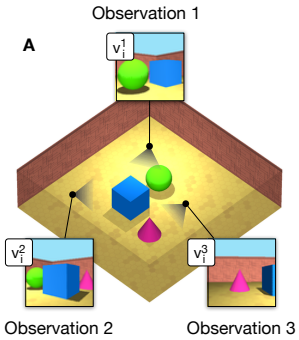
[Ho+ 20]

[Zhu + 17]

i went to the store to buy some groceries .
i store to buy some groceries .
i were to buy any groceries .
horses are to buy any groceries .
horses are to buy any animal .
horses the favorite any animal .
horses the favorite favorite animal .
horses are my favorite animal .

[Bowman+ 15]

[Li+ 19]

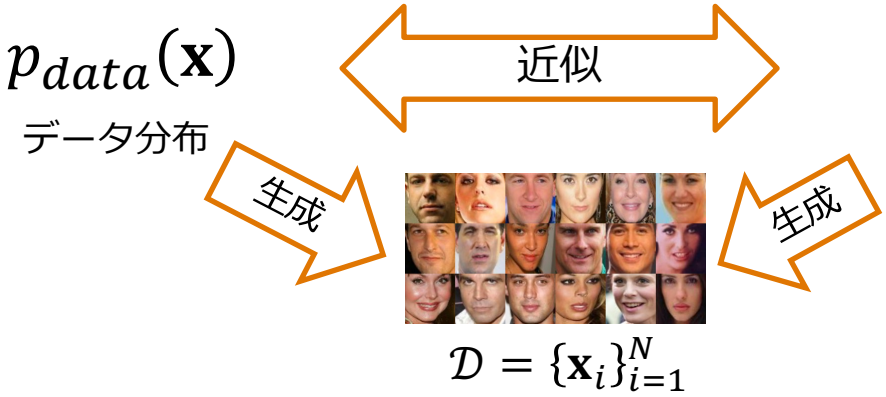


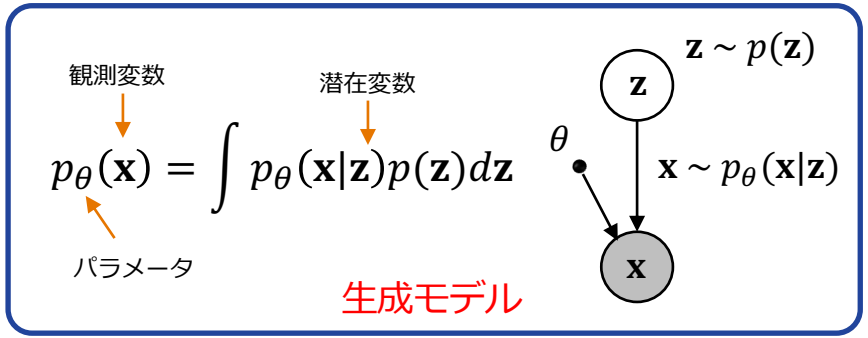
[Eslami+ 18]

[Saharia+ 22]

生成モデル

- 観測されたデータが未知のデータ分布から生成されていると仮定し，その生成過程を確率分布によってモデル化する枠組み.
  - データとして観測される観測変数の他に，その背景にある確率変数として潜在変数も仮定することが多い（潜在変数モデル）.
- 「データがどのようにできているか？」を明示的に設計することができ，モデルからデータを生成（シミュレーション）することができる.

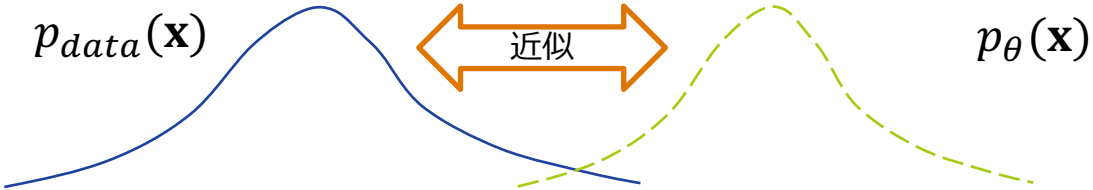




生成モデルの学習

- 目標：生成モデル$p_\theta(\mathbf{x})$がデータ分布$p_{data}(\mathbf{x})$を近似するようにしたい.
  - 生成モデルの構造を設計した上で，近似を実現するようにパラメータθを選ぶ.

=> 生成モデルの学習



- 分布間の「距離」にカルバック・ライブラーダイバージェンスを選択すると，尤度最大化に対応.

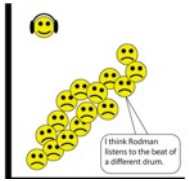
$$\hat{\theta} = \arg\max_\theta \sum_{x_i \in \mathcal{D}} \log p_\theta(\mathbf{x}_i)$$

生成モデルでできること

- 生成：
  - 生成モデルが学習できれば，未知のデータを生成できる
  - 「生成」モデルと呼ばれるのはここから

生成 → 生成モデル

- 密度推定：
  - データを入力すると，それがどれだけ生成モデルと違うかがわかる.
  - 外れ値検出や異常検知に用いられる.

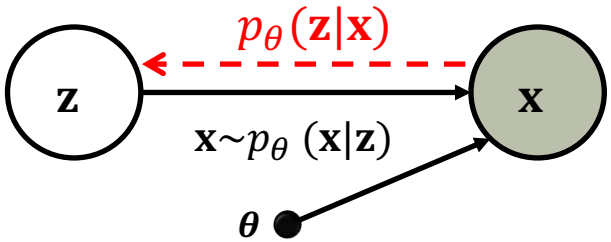


http://jblomo.github.io/datamining290/slides/2013-04-26-Outliers.html

- 欠損値補完，ノイズ除去：
  - 欠損やノイズのある入力を入れると「元のデータらしく」補完してくれる.

生成モデルにおける推論

- 推論（inference）：
  - 確率変数（観測変数）が与えられた下で，任意の確率変数（潜在変数）の事後分布を求める.
  - 潜在変数を持つ生成モデルにおける重要な概念（「結果」から「原因」を求める）.



- 一般のモデルでは，推論は計算困難なことが多い.
  - 様々な近似推論が提案されている.

※ 本発表では，潜在変数の事後分布を求めること（推論）と，モデルのパラメータ値を最適化すること（推定，学習）を区別します.

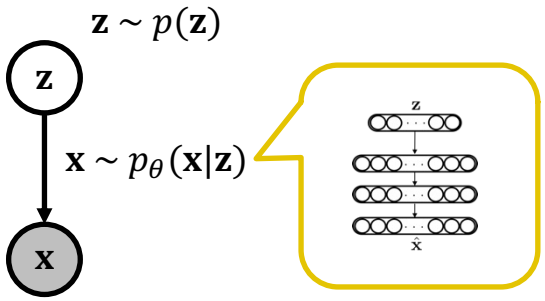
深層生成モデル

- 観測変数が複雑な場合，単純な確率分布では直接表現できない.
  - 特に観測変数がベクトルで，次元間の依存関係が非線形な場合（高解像度画像など）
  - 従来の生成モデルは，複雑な観測データを直接生成することは意図していなかった.

複雑な関係性を表すには？-> 深層ニューラルネットワーク（DNN）

- 深層生成モデル（deep generative model）
  - 確率分布をDNNで表現した生成モデル.
  - モデルパラメータは勾配情報に基づき学習.

生成モデルによって明示的に生成過程をモデル化できる
+
DNNによって複雑な変数間の関係性を捉えられる



深層ニューラルネットワークによる確率分布の表現

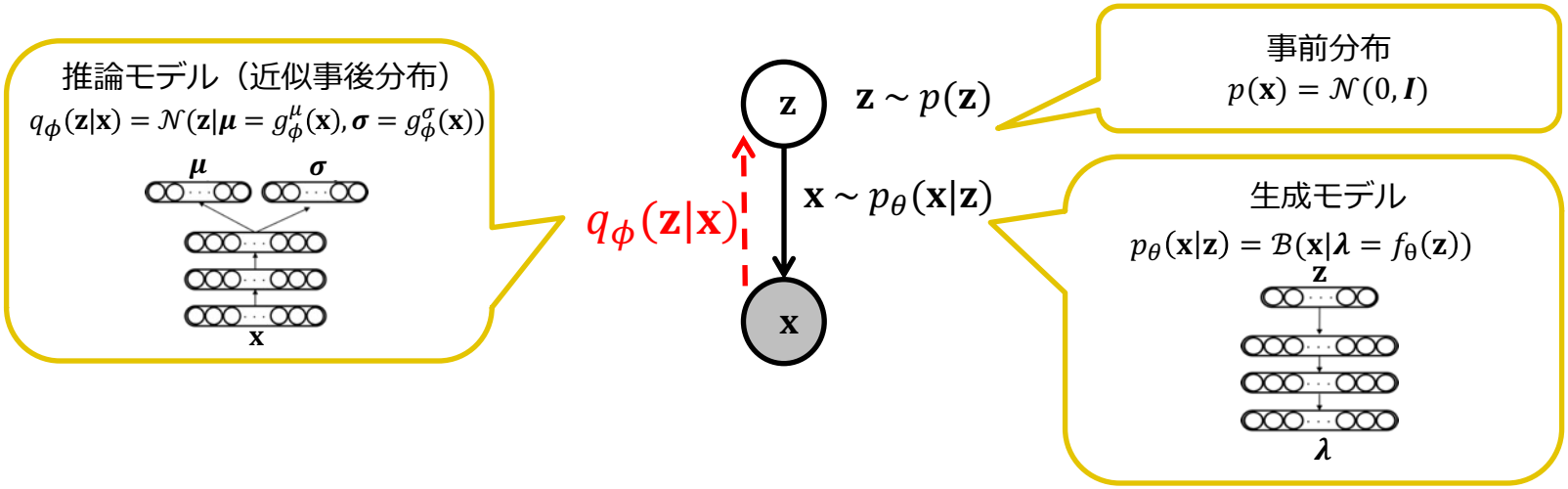
深層生成モデルの種類

- 分布間の距離の選択や，モデルの設計，分布のパラメータ化方法などによって種類は様々

	モデル	生成モデルの尤度計算	生成	推論
VAE	生成モデル： $p(\mathbf{x},\mathbf{z}) = \int p(\mathbf{x}\|\mathbf{z})p(\mathbf{z})d\mathbf{z}$ 推論モデル： $q(\mathbf{z}\|\mathbf{x})$	直接は不可能（対数尤度の下界が計算可能）	低コスト	可能（推論モデル）
GAN	生成器： $G(\mathbf{z})$ 識別器： $D(\mathbf{x})$	不可能（識別器が真のモデルとの尤度比を推定）	低コスト	不可能（エンコーダを導入すれば可能）
自己回帰モデル	条件付きモデル：$\prod_d p(x_d\|x_1,\dots,x_{d-1})$	可能	高コスト	潜在変数がない
フローベース	フロー（可逆な関数）： $\mathbf{x} = f(\mathbf{z})$	可能	低コスト	可能（逆変換）
拡散モデル	逆過程：$p(\mathbf{x}_T)\prod_t p(\mathbf{x}_{t-1}\|\mathbf{x}_t)$ 拡散過程：$\prod_t q(\mathbf{x}_t\|\mathbf{x}_{t-1})$	直接は不可能（対数尤度の下界が計算可能）	高コスト（反復）	可能（拡散過程）
スコアベース	スコアネットワーク：$s(\mathbf{x})$	直接は不可能（対数尤度の勾配が計算可能）	高コスト（反復）	潜在変数がない
エネルギーベース	エネルギー関数：$E(\mathbf{x})$	困難（分配関数の計算が困難）	高コスト（反復）	モデルの設計による

