第１回定期ミーティング

2024年4月30日(水)

早稲田大学　基幹理工学研究科
電子物理システム学専攻 史研究室
石黒将太郎・野口颯汰

拡散モデル(学習における問題)

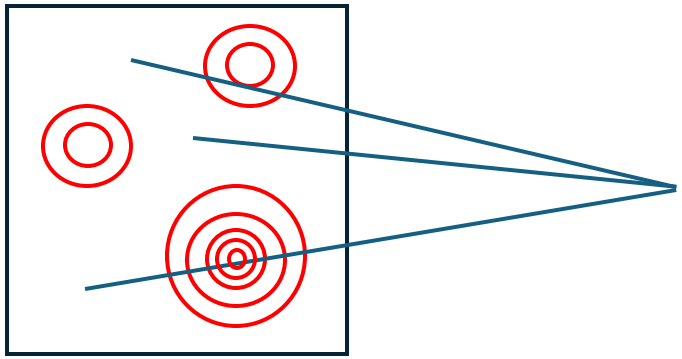
- 世の中で重要な高次元データにおいては一部のデータのみが非ゼロの確率を持つとされる⇒**多様体仮説**

  問題①：多様体仮説が成立するデータを学習する場合、確率密度が小さい領域から初期サンプルを抽出する可能性が高い

  ⇒**パラメータ更新がしづらい**

  問題②：多峰性を持つデータ分布の場合、あるモードから多モードへ移りにくい

  ⇒**最もよい学習結果**にたどり着きにくい



パラメータ更新中に、確率密度が小さい領域を通過すると、スコア(勾配のlog)推定が不正確になり、学習が行われない

図: 高次元データの確率密度関数(多様体仮説).

拡散モデル(拡散による問題解決)

- 確率が低い領域での探索がランダムウォークである問題を解決するために、高い確率密度の領域を**周辺の空間方向に拡散**する
- 確率密度の小さい領域でもスコア関数 $s_\theta(x, \sigma)$ が有意な大きさとなり、モードの中心へと近づく方向にパラメータを更新できる

