Appendix

A. Experimental Settings

Datasets. The FaceForensics++ (FF++) dataset [49] consists of 1000 original video sequences sourced from `YouTube` and 1000 synthetic videos generated using four different manipulation techniques. The Google/Jigsaw DeepFake detection (DFD) [16] dataset consists of 3,068 deepfake videos generated based on 363 original videos of 28 consented individuals of various ages, genders, and ethnic groups. Pairs of actors were selected randomly and deep neural networks swapped the face of one actor onto the head of another. The specific deep neural network synthesis model is not disclosed, but manipulation masks are also provided in this dataset. DeeperForensics-1.0 [27] is a large scale dataset consisting of 11, 000 deepfake videos generated with high-quality collected data vary in identities, poses, expressions, emotions, lighting conditions, and 3DMM blendshapes. Real-world distortions such as compressions, color saturations, contrast changes, white gaussian noises, and local block-wise distortions, are heavily applied and mixed. Celeb-DF [37] is a new DeepFake dataset of celebrities with 408 real videos and 795 synthesized video with reduced visual artifacts (Table 1).

Pre-processing. We use a standard data pre-processing technique for deepfakes and videos, i.e. dumping the video frames to images and cropping out facial areas using facial boundaries and landmarks [72] instead of using the full-frame. When cropping faces, we use a face margin of 0.3-0.5 to get a full cropping of the actors' head. Consecutive are processed in OpenCV to calculate the dense optical flows and cropped similarly to the RGB frames.

Training details. During training, each input is formed by 1 RGB frame with 10 pre-computed optical flow fields starting from that RGB frame. Given the FPS of FF++, this registers a temporal signature of roughly 0.5s. We sample 270 frames from each FF++ training video [49] and utilized the standard normalization on the RGB frame, but not the flow frames. Besides this, we do not use any other form of video data augmentations. We used a pre-trained HRNet as the backbone of our architecture. However, given the shape of our input, we also need to perform *cross modality pre-training* to initialize the weights of the first conv. layer by averaging the weights across the RGB channels and replicate it corresponding to the number of input flow channels [66]. Our network parameters θ are trained using Adam, with a learning rate of $2e^{-4}$ for f_ω and $1e^{-3}$ for $\mathbf{p}_i$.

Validation details. During validation, we sample 100 frames from each testing video, and do the same across all unseen datasets. We combine the logits using an aggregation function (`sum` or `avg`). This can be understood as combining the *similarity evidence* between the learned prototypes and the testing video across clips in the video.

B. TQTL Semantics

For completeness, we detailed the semantics for TQTL for evaluating a specification, similar to the work by Dokhanchi et al. [14]. Consider the data stream $\mathcal{D}, i \in N$ is the index of current frame, $\pi \in P$, $\phi, \phi_1, \phi_2 \in TQTL$ and evaluation function $\epsilon : V_t \cup V_p \rightarrow \mathbb{N}$, which is the environment over the time and prototype variables. The quality value of formula ϕ with respect to $\mathcal{D}$ at frame i with evaluation ϵ is recursively assigned as follows:

$$
\begin{aligned}
[\![\top]\!](\mathcal{D}, i, \epsilon) &\coloneqq +\infty \\
[\![\pi]\!](\mathcal{D}, i, \epsilon) &\coloneqq [\![f_\pi(t_{1\ldots n}, p_{1\ldots n}) \sim c]\!](\mathcal{D}, i, \epsilon) \\
[\![x.\phi]\!](\mathcal{D}, i, \epsilon) &\coloneqq [\![\phi]\!](\mathcal{D}, i, \epsilon[x \Leftarrow i]) \\
[\![\exists p_i @ x, \phi]\!](\mathcal{D}, i, \epsilon) &\coloneqq \max_{k \in \mathcal{P}} ([\![\phi]\!](\mathcal{D}, i, \epsilon[p_i \Leftarrow k])) \\
[\![x \leq y + n]\!](\mathcal{D}, i, \epsilon) &\coloneqq \begin{cases} +\infty & if\ \epsilon(x) \leq \epsilon(y) + n \\ -\infty & otherwise \end{cases} \\
[\![\neg\phi]\!](\mathcal{D}, i, \epsilon) &\coloneqq -[\![\phi]\!](\mathcal{D}, i, \epsilon) \\
[\![\phi_1 \vee \phi_2]\!](\mathcal{D}, i, \epsilon) &\coloneqq \max([\![\phi_1]\!](\mathcal{D}, i, \epsilon), [\![\phi_2]\!](\mathcal{D}, i, \epsilon)) \\
[\![\phi_1 U \phi_2]\!](\mathcal{D}, i, \epsilon) &\coloneqq \max_{i \leq j} \Big(\min \big([\![\phi_2]\!](\mathcal{D}, j, \epsilon), \\
&\qquad \min_{i \leq k < j} [\![\phi_1]\!](\mathcal{D}, k, \epsilon) \big) \Big)
\end{aligned}
$$

