Appendix for "Preserving Fairness Generalization in Deepfake Detection"

A. Related Work

Deepfake Detection. Current deepfake detection methods can be categorized into three primary groups based on the features they employ. The first category hinges on identifying inconsistencies in the *physical and physiological* characteristics of deepfakes. For example, inconsistent corneal specular highlights [67], the irregularity of pupil shapes [68, 69], eye blinking patterns [70], eye color difference [71], facial landmark locations [72], etc. The second category concentrates on *signal-level* artifacts introduced during the synthesis process, especially those from the frequency domain [73]. These methods encompass various techniques, such as examining disparities in the frequency spectrum [74, 75], utilizing checkerboard artifacts introduced by the transposed convolutional operator [76, 77]. However, the methods from the above two categories usually exhibit relatively low detection performance. Therefore, the largest portion of existing detection methods fall into the *data-driven* category, including [7–13]. These methods leverage various types of Deep Neural Networks (DNNs) trained on both authentic and deepfake videos to capture specific discernible artifacts. While these methods have achieved promising performance for the intra-domain evaluation, their performance sharply degrades during cross-domain testing.

Generalization in Deepfake Detection. To address the generalization issue, disentanglement learning [32] is widely used to extract the forgery-related features while getting rid of forgery-irrelated features for detection. For example, Hu et al. [14] propose a disentanglement framework to automatically locate the forgery-related region for detection. Based on this framework, Zhang et al. [15] add auxiliary supervision to improve the generalization ability. To enhance the independence of disentangled features, Liang et al. [16] propose a new framework by introducing content consistency constraints and global representation contrastive constraints. Such framework is later extended [17] by exclusively utilizing common forgery features, which are extracted separately from forgery-related features for detection.

Fairness in Deepfake Detection. Recent studies have delved into fairness concerns within the domain of deepfake detection [30]. Trinh et al. [26] examined biases in existing deepfake datasets and detection models across protected subgroups. They found a large error rate difference among subgroups, consistent with similar observations in the study [31]. Pu et al. [33] assessed the reliability of the deepfake detection model MesoInception-4 on FF++ and revealed its overall unfairness toward both genders. A more comprehensive analysis of deepfake detection bias, encompassing both demographic and non-demographic attributes, was presented by Xu et al. [27]. The authors significantly enriched five widely used deepfake detection datasets with diverse annotations to facilitate future research in this area. Furthermore, [29] highlighted substantial bias in both datasets and detection models. In an effort to mitigate performance bias across genders, they introduced a gender-balanced dataset. However, this approach yielded only modest improvements and required extensive data annotation efforts. More recently, Ju et al. [6] enhance fairness in testing scenarios within the same data domain, they do not maintain fairness when applied to cross-domain testing, which is the central focus of this paper.

B. Fairness Metrics

We assume a test set comprising indices $\{1, \ldots, n\}$. Y_j and $\hat{Y}_j$ respectively represent the true and predicted labels of the sample X_j. Their values are binary, where 0 means real and 1 means fake. For all fairness metrics, a lower value means better performance.