  ⇒目標確率分布に対してカバー率の高い学習が可能

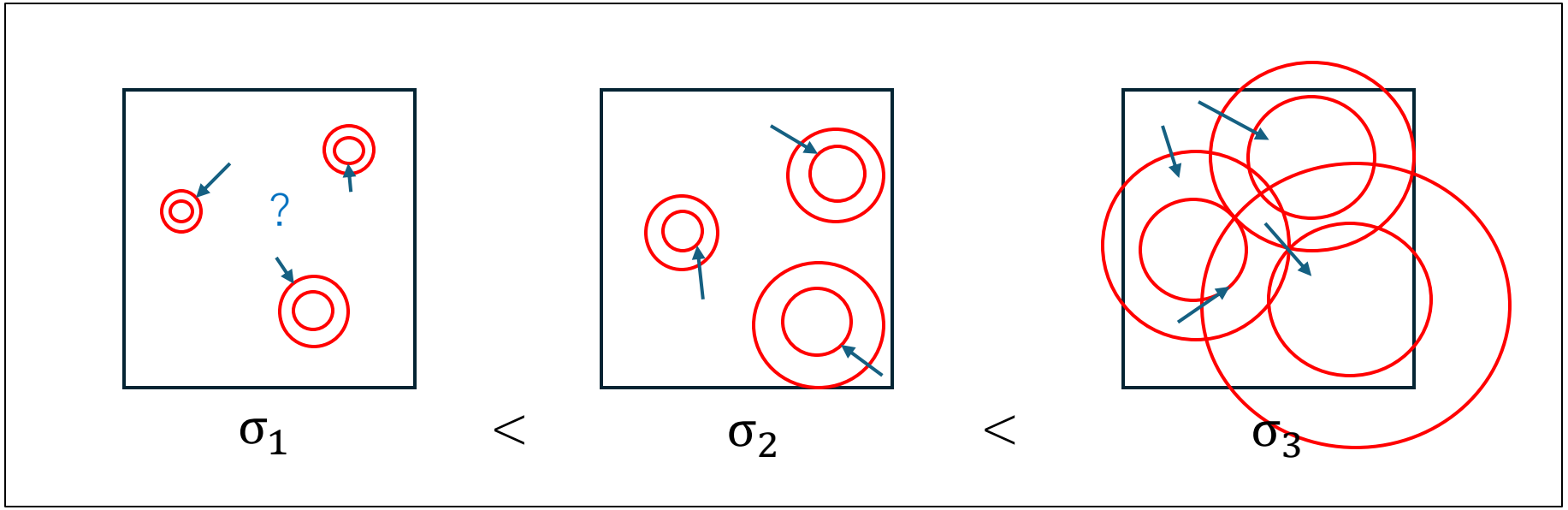


図: 異なるノイズσを用いた確率分布の攪乱.

拡散モデル(学習過程)

- DDPM(デノイジング拡散確率モデル)は、学習データの分布(x_0)を拡散することで完全なノイズ(正規分布N)に近づける**拡散過程**と、拡散過程を逆向きにたどる**逆拡散過程**(生成過程)に分けられる
- 拡散過程はマルコフ過程であり、中間層にある確率分布を解析的に求められる

  ⇒全ての層に対して誤差逆伝播法を適用する必要がないため、効率的に学習
- 逆拡散過程はノイズ付与画像$x_{1:T}$に対してT回デノイズするため、生成速度は遅い

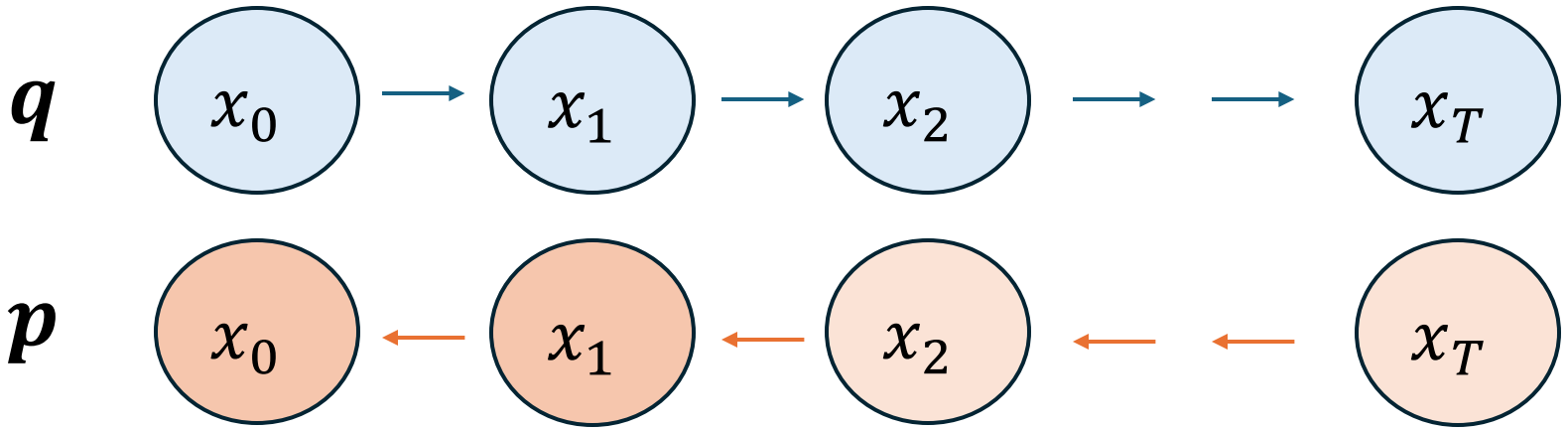


図: 拡散過程qと逆拡散過程p.

