A. Appendix

		Ratio of fake frames		Avg. Length
		One seg.	Two seg.	
M	**NT, F2F**	0.363	N/A	193.23
Random	**DF**	0.231	0.389	668.5
	FSh	0.231	0.389	668.5
	F2F	0.233	0.393	662.0
	NT	0.264	0.445	585.3
	FS	0.264	0.445	585.3
	Average	**0.243**	**0.411**	**633.9**

Table 7: Ratio of fake frames and average length of videos in the benchmark dataset. This benchmark dataset is based on FaceForensics++ (FF++) and has the same sub-datasets as FF++. The ratio of fake frames differs among sub-datasets due to the original fake videos having different number of total frames. The average length is calculated in terms of the number of frames in a video. Each segment of fake frames is contiguous.

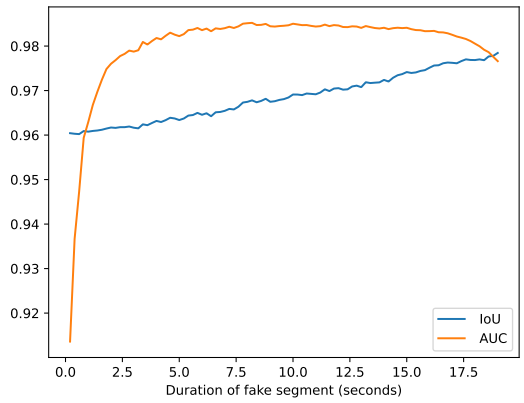


Figure 5: Performance (IoU and AUC) of the proposed approach across different lengths of deepfake segments. This is a visualization of Table 5 with more dense data points.

IoU for random guessing algorithm

Let the ground truth map be GT_{map} and predicted segmentation map be P_{map}. Both will be 1-D vectors of equal length with a predicted Boolean class (R or F) for each frame in the video.

$$GT_{map} = \{RRRRRRFFFRR...\} \tag{3}$$

$$P_{map} = \{RRRRRRFFFRR...\} \tag{4}$$

$$IoU = \frac{Intersection}{Union} = \frac{|GT_{map} \cap P_{map}|}{|GT_{map} \cup P_{map}|} \tag{5}$$

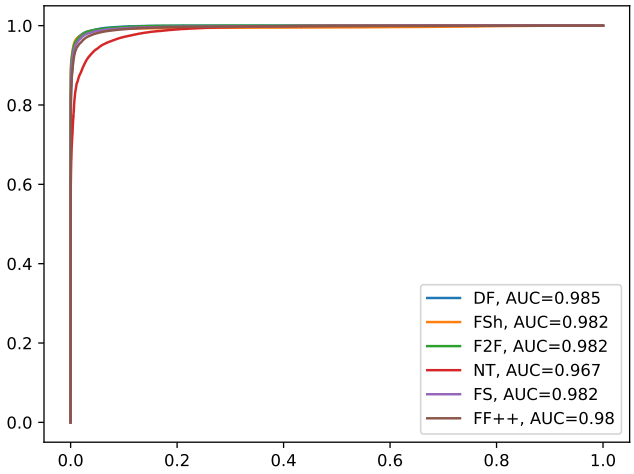
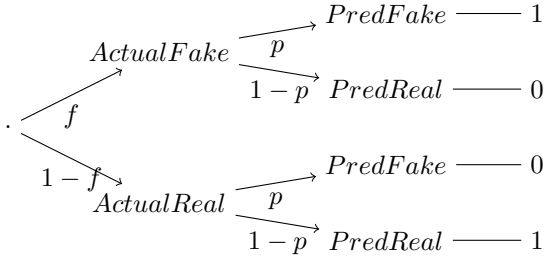


Figure 6: ROC curve for video level results. Model was trained on FaceForensics++ (FF++) and tested on the five sub-datasets within FF++ and all of FF++. This is an illustration of a part of Table 3 in the main paper.

Observation: $|GT_{map} \cap P_{map}|$ is the count of correctly predicted frames, and $|GT_{map} \cup P_{map}|$ is the count of correctly predicted frames and wrongly predicted frames $\times 2$.

