

2026年3月11日 IEICE総合大会 企画セッション

AI-6：統一的体系化によるAI・データサイエンス研究の新展望

CENTER FOR DATA SCIENCE

生成AIの意思決定写像 としての表現の試み



小林 学，中原悠太，須子統太，松嶋敏泰（早稲田大学）

1. 講演の背景と目的

- AI・データサイエンスを統一的に表現するフレームワークとして「意思決定写像」を提案
- 生成AIも「意思決定写像」として表現できなければならない

目的

- テキスト生成AIと画像生成AIで良く用いられる手法を例として、「意思決定写像」としての表現を試みる
- 生成AIを「データからの生成観測メカニズム（情報源）の模倣」と考えて、すでにあるいくつかの応用研究について改めて検討を行ってみる

1. 講演の背景と目的

プロンプトと生成されるデータの関係

- プロンプトを x , 生成されるデータを y と書く
- y の空間に対して, x は「制約」あるいは「条件」と考えることができる
- 多くの場合 $P^*(y|x)$ を目指す

2. テキスト生成AI

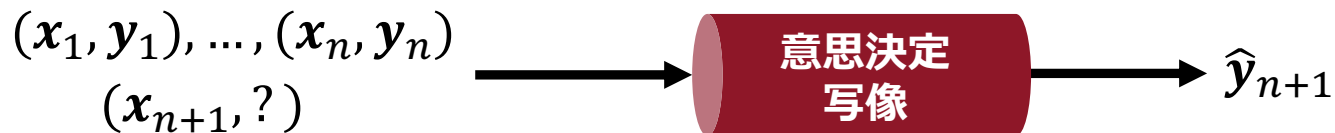
【入力】 学習時：プロンプト x_i とその回答 y_i の n 組の集合

推論時：プロンプト x_{n+1}

【出力】 プロンプト x_{n+1} に対応する回答 \hat{y}_{n+1}

$$x_i = (x_{i1}, x_{i2}, \dots, x_{il_i}), \quad y_i = (y_{i1}, y_{i2}, \dots, y_{im_i}) \quad x_{ik}, y_{ik} \in \mathcal{V}$$

トークン ボキャブラリ



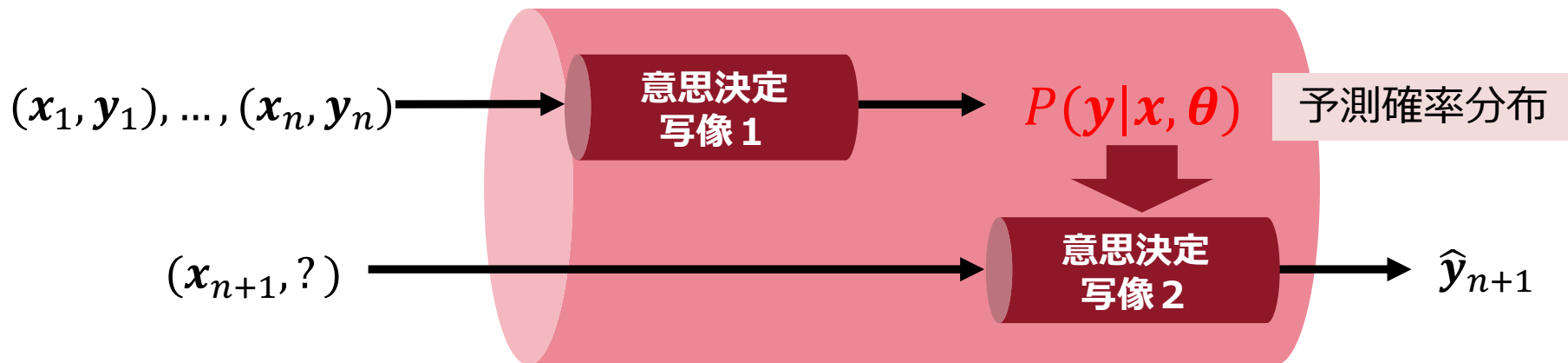
テキスト生成AIの意思決定写像

【入力】 学習時：プロンプト x_i とその回答 y_i の n 組の集合

推論時：プロンプト x_{n+1}

【出力】 プロンプト x_{n+1} に対応する回答 \hat{y}_{n+1}

予測確率分布を用いた間接予測の意思決定写像



θ は確率分布を決めるパラメータ。例えばニューラルネットワークであれば重みパラメータ

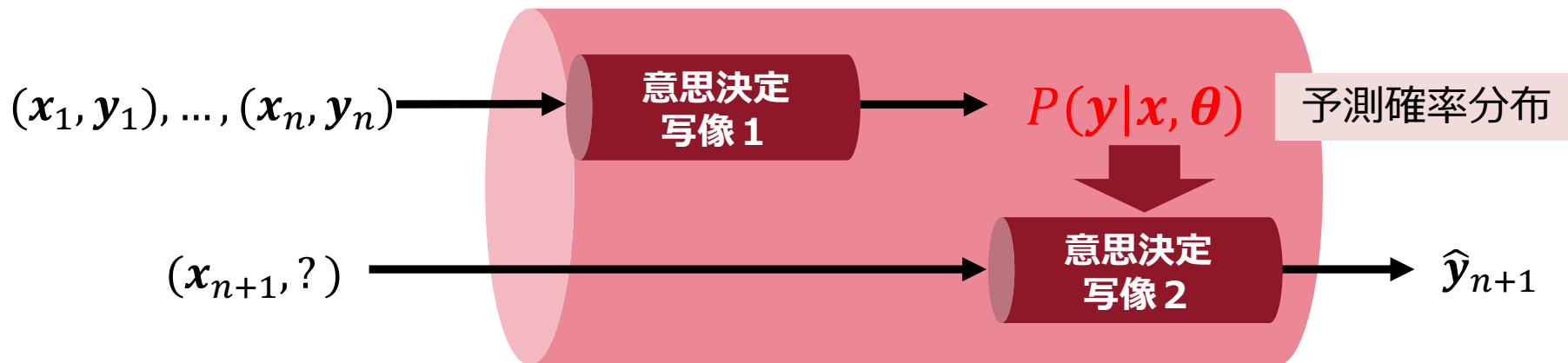
テキスト生成AIの意思決定写像

【入力】 学習時：プロンプト x_i とその回答 y_i の n 組の集合

推論時：プロンプト x_{n+1}

【出力】 プロンプト x_{n+1} に対応する回答 \hat{y}_{n+1}

予測確率分布を用いた間接予測の意思決定写像



予測確率分布 $P(y|x, \theta)$ の構築が主眼 (θ が公開されるとオープンウェイト)

テキスト生成AIの意思決定写像

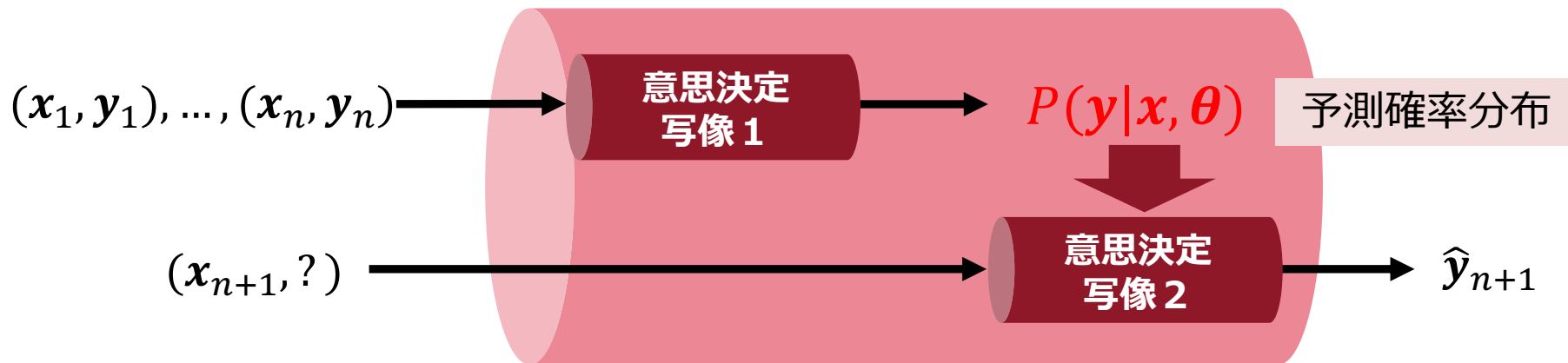
【入力】 学習時：プロンプト x_i とその回答 y_i の n 組の集合

推論時：プロンプト x_{n+1}

【出力】 プロンプト x_{n+1} に対応する回答 \hat{y}_{n+1}

【目的】 出力とする回答の「予測」の問題

予測確率分布を用いた間接予測の意思決定写像



テキスト生成AIの意思決定写像

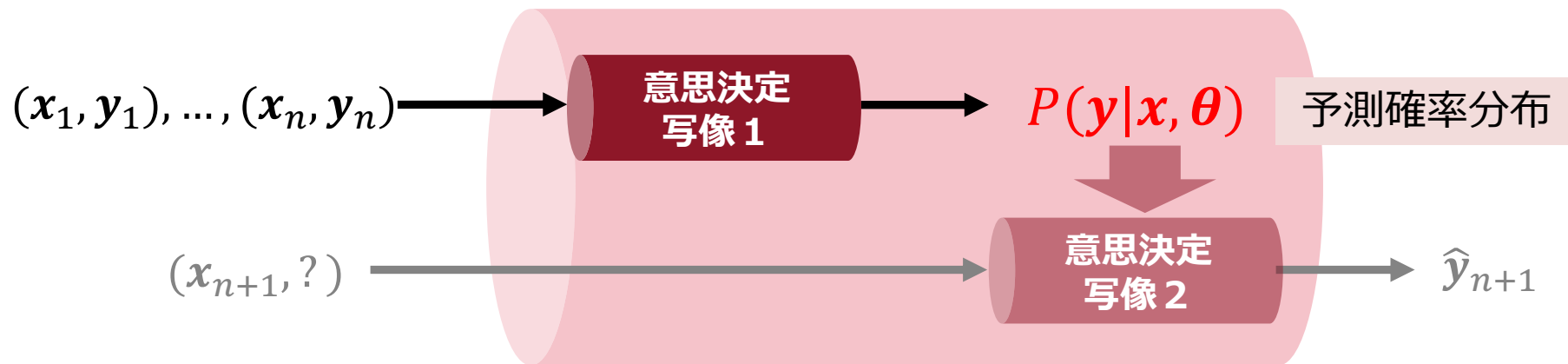
【入力】 学習時：プロンプト x_i とその回答 y_i の n 組の集合

推論時：プロンプト x_{n+1}

【出力】 プロンプト x_{n+1} に対応する回答 \hat{y}_{n+1}

【目的】 出力とする回答の「予測」の問題

簡単のため予測確率分布のパラメータを出力する「意思決定写像 1」に着目



テキスト生成AIの意思決定写像

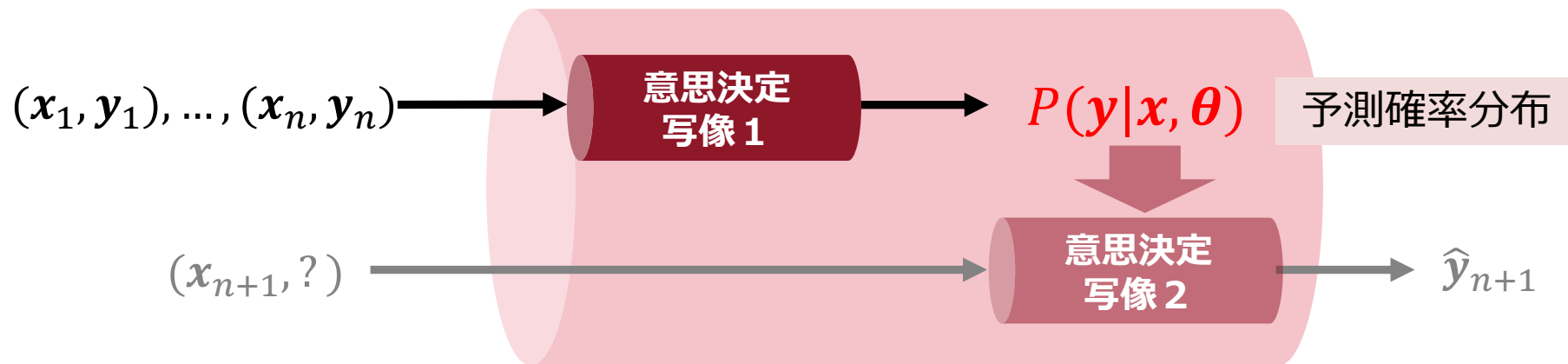
意思決定写像 1

【入力】 プロンプト x_i とその回答 y_i の n 組の集合

【出力】 $P(y|x, \theta)$ の θ を出力

【目的】 出力とする回答の「予測」の問題

簡単のため予測確率分布のパラメータを出力する「意思決定写像 1」に着目



テキスト生成AIの意思決定写像の例 1

意思決定写像 1 の例

【目的】 回答の「予測」の問題

【入力】 プロンプト x_i とその回答 y_i の n 組の集合

【出力】 $P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ の $\boldsymbol{\theta}$ を出力

