第14回定期ミーティング

2025/03/04

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

目次

1.研究テーマ

2.先行研究

- DDIMの損失関数
- DiffusionAutoencoderの損失関数
- 他のDiffusionモデルのアイデンティティ損失追加方法

3.今後の研究計画

研究の方向性

表情変換顔生成モデル(Diffsion)と2Dto3D顔形状を生成するモデルを
組み合わせることで、3Dの表情変換を軽量化かつ自然に行う

Step1：表情編集に特化したDDIMを訓練

Step2：変換前後でβ変化しないように訓練

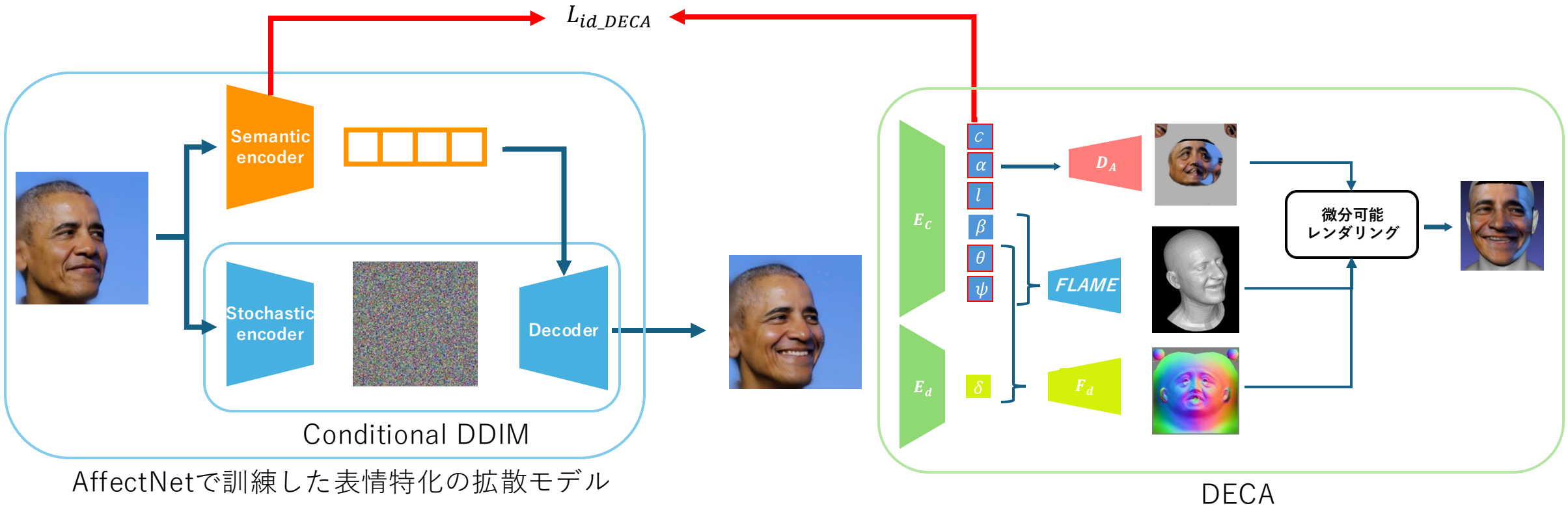


図. DECA[1]とDiffusionAutoencoders[2]を元にした提案モデル

[1] YAO FENG et al. Learning an Animatable Detailed 3D Face Model from In-The-Wild Images
[2] Konpat Preechakul et al. Diffusion Autoencoders: Toward a Meaningful and Decodable Representation

目次

DDIMの損失関数

実際のノイズと予測したノイズの二乗誤差を最小化することで学習

ノイズの付加：各時刻 t において、元の画像 x_0に対して以下の式でノイズが加えられます

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon$$

x_0：元のデータ
x_t: 時刻 tにおけるノイズが加わったデータ
$\sqrt{\bar{a}_t}$: 元のデータをどの程度残すかを決める「重み」
$\sqrt{1-\bar{a}_t}$: どの程度ノイズを混ぜるかを決める「重み」
ϵ: 実際に加えるランダムなノイズ

$\bar{a}_t$：元データの残り具合

$$\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$$

(α_s: 各ステップ s での「元のデータを残す割合」)

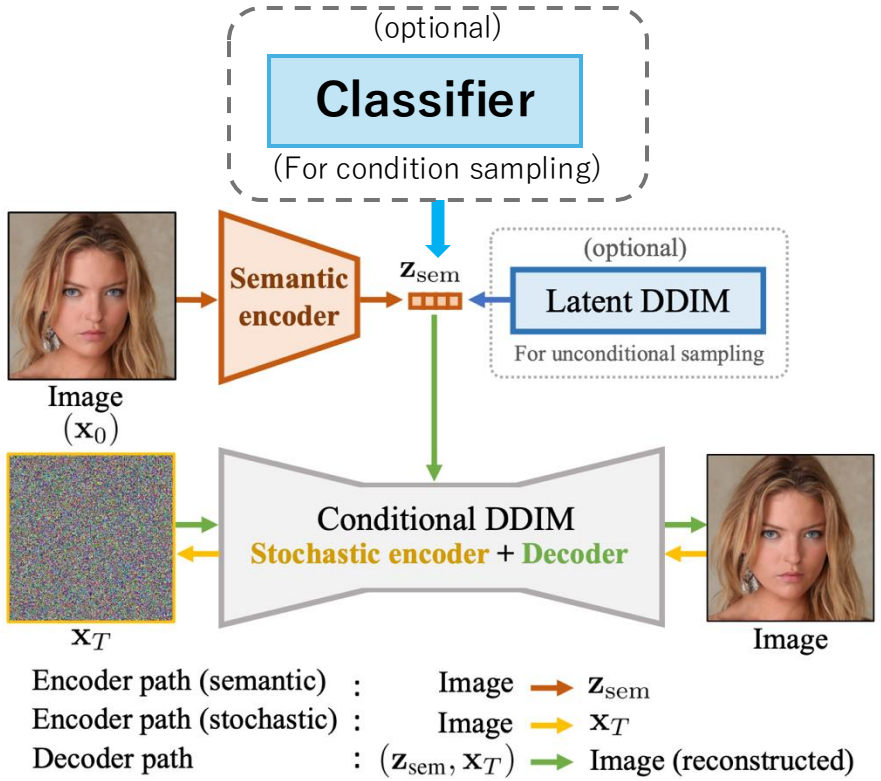
DDIM損失関数：予測されたノイズ $\epsilon_\theta(x_t, t)$と実際に加えたノイズ ϵ との差の二乗誤差（MSE）として定義

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,\, x_0,\, \epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]$$

ϵ: 実際のノイズ
ϵ_θ ：モデルが予測したノイズ

DiffAEの損失関数

DiffusionAutoencoderと線形分類器の両方を訓練する



画像生成の流れ

1. 入力画像をSemantic encoderとStochastic encoderに入力
2. Semantic encoderで意味的情報z_{sem}を抽出
3. Stochastic encoderで画像の詳細情報x_Tを抽出
4. 線形分類器から属性の方向を示す重みベクトルを取得
5. 抽出されたz_{sem}に対して、重みベクトルを加算
6. 操作後のz_{sem}とx_TをDecoderに入力し、属性変更した画像を生成

