DEEP LEARNING JP
[DL Papers]

# Alignment Algorithms for Diffusion Models

2024.12.12 Yuta Oshima, M2, Matsuo-Iwasawa Lab

# Alignment Algorithms for Diffusion Models

- **本発表について**
  - まず，前提となる拡散モデルの基礎知識について共有し，その後，拡散モデルのアラインメントのためのアルゴリズムについて展開
- **テーマの選定理由**
  - 拡散モデルは，動画像生成，プランニングなど，多岐にわたる分野に応用可能
  - 最近ではSoraのような高精度なモデルの公開により，実用化への道がますます現実味を帯びる
  - 一方で，実用に耐えるモデルを構築するには，拡散モデルの性能や挙動を適切に調整する「アラインメント」の重要性がこれまで以上に高まっている

# Alignment Algorithms for Diffusion Models

- **参考文献**
  - Alignment of Diffusion Models: Fundamentals, Challenges, and Future
    - Buhua Liu, Shitong Shao, Bao Li, Lichen Bai, Zhiqiang Xu, Haoyi Xiong, James Kwok, Sumi Helal, Zeke Xie
    - https://arxiv.org/abs/2409.07253
  - Understanding Reinforcement Learning-Based Fine-Tuning of Diffusion Models: A Tutorial and Review
    - Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, Sergey Levine
    - https://arxiv.org/abs/2407.13734

# 目次

1. 拡散モデルの準備
2. 拡散モデルのアラインメント

# 目次

1. 拡散モデルの準備
2. 拡散モデルのアラインメント

# 拡散モデルとは

- **拡散モデルは，画像生成での成功を皮切りに，大きく注目を集める生成モデル．**
  - 発展の契機となったのは， Denoising Diffusion Probabilistic Models (DDPM) [Ho+ 2018] と呼ばれるモデルの登場
  - 近年ではSora [OpenAI 2024] などの動画生成分野でも広く注目を集める

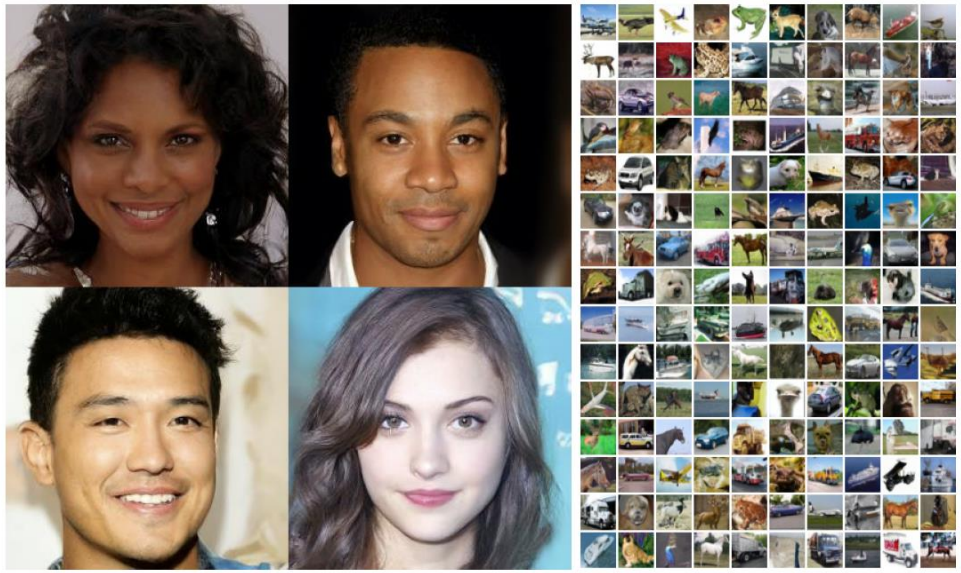
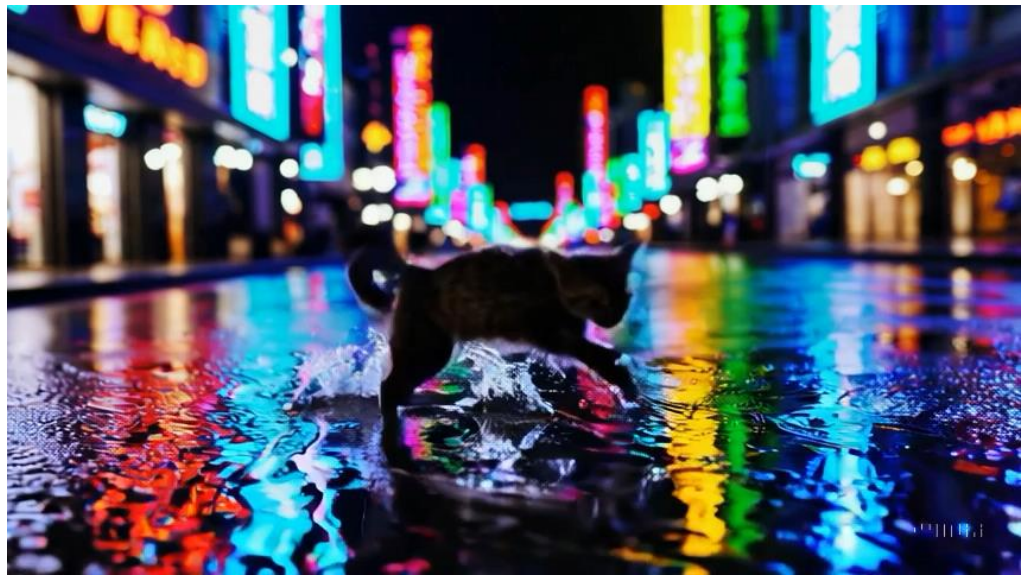
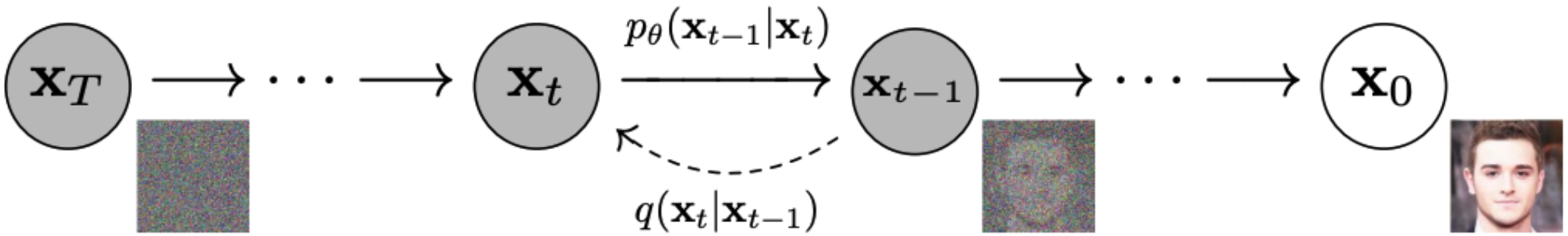