Variational Autoencoder

- Variational autoencoder（VAE） [Kingma+ 13, Rezende+ 14]
  - 潜在変数モデルの確率分布をDNNで表現（深層潜在変数モデル）．
  - 潜在変数zの近似事後分布（推論）を，観測xを入力とした関数（DNN）で表現（amortized variational inference）



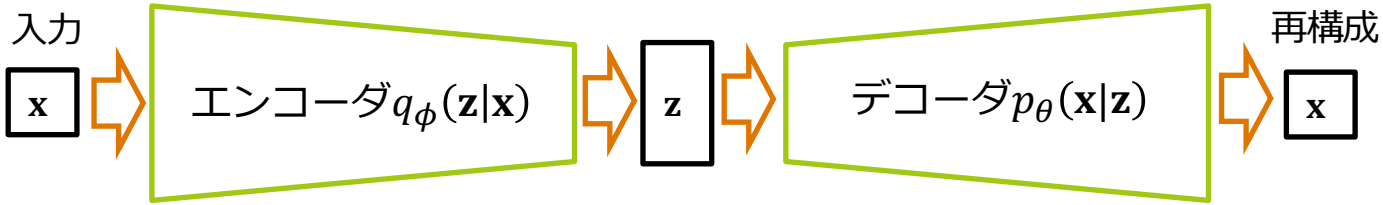
※ $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$が生成モデルだが，慣例上$p_\theta(\mathbf{x}|\mathbf{z})$を生成モデルと呼ぶ

Variational Autoencoder

- 目的関数：対数周辺尤度の変分下界（エビデンス下界，evidence lower bound；ELBO）

$$\log p_\theta(\mathbf{x}) \geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{負の再構成誤差}} - \underbrace{D_{KL}[q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})]}_{\text{推論モデルの正則化}}$$

- 分布のパラメータ化等の話を省略して，情報の流れを確認．
  - VAEでは推論モデルで入力$\mathbf{x}$を$\mathbf{z}$にエンコードし，生成モデルで$\mathbf{z}$から$\mathbf{x}$をデコード（再構成）する．
  - 推論モデルと生成モデルをオートエンコーダにおけるエンコーダとデコーダとみなせる．



生成画像

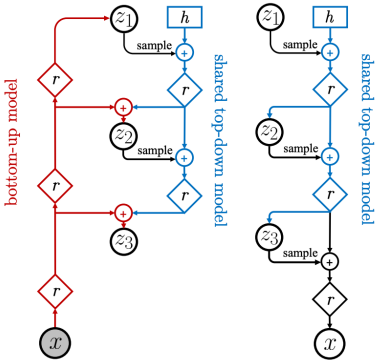
- ランダムな$\mathbf{z}$からデコーダによって画像$\mathbf{x}$をサンプリング
  - データ集合と同じような画像が生成できているが，輪郭等がぼやける傾向がある．

[Kingma+ 13]

@AlecRad

生成画像

- Nouveau VAE（NVAE）[Vahdat+ 20]
  - VAEの潜在変数を階層化する.
  - 利点：
    - 階層的な表現を獲得できる.
    - モデル全体の表現力を向上させることができる.
    - より柔軟な推論が可能となる.

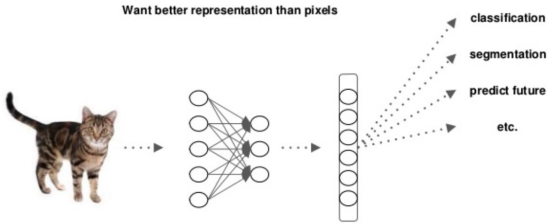


(d) CelebA HQ ($t = 0.6$)

(e) FFHQ ($t = 0.5$)

VAEと表現学習

- VAEでは，再構成だけでなく表現$\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})$も学習しているとみなせる．
  - 深層生成モデルにおいては，表現学習は推論と等価
    - 入力から潜在変数へ推論することで表現を獲得している．
  - VAEは表現学習手法として優れた手法．

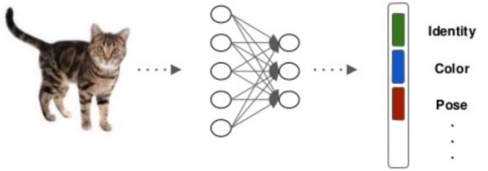


https://www.slideshare.net/lubaelliott/emily-denton-unsupervised-learning-of-disentangled-representations-from-video-creative-ai-meetup

- 表現学習（representation learning）：
  - データから「良い表現」を（できれば教師なしで）獲得する．
    - 良い表現：元のデータの性質をある程度保持しつつ，他のタスクにも使い回せるような表現．
  - Meta-Prior [Bengio+ 13, Goodfellow+ 16]
    - 同時に多くのタスクに使える表現の性質に関する仮定
    - 多様体，Disentangle，概念の階層性，半教師あり学習，クラスタ性など

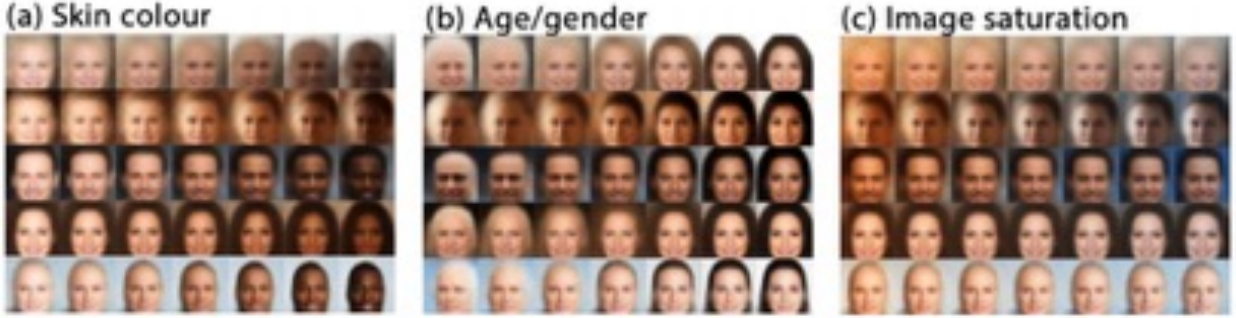
Disentangled representation

- Disentangled representation（もつれを解く表現）
  - データは独立に変化する要素から生成されているという仮定
    - 例) 物体の向き，光源の状態
  - 利点：
    - 人間が解釈しやすい表現（「概念」の獲得）
    - 様々なタスクに転用できる可能性



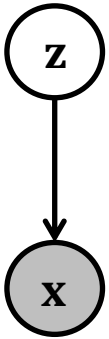
https://www.slideshare.net/lubaelliott/emily-denton-unsupervised-learning-of-disentangled-representations-from-video-creative-ai-meetup

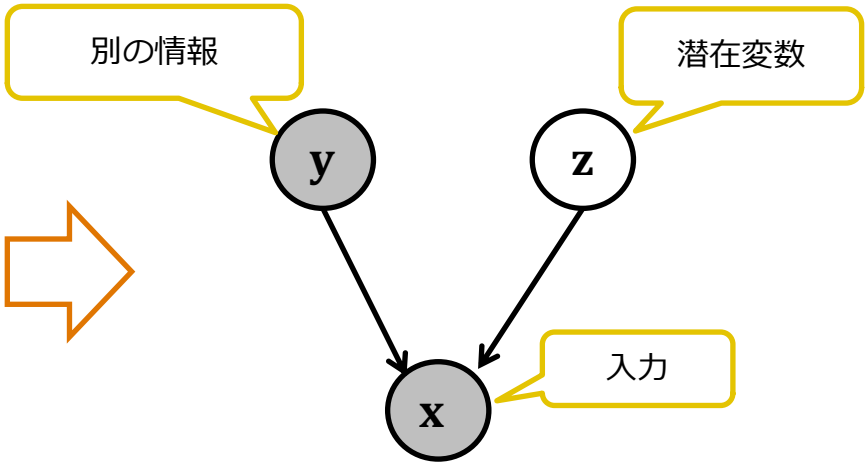
- 推論モデルへの正則化によってdisentangleな表現を獲得可能[Higgins+ 17]



条件付き深層生成モデル

- 観測変数$\boldsymbol{y}$ （$\mathbf{x}$と異なる情報）で条件づけた深層生成モデル
  - $\boldsymbol{y}$ から$\mathbf{x}$への生成過程を表現.





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

$$p(\mathbf{x}|\mathbf{y}) = \int p(\mathbf{x}|\mathbf{z},\mathbf{y})p(\mathbf{z})d\mathbf{z}$$

条件付き深層生成モデル

- 属性**y**から画像**x**の生成[Larsen+ 15]

- 文書**y**から画像**x**の生成[Mansimov+ 15]


A stop sign is flying in blue skies.

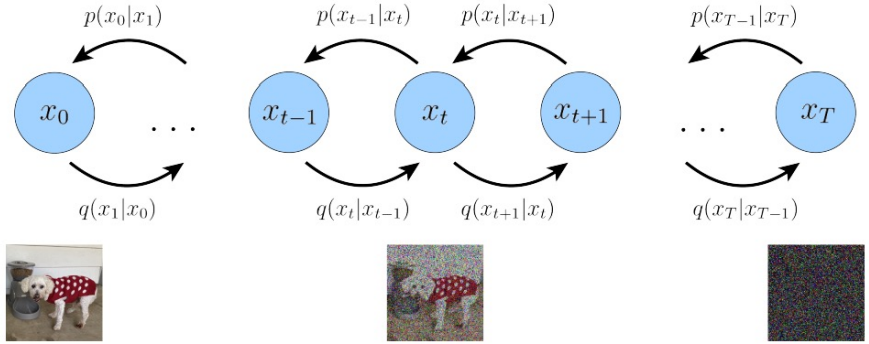

A herd of elephants flying in the blue skies.