$$
\begin{aligned}
F_{FPR} &:= \sum_{\mathcal{J}_j \in \mathcal{J}} \left| \frac{\sum_{j=1}^{n} \mathbb{I}_{[\hat{Y}_j=1, D_j=\mathcal{J}_j, Y_j=0]}}{\sum_{j=1}^{n} \mathbb{I}_{[D_j=\mathcal{J}_j, Y_j=0]}} - \frac{\sum_{j=1}^{n} \mathbb{I}_{[\hat{Y}_j=1, Y_j=0]}}{\sum_{j=1}^{n} \mathbb{I}_{[Y_j=0]}} \right|, \\
F_{OAE} &:= \max_{\mathcal{J}_j \in \mathcal{J}} \left\{ \frac{\sum_{j=1}^{n} \mathbb{I}_{[\hat{Y}_j=Y_j, D_j=\mathcal{J}_j]}}{\sum_{j=1}^{n} \mathbb{I}_{[D_j=\mathcal{J}_j]}} - \min_{\mathcal{J}_j' \in \mathcal{J}} \frac{\sum_{j=1}^{n} \mathbb{I}_{[\hat{Y}_j=Y_j, D_j=\mathcal{J}_j']}}{\sum_{j=1}^{n} \mathbb{I}_{[D_j=\mathcal{J}_j']}} \right\}, \\
F_{DP} &:= \max_{k \in \{0,1\}} \left\{ \max_{J_j \in \mathcal{J}} \frac{\sum_{j=1}^{n} \mathbb{I}_{[\hat{Y}_j=k, D_j=J_j]}}{\sum_{j=1}^{n} \mathbb{I}_{[D_j=J_j]}} - \min_{J_j' \in \mathcal{J}} \frac{\sum_{j=1}^{n} \mathbb{I}_{[\hat{Y}_j=k, D_j=J_j']}}{\sum_{j=1}^{n} \mathbb{I}_{[D_j=J_j']}} \right\}, \\
F_{MEO} &:= \max_{k,k' \in \{0,1\}} \left\{ \max_{J_j \in \mathcal{J}} \frac{\sum_{j=1}^{n} \mathbb{I}_{[\hat{Y}_j=k, Y_j=k', D_j=J_j]}}{\sum_{j=1}^{n} \mathbb{I}_{[D_j=J_j, Y_j=k]}} - \min_{J_j' \in \mathcal{J}} \frac{\sum_{j=1}^{n} \mathbb{I}_{[\hat{Y}_j=k, Y_j=k', D_j=J_j']}}{\sum_{j=1}^{n} \mathbb{I}_{[D_j=J_j', Y_j=k]}} \right\}.
\end{aligned}
$$

Where D is the demographic variable, $\mathcal{J}$ is the set of subgroups with each subgroup $\mathcal{J}_j \in \mathcal{J}$. F_{FPR} meatures the disparity in False Positive Rate (FPR) across different groups compared to the overall population. F_{OAE} meatures the maximum ACC

gap across all demographic groups. F_{DP} measures the maximum difference in prediction rates across all demographic groups. And F_{MEO} captures the largest disparity in prediction outcomes (either positive or negative) when comparing different demographic groups.

C. The Network Details

Encoder. The architecture details of the encoder in our proposed method are presented in Fig. C.1. An image pair, comprising one fake and one real image, serves as the input, which is subsequently processed by an encoder built upon the Xception [63] backbone.

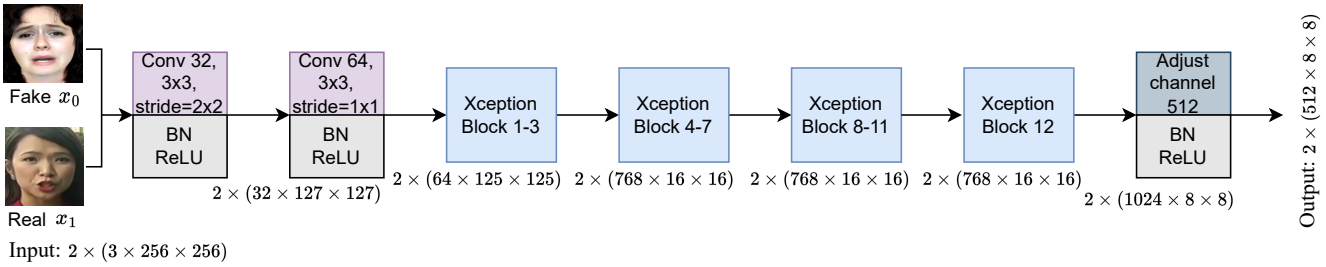

Figure C.1. The architecture details of the encoder in our proposed method.