【設定】 トランスフォーマー（この構成は省略. すみません. . .）

$$Y_1 \sim P(y|\mathbf{x}, \boldsymbol{\theta}), \dots, Y_k \sim P(y|\mathbf{x}, y_1, y_2, \dots, y_{k-1}, \boldsymbol{\theta}), \dots$$

テキスト生成AIの意思決定写像の例 1

意思決定写像 1 の例

【目的】 回答の「予測」の問題

【入力】 プロンプト x_i とその回答 y_i の n 組の集合

【出力】 $P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ の $\boldsymbol{\theta}$ を出力

【設定】 トランスフォーマー（ここの構成は省略. すみません. . .）

$$Y_1 \sim P(y|\mathbf{x}, \boldsymbol{\theta}), \dots, Y_k \sim P(y|\mathbf{x}, y_1, y_2, \dots, y_{k-1}, \boldsymbol{\theta}), \dots$$

評価基準が決まると、意思決定写像が決まる

テキスト生成AIの意思決定写像の例 1

意思決定写像 1 の例

【目的】 回答の「予測」の問題

【入力】 プロンプト x_i とその回答 y_i の n 組の集合

【出力】 $P(y|x, \theta)$ の θ を出力

【設定】 トランスフォーマー（この構成は省略. すみません. . .）

$$Y_1 \sim P(y|x, \theta), \dots, Y_k \sim P(y|x, y_1, y_2, \dots, y_{k-1}, \theta), \dots$$

【評価基準】 $P(y|x, \theta) = P(y_1|x, \theta)P(y_2|x, y_1, \theta) \cdots P(y_k|x, y_1, y_2, \dots, y_{k-1}, \theta) \cdots$
が成り立つので、以下を θ について最大化（尤度最大化）

$$G = \prod_{i=1}^n P(y_i|x_i, \theta)$$

2.2 テキスト生成AIの意思決定写像の例 2

意思決定写像 1 の例 2

【目的】，【入力】，【出力】，【設定】は同じ

【評価基準】 $P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = P(y_1|\mathbf{x}, \boldsymbol{\theta})P(y_2|\mathbf{x}, y_1, \boldsymbol{\theta}) \cdots P(y_k|\mathbf{x}, y_1, y_2, \dots, y_{k-1}, \boldsymbol{\theta}) \cdots$

報酬関数 $r(\mathbf{y}, \hat{\mathbf{y}})$: \mathbf{y} と $\hat{\mathbf{y}}$ の意味が近いほど点数が高くなる関数
(簡単な例として, 同じ意味なら1, そうでなければ0)

$$\frac{1}{n} \sum_{i=1}^n r(\mathbf{y}_i, \hat{\mathbf{y}}) P(\hat{\mathbf{y}}|\mathbf{x}_i, \boldsymbol{\theta})$$

 強化学習の活用

[従来研究例] Daya Guo, et. al. "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning", <https://arxiv.org/abs/2501.12948>, Jan. 2025

2.2 テキスト生成AIの意思決定写像の例 2

意思決定写像 1 の例 3

【目的】、【入力】、【出力】、【設定】は同じ

【評価基準】 $P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = P(y_1|\mathbf{x}, \boldsymbol{\theta})P(y_2|\mathbf{x}, y_1, \boldsymbol{\theta}) \cdots P(y_k|\mathbf{x}, y_1, y_2, \dots, y_{k-1}, \boldsymbol{\theta}) \cdots$

報酬関数 $r(\mathbf{y}, \hat{\mathbf{y}})$: \mathbf{y} と $\hat{\mathbf{y}}$ の意味が近いほど点数が高くなる関数
(簡単な例として, 同じ意味なら1, そうでなければ0)

$$\frac{1}{n} \sum_{i=1}^n \sum_{\hat{\mathbf{y}}} r(\mathbf{y}_i, \hat{\mathbf{y}}) P(\hat{\mathbf{y}}|\mathbf{x}_i, \boldsymbol{\theta})$$

➡ EMアルゴリズムの活用

[従来研究例] Avi Singh, John D Co-Reyes, Rishabh Agarwal, et.al. (Google) "Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models", <https://arxiv.org/abs/2312.06585>, Apr. 2024.

2.2 テキスト生成AIの意思決定写像の例 2

意思決定写像 1 の例 4

【目的】，【入力】，【出力】，【設定】は同じ

【評価基準】 $P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = P(y_1|\mathbf{x}, \boldsymbol{\theta})P(y_2|\mathbf{x}, y_1, \boldsymbol{\theta}) \cdots P(y_k|\mathbf{x}, y_1, y_2, \dots, y_{k-1}, \boldsymbol{\theta}) \cdots$

報酬関数 $r(\mathbf{y}, \hat{\mathbf{y}}, Z|\mathbf{x})$: Z は推論途中の出力列, \mathbf{y} と $\hat{\mathbf{y}}$ の意味だけでなく, 途中の推論の正しさも報酬に加えた関数

$$\frac{1}{n} \sum_{i=1}^n r(\mathbf{y}_i, \hat{\mathbf{y}}, \mathbf{z}_1, \dots, \mathbf{z}_s | \mathbf{x}_i) P(\hat{\mathbf{y}}, \mathbf{z}_1, \dots, \mathbf{z}_s | \mathbf{x}_i, \boldsymbol{\theta})$$

➡ 強化学習の活用, 推論結果の正しさを検証するAIの (報酬計算への) 利用

[従来研究例] Zhihong Shao, et. al. "DeepSeekMath-V2: Towards Self-Verifiable Mathematical Reasoning", <https://arxiv.org/html/2511.22570v1>, Nov. 2025

2.3 テキスト生成AIの意思決定写像の応用例（テキストデータ圧縮）

この意思決定写像の考えを応用したデータ圧縮

【前提】 オープンウェイト θ のテキスト生成AIを送信側と受信側で共有
(先ほどの意思決定写像 1 で θ は決まっている)

実験ではQwen3-4Bを利用 (かなり小さいモデル)

【プロンプト】 x はなし

【入力】 圧縮したいテキストをトークンの列 $\mathbf{y} = (y_1, y_2, \dots, y_m)$ に変換
実験では「SITA2025の小林の原稿TeXファイル」を利用
(絶対に学習には利用されていないことを保証したかっただけ)

【符号化】 $P(\mathbf{y}|\theta) = P(y_1|\theta)P(y_2|y_1, \theta) \cdots P(y_k|y_1, y_2, \dots, y_{k-1}, \theta) \cdots$
として $-\log_2 P(\mathbf{y}|\theta)$ ビットで符号化

[従来研究例] David Heurtel-Depeiges, Anian Ruoss, Joel Veness, Tim Genewein, "Compression via Pre-trained Transformers: A Study on Byte-Level Multimodal Data", ICML2025

2.3 テキスト生成AIの意思決定写像の応用例 (テキストデータ圧縮)

この意思決定写像の考えを応用したデータ圧縮

【前提】 Qwen3-4Bを利用 (かなり小さいモデル)

【入力】 「SITA2025の小林の原稿TeXファイル」を利用

【符号化】 $-\log_2 P(y|\theta)$ ビットで符号化 (理想符号長を計算)

実験結果

元のテキストサイズ: 34314 bytes (274512 bits)
トークン数: 11433
理想符号長: 20429.8643 bits (2553.7330 bytes)
トークンあたりビット数: 1.7869 bits/token
圧縮率 (元サイズ/理想符号長): 13.4368x
bzip2圧縮サイズ: 10009 bytes
bzip2圧縮率 (元サイズ/圧縮後): 3.4283x
モデル実行時間: 543.7292 sec
スループット: 21.03 tokens/sec

2.3 テキスト生成AIの意思決定写像の応用例（テキストデータ圧縮）

この意思決定写像の考えを応用したデータ圧縮

【前提】 Qwen3.5-4Bを利用（かなり小さいモデル）

【入力】 「SITA2025の小林の原稿TeXファイル」を利用

【符号化】 $-\log_2 P(y|\theta)$ ビットで符号化（理想符号長を計算）

実験結果

元のテキストサイズ: 34314 bytes (274512 bits)

トークン数: 10878

理想符号長: 17077.2732 bits (2134.6592 bytes)

トークンあたりビット数: 1.5699 bits/token

圧縮率（元サイズ/理想符号長）: 16.0747x

bzip2圧縮サイズ: 10009 bytes

bzip2圧縮率（元サイズ/圧縮後）: 3.4283x

モデル実行時間: 635.3481 sec

スループット: 17.12 tokens/sec

2.4 テキスト生成AIの意思決定写像の応用例（混合HMM情報源の予測）

混合HMM情報源に対してトランスフォーマーで学習

意思決定写像

【入力】 パラメータの異なる10個のHMMをランダムに選択し、40記号を x
41記号目を y とするペアを n 個用意

【設定】 トランスフォーマー（層正規化, ResNet, マルチヘッドアテンション, RoPE,
FNNの活性化関数はGELU）

【出力】 トランスフォーマーの θ

【評価基準】 尤度の最大化（クロスエントロピー）

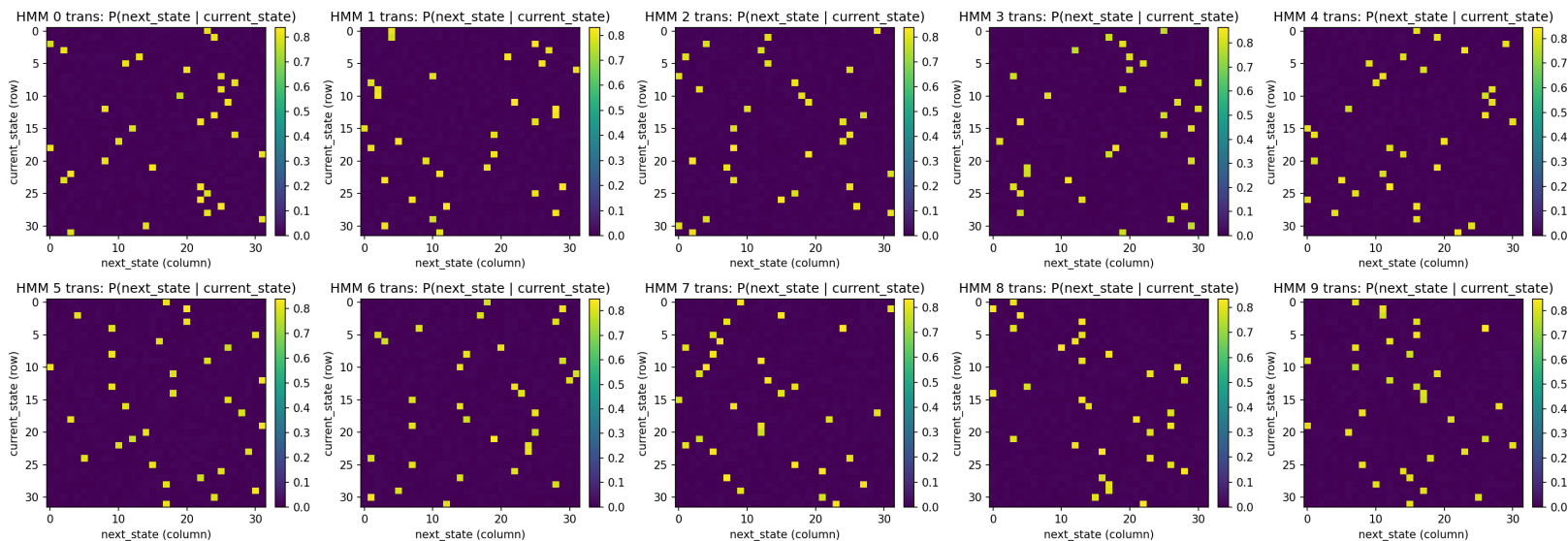
注：本当の最適は無理なのでAdamWをデフォルト設定で利用

2.4 テキスト生成AIの意思決定写像の応用例 (混合HMM情報源の予測)

混合HMM情報源に対してトランスフォーマーで学習

【入力】 パラメータの異なる10個のHMMをランダムに選択し, 40記号を x
41記号目を y とするペアを n 個用意

10個のHMMの状態遷移確率のヒートマップ (32状態)