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)



# 拡散モデルとは

- **拡散モデルでは，初期値のノイズから段階的にノイズを除去し，データを生成.**
  - 下の図における$x_T$は，ガウス分布からランダムにサンプリングされた初期値
  - ランダムな初期値テンソルからデータを生成する深層生成モデルの基本的な設計に沿う



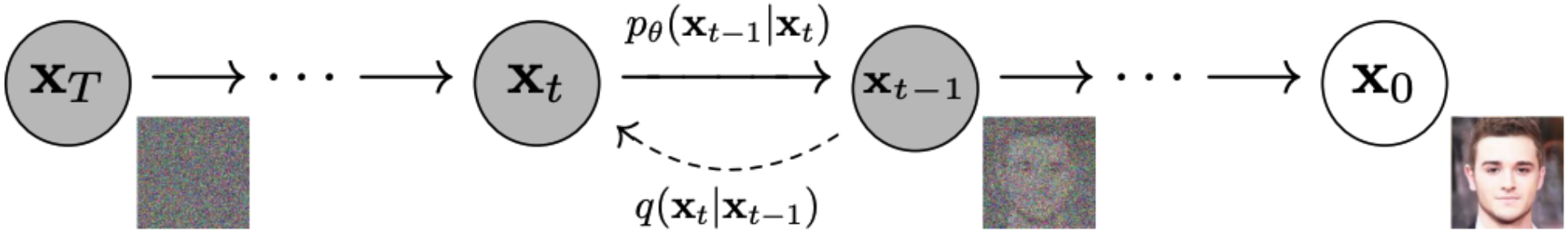
# 拡散モデルの枠組み

## Denoising Diffusion Probalistic Model (DDPM) [1]

- データ$\mathrm{x}_0$に徐々にガウシアンノイズを加えていき，完全なガウシアンノイズ$x_T \sim N(x_T; 0, I)$にする（拡散過程），ここにはNNは不使用

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

- $\beta_t$の値が大きいほど，ノイズの割合が大きい



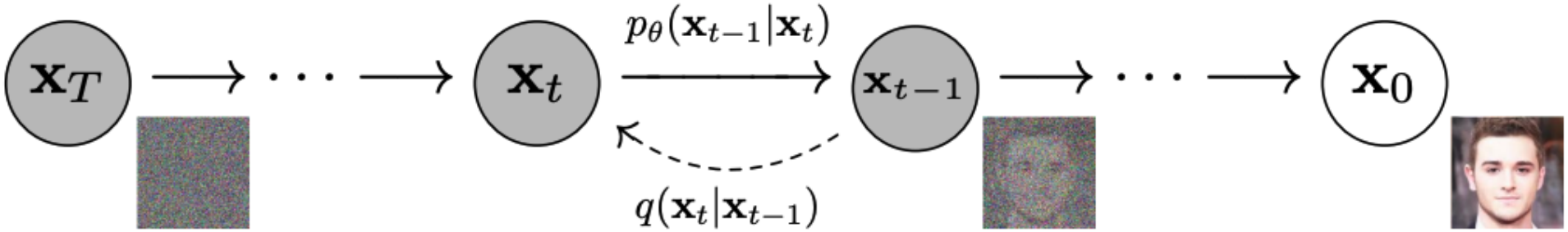
# 拡散モデルの枠組み

## Denoising Diffusion Probalistic Model (DDPM) [1]

- 生成は，拡散過程を逆向きに辿る（逆過程）．逆過程をNNでモデル化

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

- ノイズの乗ったデータ$x_t$と拡散時刻$t$を入力とし，$x_{t-1}$の平均と分散を求める形で定式化



# 拡散モデルの枠組み

## Denoising Diffusion Probalistic Model (DDPM) [1]

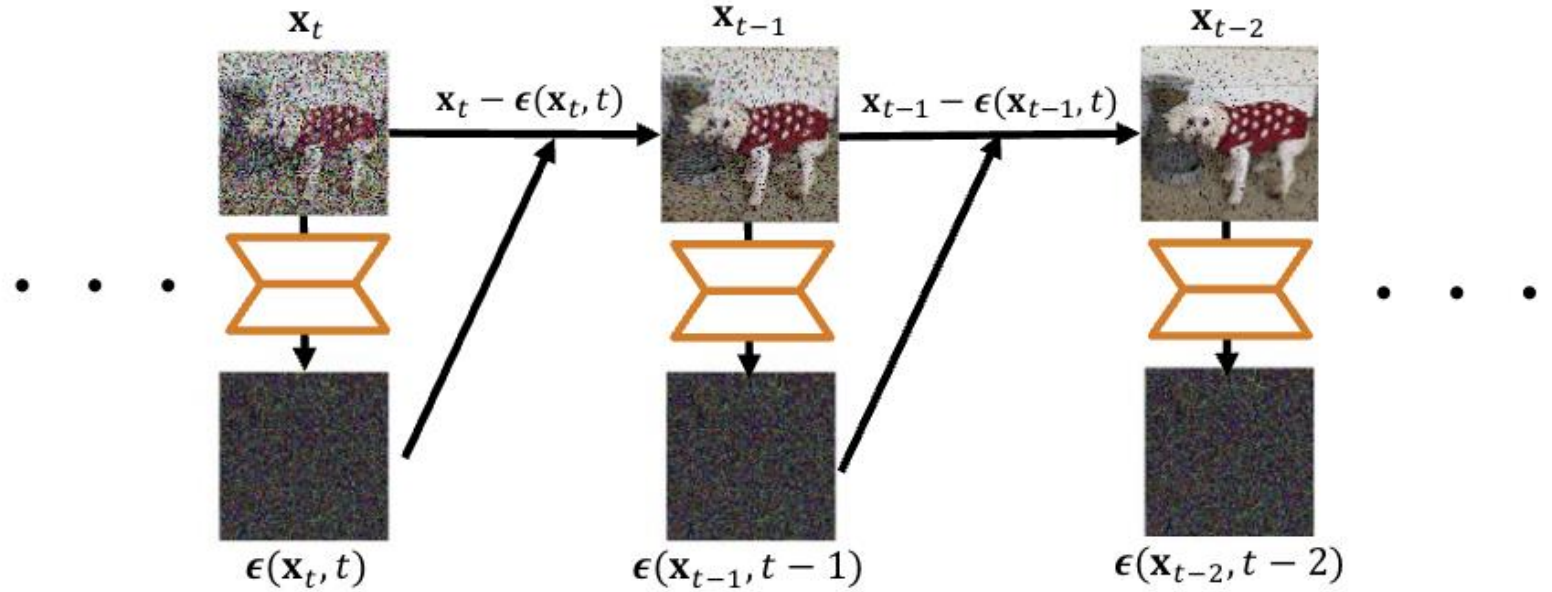
- 実際は，ノイズの乗ったデータ$x_t$と拡散時刻$t$を入力とし，$x_t$に載っているノイズを予測する．このノイズ予測器にNNが用いられている