Manipulationにおける訓練

1. Diffusion Autoencoderの訓練

$$L_{\text{simple}} = \sum_{t=1}^{T} \mathbb{E}_{x_0,\epsilon_t} \left[\|\epsilon_\theta(x_t, t, z_{\text{sem}}) - \epsilon_t\|_2^2 \right]$$

2. 線形分類器の訓練

入力されたz_{sem}に対して、以下の重みベクトルを推定

$$z_{sem\_new} = w^T z_{sem} + b$$

今回発見した新しい改善余地：

線形分類器の改善、AffectNetを用いたDifuusionAutoencoderの訓練

ID-BOOTH: IDENTITY-CONSISTENT IMAGE GENERATION WITH DIFFUSION MODELS

Darian Tomašević, Fadi Boutros, Naser Damer, Vitomir Struc, Peter Peer

アイデンティティを考慮するために新しい損失関数を導入

背景

- 厳格なプライバシー規制により、大規模な生体認証データセットの取り下げ
- 特に顔認識タスクにおいて、被験者の同意なしに収集された大規模データセットの問題
- GDPRなどのデータ保護法による顔認識データセットの収集・配布・利用の制限
- 時間のかかる手動収集と多様性に欠ける小規模データセット化
- 既存の生成アプローチが個人のアイデンティティを考慮せず、生成結果の一貫性が低い問題
- アイデンティティベースのトレーニングによる過学習と生成多様性の低下の問題

貢献

- triplet identity training objectiveの導入
  - 同一個人内での一貫性の向上（intra-identity consistency）
  - 異なる個人間での識別性の向上（inter-identity separability）
  - 生成画像の多様性の向上
  - 精度を保った上での、適切な同意を得た実世界データセットでのトレーニング

DreamBoothは拡散モデルのファインチューニングにおける欠点を解決

DreamBooth （Ruiz et al., 2023）の概念

従来の研究：入力画像に対する少数の画像を用いたファインチューニング

欠点：

- 入力画像に対する過学習
- 既存の知識が失われる可能性（人の顔とは？などの一般的な概念）

DreamBooth：

- 事前学習済みの拡散モデルを利用して、対象の概念に関連する画像セットを生成
- この画像をトレーニング中に使用し、モデルの合成能力を維持する

方法：拡散モデルの損失関数＋事前知識保持目的関数の組み合わせによるファインチューニング

事前知識保持関数：

$$\mathcal{L}_{PR} = \mathbb{E}_{z_{pr},c_{pr},\epsilon',t'}\left[\epsilon_{pr} - \epsilon_\theta(z_{pr,t'}, t', c_{pr})\|_2^2\right]$$

1. デノイジングネットワークのみの学習(コスト面のため)
2. LoRAという追加する新しいレイヤーのみ学習し、事前学習済みの重みは変更しない

PortraitBoothでは、$\mathcal{L}_{ID}$を追加することで、アイデンティティ問題を解決

DreamBooth （Ruiz et al., 2023）の問題点

既存の拡散モデル（DreamBooth含む）は、画像の再構成（画像を鮮明にすること）に注力している
→**顔のアイデンティティ**を直接的に考慮していない

引き起こされる問題：

1. **意図した入力アイデンティティとの一致性**: 生成画像が、意図した人物の顔として認識されない
2. **他の生成サンプルとの一貫性:** 同じ人物の画像を複数生成しても、毎回顔が違って見える

解決（PortraitBooth）： $\mathcal{L}_{ID}$を損失関数に加える

$$\mathcal{L}_{ID} = 1 - Sim(\varphi(x_0^f), \varphi(\hat{x}_0^f))$$

φ: 顔認識モデル（アイデンティティ特徴を出力）
$\hat{x}_0^f$ ：生成画像の顔領域
x_0^f ：入力画像の顔領域
Sim ：コサイン類似度

ID-Boothでは、$\mathcal{L}_{ID}$の問題点を解決するために$\mathcal{L}_{TID}$を追加する

$\mathcal{L}_{ID}$（PortraitBooth）の問題点

1. **表情やポーズなどの特定の特徴に過学習しやすい**
（アイデンティティ情報に不必要な要素が含まれるリスク）。
2. **学習したアイデンティティ埋め込みに、トレーニング画像のバイアスが入り込む可能性**

解決（ID-Booth）：$\mathcal{L}_{TID}$（三重項アイデンティティ損失）を損失関数に加える

$$\mathcal{L}_{TID} = max\{Sim(\varphi(x_0^f), \varphi(\hat{x}_0^f)) - Sim(\varphi(x_{pr,0}^f), \varphi(\hat{x}_0^f)) + m, 0\}$$

φ: 顔認識モデル（アイデンティティ特徴を出力）
$\hat{x}_0^f$：生成画像の顔領域（**アンカー**）
x_0^f：アンカーと同じ**アイデンティティ**を持つ顔領域（**ポジティブ**）
$x_{pr,0}^f$：アンカーと異なる**アイデンティティ**を持つ顔領域（**ネガティブ**）
Sim：コサイン類似度
m：ポジティブ・ネガティブ間の最低限の類似度差

$\mathcal{L}_{TID}$を追加することで、アイデンティティ過学習問題を解決

$$\mathcal{L}_{TID} = max\{Sim(\varphi(x_0^f), \varphi(\hat{x}_0^f)) - Sim(\varphi(x_{pr,0}^f), \varphi(\hat{x}_0^f)) + m, 0\}$$

φ: 顔認識モデル（アイデンティティ特徴を出力）
$\hat{x}_0^f$：生成画像の顔領域（**アンカー**）
x_0^f：アンカーと同じ**アイデンティティ**を持つ顔領域（**ポジティブ**）
$x_{pr,0}^f$：アンカーと異なる**アイデンティティ**を持つ顔領域（**ネガティブ**）
Sim：コサイン類似度
m：ポジティブ・ネガティブ間の最低限の類似度差

$\mathcal{L}_{ID}$（**三重項アイデンティティ損失の目的**）：
アンカーと**ポジティブ**の**類似度**を高くする
アンカーと**ネガティブ**の**類似度**を低くする

→ **ポジティブ**に近く、**ネガティブ**とは異なるアイデンティティを持った画像が生成可能

意図しない特徴（例：表情やポーズ）の過学習を引き起こす**リスク**を軽減することができる

セキュリティに担保したデータセットを用いて網羅的に実験

Dataset：

データセット	用途	枚数	特徴
TFD	微調整用	2213	実験室環境、被写体の同意取得済み
FFHQ	生成画像の評価用	70,000	野外環境で撮影、多様な顔画像

Implementation：

拡散モデル	Stable Diffusion 2.1/Stable Diffusion XL
微調整手法	LoRA
比較対象	• $\mathcal{L}_{LDM} + \mathcal{L}_{PR}$ • $\mathcal{L}_{LDM} + \mathcal{L}_{PR} + \mathcal{L}_{ID}$ • $\mathcal{L}_{LDM} + \mathcal{L}_{PR} + \mathcal{L}_{TID}$
特徴抽出	アイデンティティ抽出 ：ArcFace 顔検出 ：MTCNN
プロンプト設定	Base：クローズアップポートレート Complex：表情と周囲の環境を指定