Decoder. We further present the architecture details of the decoder in Fig. C.2, which reconstructs images in our proposed method to preserve the integrity of the extracted features. The demographic features d_0 and the content features C_0 are extracted from encoder, while f_0^a and f_0^g represent the domain-specific features and domain-agnostic features, respectively. The decoder reconstructs an image by utilizing those features separated by our disentanglement learning module as input, and passes through a series of upsampling and convolutional layers (Up-Block). AdaIN [49] is applied here for improving reconstructing and decoding. We present more visualizations of reconstruction images in different training epochs. We observe that, as the training progresses, the model learns to capture more detail features (*e.g.*, facial characteristics). This further validates our decoder successfully preserves the completeness of the extracted features.

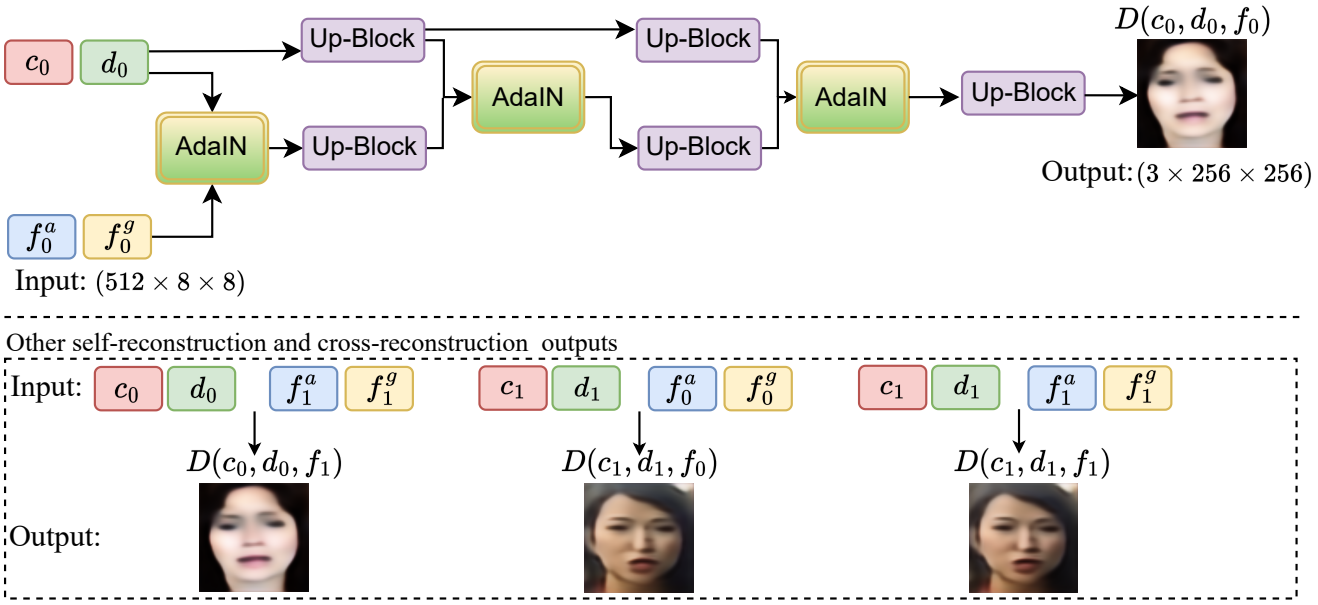

Figure C.2. The architecture details of the decoder in our proposed method.