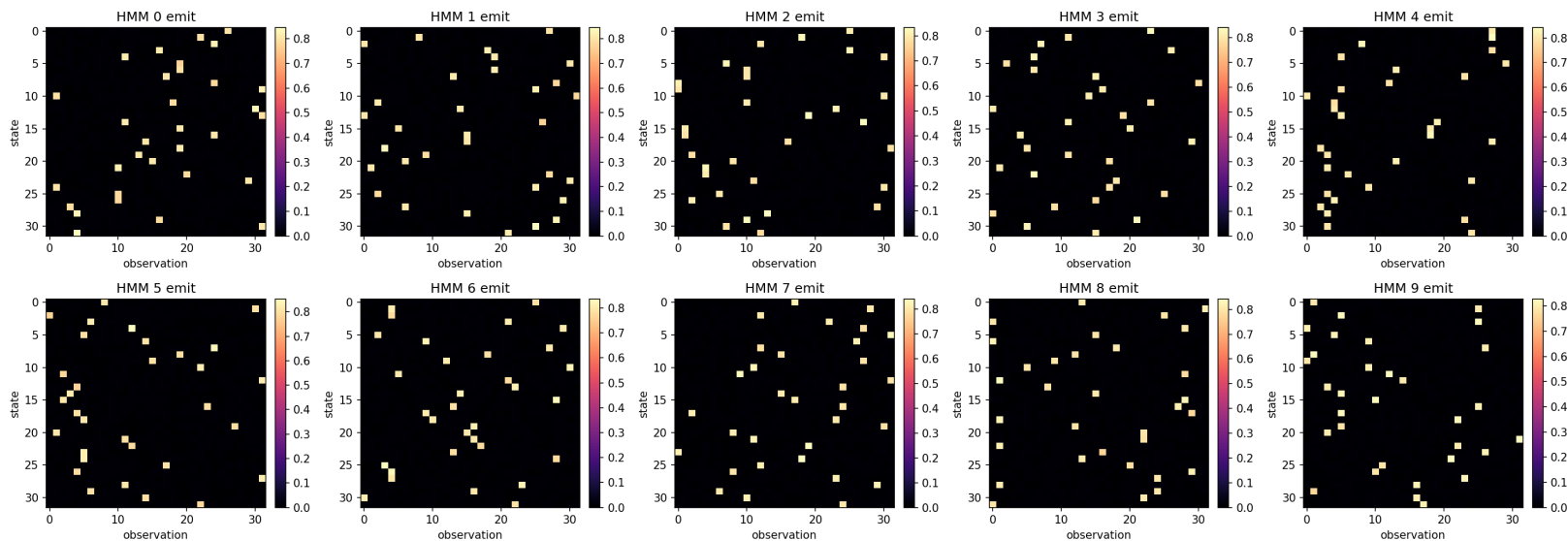


2.4 テキスト生成AIの意思決定写像の応用例（混合HMM情報源の予測）

混合HMM情報源に対してトランスフォーマーで学習

【入力】 パラメータの異なる10個のHMMをランダムに選択し、40記号を x
41記号目を y とするペアを n 個用意

10個のHMMの記号出力確率のヒートマップ（32記号）



2.4 テキスト生成AIの意思決定写像の応用例（混合HMM情報源の予測）

混合HMM情報源に対してトランスフォーマーで学習

【実験の評価方法】 学習データと同様にパラメータの異なる10個のHMMをランダムに選択し, 40記号を x 41記号目を y とするペアを m 個用意して以下を計算

(1) 情報源とパラメータを既知とした時の理想符号長の平均(M_j はHMM) :

$$L^* = -\frac{1}{m} \sum_{i=1}^m \log_2 \sum_{j=1}^{10} P^*(y_i, M_j | x_i)$$

(2) 学習したトランスフォーマーによる予測確率を用いた理想符号長の平均 :

$$L = -\frac{1}{m} \sum_{i=1}^m \log_2 P(y_i | x_i, \theta)$$

2.4 テキスト生成AIの意思決定写像の応用例（混合HMM情報源の予測）

(1) 情報源とパラメータを既知とした時の理想符号長の平均(M_j はHMM) :

$$L^* = -\frac{1}{m} \sum_{i=1}^m \log_2 \sum_{j=1}^{10} P^*(y_i, M_j | x_i)$$

(2) 学習したトランスフォーマーによる予測確率を用いた理想符号長の平均 :

$$L = -\frac{1}{m} \sum_{i=1}^m \log_2 P(y_i | x_i, \theta)$$

【評価】 HMM による $P(y|x)$ とモデルによる $P(y|x)$ の比較

データ平均のKL距離 : 0.272620

HMM理想ビット数の平均 : 3.083205

model使用時ビット数の平均 : 3.476512

Total runtime: 18483.102 seconds

3. 画像生成AI

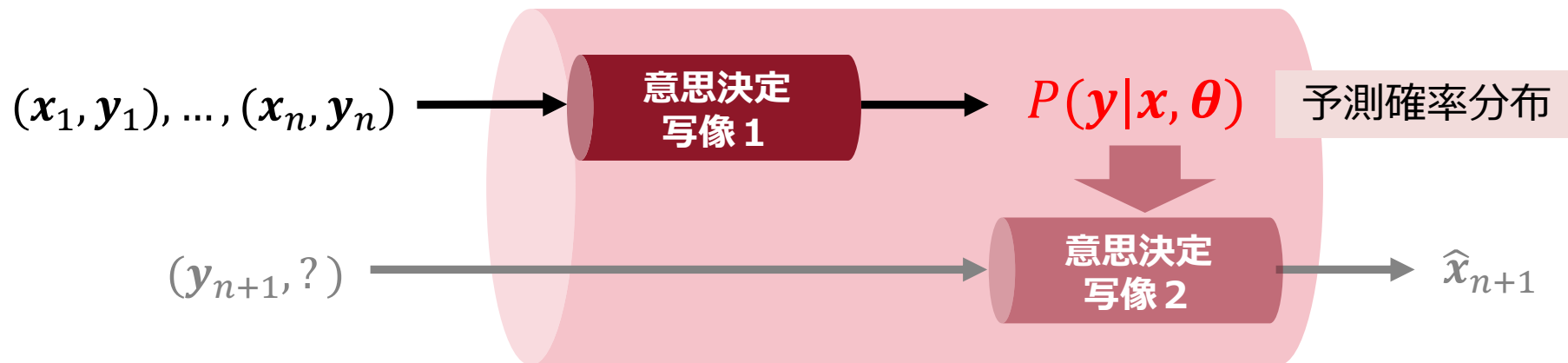
意思決定写像 1

【入力】 プロンプト y_i とその画像 x_i の n 組の集合

【出力】 $P(x|y, \theta)$ の θ を出力

【目的】 出力とする画像の「予測」の問題

簡単のため予測確率分布のパラメータを出力する「意思決定写像 1」に着目



3.1 画像生成AIの意思決定写像の例 1

意思決定写像 1 の例

【目的】 画像の「予測」の問題

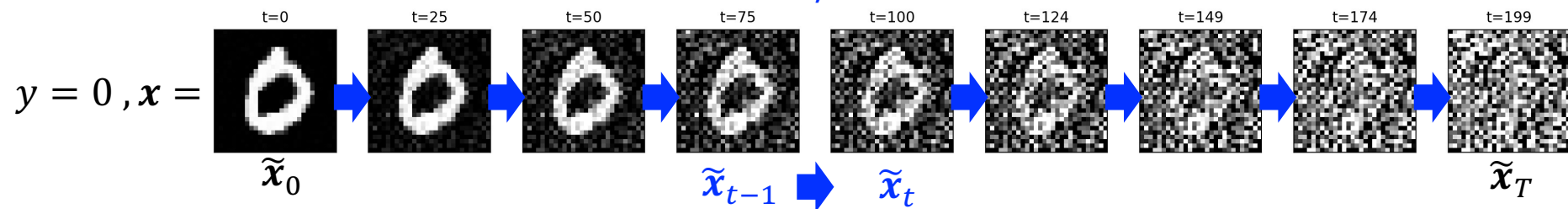
【入力】 プロンプト y_i とその画像 x_i の n 組の集合

【出力】 $P(x|y, \theta)$ の θ を出力

【設定】 拡散モデル

データ例としてMNIST画像を x とし, プロンプト y は対応する画像のラベルとする

拡散過程 $q(\tilde{x}_t|\tilde{x}_{t-1})$: エンコーダに対応 (元の画像と区別するためにチルダをつける)
元の画像の各成分を少しだけ0に近づけて, 正規ノイズを少し加えることを繰り返す



[出典] Calvin Luo, "Understanding Diffusion Models: A Unified Perspective", <https://arxiv.org/abs/2208.11970>, Aug. 2022

3.1 画像生成AIの意思決定写像の例 1

意思決定写像 1 の例

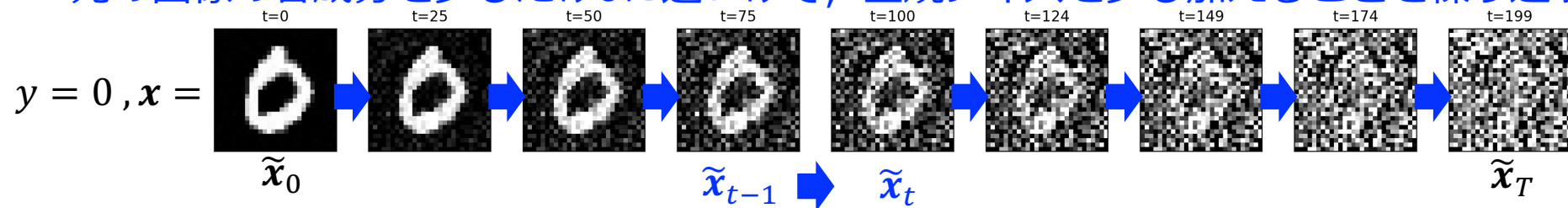
【目的】 画像の「予測」の問題

【入力】 プロンプト y_i とその画像 x_i の n 組の集合

【出力】 $P(x|y, \theta)$ の θ を出力

【設定】 拡散モデル

拡散過程 $q(\tilde{x}_t|\tilde{x}_{t-1})$: エンコーダに対応 (元の画像と区別するためにチルダをつける)
元の画像の各成分を少しだけ0に近づけて, 正規ノイズを少し加えることを繰り返す



$$\tilde{x}_t = \sqrt{\alpha_t} \tilde{x}_{t-1} + \sqrt{1 - \alpha_t} \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I)$$

α_t は定数 (あらかじめ決めておく)

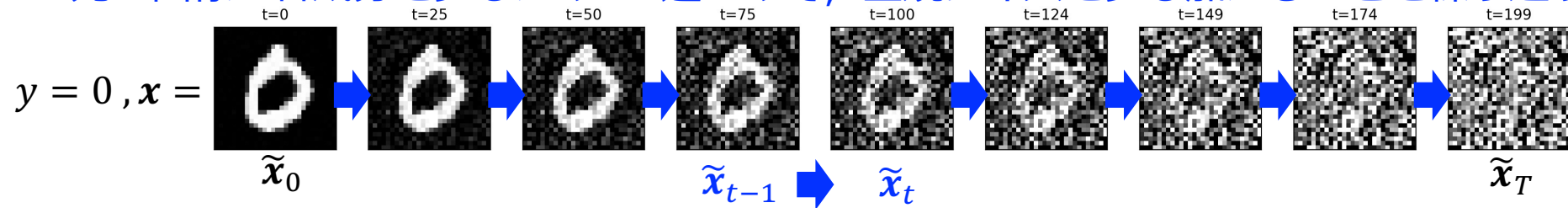
[出典] Calvin Luo, "Understanding Diffusion Models: A Unified Perspective", <https://arxiv.org/abs/2208.11970>, Aug. 2022

3.1 画像生成AIの意思決定写像の例 1

意思決定写像 1 の例

- 【目的】 画像の「予測」の問題
- 【入力】 プロンプト y_i とその画像 x_i の n 組の集合
- 【出力】 $P(x|y, \theta)$ の θ を出力
- 【設定】 拡散モデル

拡散過程 $q(\tilde{x}_t|\tilde{x}_{t-1})$: エンコーダに対応 **(注: 未知のパラメータがない!)**
元の画像の各成分を少しだけ0に近づけて, 正規ノイズを少し加えることを繰り返す



$$\tilde{x}_t = \sqrt{\alpha_t} \tilde{x}_{t-1} + \sqrt{1 - \alpha_t} \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I)$$

α_t は**定数** (あらかじめ決めておく)

[出典] Calvin Luo, "Understanding Diffusion Models: A Unified Perspective", <https://arxiv.org/abs/2208.11970>, Aug. 2022

3.1 画像生成AIの意思決定写像の例 1

意思決定写像 1 の例

【目的】 画像の「予測」の問題

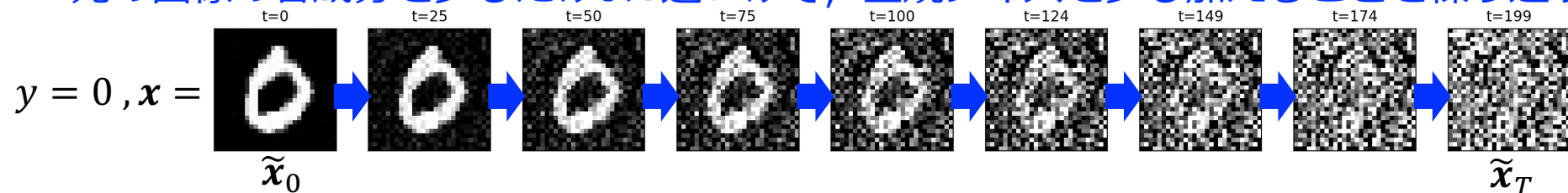
【入力】 プロンプト y_i とその画像 x_i の n 組の集合

【出力】 $P(x|y, \theta)$ の θ を出力

【設定】 拡散モデル

拡散過程 $q(\tilde{x}_t|\tilde{x}_{t-1})$: エンコーダに対応 **(注: 未知のパラメータがない!)**

元の画像の各成分を少しだけ0に近づけて, 正規ノイズを少し加えることを繰り返す



\tilde{x}_T は (ほぼ) 独立な標準正規分布

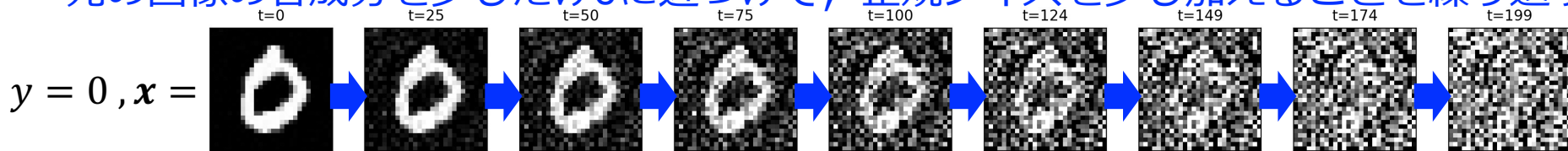
【出典】 Calvin Luo, "Understanding Diffusion Models: A Unified Perspective", <https://arxiv.org/abs/2208.11970>, Aug. 2022