IoU falls in the range $[0, 1]$; where the greater the value, the better the predicted segment map. Although the theoretical lower bound of IoU is zero, in practice it is useful to understand how a random guessing algorithm will be scored. Let f be the ratio of Real frames in the GT_{map} and p be the probability at which the randomly predicted frame in P_{map} is classified as Real. The graph below shows the possible $|GT_{map} \cap P_{map}|$ values (call it S).



For a single frame, the expected value of S is,

$$\begin{aligned} E(S) &= f.p.1 + f(1-p).0 + (1-f).p.0 + (1-f).(1-p).1 \\ &= 1 + 2.f.p - f - p = \alpha \end{aligned} \tag{6}$$

For T total frames $E(S) = T\alpha$. Using our observation above $|GT_{map} \cup P_{map}| = 2(T - T\alpha)$. Therefore IoU can be calculated as,

$$\begin{aligned} E(S) &= \frac{T\alpha}{T(2-\alpha)} \\ &= \frac{1 + 2.f.p - f - p}{1 - 2.f.p + f + p} \end{aligned} \tag{7}$$

	DF		FSh		F2F		NT		FS		FF++	
	One seg	Two seg	One seg	Two seg	One seg	Two seg	One seg	Two seg	One seg	Two seg	One seg	Two seg
DF	**0.993**	**0.987**	0.961	0.939	0.981	0.967	0.856	0.752	0.977	0.958	0.956	0.925
FSh	0.978	0.965	*0.986*	*0.98*	0.985	0.973	0.866	0.772	0.983	0.968	0.962	0.935
F2F	0.985	0.979	0.986	0.977	**0.991**	**0.987**	0.913	0.850	0.992	0.983	0.974	0.957
NT	0.985	0.980	0.984	0.979	0.981	0.977	*0.974*	*0.965*	0.972	0.965	*0.980*	*0.974*
FS	0.914	0.859	0.960	0.930	0.972	0.952	0.761	0.592	**0.993**	**0.985**	0.922	0.868
FF++	*0.987*	*0.981*	**0.987**	**0.98**	*0.987*	*0.982*	**0.979**	**0.968**	*0.987*	*0.977*	**0.986**	**0.978**

Table 8: Results in terms of accuracy for temporal segmentation on the proposed benchmark temporal deepfake dataset. This table is supplementary and identical in organization to Table 2 in the main paper. Each row indicates a model trained on a specific training sub-dataset; we have trained models with FaceForensics++ (FF++) and the five sub-datasets within FF++ i.e. Deepfakes (DF), Face-Shifter (FSh), Face2Face (F2F), Neural Textures (NT) and FaceSwap (FS). We report the best value in a column in **bold** and the second-best in *italic*.

	DF	FSh	F2F	NT	FS	FF++	C-DF	DFDC	WDF
DF	**0.993**	0.965	0.975	0.83	0.968	0.917	0.301	0.550	0.625
FSh	0.980	*0.990*	0.978	0.848	0.980	0.935	0.402	0.555	0.613
F2F	*0.990*	**0.993**	**0.990**	0.917	*0.993*	0.968	0.535	*0.589*	0.672
NT	0.973	0.968	0.968	*0.965*	0.960	*0.978*	*0.593*	0.584	*0.694*
FS	0.855	0.945	0.970	0.590	**0.995**	0.788	0.322	0.534	0.532
FF++	0.985	0.983	*0.983*	**0.968**	0.983	**0.987**	**0.799**	**0.682**	**0.694**

Table 9: Results (in Accuracy) for video level classification. This table is supplementary and identical in organization to Table 3 in the main paper. The columns constitute the test data. Along with FF++ and the sub-datasets of FF++ we have tested each model on other datasets such as CelebDF (C-DF), DFDC, and WildDeepFakes (WDF). The best value in a column is in **bold** and the second-best is in *italic*.

