Supplementary Materials

1. Further Results Comparing with SoTA

1.1. Results with Protocol 3

Method	FF++	DFDCP	DFD	CDF	Avg.
LwF [22]	67.34	67.43	84.05	87.90	76.68
CoReD [19]	74.08	76.59	93.41	80.78	81.22
DFIL [30]	86.28	79.53	92.36	83.81	85.49
DMP [41]	**91.61**	84.86	91.81	91.67	89.99
Ours	90.89	**89.33**	**93.97**	**94.34**	**92.13**

Table 4. Performance comparisons (ACC) with Protocol 3. All results of previous methods are copied from [41] and [30].

In Tab. 4, we copy the results after all tasks are incremented with P3 from their official papers [30, 41] to further compare the IFFD performance. Despite the notable distinction in experimental settings among these methods, our method still exhibits superior performance.

1.2. Evaluation with Forgetting Rate

Following [25], we compute FR based on AUC between current and first-learned models. Specifically, FR is calculated as $FR = 1 - \frac{AUC_{last}}{AUC_{first}}$, where AUC_{last} is the AUC of one dataset tested on the currently-trained model, AUC_{first} is the AUC of the model that firstly-introduced the dataset. The FR results in Tab. 5 indicate that our method has effectively tackled the issue of forgetting.

2. Further Visualization Analysis

2.1. Visualization of Model Attention via Grad-CAM

As shown in Fig 6, we deploy Grad-CAM [34] to generate saliency maps. It can be observed that our method could explore more forgery clues since we successfully accumulated forgery information. While DFIL struggles to find rich clues and cannot consistently focus on the forgery regions.

Method	SDv21	FF++	DFDCP	Avg.
Lower Bound	47.19	32.75	16.40	32.11
LwF	38.45	14.20	0.41	17.69
CoReD	12.80	11.21	2.05	8.69
DFIL	6.72	20.69	11.80	13.07
HDP	9.43	14.68	3.25	9.12
Ours	**0.28**	**10.06**	**0.94**	**3.09**

Table 5. Evaluation of Forgetting Rate ↓ (%).

2.2. Visualization of Actual Feature Distribution with Toy Models

To further investigate the learned feature distribution in IFFD, we cleverly craft toy models to visualize the **actual** feature distributions of baseline (DFIL [30]) and our method. To be specific, we train new models with features that have only two dimensions and all other settings are consistent with the standard ones. Consequently, we could directly visualize the two-dimensional features with a two-dimensional coordinate system. As shown in Fig. 7, the Baseline performs limited in distinguishing various forgeries and detecting binary Real/Fake, while our method could effectively isolate each domain and uphold a clean binary decision boundary. Notably, the two-dimensional features are insufficient to adequately represent the learned representations, resulting in the toy model performing poorly compared to the standard model. Nevertheless, it could still suggest that the actual feature distribution of the standard models is organized as we anticipated, that is, aligned feature isolation.

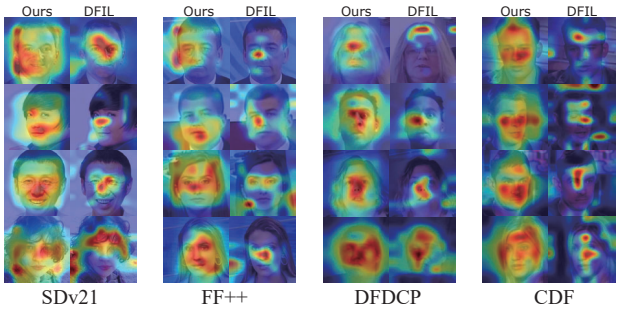

Figure 6. Saliency map visualization of DFIL [30] and the proposed method.