3.1 画像生成AIの意思決定写像の例 1

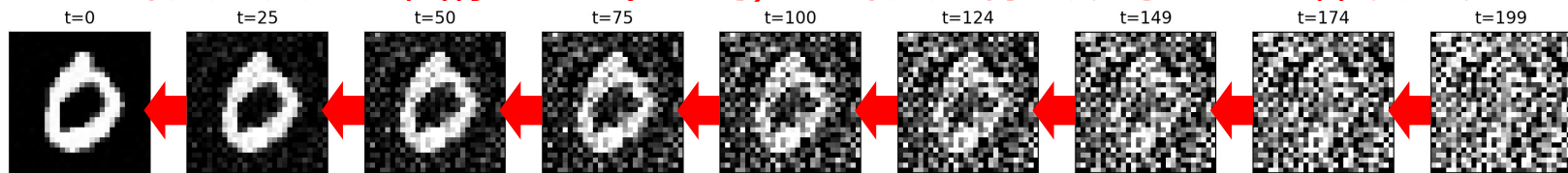
意思決定写像 1 の例

- 【目的】 画像の「予測」の問題
- 【入力】 プロンプト y_i とその画像 x_i の n 組の集合
- 【出力】 $P(x|y, \theta)$ の θ を出力
- 【設定】 拡散モデル

拡散過程 $q(\tilde{x}_t|\tilde{x}_{t-1})$: エンコーダに対応 (注: 未知のパラメータがない!)
元の画像の各成分を少しだけ0に近づけて, 正規ノイズを少し加えることを繰り返す



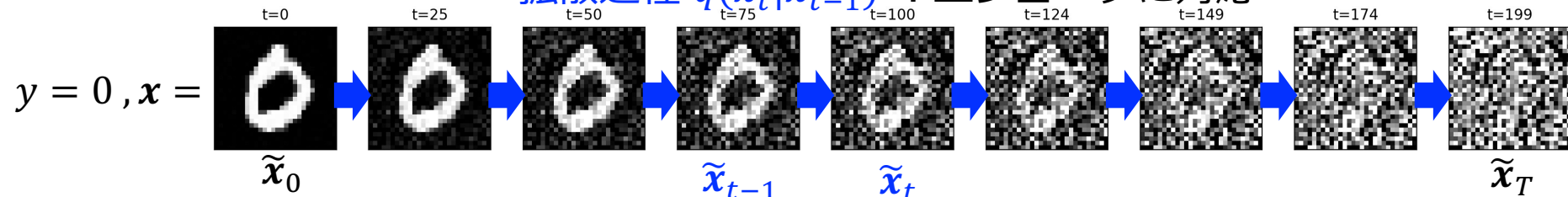
\tilde{x}_0 逆拡散過程 $P(\tilde{x}_{t-1}|\tilde{x}_t, y, \theta)$: デコーダに対応 (注: θ を学習!) \tilde{x}_T
ノイズの入った画像から (NNで) ノイズを除去することを繰り返す



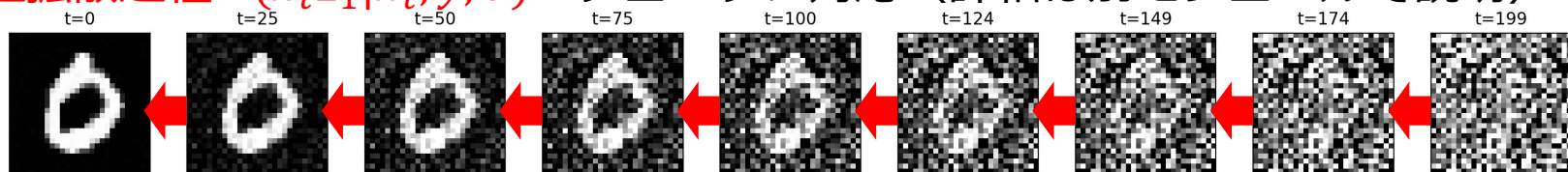
3.1 画像生成AIの意思決定写像の例 1

【設定】 拡散モデル

拡散過程 $q(\tilde{x}_t|\tilde{x}_{t-1})$: エンコーダに対応



逆拡散過程 $P(\tilde{x}_{t-1}|\tilde{x}_t, y, \theta)$: デコーダに対応 (詳細は別モジュールで説明)



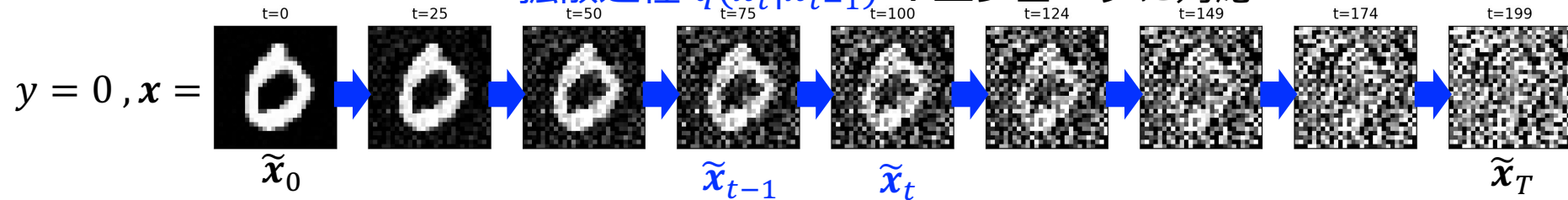
【評価基準】 評価基準 1 (厳密に式を理解する必要はない。イメージの理解で良い)
以下の対数尤度の θ に関する最大化

$$\sum \log P(x_i|y_i, \theta)$$

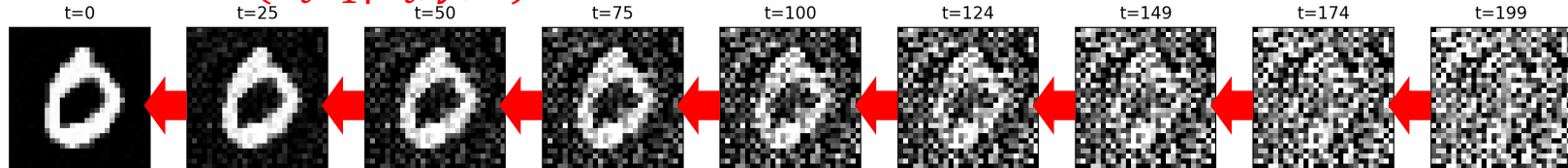
3.1 画像生成AIの意思決定写像の例 1

【設定】 拡散モデル

拡散過程 $q(\tilde{x}_t|\tilde{x}_{t-1})$: エンコーダに対応



逆拡散過程 $P(\tilde{x}_{t-1}|\tilde{x}_t, y, \theta)$: デコーダに対応 (詳細は別モジュールで説明)



【評価基準】 評価基準 1 (厳密に式を理解する必要はない。イメージの理解で良い)

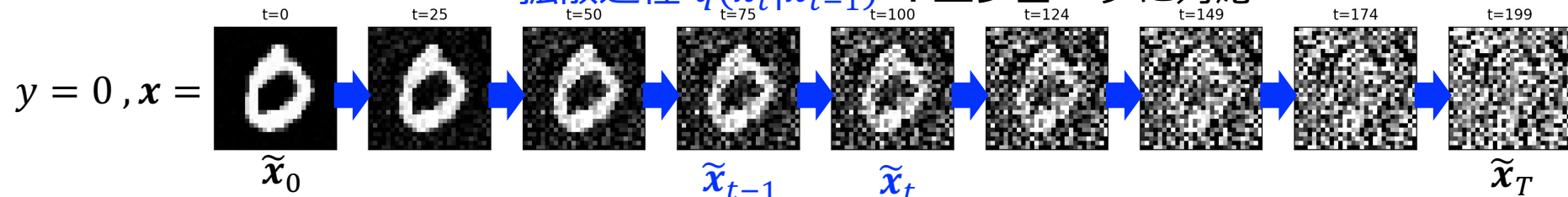
$$\log P(x|y, \theta) = \log \int P(\tilde{x}_0, \tilde{x}_{1:T}|y, \theta) d\tilde{x}_{1:T} = \log \mathbb{E}_{q(\tilde{x}_{1:T}|\tilde{x}_0)} \left[\frac{P(\tilde{x}_0, \tilde{x}_{1:T}|y, \theta)}{q(\tilde{x}_{1:T}|\tilde{x}_0)} \right]$$

【出典】 Calvin Luo, "Understanding Diffusion Models: A Unified Perspective", <https://arxiv.org/abs/2208.11970>, Aug. 2022

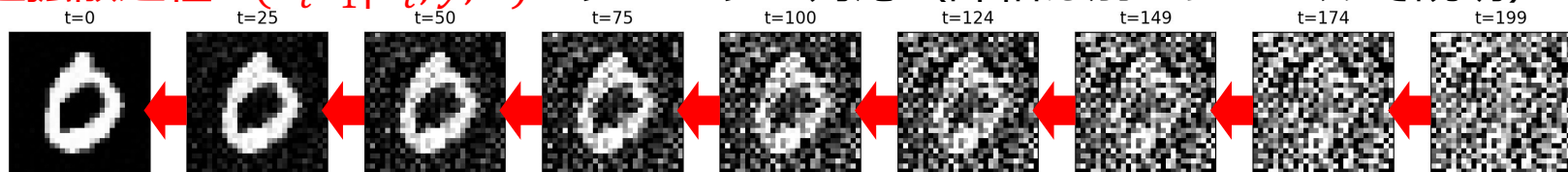
3.1 画像生成AIの意思決定写像の例 1

【設定】 拡散モデル

拡散過程 $q(\tilde{x}_t|\tilde{x}_{t-1})$: エンコーダに対応



逆拡散過程 $P(\tilde{x}_{t-1}|\tilde{x}_t, y, \theta)$: デコーダに対応 (詳細は別モジュールで説明)



【評価基準】 評価基準 1 (厳密に式を理解する必要はない。イメージの理解で良い)

$$\log P(x|y, \theta) = \log \int P(\tilde{x}_0, \tilde{x}_{1:T}|y, \theta) d\tilde{x}_{1:T} = \log \mathbb{E}_{q(\tilde{x}_{1:T}|\tilde{x}_0)} \left[\frac{P(\tilde{x}_0, \tilde{x}_{1:T}|y, \theta)}{q(\tilde{x}_{1:T}|\tilde{x}_0)} \right]$$

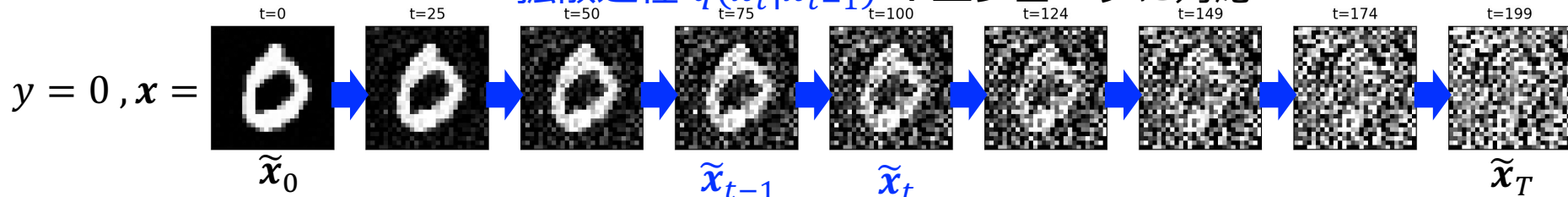


【出典】 Calvin Luo, "Understanding Diffusion Models: A Unified Perspective", <https://arxiv.org/abs/2208.11970>, Aug. 2022