	Temporal Evaluation (Accuracy)			Video level (Accuracy)	
	ViT	ViT+TsT	ViT+TsT+Algo 1	ViT	ViT+TsT
DF	0.981	0.983 (**+0.002**)	0.987 (**+0.006**)	0.973	0.985 (**+0.012**)
FSh	0.981	0.983 (**+0.002**)	0.987 (**+0.006**)	0.973	0.983 (**+0.010**)
F2F	0.982	0.984 (**+0.002**)	0.987 (**+0.005**)	0.973	0.983 (**+0.010**)
NT	0.970	0.973 (**+0.003**)	0.979 (**+0.009**)	0.965	0.968 (**+0.003**)
FS	0.979	0.982 (**+0.003**)	0.987 (**+0.008**)	0.973	0.983 (**+0.010**)
FF++	0.979	0.981 (**+0.002**)	0.986 (**+0.007**)	0.985	0.987 (**+0.002**)

Table 10: Results in Accuracy for Ablation study on temporal segmentation of deepfakes and video-level classification. This table is supplementary and identical in organization to Table 6 in the main paper. Changes in the results are reported bold and are in brackets.

For a random guessing algorithm with probability $p = 0.5$ for each class in a binary classification problem we have $IoU = 1/3$. This will be the random guessing baseline for IoU in our context. from equation (5).

Smoothing Algorithm

The predictions of the ViT for the videos are frame-level and therefore there are often some noisy predictions. These noisy predictions can be corrected (Figure 7) with a simple smoothing technique. We have used Algorithm 1 to smooth out noisy frame level prediction. In this algorithm a minimum fake-segment duration (in number of frames) is set. For each frame-prediction, majority voting is taken from predictions of past frames (on the left) and from future frames (on the right), and this helps determining the final label of that frame. Smoothing noisy predictions aids in better performance as can be seen in Table 6 in the main paper.

	Window Size					
	5		**10**		**15**	
Overlap	IoU	AUC	IoU	AUC	IoU	AUC
4	**0.974**	**0.988**	0.956	0.973	0.797	0.846
3	0.953	0.977	0.946	0.975	0.806	0.848
2	0.950	0.976	0.953	0.976	0.745	0.766
1	*0.971*	*0.985*	0.958	0.976	0.797	0.838
0	0.958	0.983	0.947	0.975	0.766	0.84

Table 11: Ablation study on varying Window sizes in terms of number of frames in a window and overlap in sliding-window. The values are from frame-level prediction on our proposed temporal segmentation dataset with one fake-segment to solve the temporal segmentation problem. We can notice that a window size of 5 with overlap of 4 gives us the optimal results for temporal segmentation.

	Window Size					
	5		**10**		**15**	
Overlap	Acc	AUC	Acc	AUC	Acc	AUC
4	*0.987*	*0.982*	**0.992**	**0.985**	0.982	0.959
3	0.987	0.974	0.983	0.972	0.983	0.966
2	0.988	0.975	0.984	0.977	0.980	0.976
1	0.982	0.981	0.983	0.974	0.985	0.969
0	0.990	0.978	0.983	0.974	0.980	0.944

Table 12: Ablation study on varying Window sizes in terms of number of frames in a window and overlap in sliding-window. The values are from video-level prediction. We can notice that a window size of 5 with overlap of 4 gives us the second-best results where the results for window size 10 with overlap of 4 frames are the best. However, our main goal is to achieve best results in frame-level performance. Hence, we chose the prior parameters for the experiments.

Algorithm 1 Smoothing noisy predictions.

Require: ρ, the list of predictions per frame
Require: $k \geq 0$, the offset
  for $i \leftarrow 0 \ldots len(\rho)$ **do**
    $\rho_{left} \leftarrow$ sub-list of size k on left of $\rho[i]$
    $\rho_{right} \leftarrow$ sub-list of size k on right of $\rho[i]$
    $M_{left} \leftarrow$ majority-vote(ρ_{left})
    $M_{right} \leftarrow$ majority-vote(ρ_{right})
    if ρ_{left} is empty $and\ \rho[i] \neq M_{right}$ **then**
      $\rho[i] \leftarrow M_{right}$
    else if ρ_{right} is empty $and\ \rho[i] \neq M_{left}$ **then**
      $\rho[i] \leftarrow M_{left}$
    else if $M_{left} = M_{right}\ and\ \rho[i] \neq M_{left}$ **then**
      $\rho[i] \leftarrow M_{left}$
    end if
  end for
  return ρ