画像引用元：深層生成モデル2024 第6回 講義資料

# 拡散モデルの枠組み

## Denoising Diffusion Probalistic Model (DDPM) [1]

- 実際は，ノイズの乗ったデータ$x_t$と拡散時刻$t$を入力とし，$x_t$に載っているノイズを予測する．このノイズ予測器にNNが用いられている
- ノイズ予測器で予測したノイズ$\varepsilon$を$x_t$から引くことで，反復的にノイズを除去



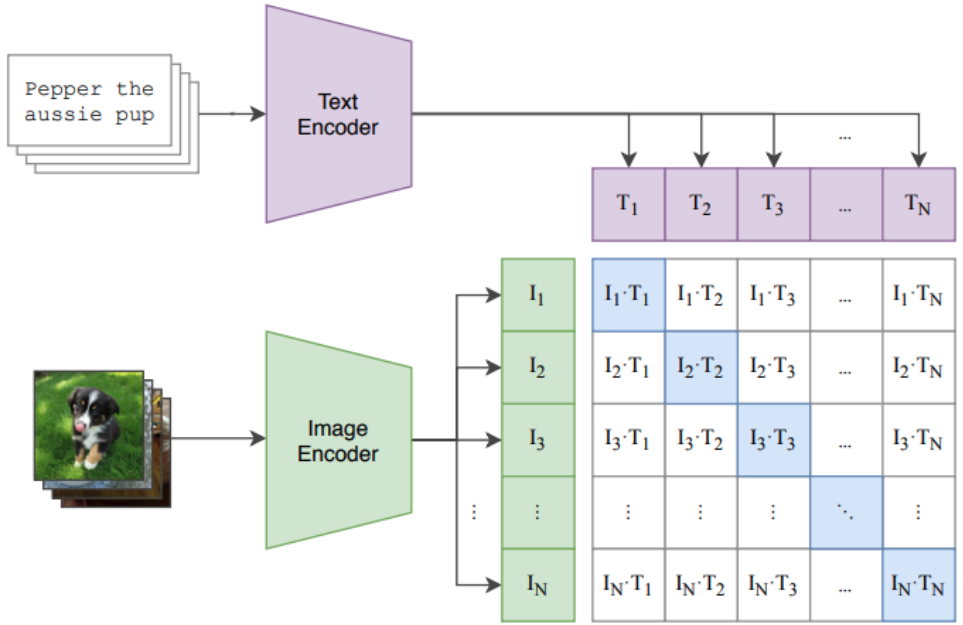

画像引用元：深層生成モデル2024 第6回 講義資料

# 目次

1. 拡散モデルの準備
2. 拡散モデルのアラインメント

# 拡散モデルのアラインメント

- 言語モデルと同様に，拡散モデルにおいても，人間にとって好ましい生成結果を得るために，モデルを**アラインメント**する研究も盛ん
- **報酬モデル**を用意し，より報酬が高い出力を生成できるように，拡散モデルを**強化学習する**方法が，最も基本的
- 動画像生成の領域では，報酬モデルとして，**人間選好のスコア** (ex. HPS v2 [Wu+ 2023]) や，**テキスト追従性スコア** (ex. CLIP [Radford+ 2021])を主に用いる

# 拡散モデルのアラインメント

## 報酬モデル

- 報酬モデルは，プロンプト$c$に対する出力$x$から，報酬を出力するモデル$r_\phi(c,x)$と書くことができる
- 報酬モデルは，プロンプト$c$に対する，好ましい出力$x^w$と好ましくない出力$x^l$の**ペア**を用いて，以下の損失関数を用いて学習される(Bradley-Terry モデル)

$$\mathcal{L}_{\mathrm{RM-BT}}(\phi) = -\mathbb{E}_{(c,x^w,x^l)\sim\mathcal{D}}\left[\log(\sigma(r_\phi(c,x^w) - r_\phi(c,x^l)))\right]$$