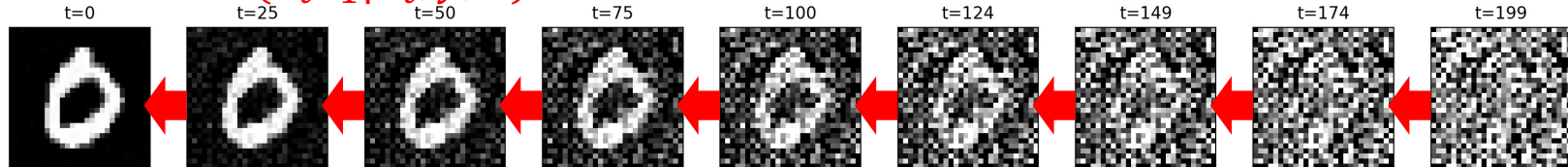
3.1 画像生成AIの意思決定写像の例 1

【設定】 拡散モデル

拡散過程 $q(\tilde{x}_t|\tilde{x}_{t-1})$: エンコーダに対応



逆拡散過程 $P(\tilde{x}_{t-1}|\tilde{x}_t, y, \theta)$: デコーダに対応 (詳細は別モジュールで説明)



【評価基準】 評価基準 2 (厳密に式を理解する必要はない。イメージの理解が良い)

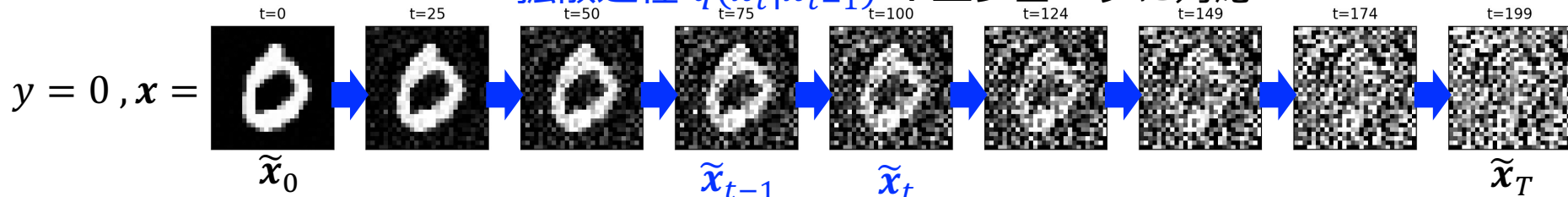
$$\log P(x|y, \theta) = \log \int P(\tilde{x}_0, \tilde{x}_{1:T}|y, \theta) d\tilde{x}_{1:T} = \log \mathbb{E}_{q(\tilde{x}_{1:T}|\tilde{x}_0)} \left[\frac{P(\tilde{x}_0, \tilde{x}_{1:T}|y, \theta)}{q(\tilde{x}_{1:T}|\tilde{x}_0)} \right]$$
$$\geq \mathbb{E}_{q(\tilde{x}_{1:T}|x)} \left[\log \frac{P(\tilde{x}_0, \tilde{x}_{1:T}|y, \theta)}{q(\tilde{x}_{1:T}|\tilde{x}_0)} \right] \quad \leftarrow \text{ELBO (1データ分. VAEでも出現)}$$

[出典] Calvin Luo, "Understanding Diffusion Models: A Unified Perspective", <https://arxiv.org/abs/2208.11970>, Aug. 2022

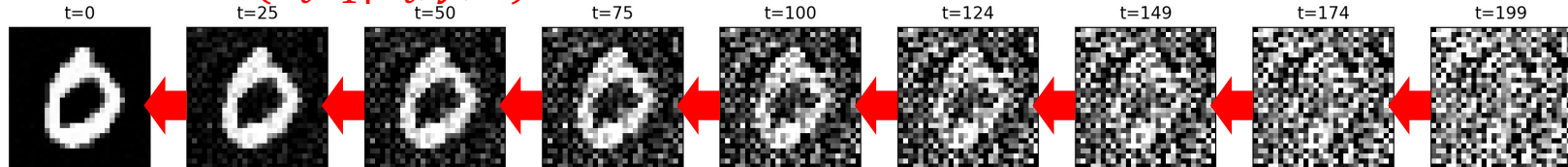
3.1 画像生成AIの意思決定写像の例 1

【設定】 拡散モデル

拡散過程 $q(\tilde{x}_t|\tilde{x}_{t-1})$: エンコーダに対応



逆拡散過程 $P(\tilde{x}_{t-1}|\tilde{x}_t, y, \theta)$: デコーダに対応 (詳細は別モジュールで説明)



【評価基準】 評価基準 2 (厳密に式を理解する必要はない。イメージの理解が良い)

$$\log P(x|y, \theta) = \log \int P(\tilde{x}_0, \tilde{x}_{1:T}|y, \theta) d\tilde{x}_{1:T} = \log \mathbb{E}_{q(\tilde{x}_{1:T}|\tilde{x}_0)} \left[\frac{P(\tilde{x}_0, \tilde{x}_{1:T}|y, \theta)}{q(\tilde{x}_{1:T}|\tilde{x}_0)} \right]$$

$$\geq \mathbb{E}_{q(\tilde{x}_{1:T}|x)} \left[\log \frac{P(\tilde{x}_0, \tilde{x}_{1:T}|y, \theta)}{q(\tilde{x}_{1:T}|\tilde{x}_0)} \right] \leftarrow \text{この式だとよくわからない}$$

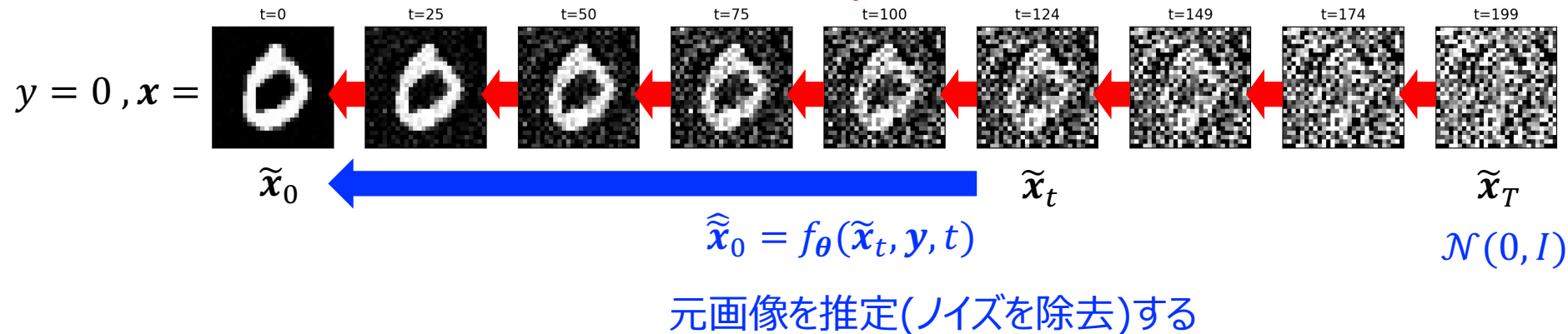
【出典】 Calvin Luo, "Understanding Diffusion Models: A Unified Perspective", <https://arxiv.org/abs/2208.11970>, Aug. 2022

3.1 画像生成AIの意思決定写像の例 1

【設定】 拡散モデル

逆拡散過程：デコーダに対応

ノイズの入った画像から（NNで）ノイズを除去することを繰り返す



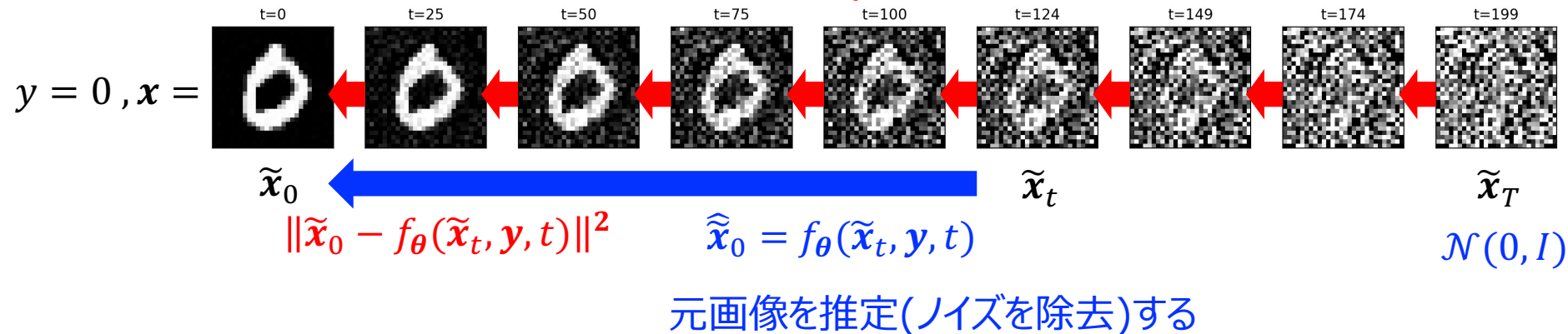
[出典] Calvin Luo, "Understanding Diffusion Models: A Unified Perspective", <https://arxiv.org/abs/2208.11970>, Aug. 2022

3.1 画像生成AIの意思決定写像の例 1

【設定】 拡散モデル

逆拡散過程：デコーダに対応

ノイズの入った画像から（NNで）ノイズを除去することを繰り返す



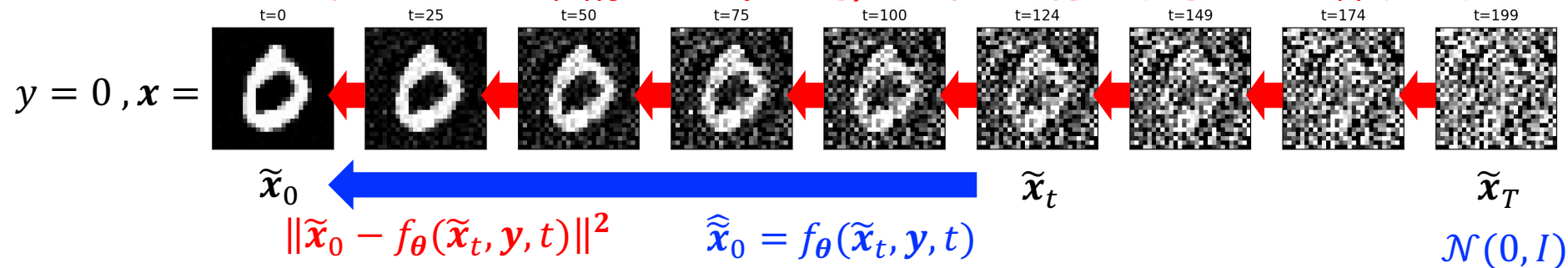
[出典] Calvin Luo, "Understanding Diffusion Models: A Unified Perspective", <https://arxiv.org/abs/2208.11970>, Aug. 2022

3.1 画像生成AIの意思決定写像の例 1

【設定】 拡散モデル

逆拡散過程：デコーダに対応

ノイズの入った画像から（NNで）ノイズを除去することを繰り返す



【評価基準】 評価基準 3

元画像を推定(ノイズを除去)する

$$\mathbb{E}_{x,y,t,q(\tilde{x}_t|x)} [C_t \|\tilde{x}_0 - f_\theta(\tilde{x}_t, y, t)\|^2] \quad C_t \text{ は } \alpha_1, \alpha_2, \alpha_3, \dots \text{ によって決まる定数}$$

意味：元画像との2乗誤差が小さくなるように，元画像を推定(ノイズを除去)する
(ステップの値 t によって2乗誤差の重み C_t が違う)

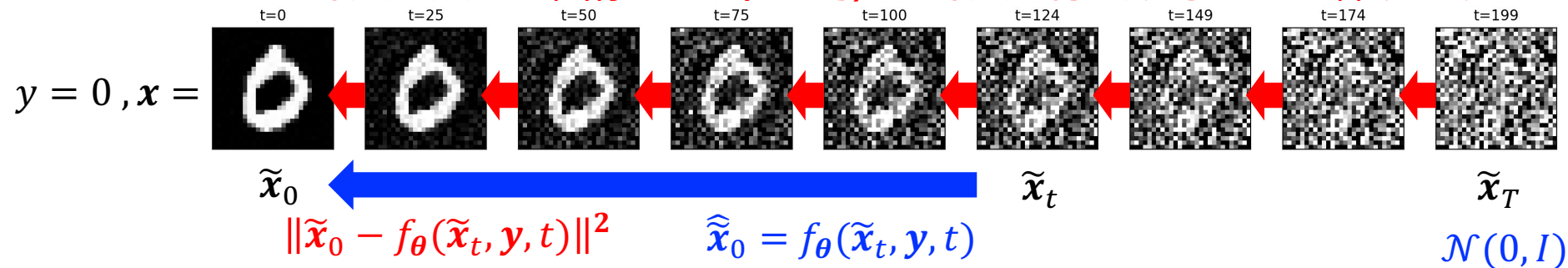
[出典] Calvin Luo, "Understanding Diffusion Models: A Unified Perspective", <https://arxiv.org/abs/2208.11970>, Aug. 2022

3.1 画像生成AIの意思決定写像の例 1

【設定】 拡散モデル

逆拡散過程：デコーダに対応

ノイズの入った画像から（NNで）ノイズを除去することを繰り返す



元画像を推定(ノイズを除去)する

【評価基準】 評価基準 3

$$\mathbb{E}_{x,y,t,q(\tilde{x}_t|x)} [C_t \|\tilde{x}_0 - f_\theta(\tilde{x}_t, y, t)\|^2] = \mathbb{E}_{x,y,q(\tilde{x}_{1:T}|x)} \left[\log \int \frac{P(\tilde{x}_0, \tilde{x}_{1:T} | y, \theta)}{q(\tilde{x}_{1:T} | \tilde{x}_0)} d\tilde{x}_{1:T} \right]$$