We say that D satisfies ϕ ($\mathcal{D} \models \phi$) iff $[\![\phi]\!](\mathcal{D}, 0, \epsilon_0) > 0$, where ϵ_o is the initial environment. On the other hand, a data stream $\mathcal{D}'$ does not satisfy a TQTL formula ϕ ($\mathcal{D}' \not\models \phi$), iff $[\![\phi]\!](\mathcal{D}, 0, \epsilon_0) \leq 0$. The quantifier $\exists id @ x$ is the maximum operation on the quality values of formula $[\![\phi]\!]$ corresponding to the prototypes IDs at frame x.

C. GIFs

For the following visualizations and figures in the main paper, we have included the corresponding gifs in folders within the supplementary materials.

D. Additional Visualizations

D.1. More examples of prototypes and classes of temporal artifacts learned.

D.2. More examples of how `DPNet` classify a video.

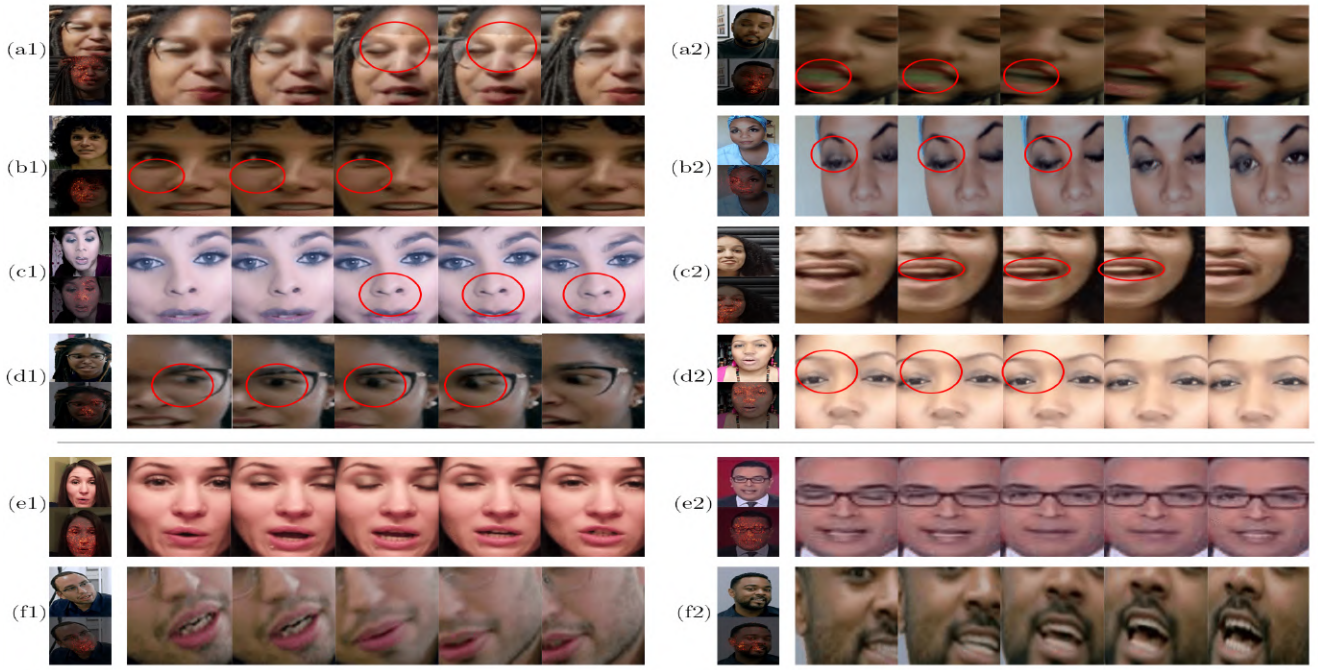
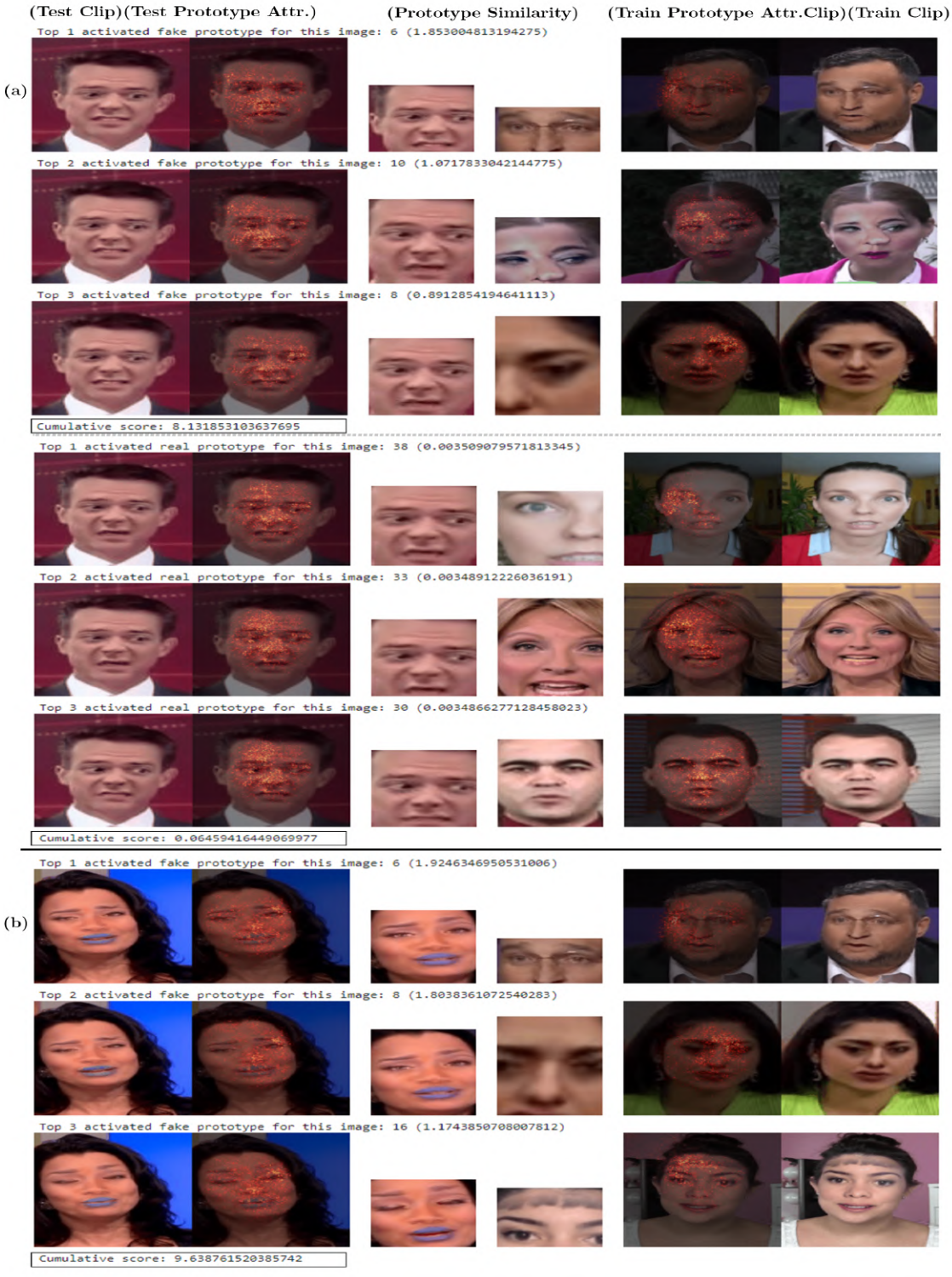


Figure 7. **Different classes of temporal artifacts and unnatural movements found by** `DPNet`. Top block are fake prototypes and bottom are real prototypes. a) heavy discolouration, b) subtle discolouration, c) subtle disappearance, d) unnatural movement, e) combined eye-mouth movement, and f) head movement. Best view as GIFs.