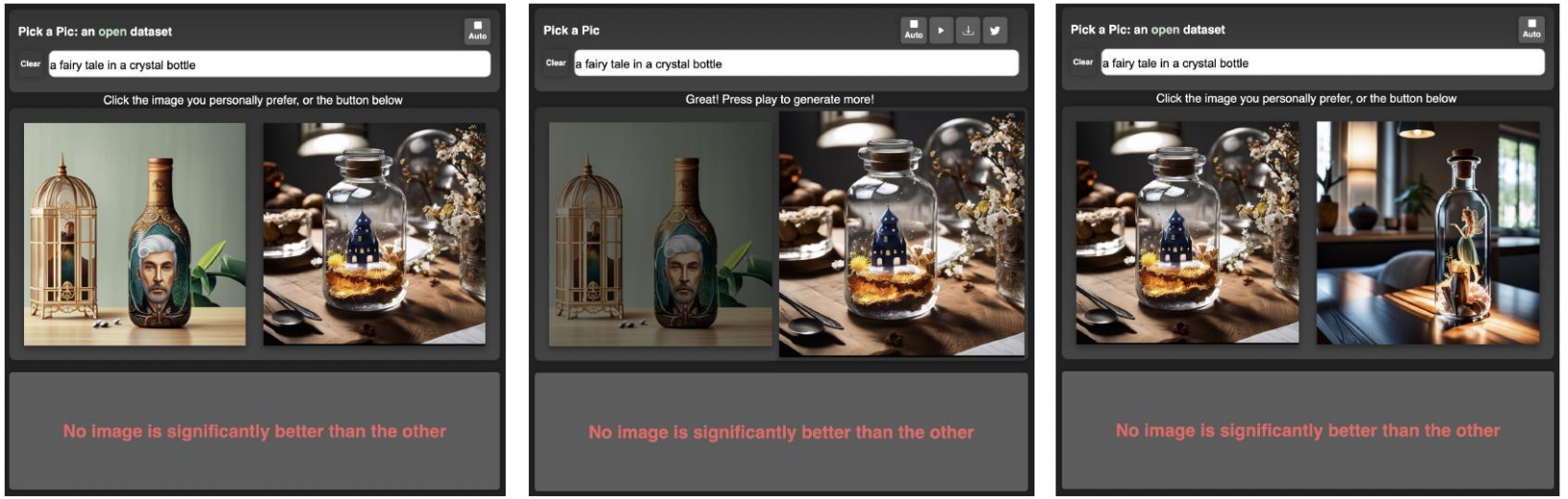
  - ただし$\sigma$はシグモイド関数

式引用元：https://arxiv.org/abs/2409.07253

# 拡散モデルのアラインメント

## 報酬モデル

- 例えば，人間選好のスコアであるPickScore [Kirstain+ 2023] では，プロンプト$c$に対する，好ましい出力$x^w$と好ましくない出力$x^l$のペアは下のようなインターフェイスで集められる




(a) (b) (c)

# 拡散モデルのアラインメント

## Reinforce Learning from Human Feedback (RLHF)

- 報酬モデルは，プロンプト$c$に対する出力$x$から，報酬を出力するモデル$r_\phi(c,x)$と書くことができた
- ナイーブには，この報酬モデルの**報酬を最大化**するように，生成モデル$p_\theta(x_0|c)$を学習できれば良い
- すなわち，以下の最適化問題を解く

$$\min_\theta \mathbb{E}_{c\sim\rho, x_0\sim p_\theta(x_0|c)}\left[-r_\phi(c,x_0)\right]$$

  - ここで$\rho$はプロンプト$c$の分布

式引用元：https://arxiv.org/abs/2409.07253

# 拡散モデルのアラインメント

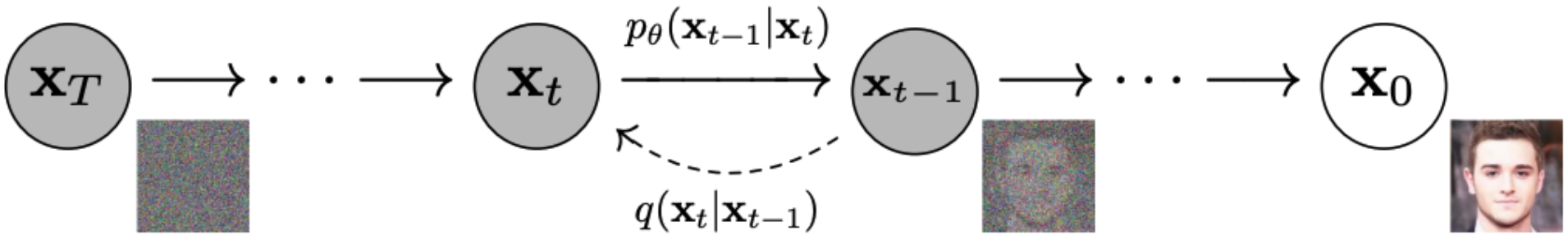
$$\min_{\theta} \mathbb{E}_{c\sim\rho, x_0\sim p_\theta(x_0|c)} \left[-r_\phi(c, x_0)\right]$$

## RLHF: Denoising Diffusion Policy Optimization (DDPO)

- DDPO [Black+ 2023] では，ノイズ除去プロセスをマルチステップ意思決定問題とみなすことで，**方策勾配定理**より，方策勾配を以下のように推定する

$$\mathbb{E}_{c\sim\rho, x_{0:T}\sim p_\theta(x_{0:T}|c)} \left[-r_\phi(c, x_0)\Sigma_{t=1}^{T} \nabla_\theta \log p_\theta(x_{t-1}|x_t, c)\right]$$