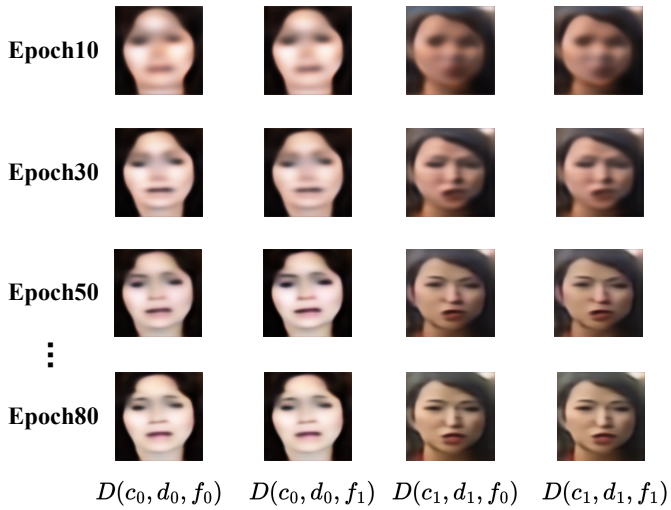

Figure C.3. Visualization of the reconstruction images during the training process.

D. End-to-end Training Algorithm

Below is the pseudocode of our joint optimization, which integrates a loss flattening strategy based on sharpness-aware minimization [39], and is implemented throughout the end-to-end training process.

Algorithm 1: Joint Optimization

Input: A training dataset $\mathcal{S}$ with demographic variable D, a set of subgroups $\mathcal{J}$, α, α', max_iterations, num_batch, learning rate β
Output: A deepfake detection model with fairness generalizability
Initialization: θ_0, $l = 0$
for $e = 1$ *to* max_iterations **do**
  for $b = 1$ *to* num_batch **do**
    Sample a mini-batch $\mathcal{S}_b$ from $\mathcal{S}$
    Compute sample loss of $(C(h(I_i), Y_i)), \forall (I_i, Y_i) \in \mathcal{S}_b$
    For each $j \in \{1, ..., |\mathcal{J}|\}$, set η_j^* to be the value of η_j that minimizes L_j as given in (2b). This minimization is solved using binary search.
    Set $L_j(\theta) \leftarrow L_j(\theta, \eta_j^*)$ using (2b), $\forall j$
    Using binary search to find η that minimizes (2a)
    Compute ϵ^* based on Eq. (3)
    Compute gradient approximation for (4)
    Update θ: $\theta_{l+1} \leftarrow \theta_l - \beta \nabla_\theta \mathcal{L}\big|_{\theta_l + \epsilon^*}$
    $l \leftarrow l + 1$
  end
end
return θ_l

E. Additional Experimental Settings

We show the total number of train, validation and test samples of each dataset and the attributes included in our experiment in Table E.1. We only use FF++ for training and validation.

Dataset	Samples			Intersection Sensitive Attributes
	Train	Validation	Test	
FF++	76,139	25,386	25,401	M-A, M-B, M-W, M-O, F-A, F-B, F-W, F-O
DFD	-	-	9,385	M-B, M-W, M-O, F-B, F-W, F-O
DFDC	-	-	22,857	M-A, M-B, M-W, M-O, F-A, F-B, F-W, F-O
Celeb-DF	-	-	28,458	M-B, M-W, M-O, F-B, F-W, F-O

Table E.1. Test sample number and Intersection attributes in each dataset. '-' means not used.

F. Additional Experimental Results

Stability Evaluation. The stability comparison of DAW-FDD with ours over 5 random runs is shown in Table F.1. Our method shows superior fairness and detection mean score out of 5 random runs compared to DAW-FDD. This suggests that our approach has a robust and formidable capacity to improve fairness.

Effect of Trade-off λ. To validate the effect of the trade-off hyperparameter in Eq. 3, we conduct sensitivity analysis on FF++ dataset. Fig. F.1 shows the fairness metrics and detection metric AUC to different λ values. Experiment results demonstrate that the model attains optimal fairness performance when λ is configured to 1.0 and also keeps fair AUC score. Notably, the analysis uncovers a trade-off between fairness and AUC score: as λ ranges from 0.4 to 0.8, there is an enhancement in AUC while the fairness (F_{DP}, F_{MEO}, and F_{OAE}) becomes worse. However, when λ changes from 0.8 to 1.0, we can see the opposite effect: AUC decreases while the fairness improves. Specifically, the behavior of F_{FPR} diverges from that of the other fairness metrics. This is because a higher AUC typically reflects an optimal balance between maximizing the TPR and minimizing the FPR. As a result, at a λ of 0.8, a lower F_{FPR} is accompanied by a higher AUC. To more clearly show the relationship between each fairness metric and AUC, we present these dynamics separately in Fig. F.2, which illustrates the trend where gains in AUC correspond to diminished fairness.