A toilet seat sits open in the grass field.


A person skiing on sand clad vast desert.

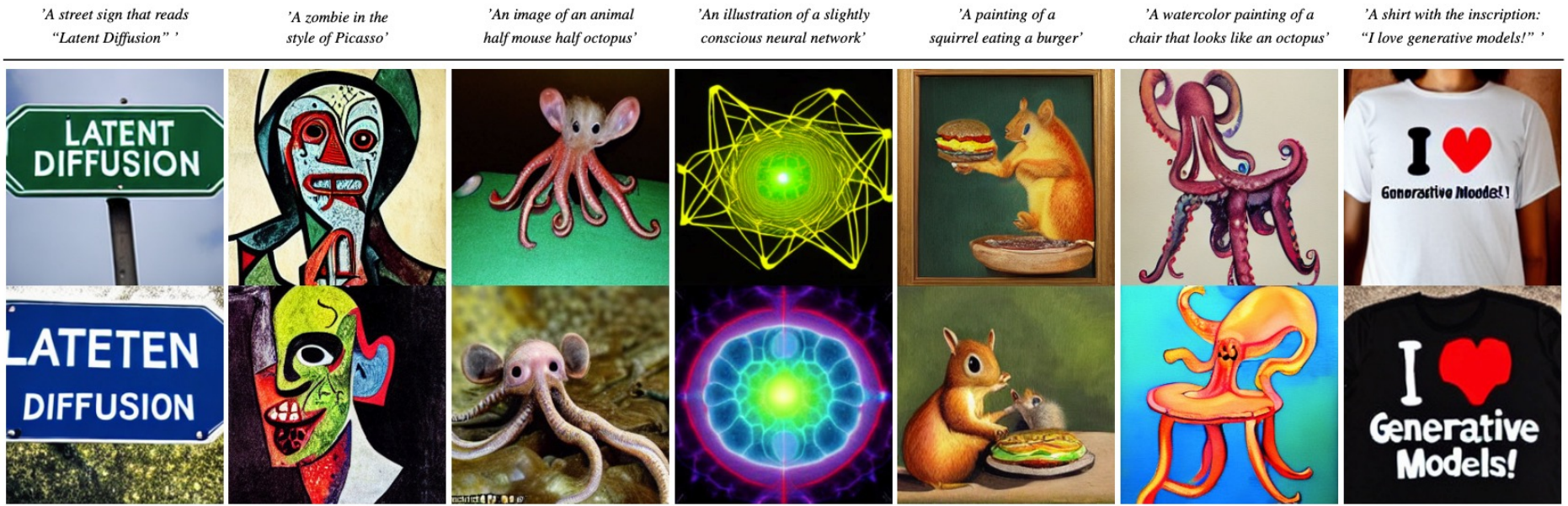
拡散モデル（diffusion model）

- 入力$\mathbf{x}_0$を画像とし，画像$\mathbf{x}_0$からランダムノイズ$\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, I)$ にする拡散過程と，ランダムノイズ$\mathbf{x}_T$から画像$\mathbf{x}_0$に戻す逆過程を考える．
  - 拡散過程と逆過程のいずれもガウス分布と仮定．
  - 拡散過程の平均・分散は学習せず，逆過程の平均を再パラメータ化して学習．
    - さらに再パラメータ化して，任意のステップtの画像$\mathbf{x}_t$からノイズ$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I)$を生成するように学習．
- VAEとは異なり，Tステップの反復的な生成を行う．
  - Tは1000など．



条件付き深層生成モデル

- 大規模言語モデルと拡散モデルを使って，文書から高解像度の画像生成.
  - Stable Diffusion [Rombach+ 22]

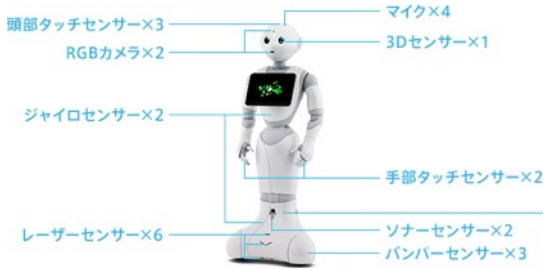


条件付き深層生成モデル

- Stable Diffusionで「A stop sign is flying in blue skies.」を生成
  - 「A stop sign」+「flying」が必ずしも適切に表現できている訳ではない.

マルチモーダル学習

- 我々はマルチモーダル情報を取り入れることで，単一のモダリティ情報よりも確実な情報処理を行っている．
- ロボットも複数のセンサから様々な種類の情報を獲得している
  - 動画，音声，角度や加速度情報，距離情報など



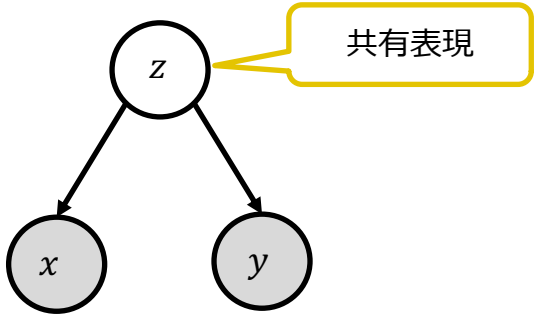
https://www.softbank.jp/robot/consumer/products/spec/

- 機械学習においても，マルチモーダルデータを活用して判断・予測を行いたい．

⇒ マルチモーダル学習

深層生成モデルによる同時モデル

- 異なるモダリティの同時分布$p(x, y)$をモデル化
  - 適切に学習できれば，任意の条件付けを行って生成できるはず（双方向生成，$p(x|y), p(y|x)$）
  - 潜在変数は2つのモダリティを統合した表現（共有表現）を獲得できるはず．

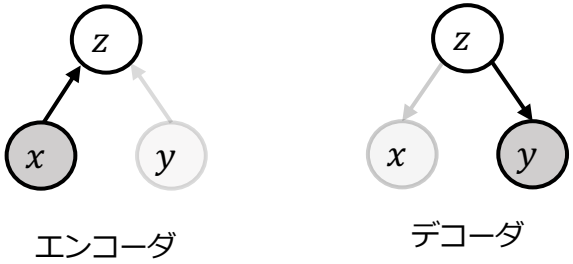


$$p(x, y) = \int p(x|z)p(y|z)p(z)dz$$

- VAEで容易に実現可能（マルチモーダルVAE）
  - モダリティごとにデコーダを増やす．

欠損モダリティ問題

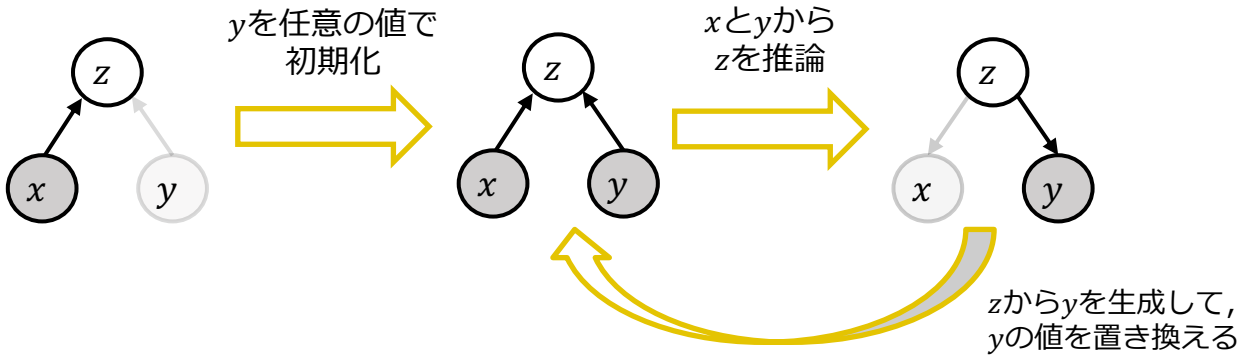
- モダリティ間を双方向変換するときは，片方のモダリティを欠損させる．
  例：xからyへの変換



- 欠損させるモダリティyの情報量が大きい場合，xだけでは適切にzを推論できずに崩れてしまう可能性がある（欠損モダリティ問題）．
  - zが適切に推論できなければ，yも適切に生成できなくなる．

既存手法：反復サンプリング手法

- VAEにおける欠損値補完
  - 反復的に欠損した部分（モダリティ）を補完する（反復サンプリング手法[Rezende 14]）



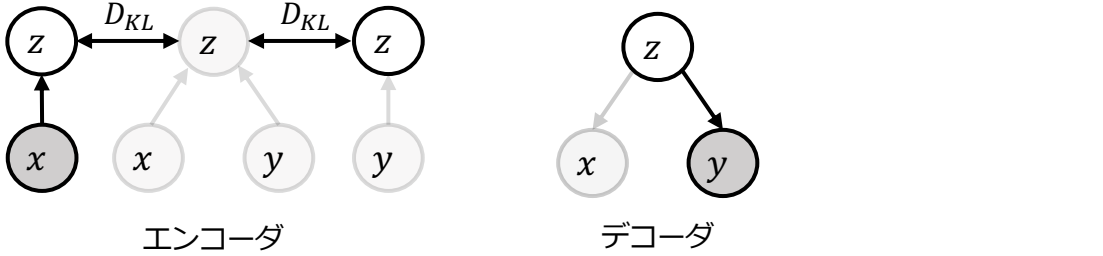
しかし欠損値の情報量が大きい場合，この手法では補完しきれないことを確認.