拡散モデル(ノイズスケジューリング)

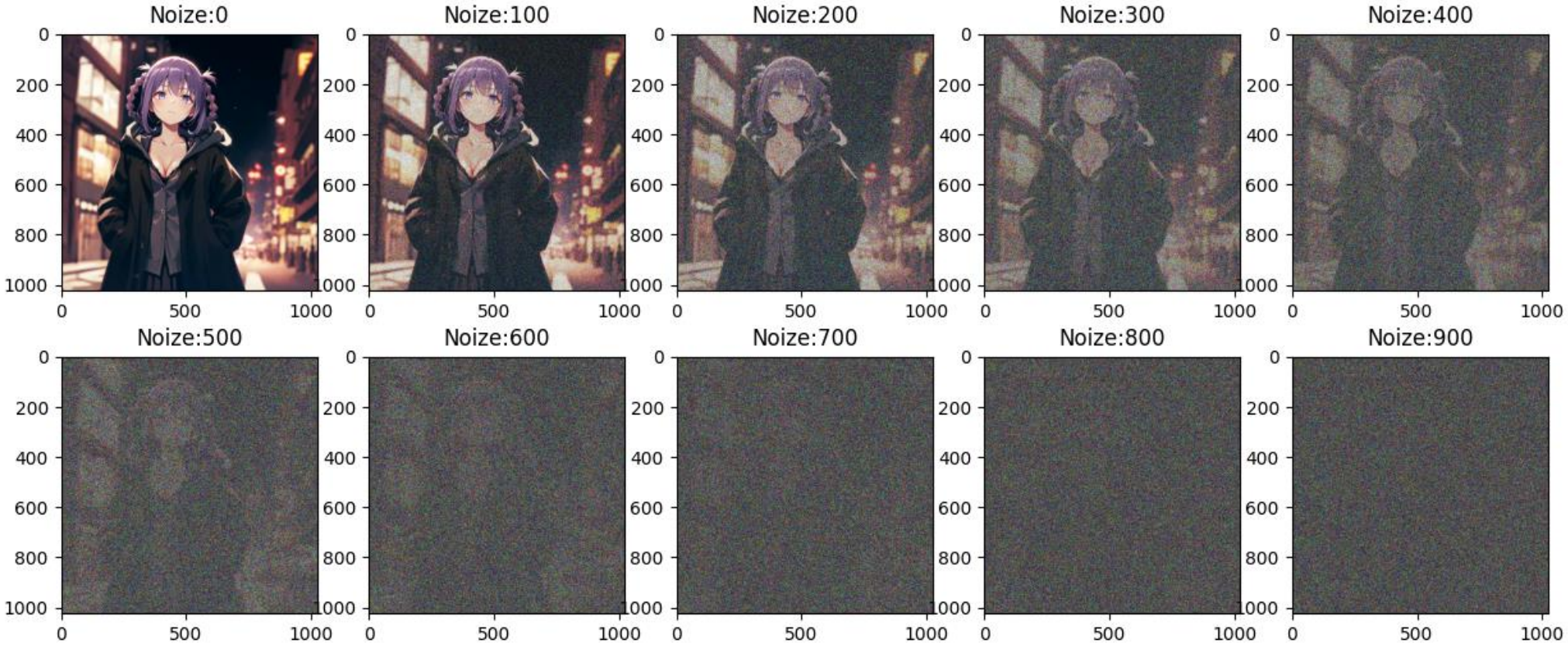


図: 実際のノイズスケジューリング.

拡散モデル(生成イメージ)