評価基準 2 と評価基準 3 は等価！

評価基準 2 : ELBO

評価基準 3 を使えば良い (2乗誤差を小さくする)

[出典] Calvin Luo, "Understanding Diffusion Models: A Unified Perspective", <https://arxiv.org/abs/2208.11970>, Aug. 2022

3.1 画像生成AIの意思決定写像の例 1

意思決定写像 1 の例

【目的】 画像の「予測」の問題

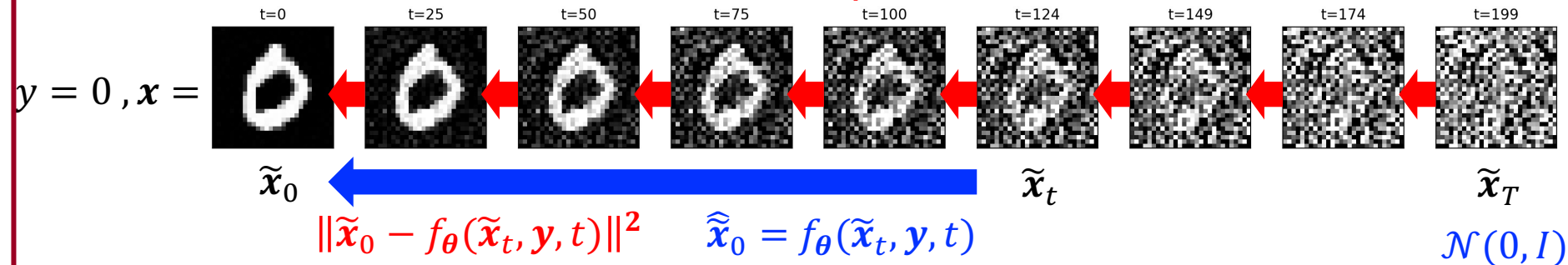
【入力】 プロンプト y_i とその画像 x_i の n 組の集合

【出力】 $f_{\theta}(\cdot)$ の θ を出力

【設定】 拡散モデル

逆拡散過程：デコーダに対応

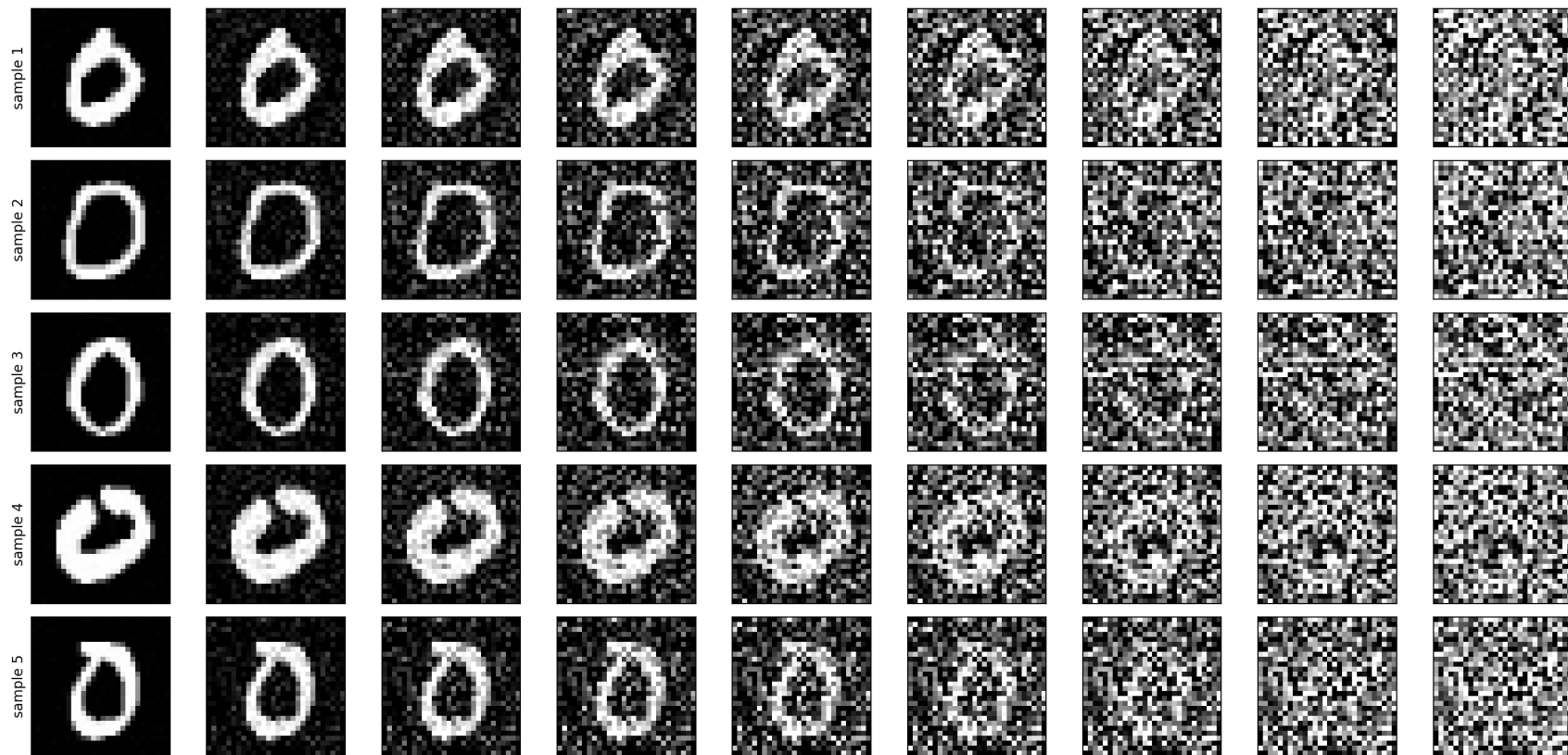
ノイズの入った画像から（NNで）ノイズを除去することを繰り返す



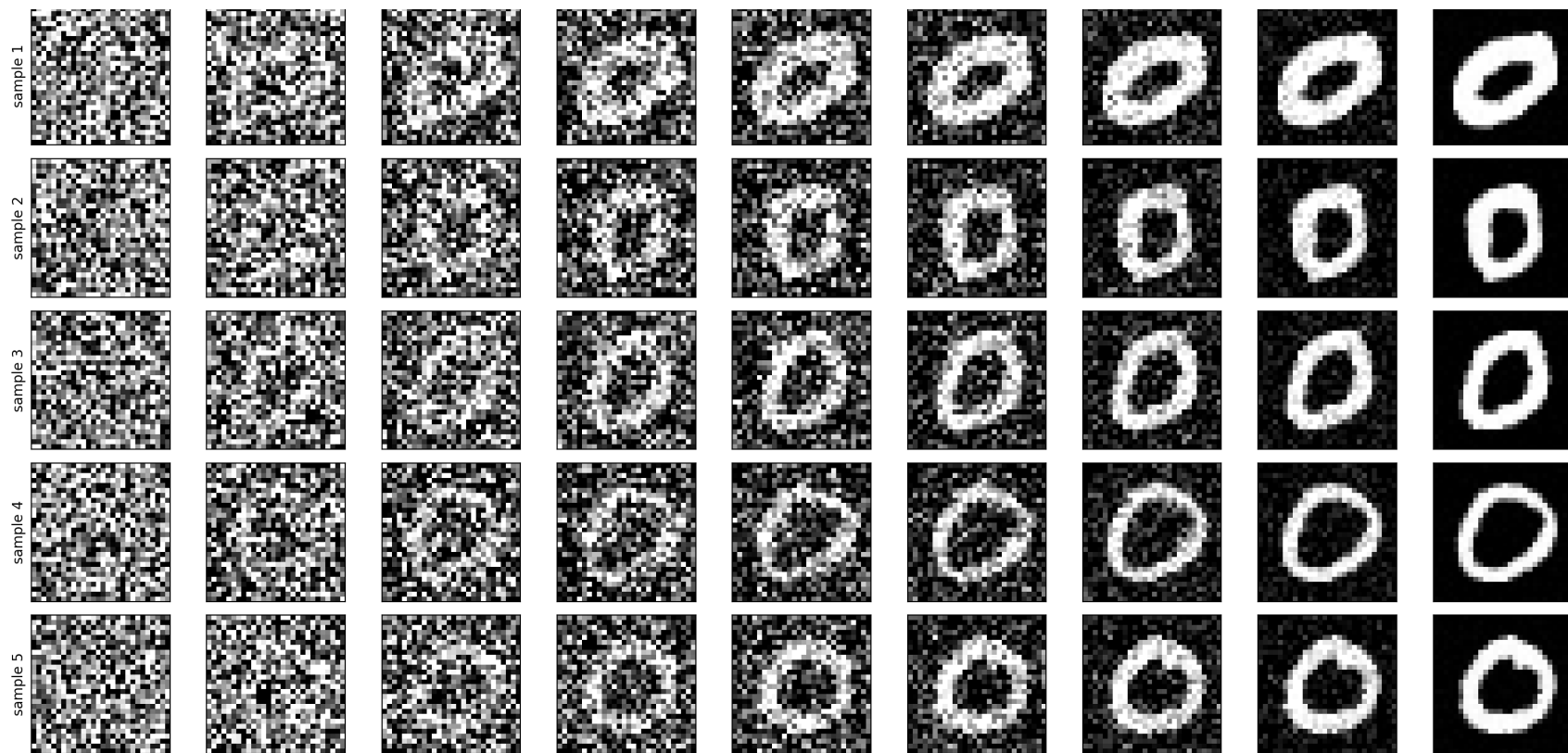
【評価基準】 $\mathbb{E}_{x,y,t,q(\tilde{x}_t|x)} [C_t \|\tilde{x}_0 - f_{\theta}(\tilde{x}_t, y, t)\|^2]$

注：やろうとしていることは対数尤度 $\mathbb{E}_{x,y} [\log P(x|y, \theta)]$ の最大化

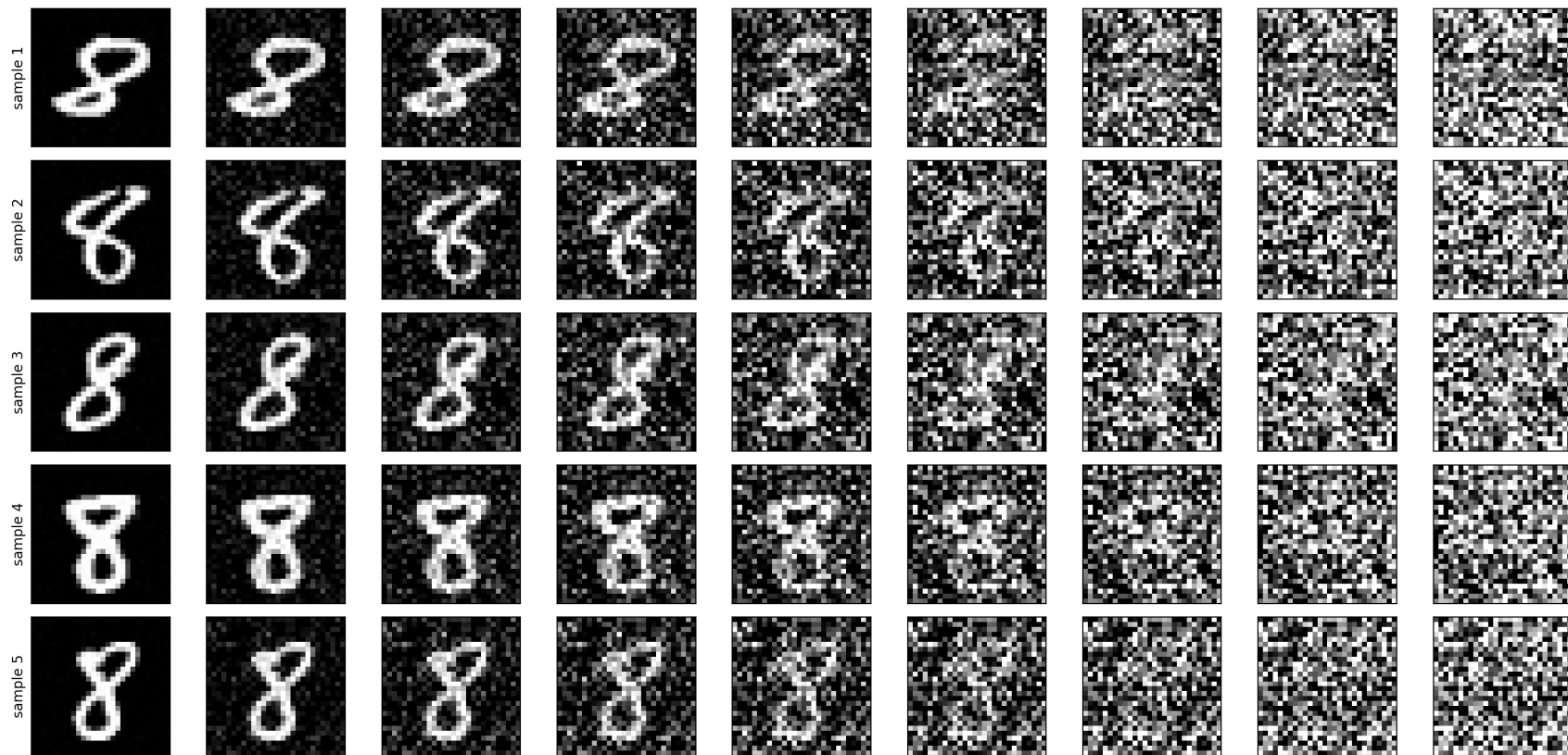
3.2 画像生成AIの意思決定写像の検証例 (MNIST画像の生成)