提案手法：JMVAE

- 単一モダリティのエンコーダ$q(z|x)$，$q(z|y)$を用意して，元のエンコーダ$q(z|x,y)$を近似する．
  - VAEの目的関数に近似のための項（KLダイバージェンス）を加える．

$$\mathrm{E}_{q_\varphi(z|x,y)}[\log p_\theta(x,y|z)] - D_{KL}[q_\varphi(z|x,y) \| p_\theta(z)] \; -D_{KL}[q_\varphi(z|x,y) \| q_\lambda(z|x)] \; -D_{KL}[q_\varphi(z|x,y) \| q_\lambda(z|y)]$$

元の目的関数　　単一モダリティのエンコーダの近似の項

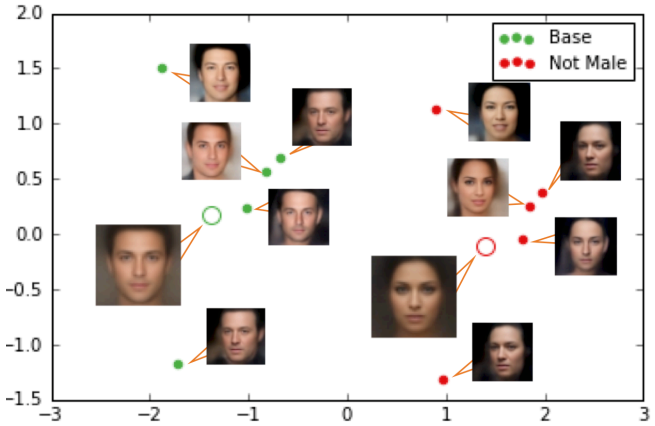


- $q(z|x)$と$q(z|y)$を単一モダリティ入力から潜在変数への写像に用いる．
  - 欠損補完をせずに潜在変数を推論できるようになる（反復サンプリングが必要ない）．

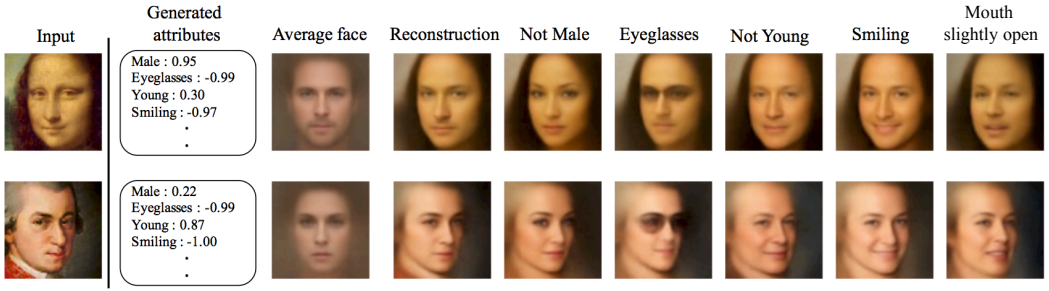
=> JMVAE [Suzuki+ 17]

JMVAE

- 双方向変換や，共有表現の学習が可能
  - e.g., 画像 (x) と属性(y)



共有表現

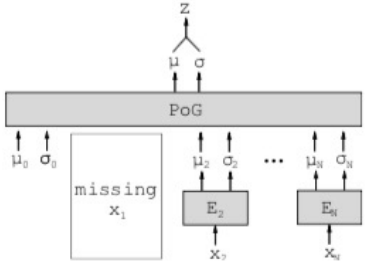


双方向の変換

- 半教師あり学習やゼロショット学習などへの応用.

マルチモーダルVAEの発展

- JMVAEの課題
  - モダリティごとに推論モデルを用意する必要がある.
  - 3つ以上のモダリティに拡張困難.

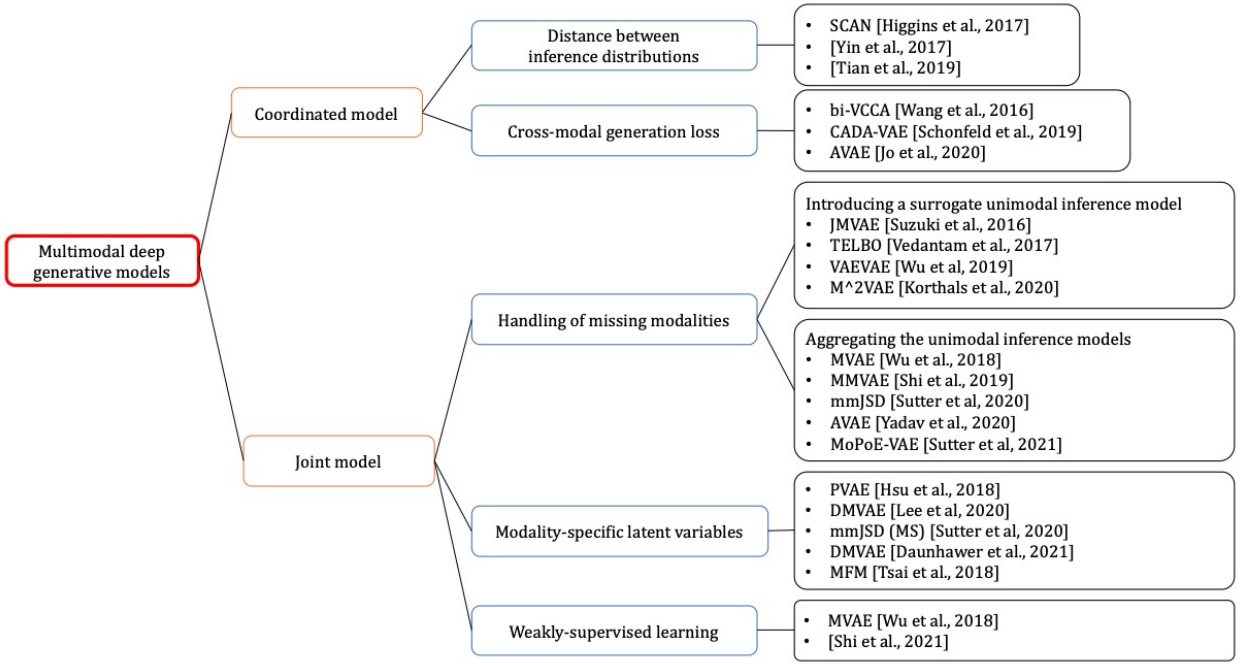


- MVAE[Wu+ 18]
  - 異なるモダリティへの推論として，エキスパートの積（PoE）を利用（右図）
  - 追加の推論モデルが不要&任意のモダリティ数に拡張可能.
  - 混合エキスパート（エキスパートの和，MoE）を用いる手法もある[Kurle+ 18, Shi+ 19].

- MoPoE-VAE[Sutter+ 21]
  - PoEとMoEを一般化したMixture of Products of Expert（MoPoE）を利用する方法を提案．任意の数のモダリティからの条件付き生成や表現の推論を実現（右図）.
  - 条件付き生成において理論的限界がある[Daunhawer+ 21]

マルチモーダルVAEの発展

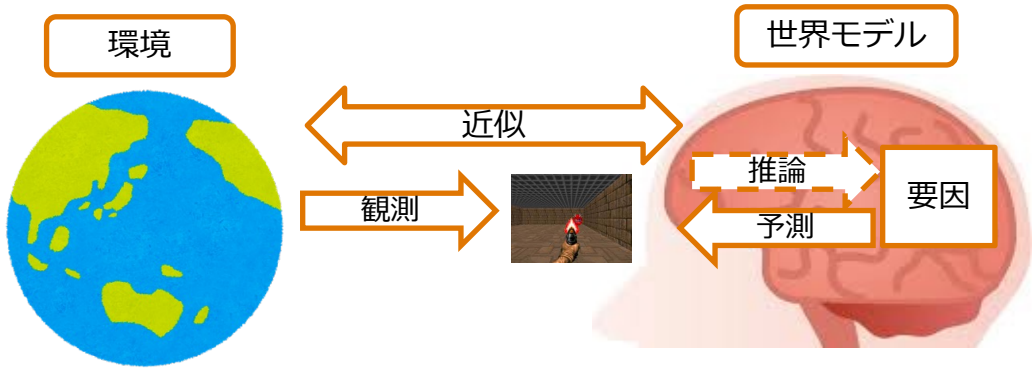
- A survey of multimodal deep generative models [Suzuki+ 22]



世界モデル

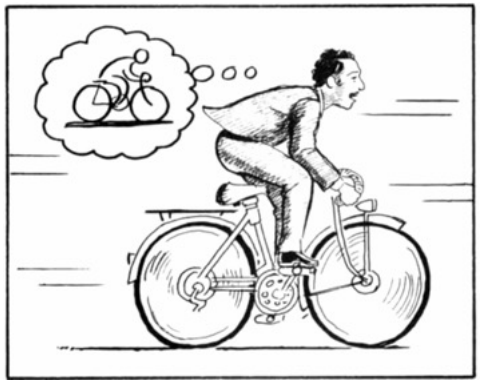
- 人間は世界のあらゆるものを知覚できるわけではない.
  - 脳に入ってくる情報は非常に限られている.
  - したがって脳の内部では，限られた情報から現実世界をモデル化している.

- 世界モデル（world model）：
  - 外界からの限られた観測を元に，世界の構造を近似するように学習するモデル.
  - 観測から要因を推論し，推論した要因から未来や未知のことを予測（生成）する.

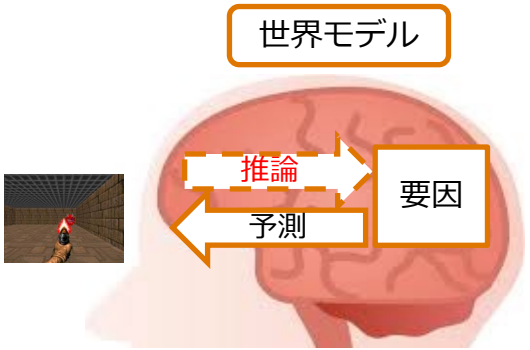


世界モデルにおける表現の獲得

- 脳内では，外界からの情報を空間的・時間的な表現に圧縮している．
  - 例：自電車を漕いでいる人は「自転車を漕いでいる」という表現を時間的・空間的に圧縮している．

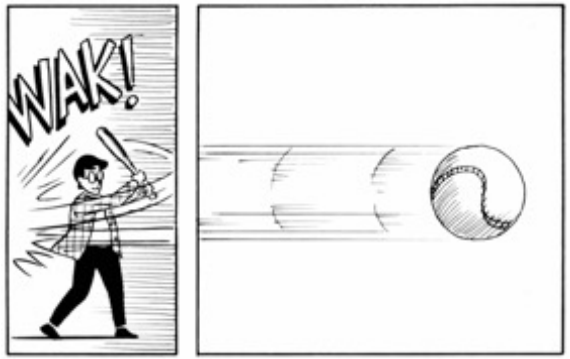


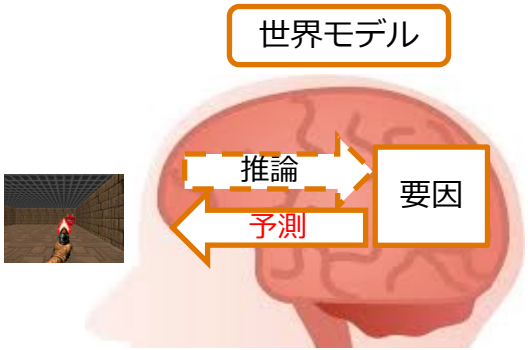
- これは，表現の階層的な推論に対応する．
  - 世界モデルは推論による表現学習を行っている．