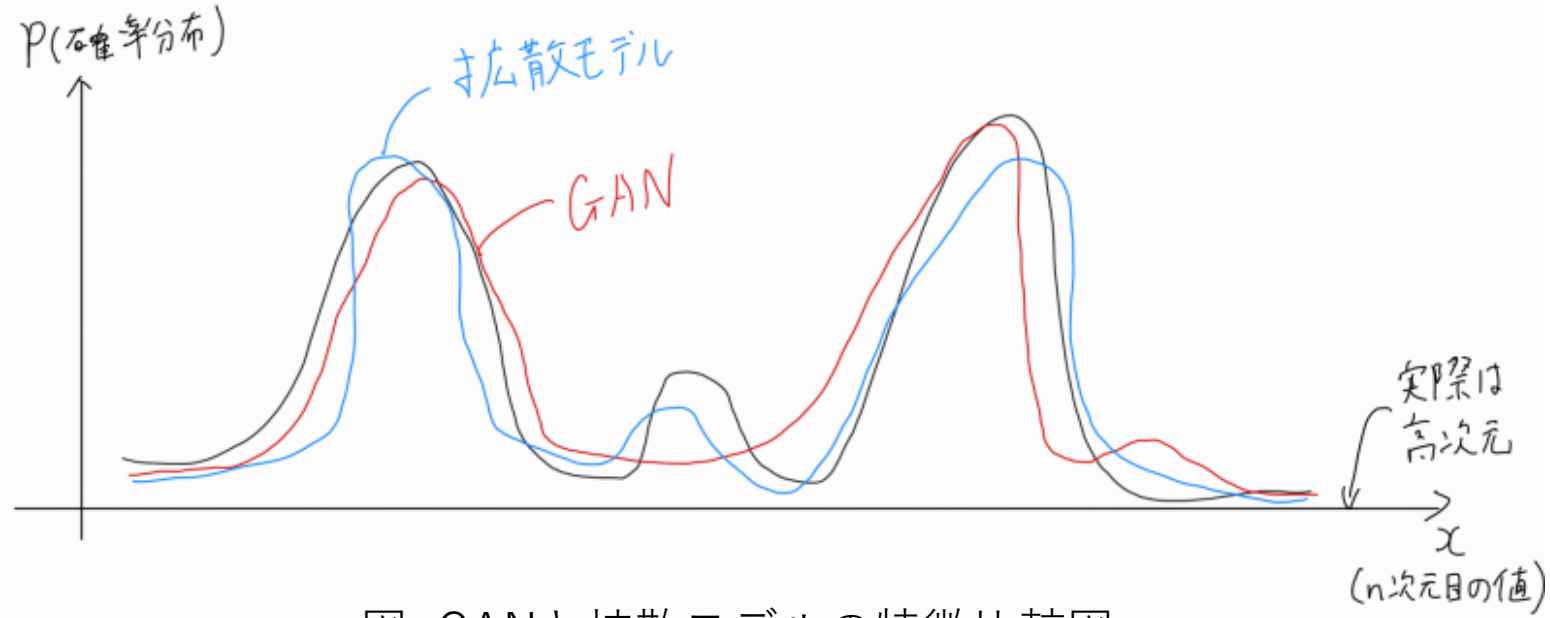


図: GANと拡散モデルの特徴比較図.

図: GANと拡散モデルの生成画像.[1]

モデル紹介(Stable Diffusion)

- Stable Diffusion[2]は、最も有名な拡散モデルで、ノイズを予測するNNとしてU-Netを採用し、画像とガイドプロンプトを紐づけるためにCLIPを利用