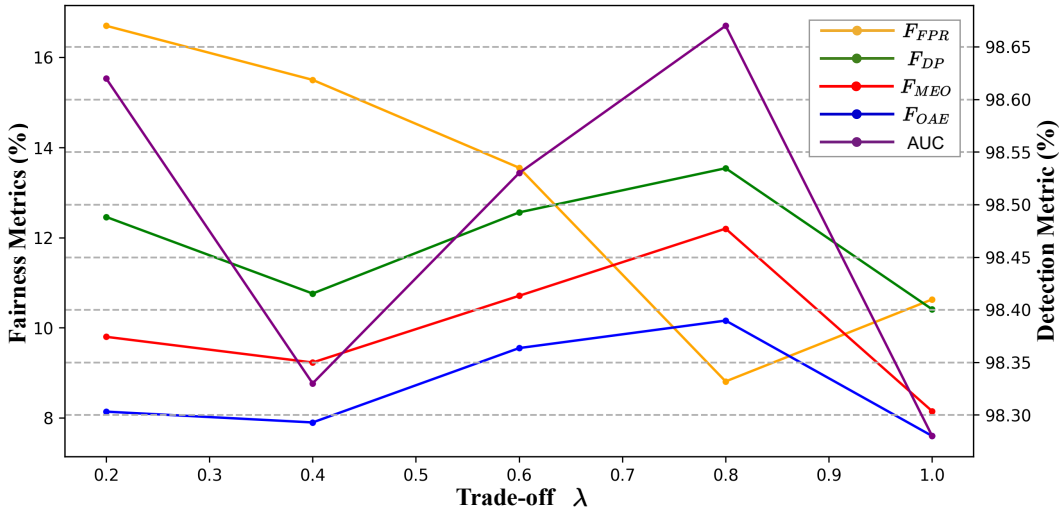

Figure F.1. Sensitivity analysis of parameter λ on the trade-off between fairness and detection accuracy on FF++.

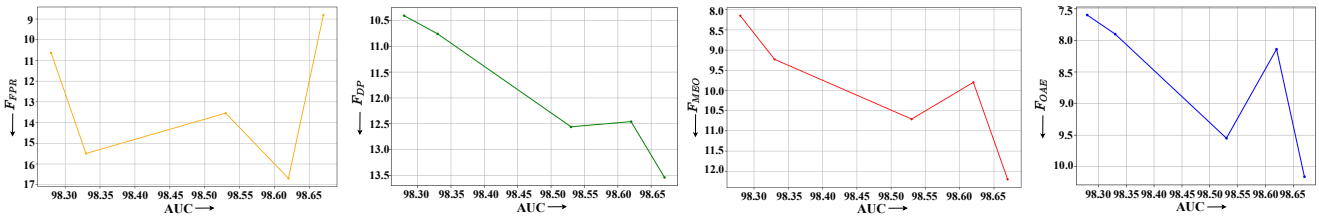

Figure F.2. Trends in Fairness Metrics vs. AUC Score. From left to right, the graphs show how F_{FPR}, F_{DP}, F_{MEO}, and F_{OAE} change with AUC, illustrating the trade-off between accuracy and fairness.

Comparison of the Loss Convergence. In Fig. F.3, we present a comparison of training loss convergence between our method and DAW-FDD, both utilizing Xception as the backbone on the FF++ dataset. It is evident that while DAW-FDD exhibits fluctuating convergence, our method demonstrates a more stable and consistent reduction in training loss. This stability indicates potential advantages in the robustness and reliability of our approach during the training process.