世界モデルによる予測

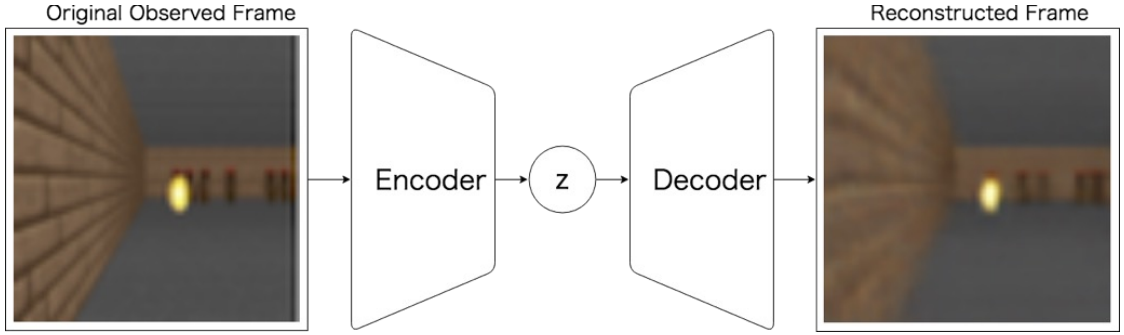
- 学習した世界モデルによって未来をシミュレーションしている.
  - 人間はこれを常に行っていると考えられる.
- 例：バットを振ってボールに当てる
  - ボールが飛び去る時間は，視覚情報が脳に到達してバットの振り方を決めて筋肉を動かす時間よりも短い.
  - 世界モデルによって無意識に予測を行い，それにしたがって筋肉を動かしている.

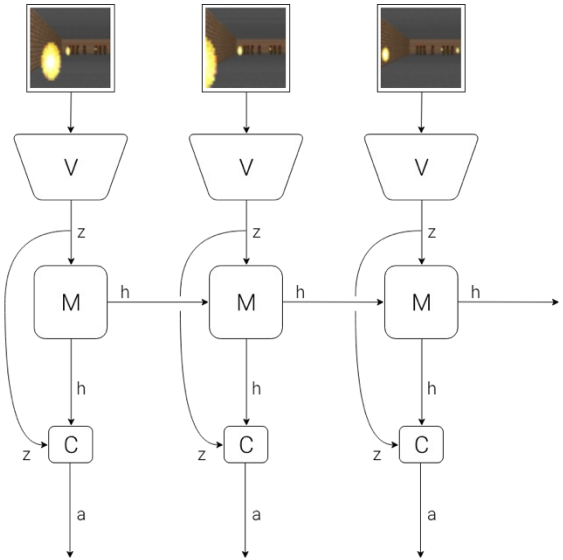




深層生成モデルを用いた世界モデル

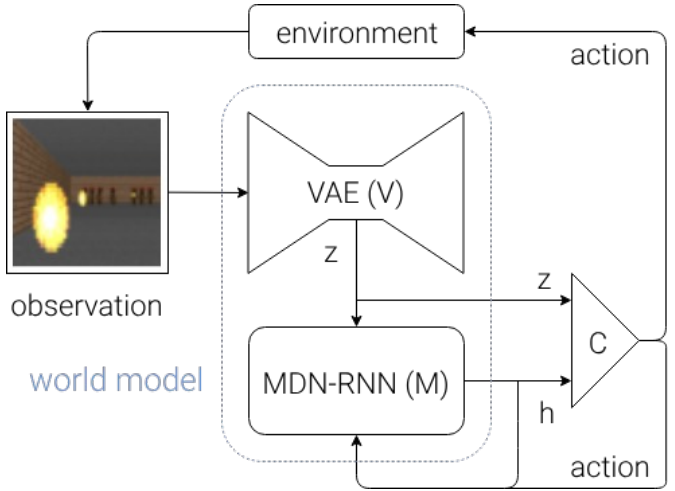
- World Model[Ha+ 18]
  - VAEとMDN-RNN[Graves + 13, Ha+ 17]で，ゲーム環境の世界モデルを学習





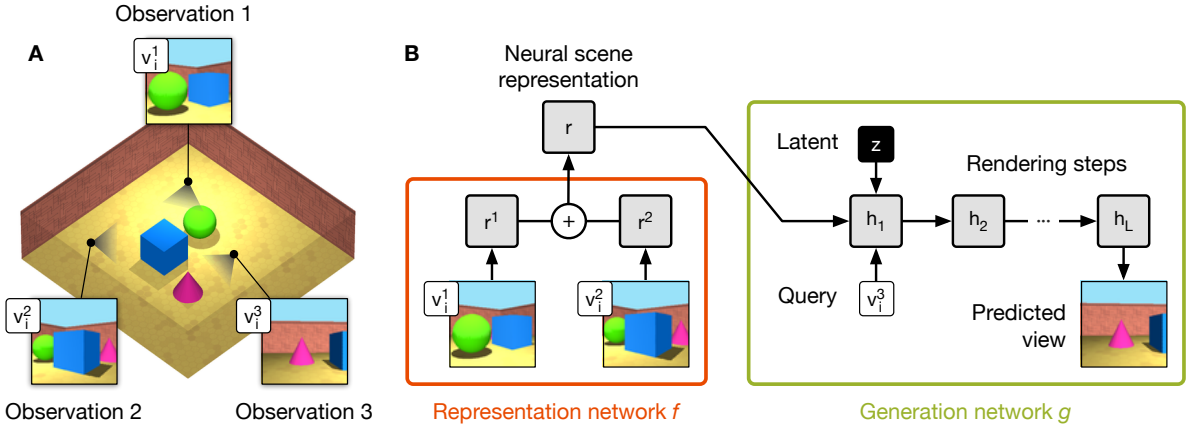
世界モデル内での強化学習

- 学習した世界モデルの中で強化学習を行う
  - 人間でいうイメージトレーニングや睡眠学習のようなもの
  - 実世界とは違い，何回でも学習できる.
- 実世界（ゲーム）でテストすると，正しく行動できていることがわかる.



複雑な環境での世界モデル

- Generative Query Network（GQN）[Eslami+ 18]
  - ある複数の視点における画像を元に，別の視点の画像を予測する世界モデル.
  - 条件付け深層生成モデルの利用.

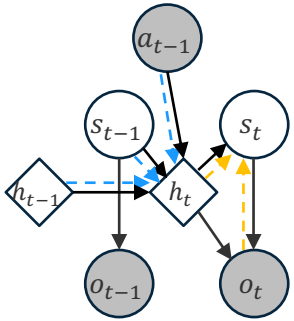


表現ネットワーク

生成ネットワーク
（生成モデル）

時系列情報からの世界モデルの獲得

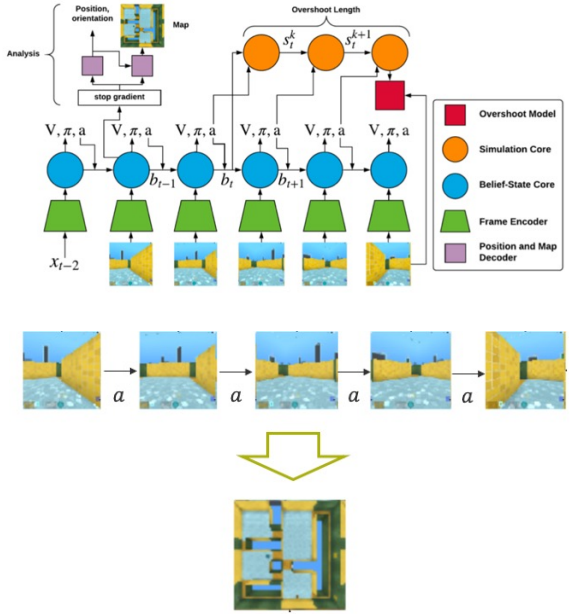
- Dreamer [Hafner+ 20]
  - 時系列情報から（回帰結合型）状態空間モデルによって世界モデルを学習し強化学習を行う.
    - 潜在空間上での想像に基づき，長期的な価値を推定.
    - DNNで方策と価値関数をパラメータ化し，交互に学習.
  - 微分可能な世界モデル上の想像に基づいて計算されるので，方策の勾配が計算できる.
  - 難しい視覚的制御において，高いサンプル効率や性能を発揮.



回帰結合型状態空間モデル（RSSM）[Hafner+ 18]

時系列情報からの世界モデルの獲得

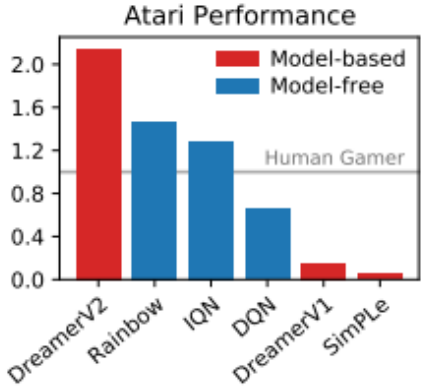
- エージェントの視点と行動から，世界の不変的な構造を学習 [Gregor+ 19]
- 状態でどのような表現が獲得されているかを確認
  - エージェントが歩き回ることによって，環境の地図が作成される．
  - 表現空間上で，一貫性のある表現が獲得できていることがわかる．

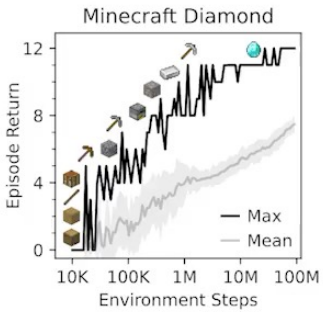


https://www.youtube.com/watch?v=dOnvAp_wxv0

Dreamerの発展

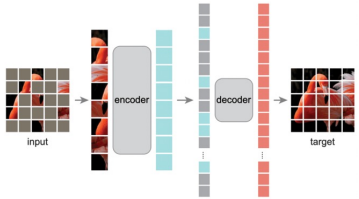
- Dreamer v2[Hafner+ 21]
  - Dreamerの状態表現を離散表現にして，正則化の学習部分を工夫する（Priorの学習率を大きくする）ことで，Atariにおいてモデルフリーを大幅に上回る結果を出した.
- Dreamer v3[Hafner+ 23]
  - モデルを大規模にし，symlog関数による予測対象の正規化などの工夫を入れることで， Minecraftのダイヤモンド収集タスクのような複雑なタスクを（人間のデモ等を使わずに）初めて解くことができた.