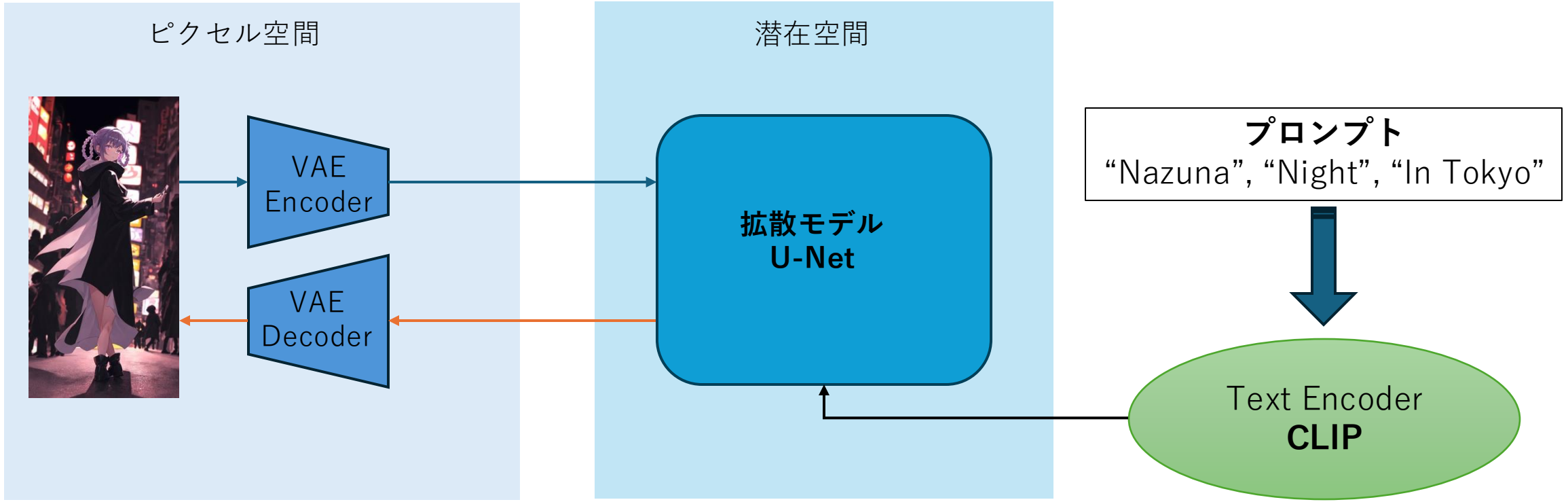


図:Stable Diffusionの模式図.

論文紹介

- *ユーザ制御可能な二次元感情パラメータを用いた人物顔画像の表情生成*[3]では、感情の正負と激しさの2次元軸に基づいて、人物の表情を連続的に変化させる手法を提案
- 提案手法では、顔の外形を制御可能にする3次元頭部モデル(3DMM)で制御し、拡散モデル(DiffusionRig[4])によって顔画像を出力

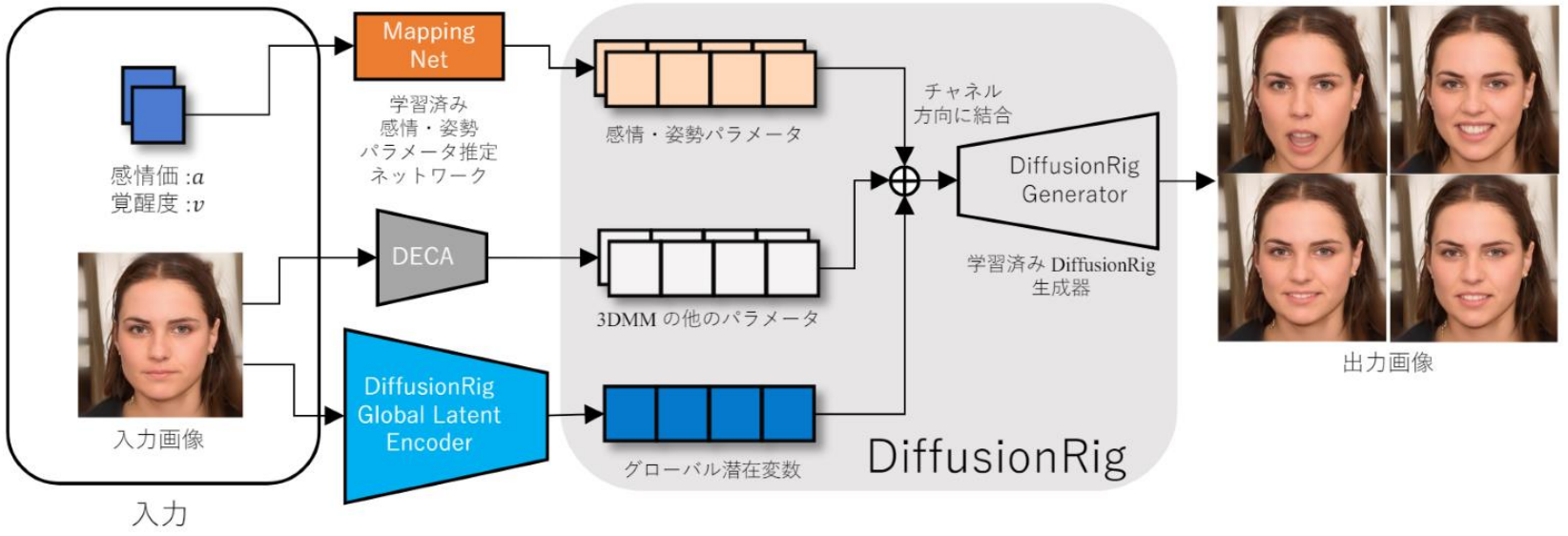


図: ネットワーク概要.[3]