Comparison of AUC on Intersectional Subgroups. We further show the AUC comparison results on FF++, DFDC, DFD, and Celeb-DF datasets with detailed performance in subgroups in Fig. F.4. Our method evidently improves the AUC of each subgroup and narrows the disparity between subgroups. Notably, in DFD and Celeb-DF, the AUC difference between subgroups is much lower than DAW-FDD's.

Method	FF++					DFDC					Celeb-DF					DFD				
	Fairness Metrics(%)↓				Detection Metric(%)↑	Fairness Metrics(%)↓				Detection Metric(%)↑	Fairness Metrics(%)↓				Detection Metric(%)↑	Fairness Metrics(%)↓				Detection Metric(%)↑
	F_{FPR}	F_{MEO}	F_{DP}	F_{OAE}	AUC	F_{FPR}	F_{MEO}	F_{DP}	F_{OAE}	AUC	F_{FPR}	F_{MEO}	F_{DP}	F_{OAE}	AUC	F_{FPR}	F_{MEO}	F_{DP}	F_{OAE}	AUC
DAW-FDD	15.81 (1.62)	11.19 (2.48)	12.57 (2.15)	9.66 (2.11)	97.54 (0.23)	44.97 (1.62)	35.07 (2.23)	16.19 (2.03)	18.59 (3.24)	60.28 (1.11)	21.32 (4.63)	19.96 (5.34)	16.17 (7.01)	49.44 (8.43)	69.97 (0.84)	34.69 (1.75)	29.36 (1.77)	18.59 (2.64)	12.05 (1.38)	73.54 (2.45)
Ours	**11.70** (1.89)	**10.40** (1.96)	**11.93** (1.46)	**8.73** (1.38)	**98.17** (0.28)	**39.22** (4.04)	**35.03** (1.83)	**10.10** (0.92)	**17.10** (2.37)	**61.84** (0.66)	**10.93** (4.79)	**12.58** (2.56)	**13.52** (4.12)	**34.05** (7.37)	**75.23** (1.81)	**27.14** (0.94)	**22.86** (1.52)	**17.58** (4.36)	**8.38** (0.89)	**82.79** (2.50)

Table F.1. Detection mean and standard deviation (in parentheses) on intra-domain and cross-domain testing sets across 5 experimental repeats. Each method is trained only on FF++.

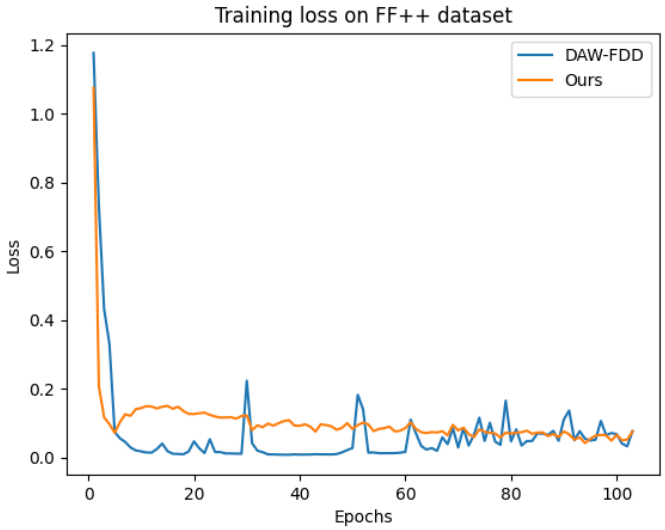

Figure F.3. Training loss convergence.