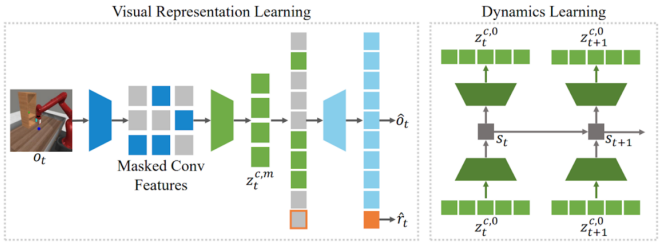


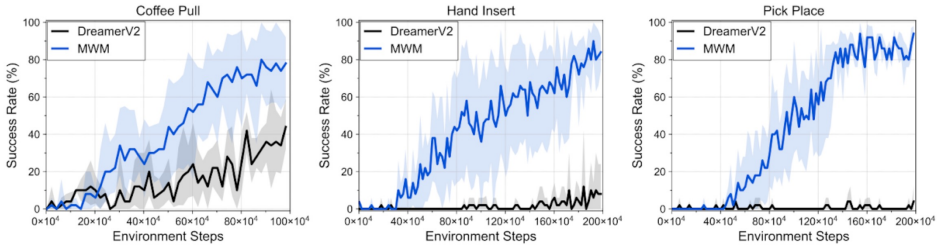
Masked World Model

- Masked Autoencoder[He+ 21]
  - Vision Transformer[Dosovitskiy+ 20]の事前学習として，入力画像のパッチをランダムにマスクして復元するように学習する.



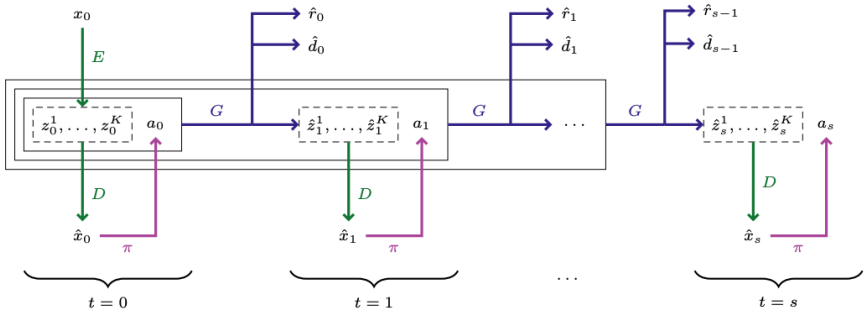
- Masked World Model [Seo+ 22]
  - Masked Autoencoderを用いて表現学習を行うことで，小さい物体との相互作用のモデル化性能が向上し，操作系の実験で高い性能.





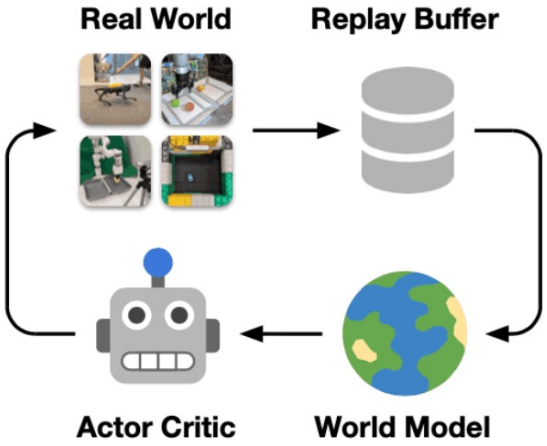
大規模言語モデルによる世界モデル

- Transformers are Sample-Efficient World Models [Micheli+ 23]
  - 潜在空間での遷移や予測を系列モデリング問題と捉えて，Transformerを利用
  - 2時間程度でAtariのベンチマークで人間レベルを達成

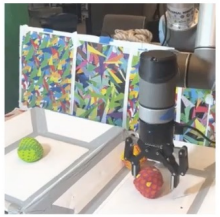


DayDreamer

- DayDreamer [Wu+ 22] ：Dreamer v2の実ロボットへの応用
  - ロボットが環境と相互作用して収集したデータから世界モデルを学習する.
  - ロボットは方策を世界モデル上のみで学習する.
  - 世界モデルを用いることで効率的に学習でき，新しいタスクや摂動に対しても対応できる.

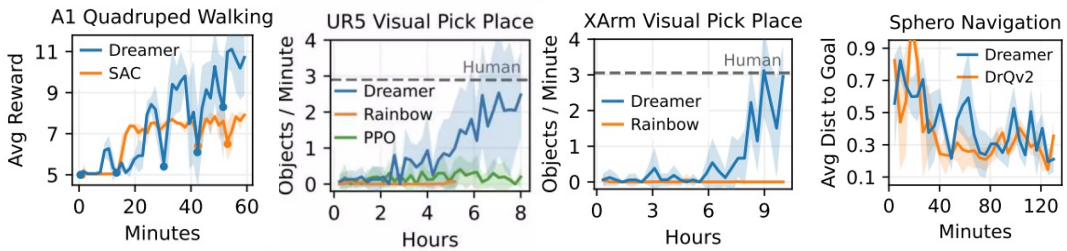



A1 Quadruped Walking


UR5 Multi-Object Visual Pick Place

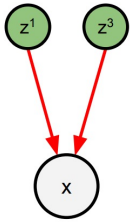

XArm Visual Pick and Place

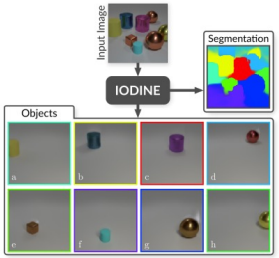

Sphero Ollie Visual Navigation



物体中心表現学習

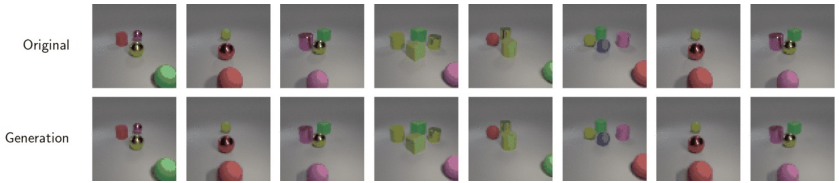
- 物体中心表現学習（object-centric representation learning）
  - 画像から物体ごとの表現を教示なしで認識（推論）・生成する枠組み.



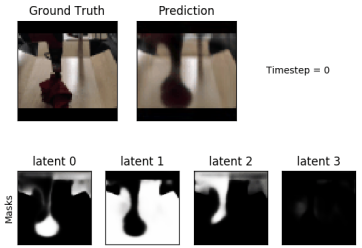


[Greff+ 20]

- 物体が時間変化する動画から物体ごとの表現と予測モデルを学習する場合は，物体中心世界モデル [Lin+ 20, Jiang+ 20]とも呼ばれる.



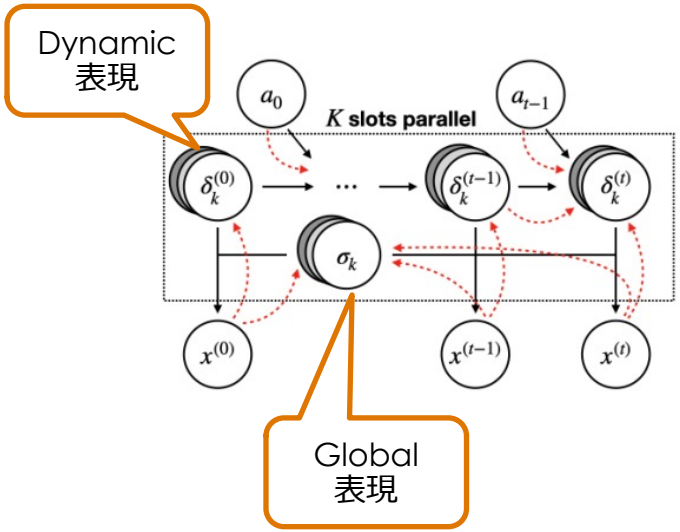
[Lin+ 20]

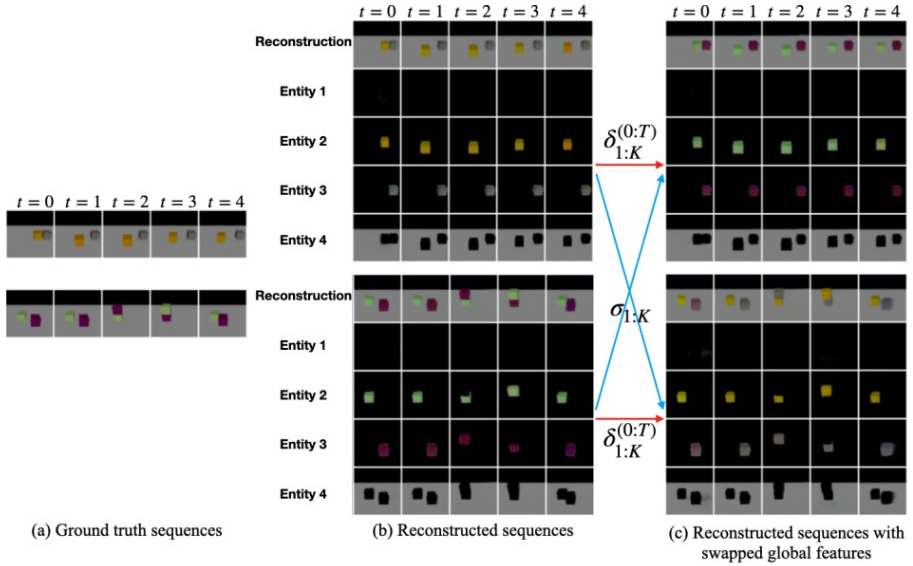


[Veerapaneni + 19]