品質、多様性、アイデンティティの一貫性、および顔認識モデルの性能を評価

指標名	略称	目的	使用モデル・手法	評価内容
Fréchet Inception Distance	FID	画像品質評価	Inception-v3	生成画像と実画像の分布の類似度（低いほど高品質）
CLIP Maximum Mean Discrepancy	CMMD	画像品質評価	CLIP	生成画像と実画像の視覚的・概念的類似度
Improved Precision	-	忠実度評価	Inception-v3	合成画像のリアルさ（高いほど本物に近い）
Improved Accuracy	-	多様性評価	Inception-v3	生成画像の多様性（高いほどバリエーション豊富）
Certainty Ratio Face Image Quality Assessment	CR-FIQA	顔画像品質評価	ResNet-101	生成画像の識別可能性（高いほど高品質）
Equal Error Rate	EER	アイデンティティの一貫性評価	ArcFace	認証誤り率（低いほど良い）
False Match Rate @ 1.0% False Non-Match Rate	FMR100	アイデンティティの識別性評価	ArcFace	1%のFNRでの誤受理率（低いほど良い）
False Match Rate @ 0.01% False Non-Match Rate	FMR1000	アイデンティティの識別性評価	ArcFace	0.01%のFNRでの誤受理率（低いほど良い）
Fisher Discriminant Ratio	FDR	アイデンティティの分離度評価	ArcFace	同一・異なるアイデンティティの識別性（高いほど良い）
ResNet-18 CosFace認識精度	-	顔認識性能評価	ResNet-18 CosFace	生成画像を用いた顔認識精度（高いほど良い）

既存の微調整手法よりもID-Boothは意図した表情や環境を正しく再現できる

Data from	Prompt	Fine-tuning	FID ↓ TFD	FID ↓ FFHQ	CMMD ↓ TFD	CMMD ↓ FFHQ	Precision ↑ TFD	Precision ↑ FFHQ	Recall ↑ TFD	Recall ↑ FFHQ	CR-FIQA ↑ –
TFD	–	–	17.446	79.884	0.008	0.929	0.507	0.684	0.316	0.003	2.131 ± 0.093
FFHQ	–	–	79.884	10.425	0.929	0.005	0.684	0.786	0.003	0.781	2.089 ± 0.145
SD-2.1	Base	–	98.010	79.240	1.485	0.923	0.002	0.413	0.356	0.317	1.737 ± 0.643
		DreamBooth	35.285	61.511	0.530	**1.242**	0.178	**0.614**	0.462	0.040	2.079 ± 0.351
		PortraitBooth	**32.991**	**60.551**	**0.519**	1.260	**0.191**	0.581	0.469	**0.049**	**2.109 ± 0.298**
		ID-Booth	33.488	61.000	0.519	1.255	0.191	0.570	**0.477**	0.038	2.097 ± 0.321
	Complex	–	89.890	44.860	1.514	1.048	0.000	0.513	0.497	0.475	1.857 ± 0.521
		DreamBooth	65.758	51.901	1.432	1.115	0.001	0.597	**0.438**	**0.186**	1.991 ± 0.487
		PortraitBooth	**62.038**	51.912	1.412	1.104	**0.004**	0.614	0.332	0.147	**2.028 ± 0.452**
		ID-Booth	62.114	**51.815**	**1.407**	**1.103**	0.001	**0.622**	0.320	0.184	2.019 ± 0.471
SD-XL	Base	–	91.633	74.058	3.236	2.379	0.000	0.464	0.106	0.225	1.986 ± 0.381
		DreamBooth	**40.806**	**69.788**	0.812	1.482	**0.155**	**0.543**	**0.306**	0.007	**2.178 ± 0.106**
		PortraitBooth	41.664	72.679	**0.807**	**1.454**	0.135	0.515	0.226	0.007	2.175 ± 0.102
		ID-Booth	41.637	72.865	0.809	1.458	0.129	0.531	0.175	**0.009**	2.175 ± 0.097
	Complex	–	86.399	41.411	2.328	1.590	0.001	0.663	0.314	0.296	2.083 ± 0.210
		DreamBooth	67.919	**51.590**	0.722	**0.982**	0.015	**0.494**	0.488	**0.155**	2.139 ± 0.228
		PortraitBooth	66.278	53.502	0.710	0.999	0.016	0.466	**0.531**	0.102	2.141 ± 0.227
		ID-Booth	**66.259**	53.488	**0.709**	0.999	**0.017**	0.459	0.493	0.137	**2.144 ± 0.220**

(↓) – Lower is better; (↑) – Higher is better; (**Bold**) – Best fine-tuning result; (Underline) – Second best fine-tuning result

ID-BoothとPortraitBoothはどちらも優れている

ID-BoothとPortraitBoothの比較をすると…

PortraitBooth：FFHQのComplexプロンプトで大幅にRecallが低下（多様性が損なわれる）
ID-Booth　　：Complexプロンプトにおいても高いRecallを維持
結論：ID-Boothは意図した表情や環境を正しく再現できる

ID-Boothは、PortraitBoothと同等のアイデンティティ性能を維持しつつ、より柔軟なプロンプト制御が可能

	Data from	Prompt	Fine-tuning	EER ↓	FMR100 ↓	FMR1000 ↓	Imposter $\mu \pm \sigma$ ↓	Genuine $\mu \pm \sigma$ ↑	FDR ↑
	TFD	–	–	0.001	0.001	0.001	0.021 ± 0.073	0.871 ± 0.065	75.753
vs. real identities	SD-2.1	Base	DreamBooth	0.039	0.052	0.066	**0.022 ± 0.075**	0.638 ± 0.170	10.916
			PortraitBooth	**0.029**	**0.034**	**0.041**	0.024 ± 0.073	**0.653 ± 0.148**	**14.531**
			ID-Booth	0.031	0.038	0.044	0.023 ± 0.073	0.650 ± 0.154	13.541
		Complex	DreamBooth	0.153	0.364	0.500	**0.014 ± 0.072**	**0.244 ± 0.144**	2.030
			PortraitBooth	0.137	**0.322**	0.489	0.015 ± 0.072	0.255 ± 0.142	**2.268**
			ID-Booth	**0.137**	0.326	**0.465**	0.016 ± 0.071	0.254 ± 0.141	2.260
	SD-XL	Base	DreamBooth	0.002	0.002	0.002	0.022 ± 0.074	0.782 ± 0.071	54.952
			PortraitBooth	0.003	0.003	0.003	0.021 ± 0.075	0.786 ± 0.074	53.233
			ID-Booth	**0.002**	**0.002**	**0.002**	**0.021 ± 0.074**	**0.786 ± 0.067**	**58.578**
		Complex	DreamBooth	0.019	0.023	0.035	0.019 ± 0.074	0.635 ± 0.144	14.537
			PortraitBooth	0.016	0.018	0.031	0.019 ± 0.074	0.646 ± 0.138	16.087
			ID-Booth	**0.015**	**0.016**	**0.024**	**0.019 ± 0.074**	**0.647 ± 0.135**	**16.473**
vs. synthetic identities	SD-2.1	Base	DreamBooth	0.067	0.090	0.106	**0.057 ± 0.079**	0.684 ± 0.224	6.955
			PortraitBooth	**0.052**	**0.063**	**0.071**	0.062 ± 0.077	**0.713 ± 0.196**	**9.596**
			ID-Booth	0.061	0.073	0.082	0.061 ± 0.079	0.702 ± 0.209	8.224
		Complex	DreamBooth	0.242	0.596	0.803	**0.087 ± 0.098**	0.285 ± 0.174	0.985
			PortraitBooth	0.227	0.544	0.766	0.097 ± 0.100	**0.314 ± 0.176**	1.138
			ID-Booth	**0.226**	**0.529**	**0.734**	0.096 ± 0.099	0.312 ± 0.177	**1.139**
	SD-XL	Base	DreamBooth	0.002	0.002	0.002	0.040 ± 0.075	0.851 ± 0.078	56.614
			PortraitBooth	0.003	0.003	0.003	0.037 ± 0.075	0.856 ± 0.073	61.196
			ID-Booth	**0.001**	**0.001**	**0.002**	**0.037 ± 0.075**	**0.856 ± 0.066**	**67.493**
		Complex	DreamBooth	0.037	0.052	0.078	0.051 ± 0.077	0.629 ± 0.176	9.045
			PortraitBooth	0.035	0.047	0.072	0.050 ± 0.078	0.643 ± 0.173	9.715
			ID-Booth	**0.031**	**0.040**	**0.063**	**0.050 ± 0.078**	**0.648 ± 0.167**	**10.492**