  - この定式化の場合は，REINFORCE [Williams 1992] に対応

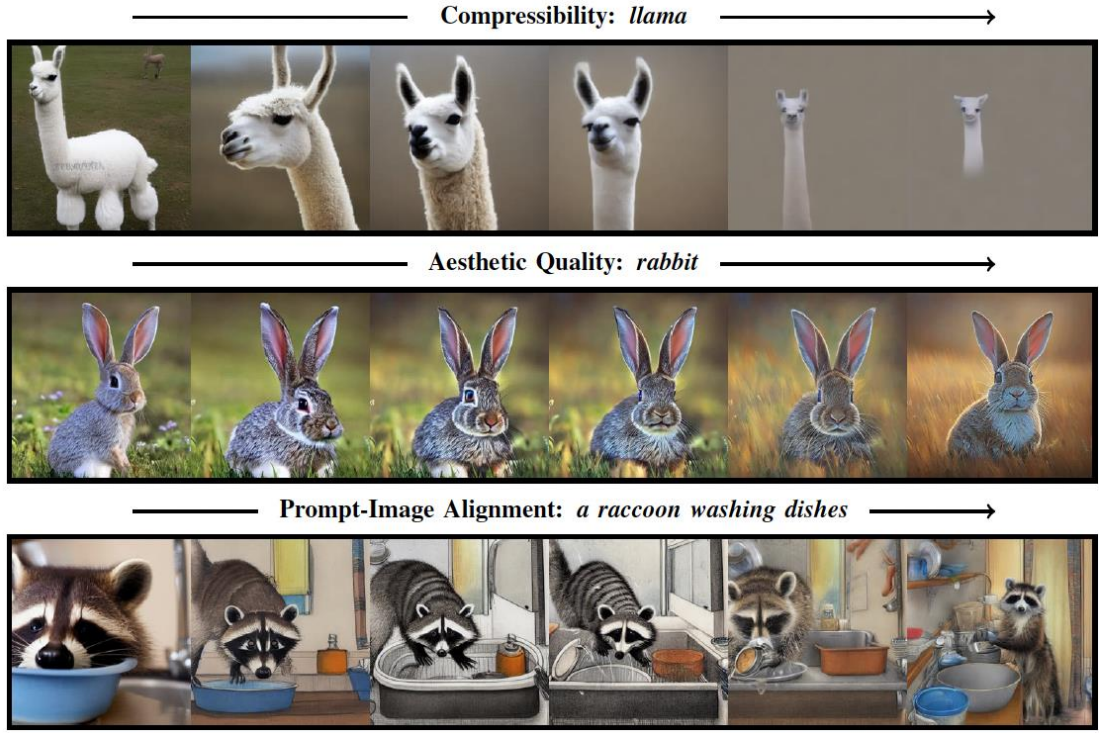


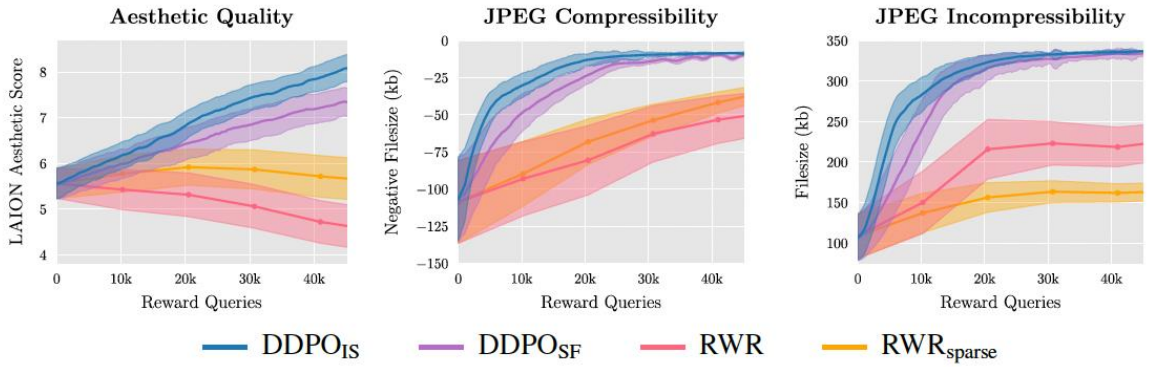

式引用元：https://arxiv.org/abs/2409.07253

# 拡散モデルのアラインメント

## RLHF: Denoising Diffusion Policy Optimization (DDPO)

- DDPOにより，各報酬 (JPEG圧縮可能性，美的魅力，テキスト追従性) に沿う生成ができるようになる



# 拡散モデルのアラインメント

$$\min_{\theta} \mathbb{E}_{c\sim\rho, x_0\sim p_\theta(x_0|c)} \left[-r_\phi(c, x_0)\right]$$

## RLHF: Diffusion Policy Optimization with KL regularization (DPOK)

- 報酬モデル$r_\phi(c, x)$の報酬を最大化するだけでは，実際には**過剰最適化**が起こる
- DPOK [Fan+ 2023] では，報酬最大化の項に，事前学習済みモデル$p_{ref}(x_0|c)$との **KL正則化項**を導入することで，これを回避する

$$\min_{\theta} \mathbb{E}_{c\sim\rho, x_0\sim p_\theta(x_0|c)} \left[-r_\phi(c, x_0) + \beta D_{\mathrm{KL}}(p_\theta(x_0|c)||p_{\mathrm{ref}}(x_0|c))\right]$$

- 実際には，**KL項の上界**を用いて勾配を計算する (DPOK 補題4.2)

$$\min_{\theta} \mathbb{E}_{c\sim\rho, x_{0:T}\sim p_\theta(x_{0:T}|c)} \left[-r_\phi(c, x_0)\right] + \beta \Sigma_{t=1}^{T} \mathbb{E}_{x_t\sim p_\theta(x_t|c)} \left[D_{\mathrm{KL}}(p_\theta(x_{t-1}|x_t, c)||p_{\mathrm{ref}}(x_{t-1}|x_t, c)\right]$$

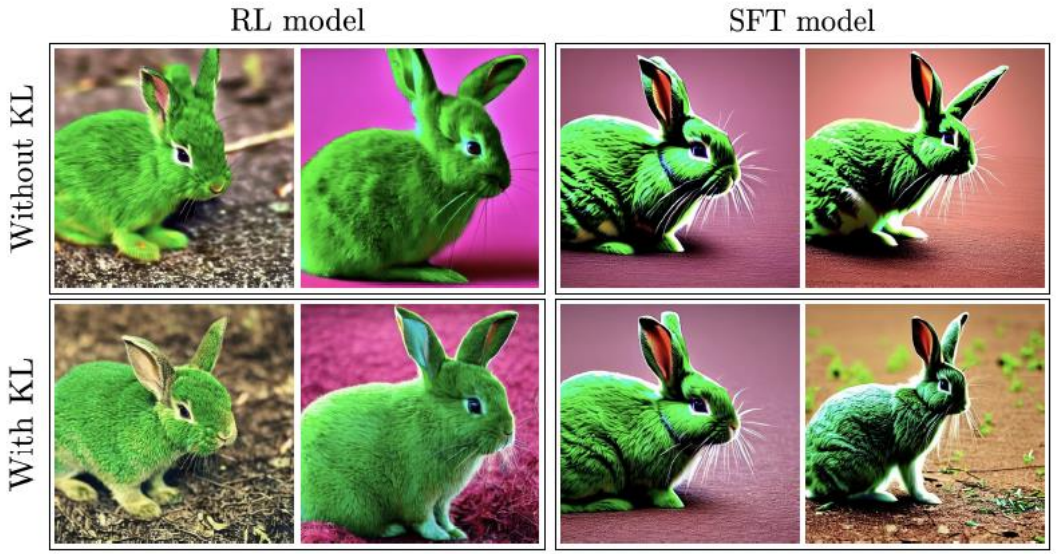
式引用元：https://arxiv.org/abs/2409.07253

# 拡散モデルのアラインメント

## RLHF: Diffusion Policy Optimization with KL regularization (DPOK)

- KL正則化により，過剰最適化により画像の質を落とすことなく，報酬関数にアラインすることができる


(a) ImageReward and Aesthetic scores




(b) Generated images

# 拡散モデルのアラインメント

## Direct Preference Optimization (DPO)

- 報酬モデルは，プロンプト$c$に対する出力$x$から，報酬を出力するモデル$r_\phi(c,x)$と書くことができた
- この報酬モデルに対して，**KL正則化付き報酬最大化**を達成するような生成モデル$p^*_\theta(x|c)$は下のように表された

$$\max_{p_\theta} \mathbb{E}_{c\sim\rho, x\sim p_\theta(x|c)} \left[ r_\phi(c,x) - \beta D_{\mathrm{KL}}\left(p_\theta(x|c) || p_{\mathrm{ref}}(x|c)\right)\right]$$