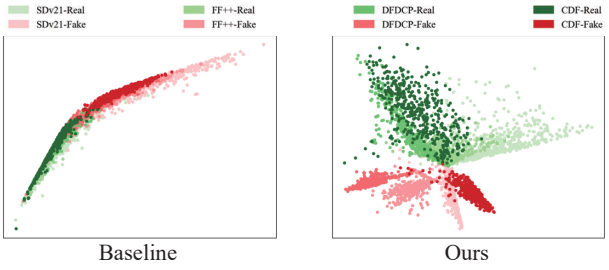

Figure 7. **Actual** two-dimensional feature distributions of toy models with Protocol 1.

3. Experiments of Generalization Ability

3.1. Generalization to Other Unseen Datasets

To validate that the accumulated forgery information enables our method to learn more about forgery generality, we

Method	DFD [10]	UniFace [42]	SDv15 [32]	FakeAVCeleb [17]	Avg.
Lower Bond	0.6705 / 0.7038	0.6058 / 0.6216	0.5319 / -	0.5841 / 0.5995	0.5981 / 0.6416
DFIL [30]	0.7719 / 0.8293	0.5637 / 0.6001	0.7786 / -	0.6111 / 0.6306	0.6813 / 0.6867
HDP [39]	0.8039 / 0.8441	0.5971 / 0.6714	0.7211 / -	0.6535 / 0.6917	0.6939 / 0.7357
Ours	**0.8225 / 0.8803**	**0.7269 / 0.7667**	**0.8110** / -	**0.7663 / 0.8304**	**0.7817 / 0.8258**

Table 6. Cross-dataset evaluations for generality with *frame-level* / *video-level* AUC. SDv15 has no video-level result since it is an image-level dataset. All methods are trained based on Protocol 1 (SDv21, FF++, DFDCP, CDF) and tested on other unseen datasets. The best results are highlighted in **bold**.

conduct cross-dataset experiments for generalization ability evaluation. As shown in Tab. 6, we apply the model trained on Protocol 1 to be evaluated on DeepFakeDetection (DFD) [10], UniFace [42] from DF40 [47], SDv15 from DiffusionFace [3], and FakeAVCeleb [17]. The experimental results substantially demonstrate that our method exhibits superior generalization ability attributable to the accumulated forgery information during incremental learning.

3.2. Generalization to Other Backbone

We additionally deployed our method on two mainstream backbones (ResNet and Xception) and compared the results with those of the original backbones under the same replay size. As shown in Tab. 7, our method also significantly improves the performance of these backbones.

4. Algorithm for Sparse Uniform Replay

As shown in Algorithm 1, we provide a concisely summarized algorithm for better comprehension in the detailed implementation of the proposed sparse uniform replay (SUR).

Algorithm 1: Sparse Uniform Replay (SUR)

Input: t-th Dataset: $\mathbf{X}^t_{all} = \{\mathbf{X}^t_{real}, \mathbf{X}^t_{fake}\}$;
Feature Extractor Trained on t-th Dataset: $\mathcal{E}^t$;
Replay size: n_r.
Initialize the t-th replay set $\mathbf{X}^t_{replay}$ as empty;
for $\mathbf{X}^t \sim \mathbf{X}^t_{all}$ **do**
  extract features of $\mathbf{X}^t$
  $\mathbf{F}^t = \mathcal{E}(\mathbf{X}^t)$
  calculate feature centroid
  $\mathbf{c}^t = avg(\mathbf{F}^t)$
  calculate magnitude matrix from $\mathbf{F}^t$ to $\mathbf{c}^t$
  $\mathbf{M}^t = \|\mathbf{F}^t - \mathbf{c}^t\|_2$
  calculate angularity matrix from $\mathbf{F}^t$ to $\mathbf{c}^t$
  $\mathbf{A}^t = \frac{(\mathbf{F}^t - \mathbf{c}^t)}{\|\mathbf{F}^t - \mathbf{c}^t\|_2}$
  rearrange $\mathbf{F}^t$ in ascending order based on $\mathbf{M}^t$
  divide $\mathbf{F}^t$ into $\frac{n_r}{2}$ equal-length segments
  $\mathbf{F}^t = \{\mathbf{F}^t_{1:\frac{2n}{n_r}}, \ldots, \mathbf{F}^t_{(n-\frac{2n}{n_r}):n}\}$
  for $\mathbf{F}^t_{seg} \sim \{\mathbf{F}^t_{1:\frac{2n}{n_r}}, \ldots, \mathbf{F}^t_{(n-\frac{2n}{n_r}):n}\}$ **do**
    calculate similarity of each feature $\mathbf{f}^t_i$ in $\mathbf{F}^t_{seg}$ with its shuffled $\tilde{\mathbf{f}}^t_i$ as stability score
    $s^t_i = \frac{\tilde{\mathbf{f}}^t_i \cdot (\mathbf{f}^t_i)^{\mathrm{T}}}{\|\tilde{\mathbf{f}}^t_i\|_2 \cdot \|\mathbf{f}^t_i\|_2}$
    store the $\mathbf{x}^t_m$ corresponding to $\mathbf{f}^t_m$ with largest s^t_m into $\mathbf{X}^t_{replay}$
    calculate angularity similarity of each feature $\mathbf{f}^t_j$ in $\mathbf{F}^t_{seg}$ with $\mathbf{f}^t_m$ based on $\mathbf{A}^t$
    store the $\mathbf{x}^t_a$ corresponding to $\mathbf{f}^t_a$ with largest angularity similarity into $\mathbf{X}^t_{replay}$