(↓) – Lower is better; (↑) – Higher is better; (**Bold**) – Best result; (Underline) – Second best result

ID-Boothは、DreamBoothよりアイデンティティの一貫性が高く、PortraitBoothよりプロンプト制御が良好

合成データでの訓練によって、汎化性能がある

Training setting			Val. ↑	Verification accuracy on benchmarks ↑				
Data from	Prompt	Fine-tuning	LFW	AgeDB-30	CA-LFW	CFP-FP	CP-LFW	Average
TFD	–	–	0.672	0.501	0.548	0.598	0.542	0.547 ± 0.034
SD-2.1	Base	DreamBooth	**0.665**	**0.525**	0.539	0.572	0.542	0.544 ± 0.017
		PortraitBooth	0.664	0.507	0.532	**0.602**	0.548	0.547 ± 0.035
		ID-Booth	0.664	0.509	**0.541**	0.579	**0.565**	**0.548 ± 0.027**
	Complex	DreamBooth	0.681	0.499	**0.553**	0.591	**0.565**	0.552 ± 0.034
		PortraitBooth	**0.682**	0.492	0.551	**0.615**	0.552	**0.553 ± 0.043**
		ID-Booth	0.668	**0.500**	0.537	0.602	0.561	0.550 ± 0.037
SD-XL	Base	DreamBooth	0.674	0.515	0.550	0.582	0.540	0.547 ± 0.024
		PortraitBooth	0.679	0.491	**0.558**	0.611	**0.555**	0.554 ± 0.042
		ID-Booth	**0.688**	**0.529**	0.550	**0.611**	0.539	**0.557 ± 0.032**
	Complex	DreamBooth	**0.745**	0.496	0.579	**0.615**	**0.579**	0.567 ± 0.044
		PortraitBooth	0.726	0.507	0.584	0.608	0.569	0.567 ± 0.037
		ID-Booth	0.732	**0.532**	**0.599**	0.605	0.567	**0.575 ± 0.029**

(↑) – Higher is better; (**Bold**) – Best result; (Underline) – Second best result

本物のTFDデータを使うよりも、合成データで学習したモデルの方が高精度を達成
→合成データは、制約の少ない多様な環境の画像を含むため、より汎化性能が高い可能性がある。

目次

ID損失関数の追加

目標：個人の特徴を維持することで、自然な表情変換を達成

1. DiffusionAutoencodersの**線形分類器の改善**案を調査

→ 表情認識精度が向上

2. DiffusionAutoencodersの損失関数に**アイデンティティ損失**を追加
（ $\mathcal{L}_{simple} + \mathcal{L}_{PR} + \mathcal{L}_{TID}$ ）

→ 2次元上でのアイデンティティを保持した表情変換が可能

要検討：2段階で学習するイメージ

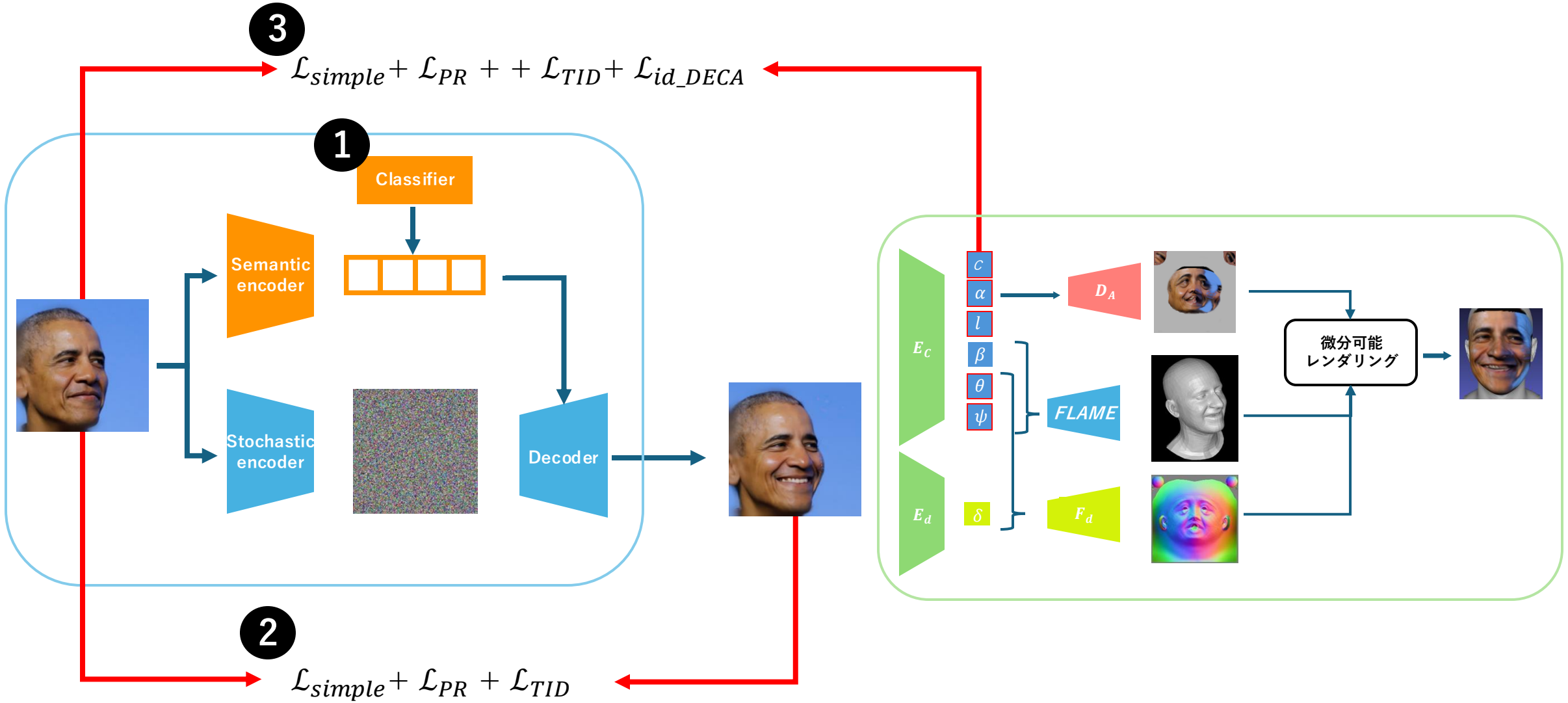
3. DiffusionAutoencodersの損失関数に**DECA特化アイデンティティ**損失を追加
（ $\mathcal{L}_{simple} + \mathcal{L}_{PR} + + \mathcal{L}_{TID} + \mathcal{L}_{id\_DECA}$ ）