(Test Clip)(Test Prototype Attr.)
(Prototype Similarity)
(Train Prototype Attr.Clip)(Train Clip)
(a)
Top 1 activated fake prototype for this image: 6 (1.853004813194275)
Top 2 activated fake prototype for this image: 10 (1.0717833042144775)
Top 3 activated fake prototype for this image: 8 (0.8912854194641113)
Cumulative score: 8.131853103637695
Top 1 activated real prototype for this image: 38 (0.003509079571813345)
Top 2 activated real prototype for this image: 33 (0.00348912226036191)
Top 3 activated real prototype for this image: 30 (0.0034866277128458023)
Cumulative score: 0.06459416449069977
(b)
Top 1 activated fake prototype for this image: 6 (1.9246346950531006)
Top 2 activated fake prototype for this image: 8 (1.8038361072540283)
Top 3 activated fake prototype for this image: 16 (1.1743850708007812)
Cumulative score: 9.638761520385742

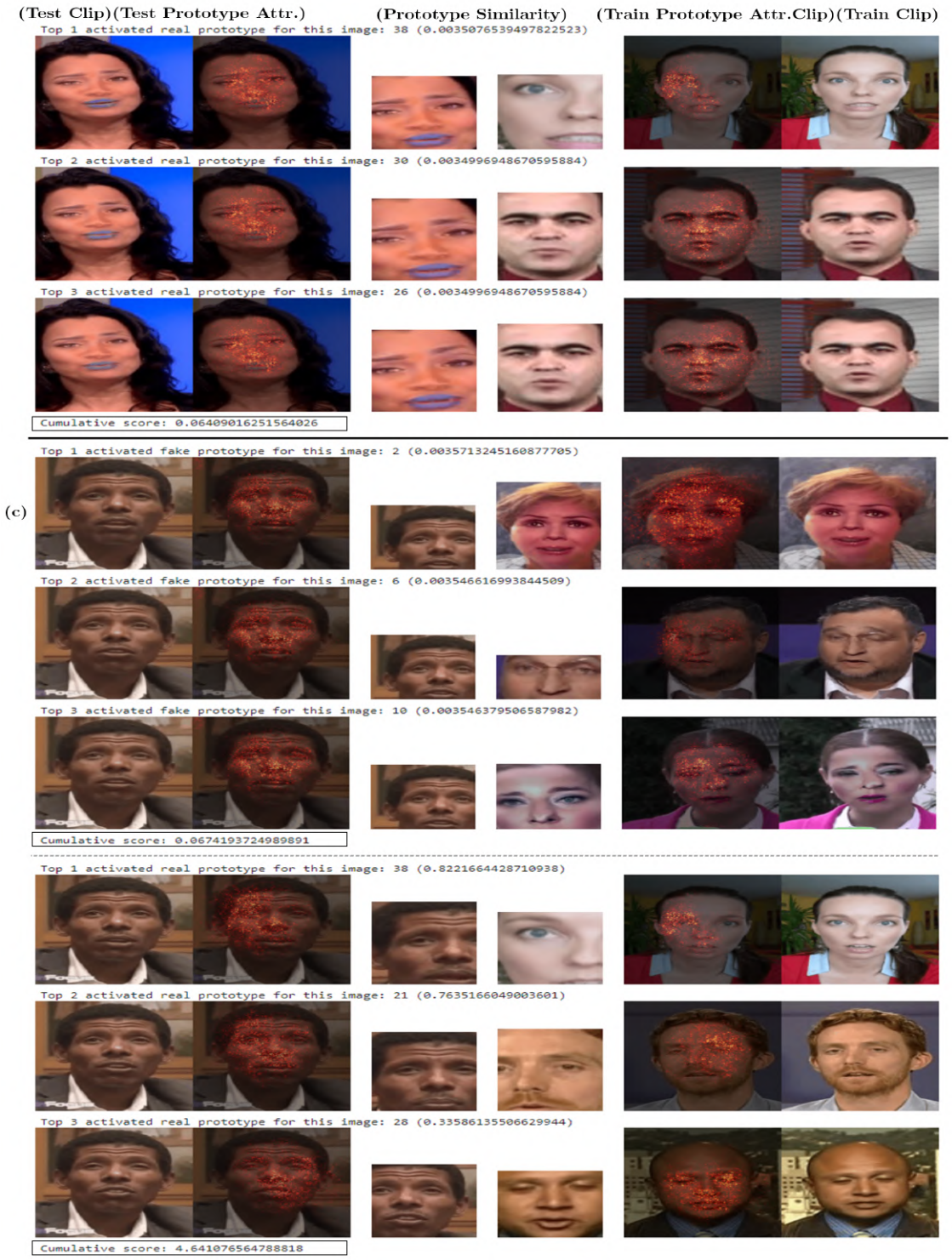


Figure 8. **More examples of how** `DPNet` **classify a video.** (a) and (b) are deepfakes, and (c) is genuine. Best view as GIFs. The prediction for each class is based on the evidence between the dynamics of the input and a small set of dynamic prototypes.