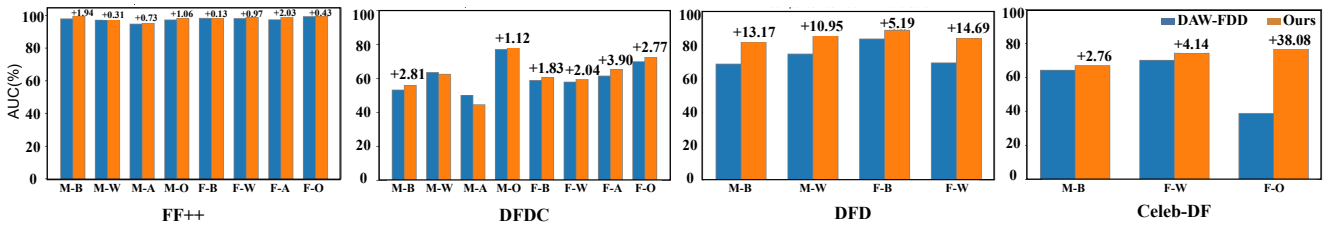

Figure F.4. AUC comparison of DAW-FDD and Ours on the Intersectional subgroups. The subgroups not represented in DFD and Celeb-DF are inapplicable.

Comparison on Cross-demographic Subgroup. DAW-FDD and our model are trained on FF++ with Intersection demographic information, tested on Celeb-DF and DFD, we report the fairness performance on the Race subgroup. The results shown in Fig. F.5 clearly demonstrate that our method exhibits substantial improvements on F_{FPR}, F_{MEO}, and F_{OAE} fairness metrics, particularly noticeable on the F_{FPR} and F_{MEO} in DFD. This suggests that our approach can maintain fairness generalization ability among different demographic subgroups.

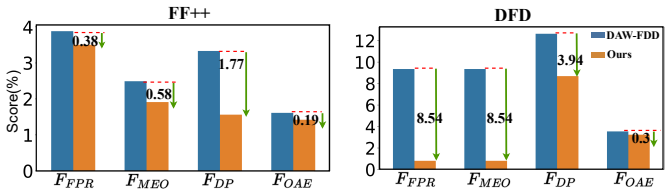

Figure F.5. Comparison of fairness performance on Race subgroup (cross-domain and cross-subgroup). Models are trained on FF++ using Intersection attribute, tested on Celeb-DF and DFD under Race subgroup.

Visualization. **1)** Detailed feature visualization of our disentangled forgery features and demographic features are presented in

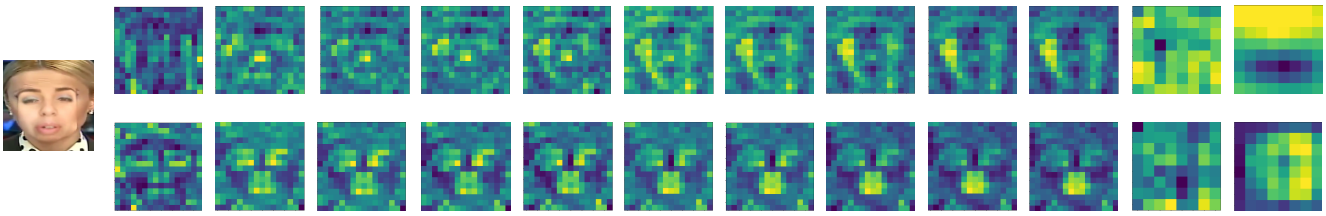

Figure F.6. More visualization of our disentangled forgery features (first row) and demographic features (second row) from our method on FF++.

Figure F.7. The UMAP [78] visualization of demographic features extracted from our method on FF++.

Fig. F.6. From left to right, the visualization demonstrates how our network builds up its understanding from original image. **2)** In addition, we show the UMAP [78] visualization of demographic features extracted from our method on FF++ in Fig. F.7. In the visualization, images with different intersectional demographic attributes locate separately in the latent space, which reveals that our model's capability to distinguish and disentangle features from different demographic backgrounds effectively. The result also aligns with demographic feature visualization in Fig. F.6, that our model actually captures demographic features for fair learning. The UMAP result further shows that the majority of subgroups in FF++ are Male-White and Female-White, the bias in the dataset makes it challenging for fair detection, suggesting the necessity of the demographic distribution-aware margin loss [47] we apply in our method for improving generalization for minority subgroups.