論文紹介

- 既存手法(TensorGAN)では、離散感情カテゴリの表情の間でしか表情を変化させられないが、提案モデルはその間を補完可能

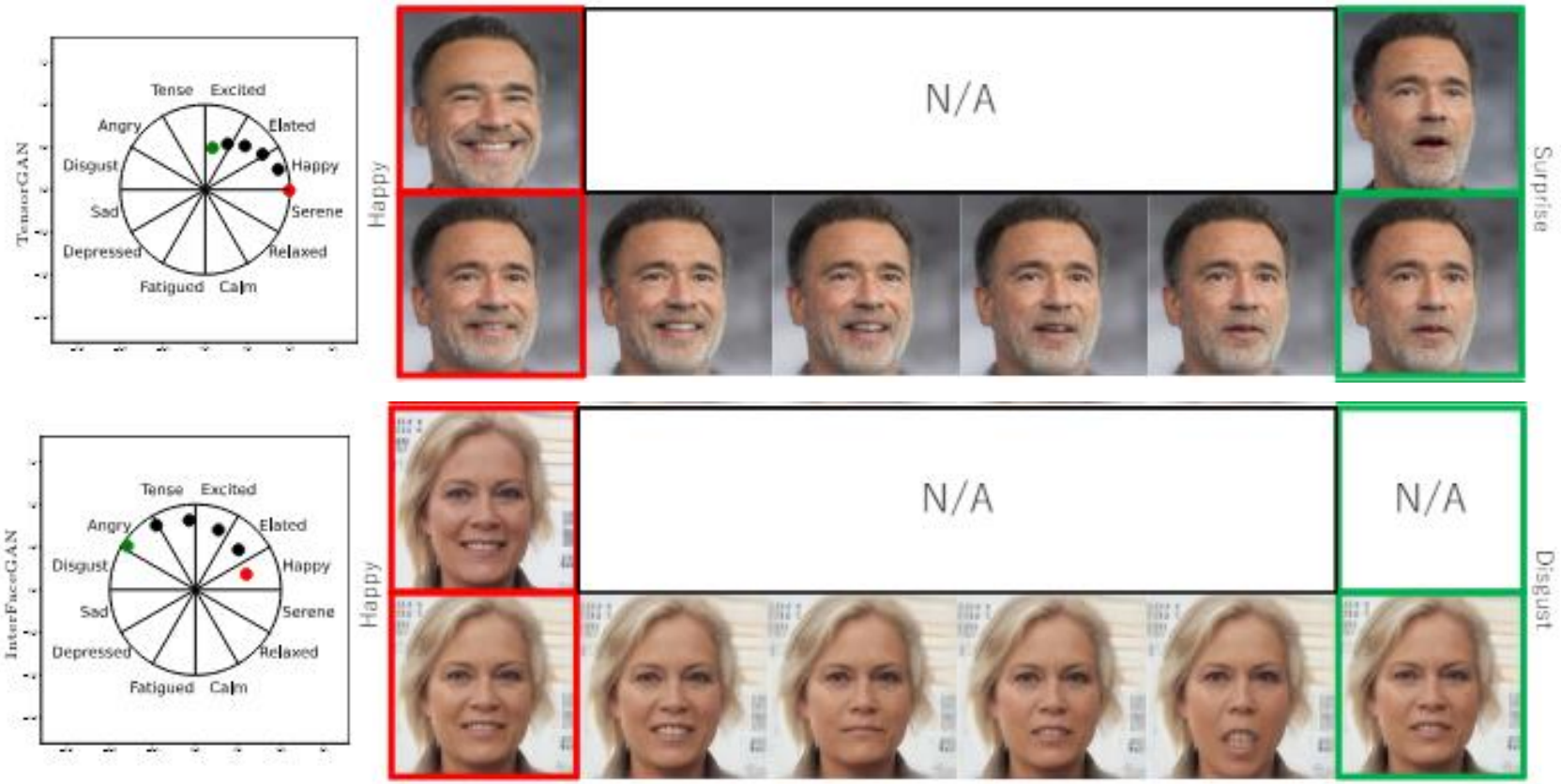


図: 出力画像の比較.[3]