- これを変形すると，以下のようになる ($\mathcal{Z}$は規格化定数) [Rafailov+ 2023]

$$p^*(x|c) = \frac{1}{\mathcal{Z}(c)} p_{\mathrm{ref}}(x|c) \exp\left(\frac{1}{\beta} r(c,x)\right)$$

式引用元：https://arxiv.org/abs/2409.07253

# 拡散モデルのアラインメント

$$\mathcal{L}_{\mathrm{RM-BT}}(\phi) = -\mathbb{E}_{(c,x^w,x^l)\sim\mathcal{D}}\left[\log(\sigma(r_\phi(c,x^w) - r_\phi(c,x^l)))\right]$$

## Direct Preference Optimization (DPO)

- さらに変形し，報酬モデルについて解くと，下のようになる

$$r(c,x) = \beta \log \frac{p^*(x|c)}{p_{\mathrm{ref}}(x|c)} + \beta \log \mathcal{Z}(c)$$

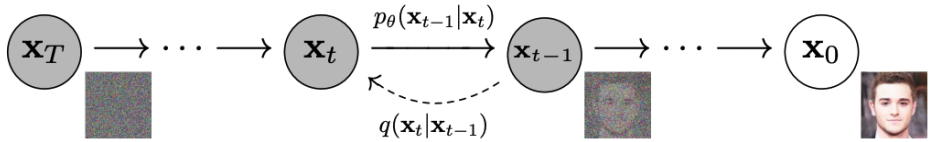
- これを，報酬モデルの損失関数に代入すると，損失関数から**報酬モデルの項を削除**することができ，このようになる
  - これがDPOの損失関数，報酬関数を学習せずとも，好ましい出力$x^w$と好ましくない出力$x^l$のペアから**直接**選好を学習できる

$$\mathcal{L}_{\mathrm{DPO}}(p_\theta; p_{\mathrm{ref}}) = -\mathbb{E}_{(c,x^w,x^l)\sim\mathcal{D}}\left[\log\sigma\left(\beta \log \frac{p_\theta(x^w|c)}{p_{\mathrm{ref}}(x^w|c)} - \beta \log \frac{p_\theta(x^l|c)}{p_{\mathrm{ref}}(x^l|c)}\right)\right]$$



式引用元：https://arxiv.org/abs/2409.07253

# 拡散モデルのアラインメント




## DPO: Diffusion-DPO

- DPOの定式化を拡散モデルに適用すると，以下のようになる

$$\mathcal{L}_{\text{Diffusion-DPO}}(p_\theta; p_{\text{ref}}) = -\mathbb{E}_{(c,x_0^w,x_0^l)\sim\mathcal{D}} \log \sigma \Bigg( \beta \mathbb{E}_{x_{1:T}^w \sim p_\theta(x_{1:T}^w|x_0^w,c), x_{1:T}^l \sim p_\theta(x_{1:T}^l|x_0^l,c)} \left[ \log \frac{p_\theta(x_{0:T}^w|c)}{p_{\text{ref}}(x_{0:T}^w|c)} - \log \frac{p_\theta(x_{0:T}^l|c)}{p_{\text{ref}}(x_{0:T}^l|c)} \right] \Bigg)$$

- イェンゼンの不等式を用いることで，この損失関数は上から抑えられる

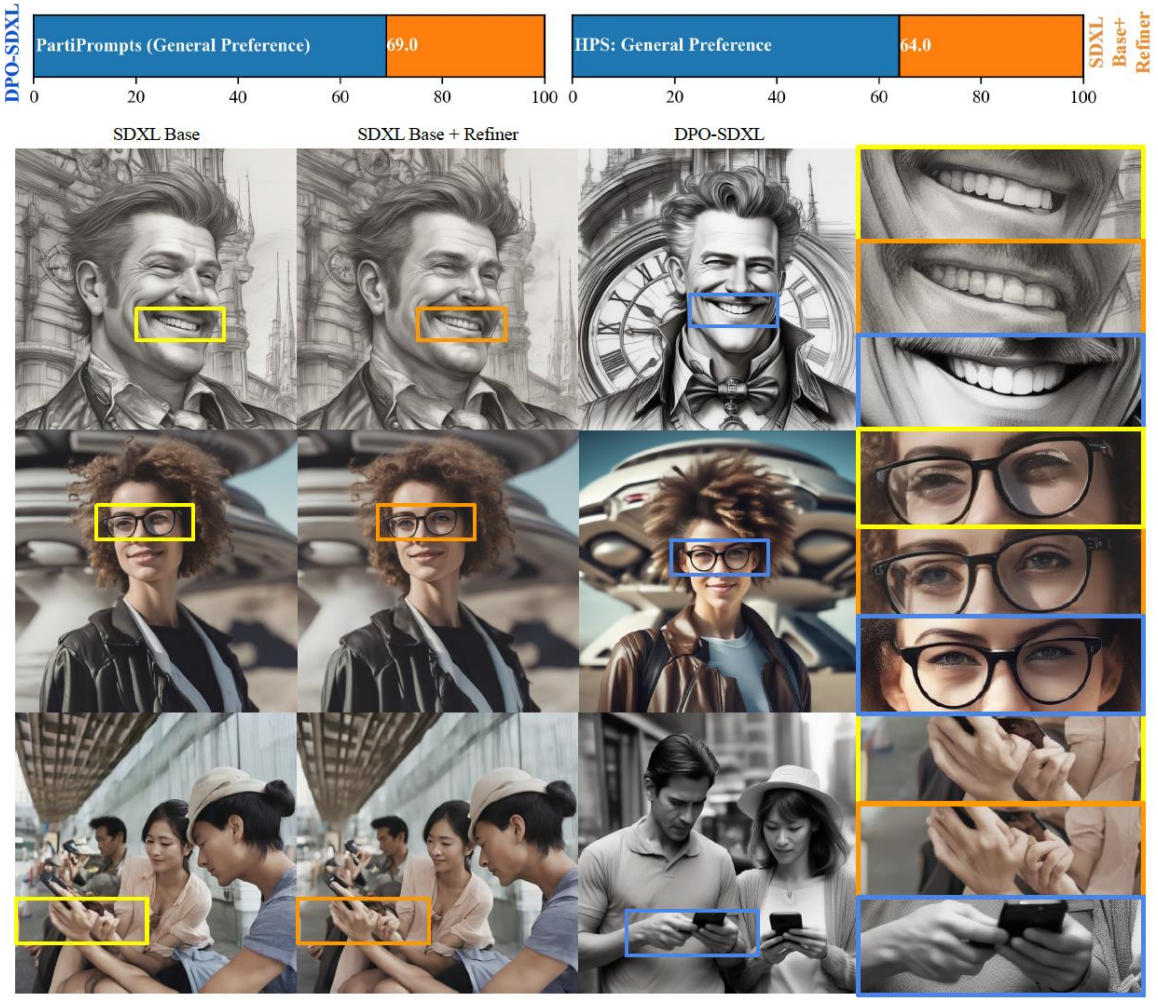
$$\mathcal{L}_{\text{Diffusion-DPO}}(p_\theta; p_{\text{ref}}) \leq -\mathbb{E}_{(c,x_0^w,x_0^l)\sim\mathcal{D}, t\sim\mathcal{U}(0,T), x_{t-1,t}^w \sim p_\theta(x_{t-1,t}^w|x_0^w,c), x_{t-1,t}^l \sim p_\theta(x_{t-1,t}^l|x_0^l,c)} \log \sigma \Bigg( \beta T \left[ \log \frac{p_\theta(x_{t-1}^w|x_t^w,c)}{p_{\text{ref}}(x_{t-1}^w|x_t^w,c)} - \log \frac{p_\theta(x_{t-1}^l|x_t^l,c)}{p_{\text{ref}}(x_{t-1}^l|x_t^l,c)} \right] \Bigg).$$