→ 3次元上でのアイデンティティを保持した表情変換が可能

[1] YAO FENG et al. Learning an Animatable Detailed 3D Face Model from In-The-Wild Images
[2] Konpat Preechakul et al. Diffusion Autoencoders: Toward a Meaningful and Decodable Representation

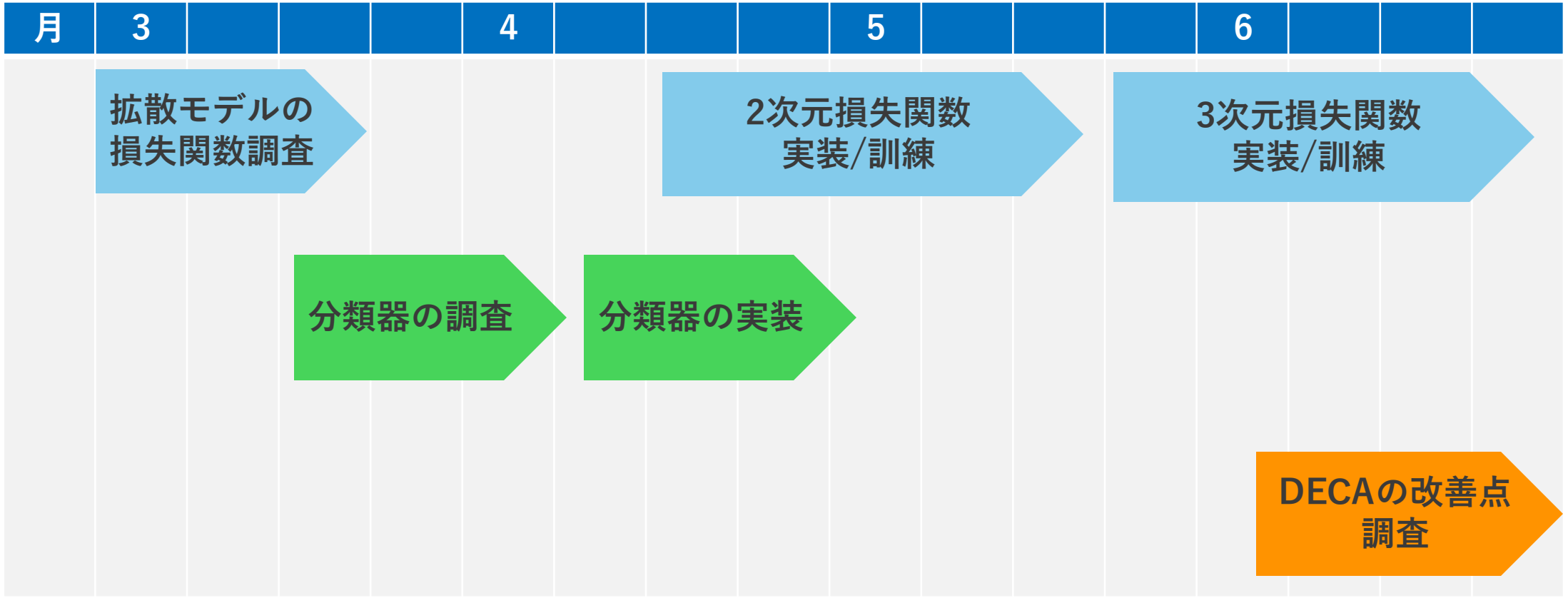
提案アーキテクチャー図

今までのアーキテクチャ図から3点の変更



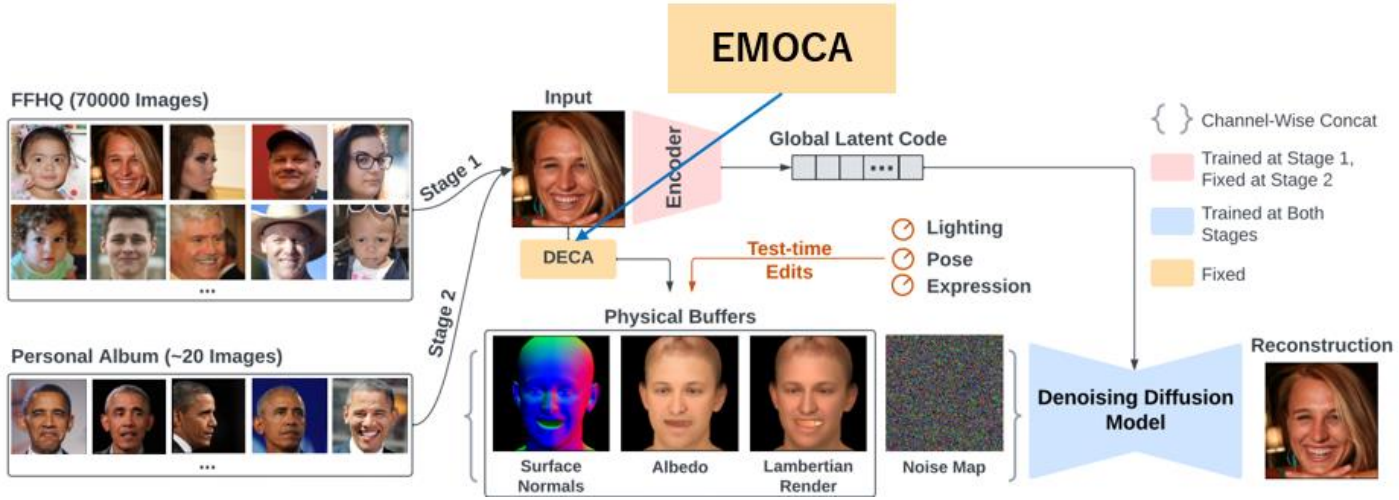
今後の研究計画

今回発表した改善点を随時実装



目次

新たな評価用画像を用いたEmoNet評価を実施中



やったことリスト