今後の計画

1. 基礎的な拡散モデル(実装済み)の検証
2. Stable Diffusion・DiffusionRigのコード解析及びローカル環境での実装
3. 3D点群データ・メッシュ構造を入力とした拡散モデルの文献調査
4. 拡散モデルの軽量化(枝刈り・量子化・ノイズスケジューリング・部分拡散モデル)の文献調査
5. 顔及び頭部を表現できる3Dデータの調査

参考文献

[1] Prafulla Dhariwal, Alex Nichol, OpenAI, "*Diffusion Models Beat GANs on Image Synthesis*", 35th Conference on Neural Information Processing Systems, 2021

[2] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, Björn Ommer, "*High-Resolution Image Synthesis with Latent Diffusion Models*", CVPR, 2022

[3] 金森透有, 金森由博, 遠藤結城, *"ユーザ制御可能な二次元感情パラメータを用いた人物顔画像の表情生成"*, 情報処理学会研究報告, 2023

[4] Zheng Ding1, Xuaner Zhang2, Zhihao Xia2, Lars Jebe2, Zhuowen Tu1, Xiuming Zhang2, *"DiffusionRig: Learning Personalized Priors for Facial Appearance Editing"*, CVPR, 2023

関連研究紹介　ARKitで取得する情報

- 条件：LiDARを搭載したiOS 14以上のiPhone

	TrueDepthカメラ	LiDARセンサー
場所	フロントカメラ（内カメラ）	バックカメラ（外カメラ）
仕組み	①赤外線ドットプロジェクターと赤外線カメラの構成 ②プロジェクターから顔に数十万個のドットを投射し、その反射光を取得 ③ドットの歪みから顔の深度情報を計算	①レーザー光を照射 ②反射光を受信して距離を周囲の物体までの距離を正確に測定できる
測定範囲	数十cm程度	数メートルから数十メートル程度
応用例	顔認証やポートレート撮影	AR機能や自動運転

関連研究紹介　デプスデータ比較

表１.ARKitで取得可能な深度データ

デプスデータ	概要	ボトルネック	データの型
capturedDepthData	True-Depthカメラ由来のデプスデータで、フェイストラッキング利用時のみ取得	capturedDepthData	AVDepthData
estimatedDepth	デュアルカメラやTrue-Depthカメラ由来のデプスではなく、機械学習ベースの推定デプスデータ	カメラフレーム	CVPixelBuffer
sceneDepth	ワールドトラッキングのみ、デバイスとしてLiDARを搭載している必要があり	カメラフレーム	ARDepthData

capturedDepthData　OR　sceneDepth

関連研究紹介　smoothedSceneDepth

- 概要：

  フレーム間の LiDAR 読み取り値（sceneDepth）の差を最小化するためにデータを平均として処理する

  ⇒時間軸上で滑らかに補正する

- 効果：

  遠景をぼかして霞ませることで、奥行き感や空気感を表現する手法であるフォグ効果で証明されている奥行きのあるオブジェクトを描写する際にフリッカーを低減し、より滑らかなモーション効果を達成