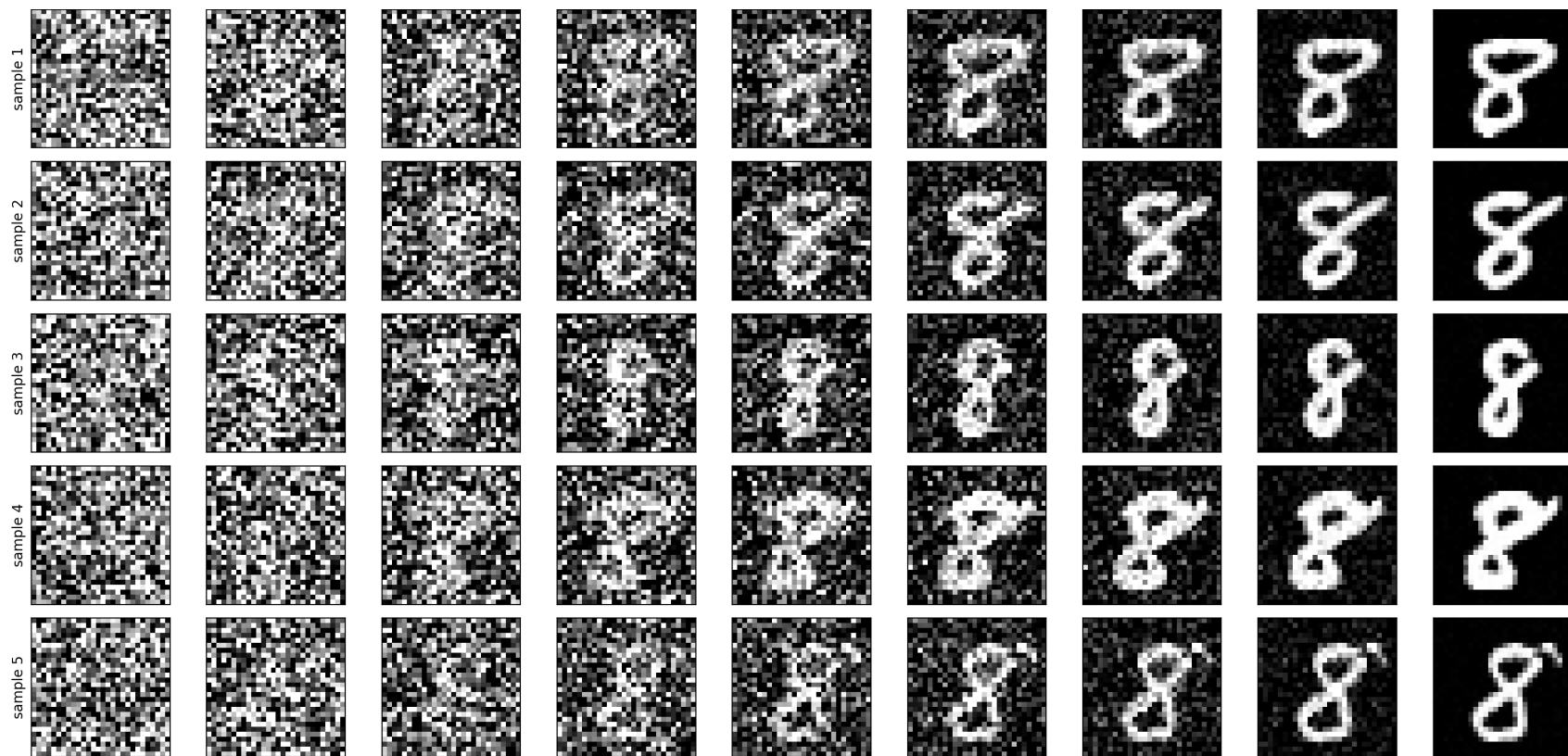
3.2 画像生成AIの意思決定写像の検証例 (MNIST画像の生成)



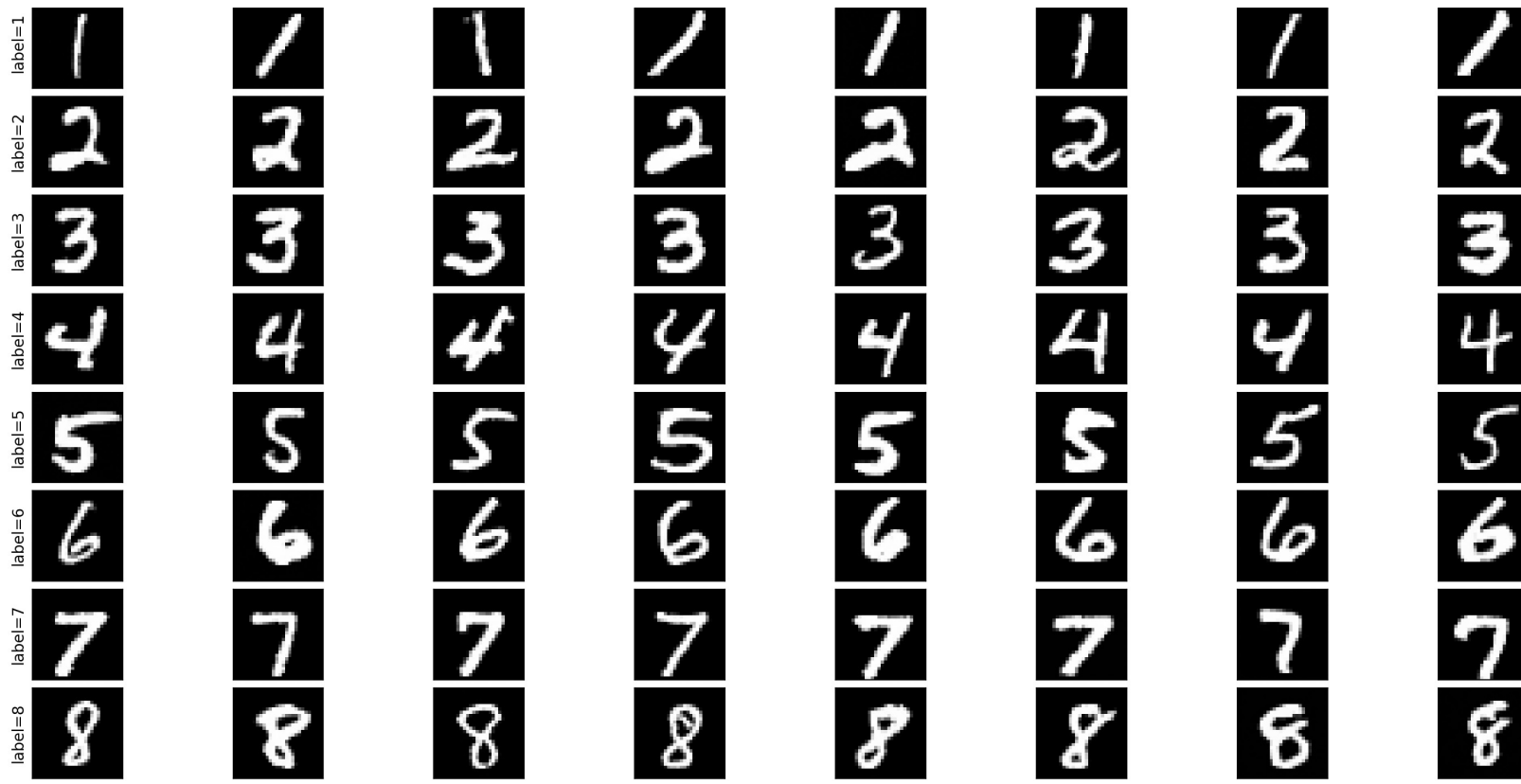
3.2 画像生成AIの意思決定写像の検証例 (MNIST画像の生成)



3.2 画像生成AIの意思決定写像の検証例 (MNIST画像の生成)



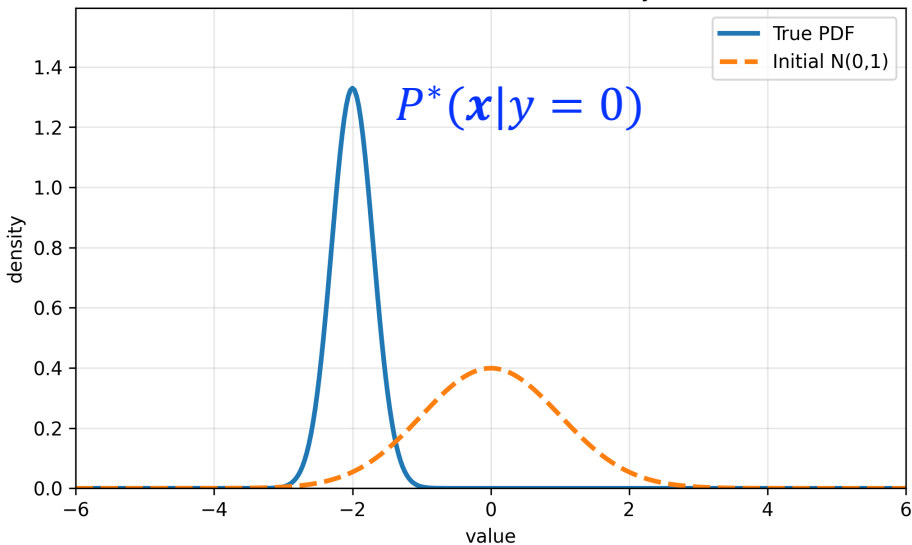
3.2 画像生成AIの意思決定写像の検証例（MNIST画像の生成）



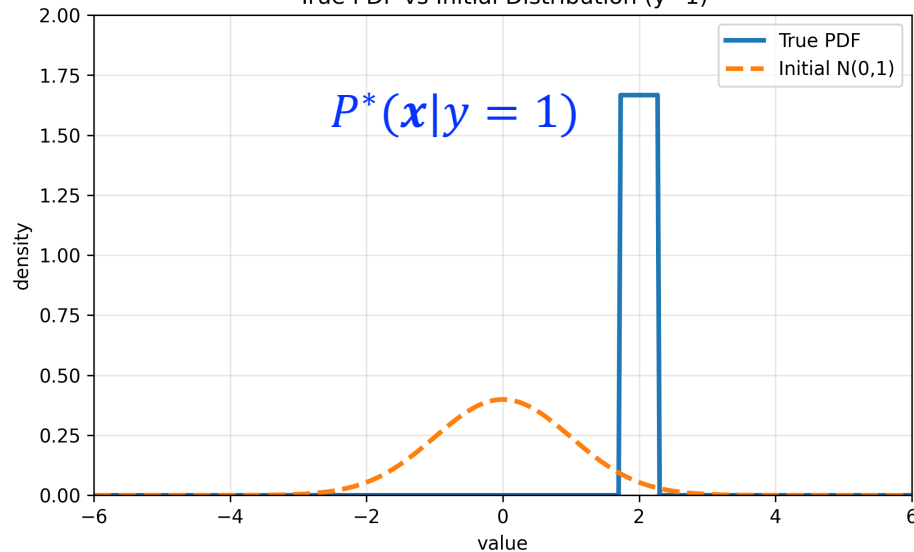
3.3 画像生成AIの意思決定写像の検証例 (条件付き確率分布の再現性)

$P^*(x|y=0)$ と $P^*(x|y=1)$ の再現性の実験 1

True PDF vs Initial Distribution (y=0)



True PDF vs Initial Distribution (y=1)