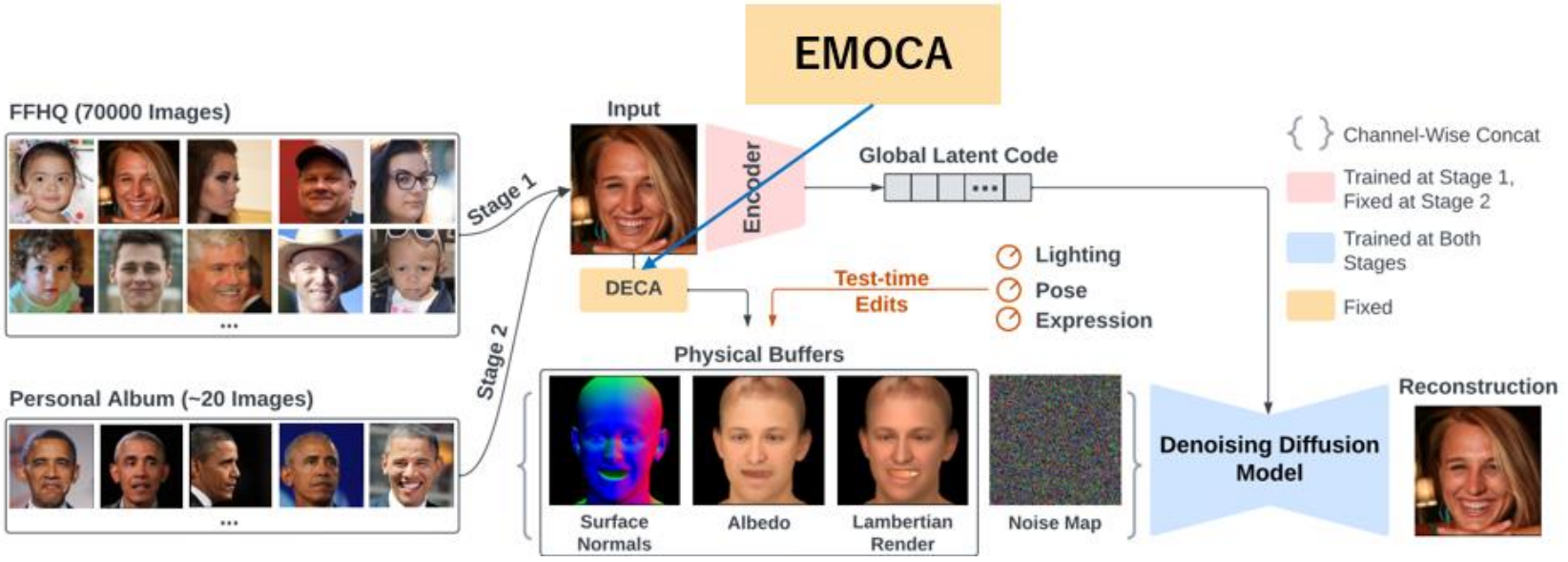
- V/A出力モデル(EmoNet, MaxVit)をローカルで実装
- EmoNetを用いたV/A評価
- 評価データセットの拡張

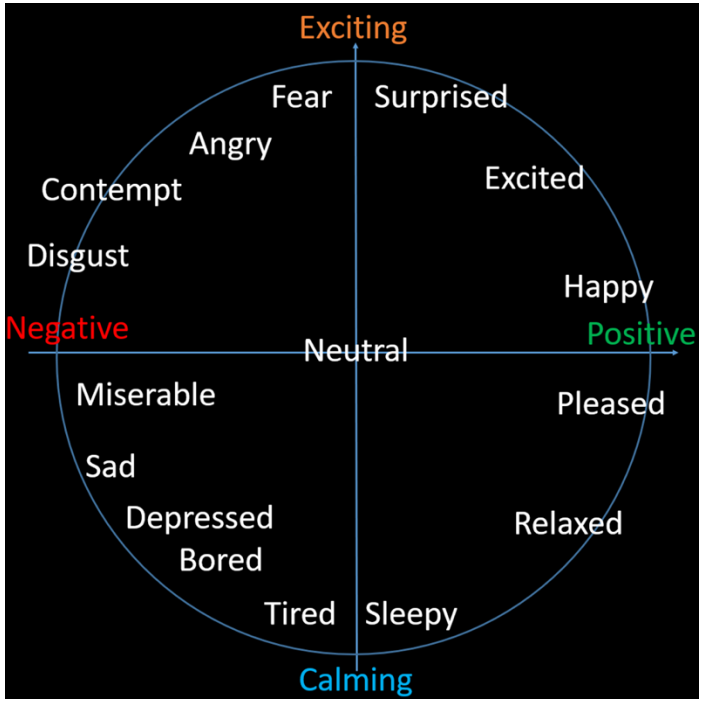
やることリスト

- 複数の指標を使用した評価
  - V/Aの出力
  - アイデンティティの保持（野口）
- 感情認識用モデルの調査
- 編集を行う人物や表情に対して得意・不得意があるかを調査

評価対象の表情変換モデル

データセット	Stage1 stage2	resnet	target expパラ	source expパラ	評価対象 モデル
FFHQ or AffectNet	DECA or EMOCA	18 or 50	DECA or EMOCA	DECA or EMOCA	32種類