今後の３Dデータ取得

- 目標：既存コードを参考にsceneDepthを実装
- 課題：

①ARKit・Swift・Metal・Xcode・UIKitの理解

ARKit： Appleが提供する拡張現実（AR）用のフレームワーク

Swift： Appleが開発したプログラミング言語

Metal：　　ハードウェアアクセラレーターを使用するグラフィックスを Apple プラットフォームで強化する技術

Xcode：Appleが提供する統合開発環境（IDE）

②点群データ前処理

位置合わせ、顔画像部分の切り取り、ノイズ除去、サンプリング

今後の３Dデータ取得

- 対策：

  拡散モデルの開発を優先的に進めていくために

  ①自分で点群データセットを生み出せるようになる

  ②外部サービスを用いて拡散モデルの研究を進める

  外部サービスとは…Sakura3D SCAN

  2,500万点の点群取得

  出力フォーマット：ASC、TXT

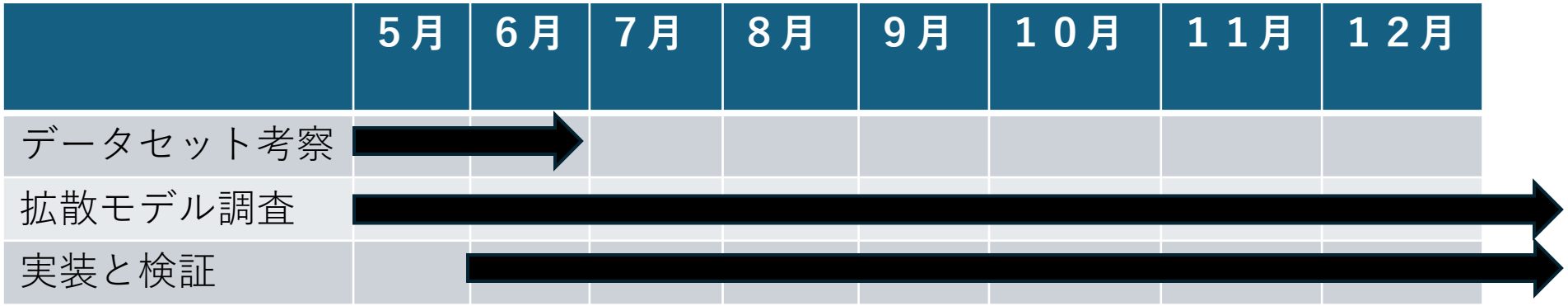
  他ソフト読み込み可能

「Sakura3D SCAN」で取得した3次元データ

実物の写真

今後の研究計画

表２.今後の研究計画



・３Dデータセット
　　capturedDepthDataとsceneDepthの実装・考察
　　既存拡散モデルデータセットの調査

・５月末までに一度既存モデルで実装を行い、拡散モデルにおける実現可能性を検証
　　学習時間、推論時間、精度など
⇒拡散モデルの妥当性・改善点・テーマ考察

参考文献

- https://zenn.dev/shu223/articles/arkit-lidar-depth
- https://developer.apple.com/documentation/arkit/arkit_in_ios/environmental_analysis/displaying_a_point_cloud_using_scene_depth
- https://developer.apple.com/documentation/arkit/arframe/3674209-smoothedscenedepth?changes=latest_beta

GAN(補助スライド)

- Generator(G)とDiscriminator(D)の2つのネットワークを利用
- Gは生成元(ノイズ・画像)から画像を生成
- Dは画像が本物か偽物(生成画像)かどうかを識別
- GはDを騙せるような画像を生成するように学習し、Dは本物と偽物を間違えないように学習

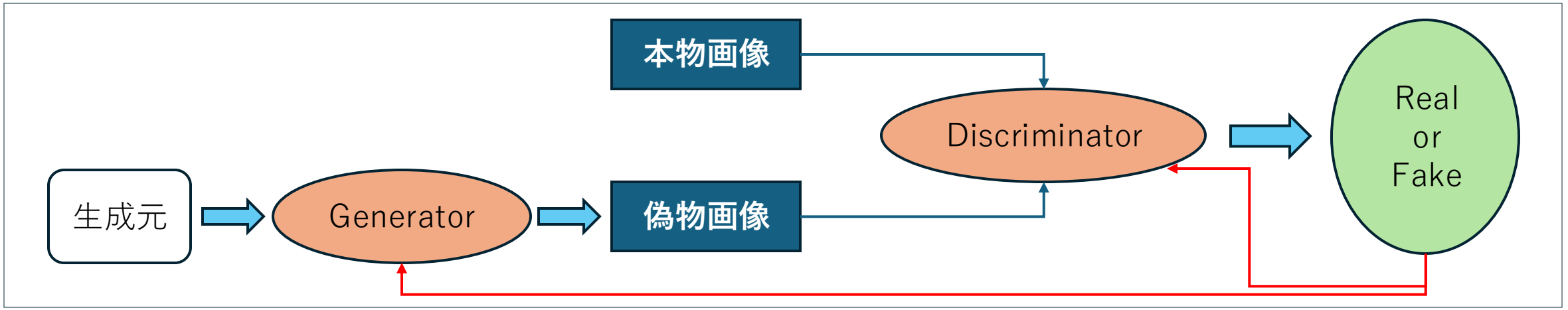


図: GANの構造.