Output: t-th replay set $\mathbf{X}^t_{replay}$.

5. Sensitivity Evaluation

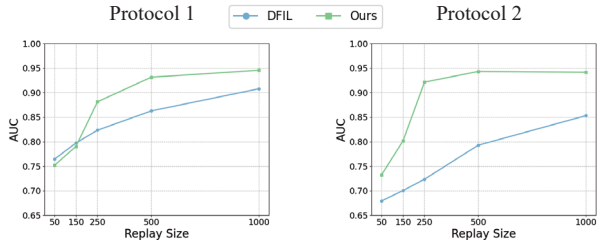

Figure 8. Sensitivity of replay size. The shown AUCs are the average values on four datasets after training with Protocol 1 or 2.

5.1. Effect of Replay Size

In Fig. 8, we examine the effect of the replay set size on model performance. It can be observed that the impact of replay set size on DFIL is relatively smooth, with performance gradually improving as the set size increases. In contrast, our method exhibits limited performance when the replay set size is small (*i.e.*, 50, 150). This is because the constraints employed for the proposed aligned feature isolation rely heavily on the replayed global distribution. Nonetheless, once the replay set reaches a more standard size, the performance of our approach becomes superior and promising.

Method	SDv21	FF++	DFDCP	CDF	Avg.
Xception+Ours	$0.996^{\uparrow 65.8\%}$	$0.767^{\uparrow 24.9\%}$	$0.852^{\uparrow 13.6\%}$	$0.951^{\uparrow 0.75\%}$	$0.892^{\uparrow 22.6\%}$
ResNet+Ours	$0.993^{\uparrow 85.8\%}$	$0.688^{\uparrow 16.0\%}$	$0.861^{\uparrow 20.0\%}$	$0.935^{\uparrow 0.73\%}$	$0.869^{\uparrow 25.4\%}$

Table 7. Generalization to other backbones (AUC). ↑ denotes the improvement compared with vanilla backbones.