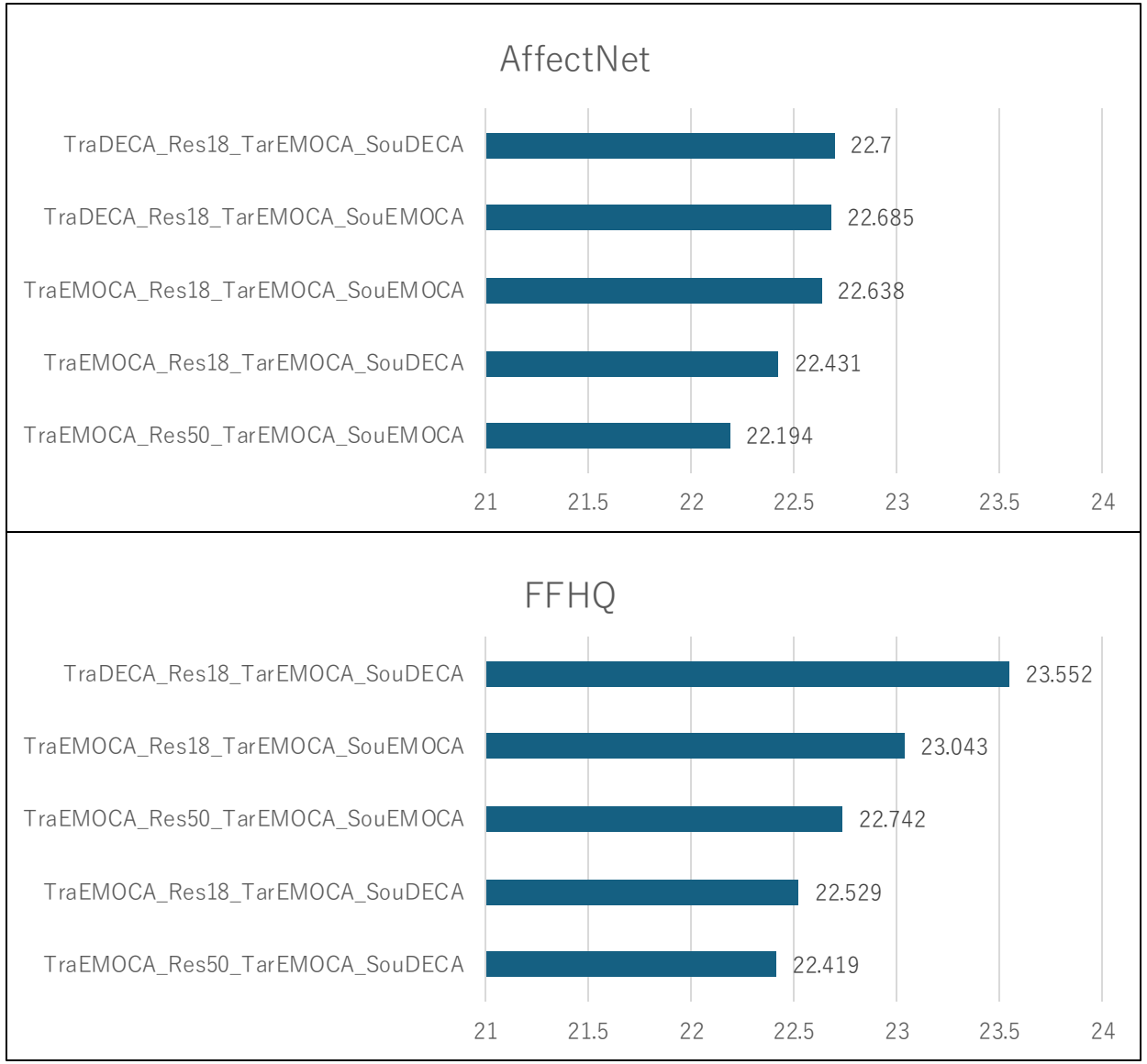
Source画像とTarget画像間のArousal

評価実験概要

- 訓練データセットとしてAffectNetと従来のFFHQを用いた場合、それぞれ16種類のモデルを比較
- オバマ大統領の表情を転送した場合の、ターゲット画像とソース画像のArousalの差を計算
- ターゲット画像の表情をうまく転送できていれば、Arousalは同値になるはず

考察

- AffectNetで訓練した方がArousalの差が小さい
- 訓練時の物理条件を付与するモデルとしてEMOCAの方が適している
- 顔以外の特徴を保持するResNetは18と50は訓練時間とのトレードオフで選択するべき
- 推論時において、ターゲット人物の特徴を抽出するモデルはEMOCA、ソース人物の特徴を抽出するモデルはDECAを選択するべき



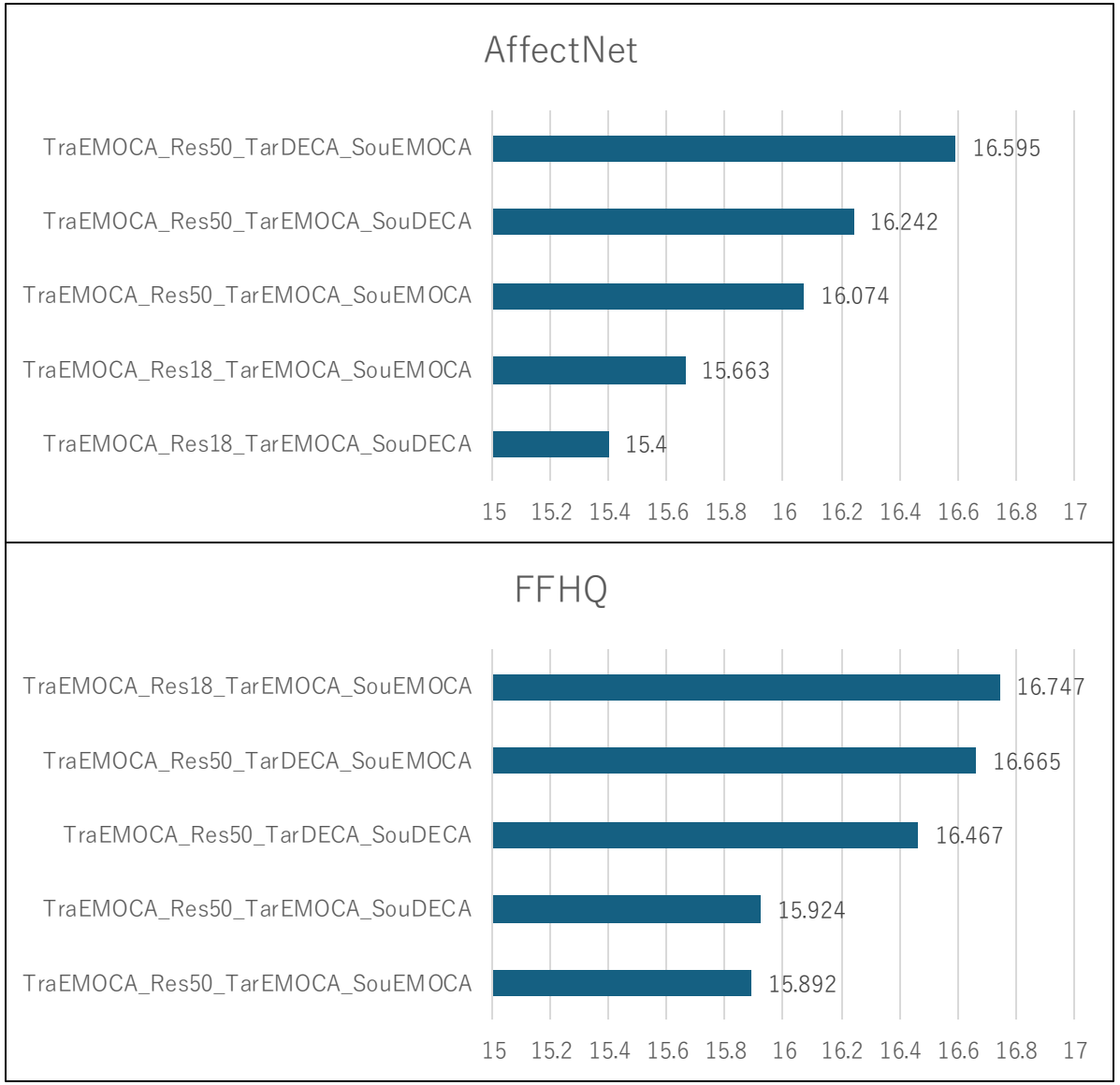
Source画像とTarget画像間のValence

評価実験概要

- 訓練データセットとしてAffectNetと従来のFFHQを用いた場合、それぞれ16種類のモデルを比較
- オバマ大統領の表情を転送した場合の、ターゲット画像とソース画像のArousalの差を計算
- ターゲット画像の表情をうまく転送できていれば、Valenceは同値になるはず

考察

- AffectNetで訓練した方がArousalの差が小さい
- 訓練時の物理条件を付与するモデルとしてEMOCAの方が適している
- 顔以外の特徴を保持するResNetは、18の方が性能が向上している場合が多い
- 推論時において、ターゲット人物の特徴を抽出するモデルはEMOCA、ソース人物の特徴を抽出するモデルはDECAを選択するべき
- Valenceの方がArousalより差が小さい