式引用元：https://arxiv.org/abs/2409.07253

# 拡散モデルのアライメント

## DPO: Diffusion-DPO

- Diffusion-DPOによりfine-tuneされたSDXLは，テスターによる評価で，元のモデルよりもより好まれる
- 定性的にも，画像の不自然な部分(歯，目，手などの細部)を改善



# 拡散モデルのアラインメント

## ここまでのまとめ

- 拡散モデルにも．LLMのアラインメントで主要な方法である，RLHFとDPOを適用することができる
- RLHFでは報酬モデルを学習し，拡散モデルを強化学習で修正する
- 一方で，DPOでは好ましい出力$x^w$と好ましくない出力$x^l$のペアのみを用いて，拡散モデルを修正する

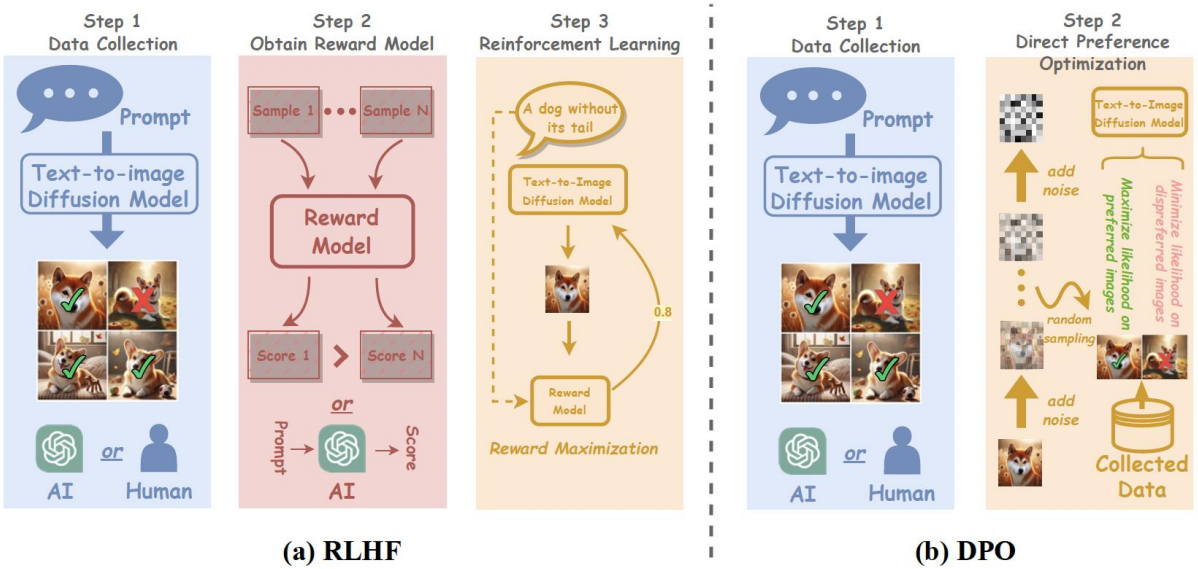



Fig. 5. The overview of RLHF and DPO of diffusion models.

図引用元：https://arxiv.org/abs/2409.07253

# 拡散モデルのアラインメント

## Test-timeのアラインメント

- Classifier guidance [Dhariwal et al. 2021]
- DOODL [Wallace et al. 2023]
- これらの手法は，勾配を用いて生成を改善するため，評価器が微分可能でないといけない