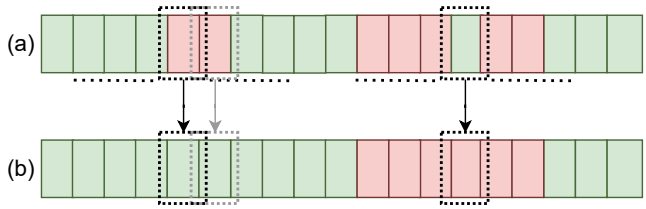
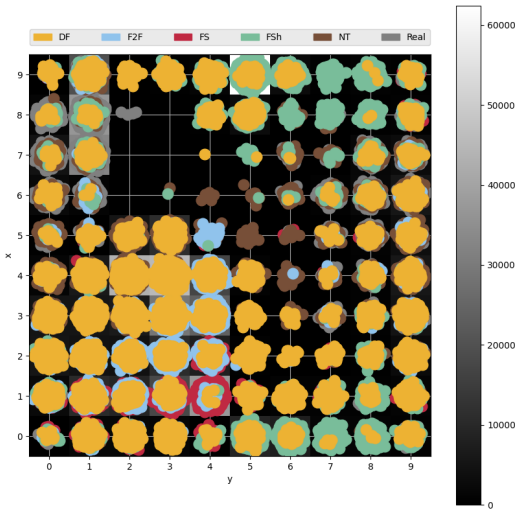
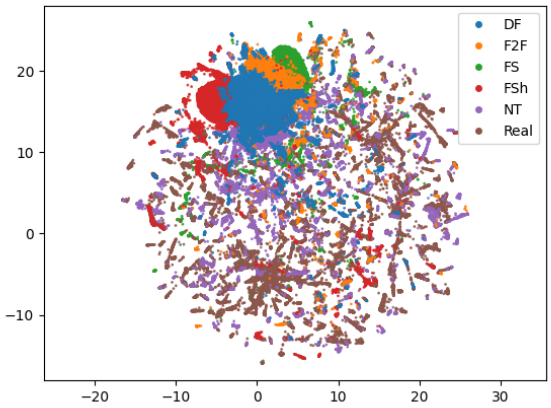


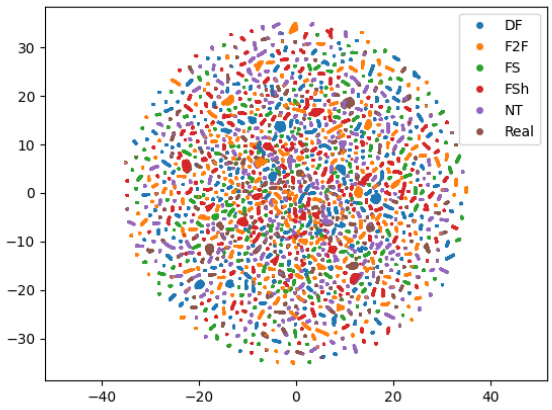
Figure 7: This figure depicts the visualization of our proposed approach for smoothing out noisy predictions. The first image (a) illustrates the raw frame-level predictions for a video, while the second image (b) shows the output after applying Algorithm 1. Each small block in the images represents the model's prediction for a frame, with green indicating 'real' and red indicating 'fake' prediction. The frames surrounded by dotted rectangles get their prediction changed based on the majority vote from past (left) and future (right) predictions, indicated by the dotted lines.



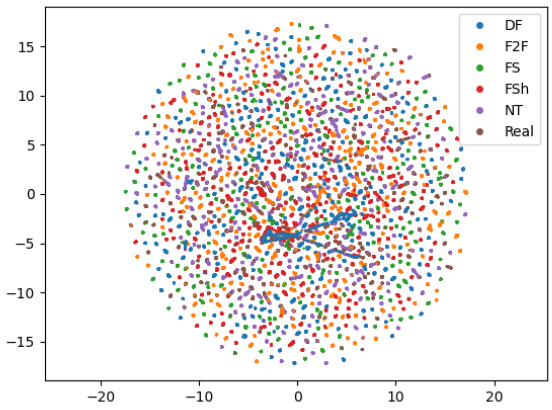
(a) Self-organizing map (SOM)



(b) Self-Organizing Nebulous Growths (SONG)



(c) t-distributed Stochastic Neighbor Embedding (t-SNE)



(d) Uniform Manifold Approximation & Projection (UMAP)

Figure 8: Visualizations on the spatial embeddings (from ViT) on the sub-datasets in FF++.