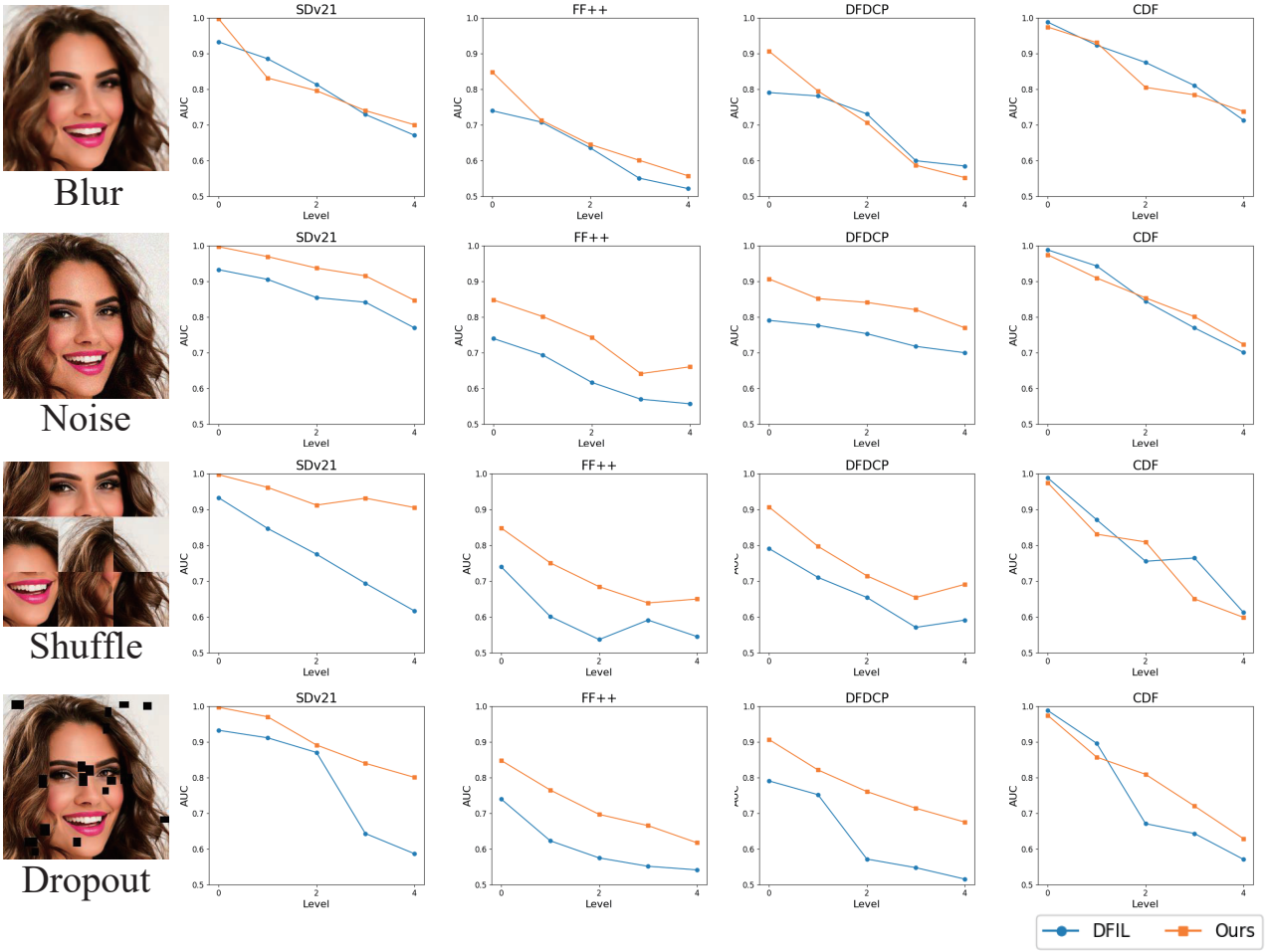

Figure 9. Robustness evaluations. The images in the first column are visualized illustrations of different types of applied perturbations. The models are trained based on Protocol 1.

5.2. Robustness against Unseen Perturbations

Considering the importance of robustness for real-world applications, we evaluate the robustness of different IFFD methods against unseen perturbations. Specifically, based on Protocol 1, we assess robustness against Block-wise Dropout (Dropout), Grid Shuffle (Shuffle), Gaussian Noise (Noise), and Median Blur (Blur), each applied at multiple intensity levels. As shown in Fig. 9, our method demonstrates consistent superiority in Noise, Shuffle, and Dropout, and also being comparable in Blur. The robustness superiority of our method may be attributed to the effective accumulation and utilization of forgery information achieved by our method, which enables the extracted and organized latent space to be more stable and representative.