物体中心表現の分離

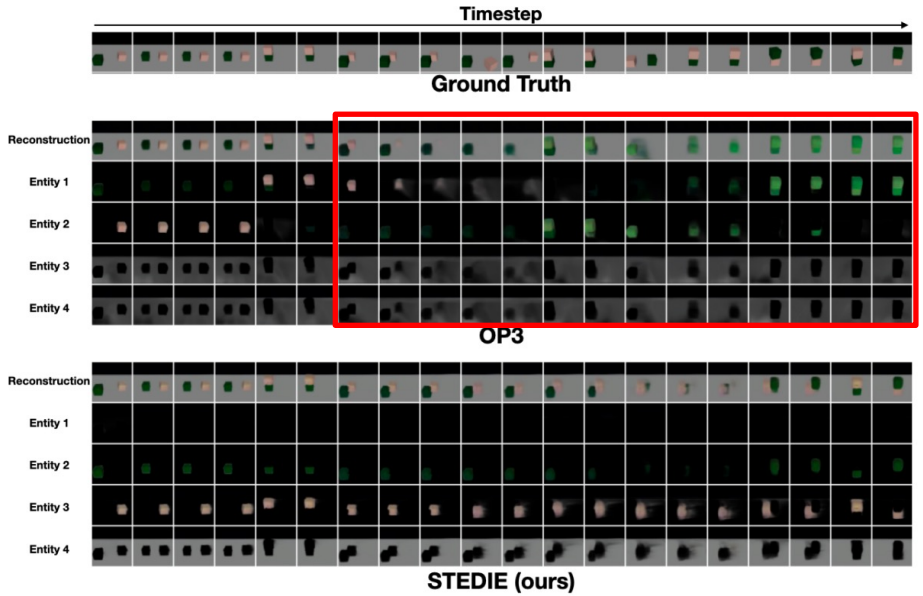
- 物体の表現のうち，相互作用に関係する表現（位置など，dynamic表現）と関係しない表現（色など，global表現）を分離するモデルを提案[Nakano+ 23; ICLR2023]
  - 相互作用に関係しない表現を分離して獲得することに成功（物体の色だけを交換することができる）．



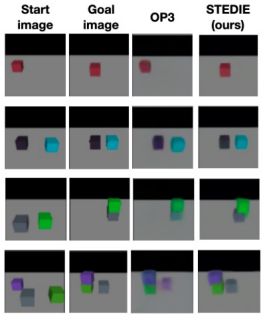


物体中心表現の分離

- 表現を分離することで，世界モデル上での長期予測やプランニング性能が向上[Nakano+ 23; ICLR2023]
  - 長期予測
    - 既存手法（OP3）は途中で予測に失敗する
    （赤で囲った部分）



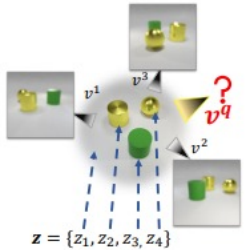
- プランニング
  - ゴール画像に近づくように世界モデル内で操作を計画し，実際の環境で実行．
  - 物体の数が増えると，提案手法の方が高い性能



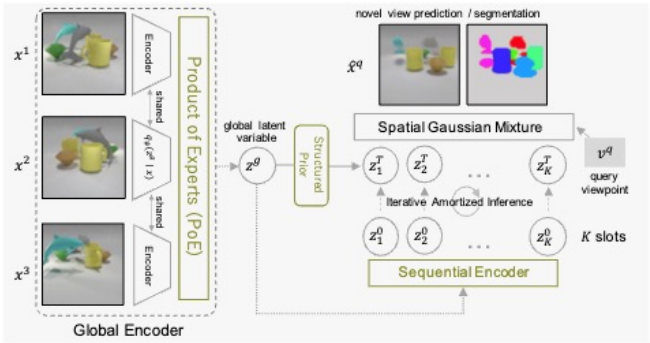
# Blocks	Model	Acc. (↑)	MSE (↓)	Average CEM steps (↓)
1	OP3	**85%**	0.0014	1.17
	STEDIE (ours)	84%	**0.0012**	**1.00**
2	OP3	58%	0.0049	1.63
	STEDIE (ours)	**63%**	**0.0041**	**1.45**
3	OP3	42%	0.0102	3.50
	STEDIE (ours)	**55%**	**0.0075**	**1.82**

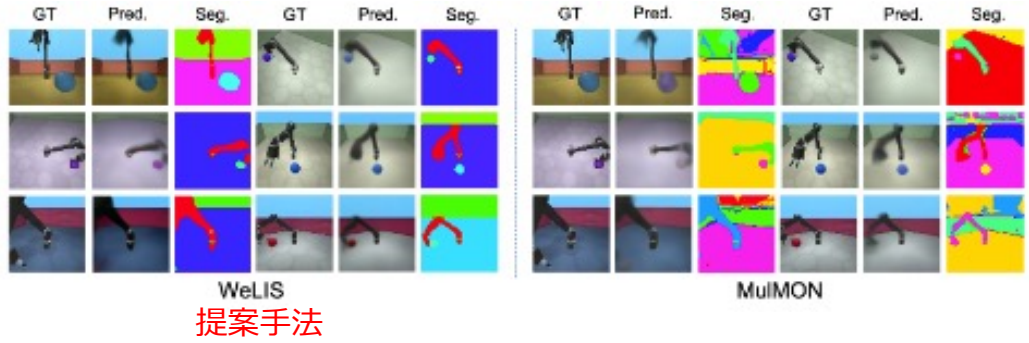
物体中心表現学習

- 複数視点の情報を元に，物体の表現を獲得する（多視点物体中心表現学習）[Nanbo+ 2020].
  - GQN+物体中心表現学習



- 物体ごとの表現だけでなく空間の表現（global表現）も獲得する方法を提案[Kobayashi+ 23] .
  - 従来手法より推論の性能が向上したり，新規のシーン生成も可能.

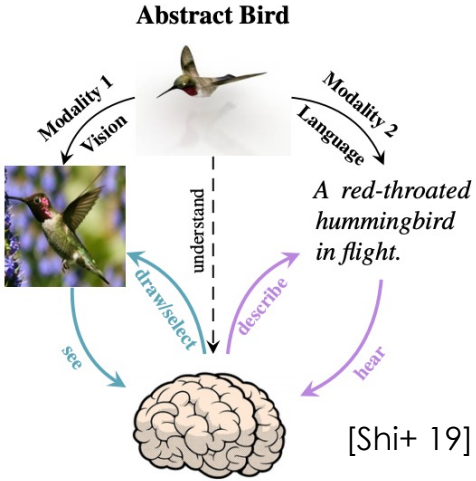




提案手法

世界モデルとマルチモーダル学習

- 従来の世界モデル研究では，単一モダリティ（主に画像）のみを扱っていた．
  - 人間は様々なモダリティ情報を元に，脳内に抽象的な表現を獲得している．

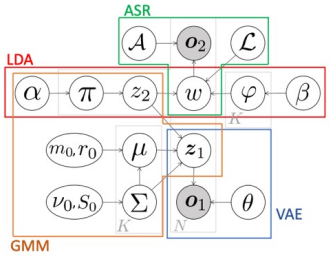


[Shi+ 19]

⇒ 世界モデルにおけるマルチモーダル学習の重要性

Neuro-SERKET・WB-PGM

- Neuro-SERKET [Taniguchi+ 19]
  - 深層生成モデルを含む確率的生成モデルによるマルチモーダル認知アーキテクチャの提案
  - VAE+GMM+LDA+ASRの例（数字の画像（MNIST）と音声から学習）



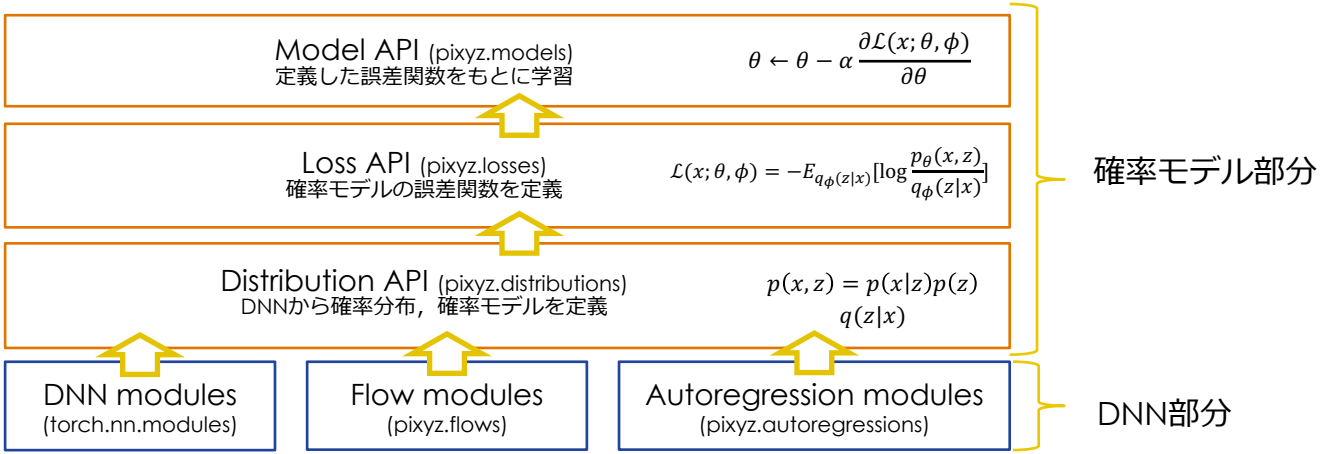
model	Accuracy (%)		Features introduced in Neuro-SERKET		
	GMM	LDA	Head-to-head	Tail-to-tail	Neural net
VAE GMM LDA ASR	63.7	27.5			
VAE GMM LDA+ASR	63.7	92.7	✓		
VAE+GMM LDA+ASR	66.7	92.7	✓		✓
VAE+GMM+LDA+ASR	**91.0**	**93.7**	✓	✓	✓

- 同様の枠組みで脳全体を確率的生成モデルで構築するwhole-brain PGM（WB-PGM）の提案 [Taniguchi+ 21]
- 今後の課題：
  - 複雑な認知アーキテクチャを全て深層生成モデルで設計・学習することは可能か？
  - そうした複雑な深層生成モデルを簡潔に実装することは可能か？

Pixyz

- Pixyz [Suzuki+ 22]：深層生成モデルに特化した確率プログラミングライブラリ
  - 複雑な深層生成モデルを簡潔に実装できることが特徴
  - PyTorch（深層学習ライブラリ）ベース

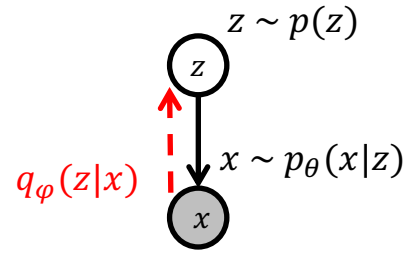
- 直感的な実装を実現するために，DNNと確率モデルの設計が分離していることが特徴.
  - DNNを意識せずに確率モデルを実装に集中できる.
  - Pixyzの構成図（実装の流れ）：



Pixyzの実装例

- Variational autoencoder (VAE) ：

$$-E_{p_{data}(x)}[D_{KL}[q_\varphi(z|x) \parallel p(z)] + E_{q_\varphi(z|x)}[\log p_\theta(x|z)]]$$