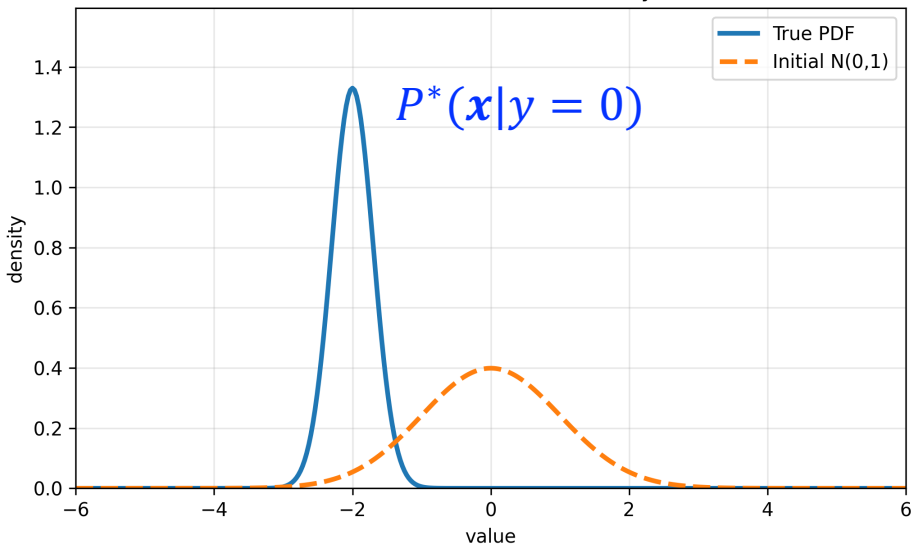


実験結果は直接デモプログラムで確認

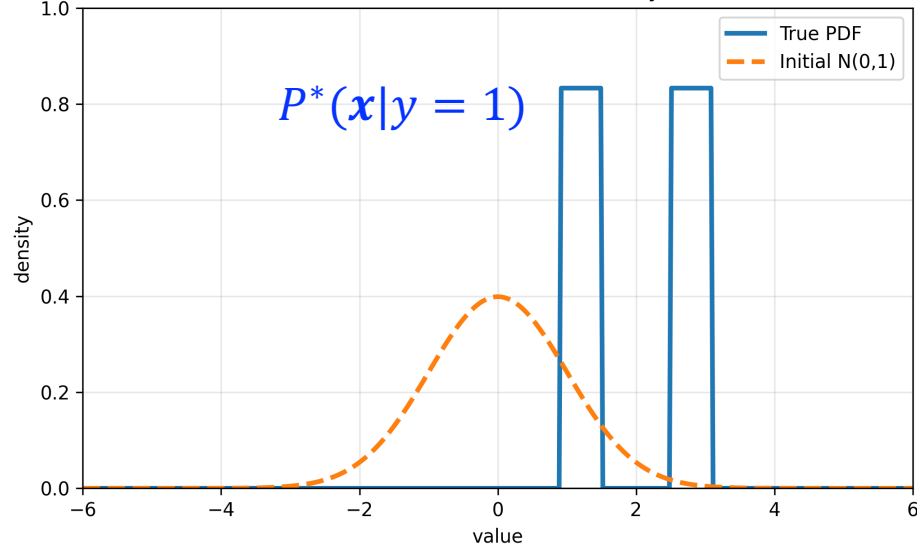
3.3 画像生成AIの意思決定写像の検証例 (条件付き確率分布の再現性)

$P^*(x|y=0)$ と $P^*(x|y=1)$ の再現性の実験 2

True PDF vs Initial Distribution (y=0)



True PDF vs Initial Distribution (y=1)

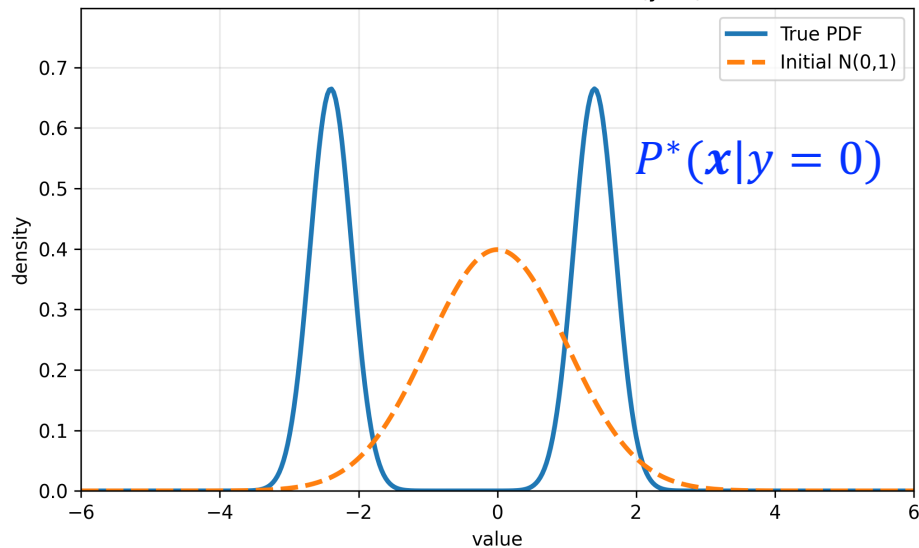


実験結果は直接デモプログラムで確認

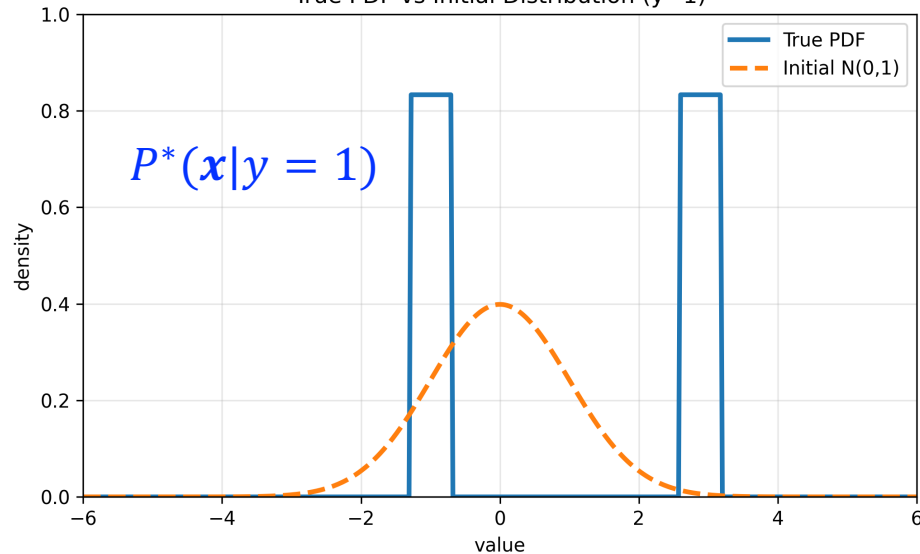
3.3 画像生成AIの意思決定写像の検証例（条件付き確率分布の再現性）

$P^*(x|y=0)$ と $P^*(x|y=1)$ の再現性の実験 3

True PDF vs Initial Distribution (y=0)



True PDF vs Initial Distribution (y=1)



実験結果は直接デモプログラムで確認

3.4 画像生成AIの意思決定写像の応用例（微分方程式の初期値から解の生成）

意思決定写像

【目的】 微分方程式の初期値が与えられて、解を予測

【入力】 初期条件・境界条件 \mathbf{y}_i と微分方程式の解 x_i の n 組の集合

【出力】 $\mathbf{x} = f_{\theta}(\mathbf{y})$ の θ を出力

【設定】 拡散モデル

【評価基準】 $\mathbb{E}_{\mathbf{x}, \mathbf{y}, t} [C_t \|\tilde{\mathbf{x}}_0 - f_{\theta}(\tilde{\mathbf{x}}_t, \mathbf{y}, t)\|^2 + D_t \|\mathcal{R}(f_{\theta}(\tilde{\mathbf{x}}_t, \mathbf{y}, t))\|^2]$

$\mathcal{R}(f_{\theta}(\tilde{\mathbf{x}}_t, \mathbf{y}, t))$: 微分方程式および境界条件からの逸脱度（方程式なので「=0」からの差分）

[出典] Jan-Hendrik Bastek, "PHYSICS-INFORMED DIFFUSION MODELS", ICLR 2025, Mar 2025

3.4 画像生成AIの意思決定写像の応用例（微分方程式の初期値から解の生成）

簡単な例：熱伝導の微分方程式に対する実験

【目的】 微分方程式の初期値が与えられて、解を予測

【入力】 初期条件・境界条件 y_i と微分方程式の解 x_i の n 組の集合

【評価基準】 $\mathbb{E}_{x,y,t,q(\tilde{x}_t|x)} [C_t \|\tilde{x}_0 - f_\theta(\tilde{x}_t, y, t)\|^2 + D_t \|\mathcal{R}(f_\theta(\tilde{x}_t, y, t))\|^2]$

$\mathcal{R}(f_\theta(\tilde{x}_t, y, t))$: 微分方程式からの逸脱度（方程式なので「=0」からの差分）

棒に対する熱伝導

位置 i の時刻 j の温度 $x(i, j)$, 初期温度分布 $x(i, 0) = y(i)$

端の温度 $x(0, j) = T_0$, $x(L, j) = T_L$

微分方程式

$$\frac{\partial x(i, j)}{\partial j} = \kappa \frac{\partial^2 x(i, j)}{\partial i^2}$$

プロンプト y は制約として働く

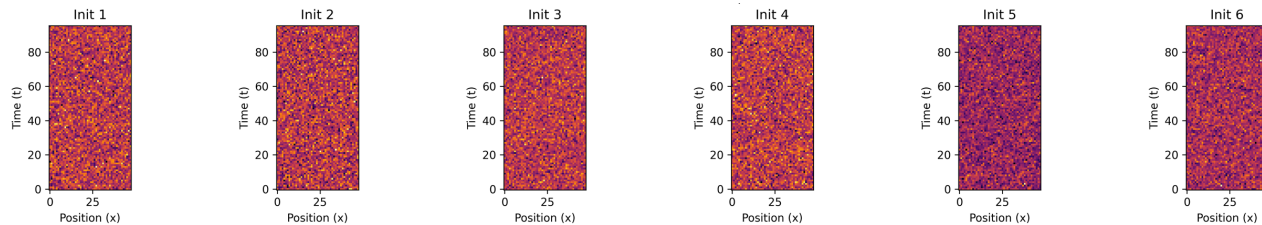
実験では位置や時刻を離散に量子化。微分方程式を「差分方程式」にする

[出典] Jan-Hendrik Bastek, "PHYSICS-INFORMED DIFFUSION MODELS", ICLR 2025, Mar 2025

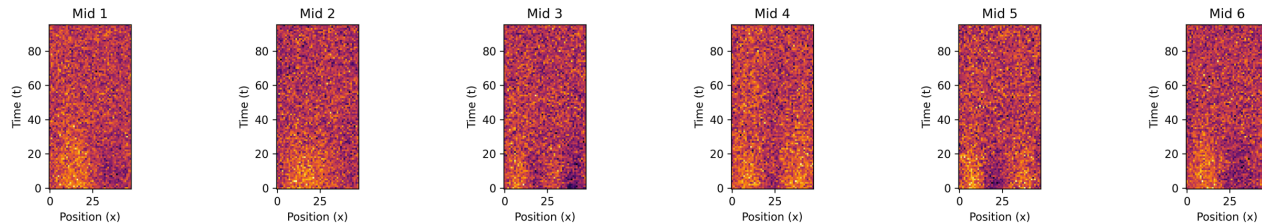
3.4 画像生成AIの意思決定写像の応用例 (微分方程式の初期値から解の生成)

実験結果

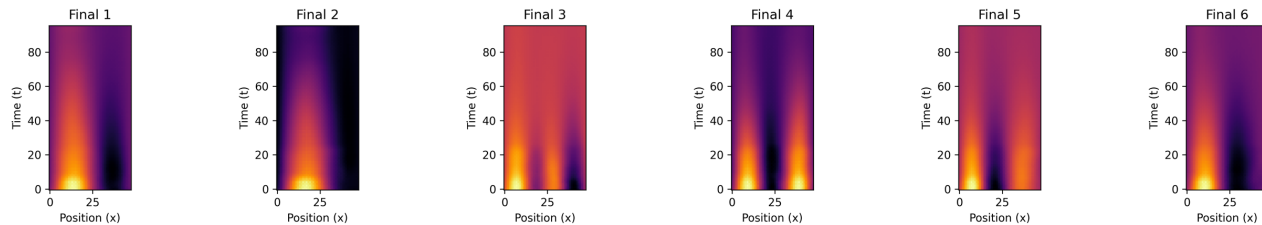
$$t = 199, \widehat{\mathbf{x}}_t$$



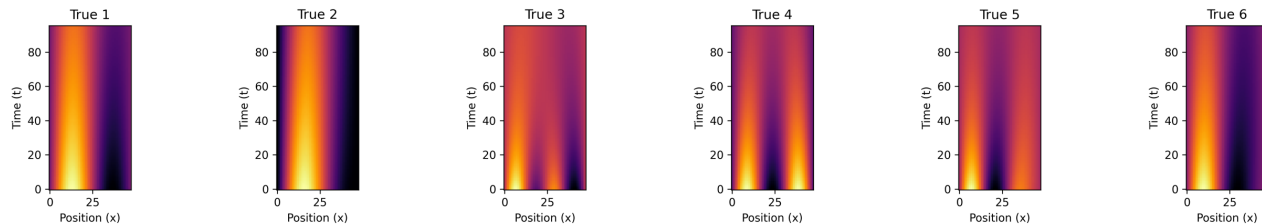
$$t = 99, \widehat{\mathbf{x}}_t$$



$$t = 0, \widehat{\mathbf{x}}_0$$



$$\mathbf{x}^*$$



4. まとめ

- テキスト生成AIと画像生成AIで良く用いられる手法を例として、「意思決定写像」としての表現を試みた
- 生成AIは「データからの生成観測メカニズム（情報源）の模倣」と考えることができる
- プロンプトはデータ生成における「制約」あるいは「条件」と考えることができる

「意思決定写像」はデータサイエンスの種々の手法や研究を見通しよく統一的に考える表現方法の一つ