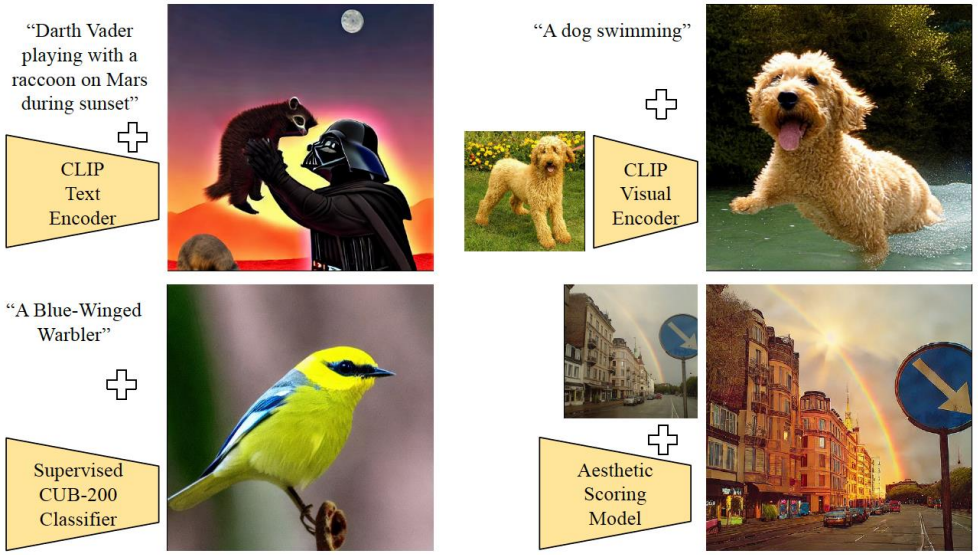



Figure 1: We propose DOODL - a process that directly optimizes diffusion latents w.r.t. a model-based loss on the final generation. Our method improves on vanilla classifier guidance in all tested settings and we demonstrate capabilities novel to this class of methods such as vocabulary expansion, entity personalization, and perceived aesthetic value improvement.

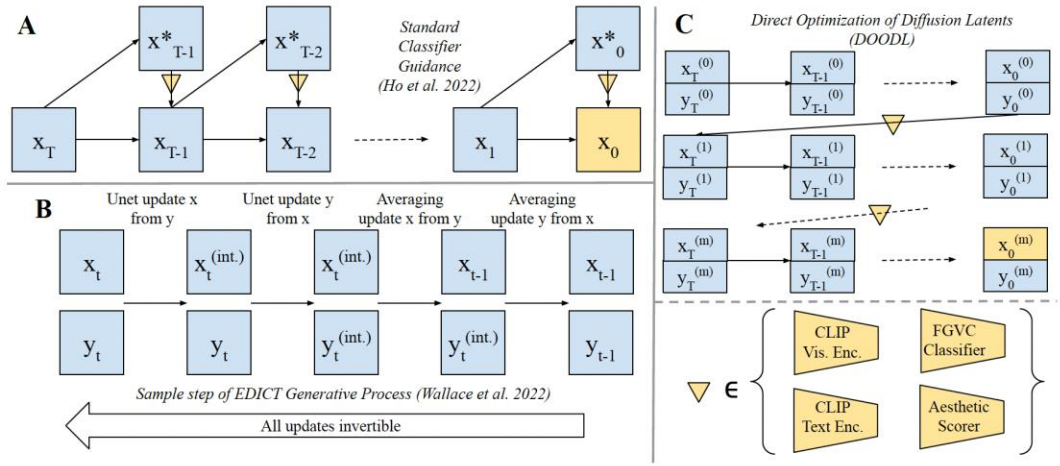
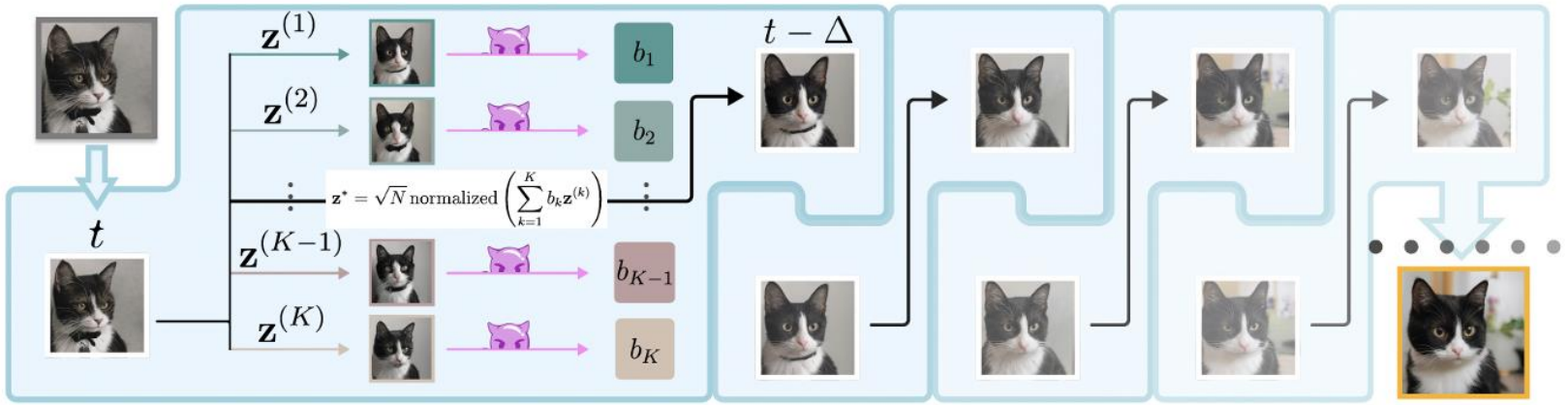



Figure 3: Method diagrams. **A** Standard classifier guidance: at each timestep, $t$, a one-step denoising approximation of $x_0$ is computed and the loss is calculated w.r.t the pixels of this generation. The gradient of this loss is incorporated into the subsequent diffusion step. **B** EDICT [45], an invertible variant of the diffusion process which admits backpropagation through the entire chain with no additional memory cost. **C** DOODL, our proposed method. We leverage EDICT and demonstrate that the gradients of model losses computed w.r.t. the final generation can be used to optimize the fully noised $x_T$ directly. ▼ indicates a gradient calculation from a differentiable model-based loss with networks employed in this work displayed.

# 拡散モデルのアラインメント

- Sampling Demon [Yeh+ 2024]
  - 再訓練や，評価器の微分可能性が必要ない
  - Demonとは，熱力学的プロセスを操作する架空の存在であるマクスウェルのデーモンに由来



# 結論

- 拡散モデルにも．LLMのアラインメントで主要な方法である，RLHFとDPOを適用することができる
  - RLHFでは報酬モデルを学習し，拡散モデルを強化学習で修正する
  - 一方で，DPOでは好ましい出力$x^w$と好ましくない出力$x^l$のペアのみを用いて，拡散モデルを修正する
- また，Test-timeにアラインメントする手法も研究がなされている
  - 評価器に対するアラインメントという観点からは，条件付けの手法として知られていたClassifier guidanceも，拡散モデルのTest-timeアラインメント手法とみなせる
  - さらに，拡散モデルのパスを制御しようとする手法も出てきている