PyTorch

```
class VAE(nn.Module):

    def __init__(self):
        super(VAE, self).__init__()

        self.fc1 = nn.Linear(784, 512)
        self.fc21 = nn.Linear(512, 2)
        self.fc22 = nn.Linear(512, 2)

        self.fc3 = nn.Linear(2, 512)
        self.fc4 = nn.Linear(512, 784)

        self.relu = nn.ReLU()
        self.sigmoid = nn.Sigmoid()

    def encode(self, x):
        h = self.relu(self.fc1(x))
        return self.fc21(h), self.fc22(h)

    def reparameterize(self, mu, logvar):
        if self.training:
            std = logvar.mul(0.5).exp_()
            eps = Variable(std.data.new(std.size()).normal_())
            return eps.mul(std).add_(mu)
        else:
            return mu

    def decode(self, z):
        h = self.relu(self.fc3(z))
        return self.sigmoid(self.fc4(h))

    def forward(self, x):
        x = x.view(-1, 784)
        mu, logvar = self.encode(x)
        z = self.reparameterize(mu, logvar)
        return self.decode(z), mu, logvar
```

Pixyz

```
class Inference(Normal):
    def __init__(self):
        super(Inference, self).__init__(cond_var=["x"], var=["z"], name="q")
        self.model = nn.Sequential(nn.Linear(2, 512), nn.ReLU())
        self.loc, self.scale = nn.Linear(512, z_dim), nn.Linear(512, z_dim)
    def forward(self, x):
        return {"loc": self.loc(self.model(x)), "scale": F.softplus(self.scale(self.model(x)))}
q = Inference()
```

```
class Generator(Bernoulli):
    def __init__(self):
        super(Generator, self).__init__(cond_var=["z"], var=["x"], name="p")
        self.model = nn.Sequential(nn.Linear(z_dim, 512), nn.ReLU(), nn.Linear(512, x_dim))
    def forward(self, z):
        return {"probs": torch.sigmoid(self.model(x))}
p = Generator()
```

DNN部分（確率分布を定義）

```
elbo = (-KullbackLeibler(q, prior) + E(q, LogProb(p))).mean()
```

確率モデル部分
（1行で書ける）

-> 確率モデルの式と対応した簡潔な実装

複雑な深層生成モデルの実装

- TD-VAE[Gregor+ 18]：深層生成モデルによる時系列モデルの一つ

$$\mathop{\mathbb{E}}_{q\left(z_{t_1}, z_{t_2} | b_{t_1}, b_{t_2}\right)} \left[\log p\left(x_{t_2} | z_{t_2}\right) + \log p_B\left(z_{t_1} | b_{t_1}\right) + \log p\left(z_{t_2} | z_{t_1}\right) - \log p_B\left(z_{t_2} | b_{t_2}\right) - \log q\left(z_{t_1} | z_{t_2}, b_{t_1}, b_{t_2}\right)\right]$$

  - 従来の深層確率プログラミング言語では実装困難

- Pixyzでの実装（確率モデリング部分のみ表示）

```
kl = KullbackLeibler(q, p_b1)
reconst = E(q,  -p_t.log_prob() - p_d.log_prob() + p_b2.log_prob())
step_loss = E(p_b2, reconst + kl)
_loss = IterativeLoss(step_loss, max_iter=seq_len-1,
                      series_var=["x", "b"], timestep_var=["t"],
                      slice_step=slice_step)
loss_cls = E(belief_state_net, _loss).mean()
print_latex(loss_cls)
```

実装したモデルは
式として可視化できる

$$mean\left(\mathbb{E}_{p(b|x)}\left[\sum_{t=1}^{19} \mathbb{E}_{f(x_{t2}, b_{t1}, b_{t2}|t,x,b)}\left[\mathbb{E}_{p_b(z_{t2}|b_{t2})}\left[D_{KL}\left[q(z_{t1}|z_{t2}, b_{t1}, b_{t2})||p_b(z_{t1}|b_{t1})\right] + \mathbb{E}_{q(z_{t1}|z_{t2}b_{t1}, b_{t2})}\left[\log p_b(z_{t2}|b_{t2}) - \log p_d(x_{t2}|z_{t2}) - \log p_t(z_{t2}|z_{t1})\right]\right]\right]\right)$$

Pxyzの学習速度

- PyTorch上に実装されている既存の確率プログラミング言語Pyroと比較.
  - PyTorchでの実装（確率プログラミング言語を使わない実装）とも比較.
  - 1ステップあたりのVAEの学習時間の比較（z：潜在変数の次元，h：隠れ層の次元）

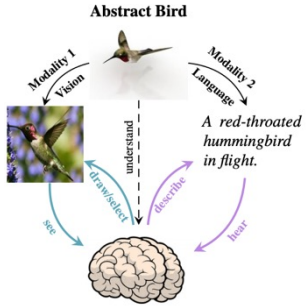
$\#z$	$\#h$	PyTorch (ms)	Pyro (ms)	Pixyz (ms)
10	400	2.47 ± 0.11	4.91 ± 0.12	3.61 ± 0.11
30	400	2.49 ± 0.10	4.94 ± 0.13	3.58 ± 0.10
10	2000	3.26 ± 0.11	4.93 ± 0.12	3.62 ± 0.09
30	2000	3.28 ± 0.10	4.95 ± 0.12	3.65 ± 0.09

- 結果：
  - Pyroと比べて高速.
  - PyTorchでの実装と比べても大きく速度が落ちていない.

速度面からも，Pixyzは複雑な深層生成モデルの実装に適している

世界モデルの今後の課題（一部）

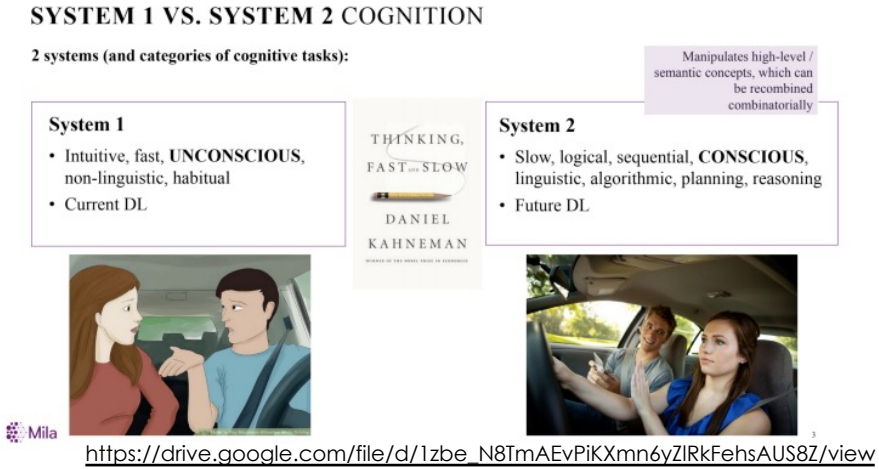
- それぞれの設計方法の世界モデルをどのように利用していくか？
- 時系列情報をどのように抽象化するか？（時間的抽象化）
  - タスクに応じたより高度な抽象化が重要.
- 不完全な世界モデルの中でどのように振る舞うべきか？
  - 世界モデルは常に「不完全」（世界の全てを知っているわけではない）.
  - 世界モデルの利用と更新のバランス.
- 他者をどのようにモデル化するか？
  - 環境のダイナミクスの中から「他者」を特定して，内部的にモデル化するには？
- マルチモーダル情報をどのように扱うか？
  - 我々は視覚だけでなく，言語など様々な種類の情報（マルチモーダル情報）から世界を構築している.
  - あるモダリティから別のモダリティを推論することもできる（例：本を読んで情景を思い浮かべる）



[Shi+ 19]

システム1とシステム2

- 「ファスト&スロー（by ダニエル・カーネマン）」より



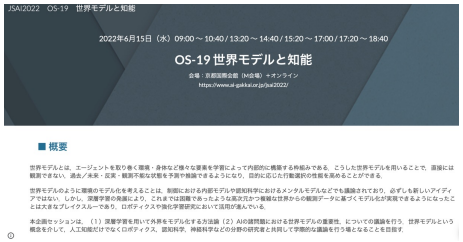
https://drive.google.com/file/d/1zbe_N8TmAEvPiKXmn6yZlRkFehsAUS8Z/view

- 古典的な人工知能は，探索やシンボルに基づく推論が中心的（システム2）
- 深層学習の登場によって，（人間でいう）直感的な振る舞いを学習できるようになった（システム1）

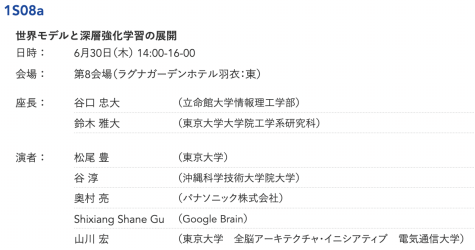
世界モデルと古典的な人工知能の融合が重要になる（システム1側からシステム2を再構築する）

世界モデル研究に向けた活動

- OSやワークショップの開催



JSAI OS「世界モデルと知能」



NEURO 2022 ワークショップ
「世界モデルと深層強化学習の展開」

- サーベイ論文の執筆
  - World Models and Predictive Coding for Cognitive and Developmental Robotics: Frontiers and Challenges [Taniguchi+ 23]
    - 世界モデルや予測符号化に関する包括的なサーベイ論文
- 雑誌での世界モデル特集号（NGC，Advanced robotics）

世界モデル・シミュレータ寄附講座（2021年7月～）

- 講座概要
  - 「世界モデル」の構築とその発展に必要なシミュレータ技術の進化は，今後のAI進化の中心的技術課題.
  - 特に，ロボティックスや言語の意味理解などのAI活用分野において，世界モデル技術は不可欠になる.
  - 日本国内で最先端の世界モデルを研究・実装する場を確保し，産業界への技術啓蒙を行うことで，実用化準備を進める.
  - また，講義を始めとする教育活動を通じて，世界モデル技術を扱える人材を育成する.
    - 2022年1月～5月にかけて「世界モデルと知能」講義を初実施．現在2期目を開講中.
- 参画企業

SQUARE ENIX AI&ARTS Alchemy　SONY　NEC

まとめ

- 今回の内容
  - 深層生成モデル
  - 深層生成モデルとマルチモーダル学習
  - 世界モデルと深層生成モデル

- 謝辞：
  - この成果は，JSPS科研費18H06458および国立研究開発法人新エネルギー・産業技術総合開発機構（ＮＥＤＯ）の委託業務の結果得られたものです。