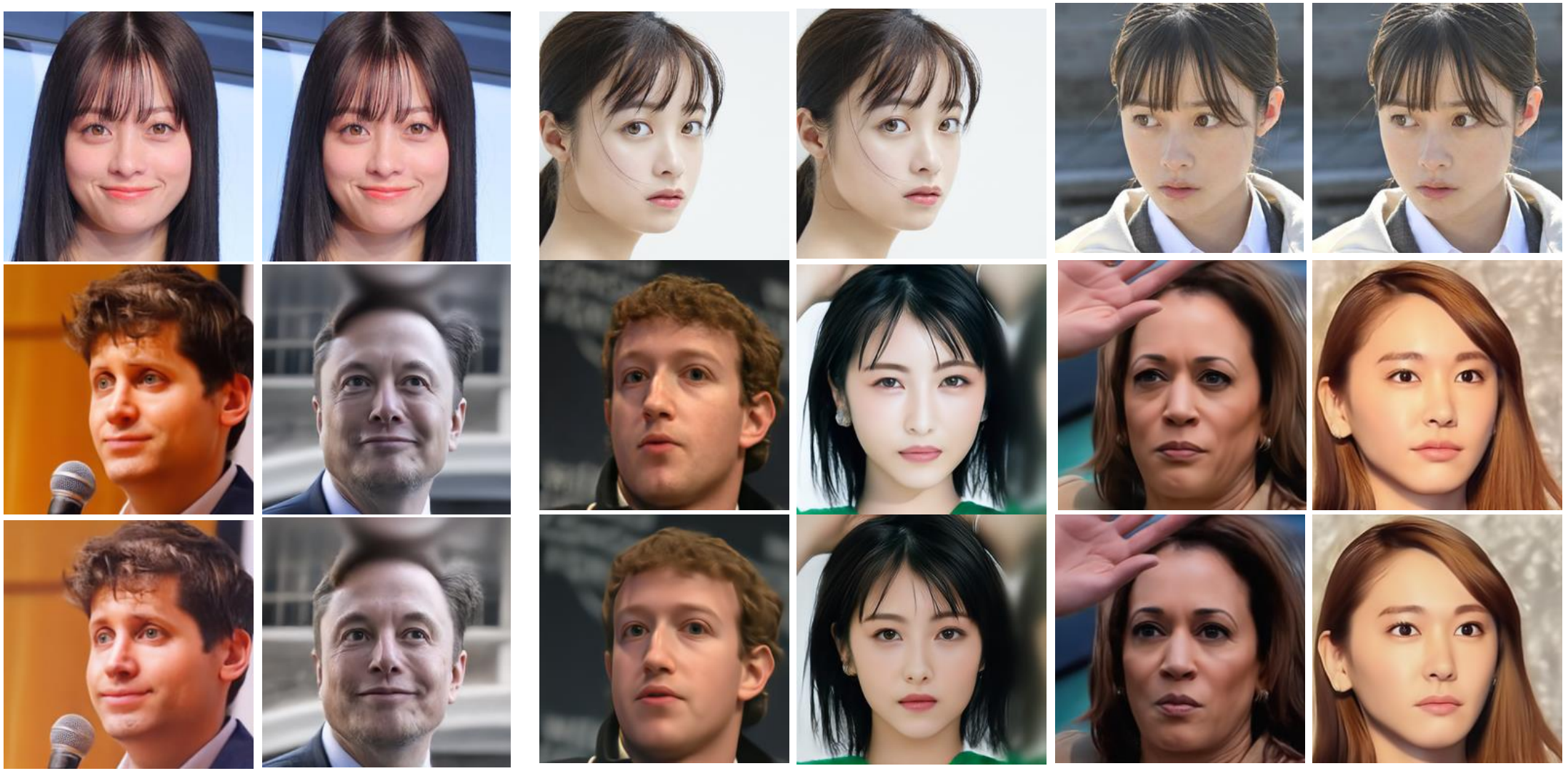


異なる人種間での編集はどうか？

target

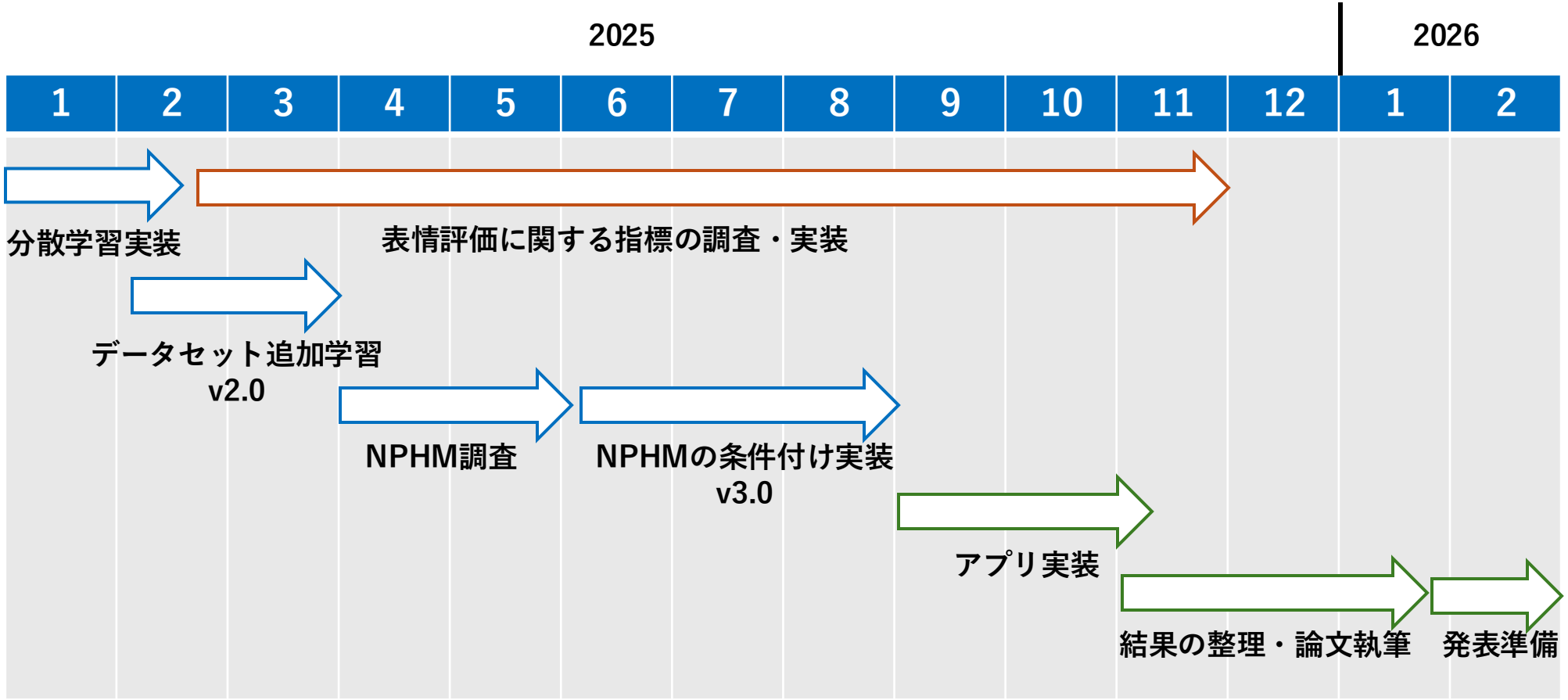
DiffusionRig

DiffusionRig Emo



- 主観的には人種や男女によって表情の再現性に変化はなさそうだが、V/A評価ではどうか？
- DiffusionRigに再現が不得意な表情はあるか？DiffusionRig-EMOにより改善されるか？

今後の研究計画



https://github.com/kdhht2334/awesome-SOTA-FER?tab=readme-ov-file#affect