A. Implementation details

Linear probing. We train the linear probing model for 100 epochs with early stopping (training is stopped if the validation loss does not improve for 10 consecutive epochs). We use the Adam optimizer with a learning rate of 10^{-3}. For AV1M, we select a training set of 22972 videos and a validation set of 2527 videos, both sampled from the official training set. For FAVC, we split the entire dataset into 63% training, 7% validation, and 30% test sets and used only the real video real audio (RVRA) and fake video fake audio (FVFA) samples.

Next-token prediction. The training setup for the next-token prediction is similar to the one used for the linear probing experiments. The main difference is that we now anneal the learning rate using a cosine scheduler. The model is trained on 50k real videos randomly sampled from the AV1M training set, using 45k samples for training and 5k for validation. The training set includes the real samples used for training the linear probes.

Audio-video synchronization. For the audio-video synchronization task, we use the publicly released implementation of [73] with its default settings. We use a temporal neighborhood of 30 frames and a learning rate scheduler with a patience of 5 epochs, a reduction factor of 0.1, and a starting learning rate of 10^{-5}. The training set is identical to the one used for the next-token prediction task.

B. Model checkpoints

For AV-HuBERT we use the `self_large_vox_433h` checkpoint which was pretrained on LRS3 and VoxCeleb2 and finetuned on 433 hours of LRS3 samples for the task of visual speech recognition. For CLIP we use the `openai/clip-vit-large-patch14` checkpoint, and for Video MAE the `MCG-NJU/videomae-large`, both which can be found on HuggingFace. For FSFM we use the `FF++_c23_32frames` checkpoint, which was trained on the FaceForensics++ dataset. For Wav2Vec2, we use the `facebook/wav2vec2-xls-r-2b` HuggingFace checkpoint of the 2-billion parameters model, which was pretrained on 436k hours of multilingual data. For Auto-AVSR, we use the best models trained on 3,448 hours: `LRS3_V_WER19.1` for visual features, `LRS3_A_WER1.0` for audio features, `LRS3_AV_WER0.9` for multimodal features. For BRAVEn we use the checkpoints from the high-resource setup, with self-training, finetuned for ASR and VSR, respectively. Input face crops and audio files are extracted using the AV-HuBERT code.

C. Datasets

General preprocessing. We follow the same preprocessing steps used for each backbone network during its pretraining stage. Visual cropping is applied according to the "pre-training content" column in Tab. 1. Specifically, models pretrained on generic visual content use uncropped frames, those pretrained on faces use face-cropped frames, and those pretrained on lips use lip-cropped frames.

DeepfakeEval 2024 preprocessing. We use TalkNet-ASD [76] to identify which segments in an audio-video media file contain a single person speaking. We selected audio-video segments that met these criteria: (i) each video segment that has an associated audio stream; (ii) a video segment should contain a single speaking face that was tracked in every frame. (Some videos had static images or background music instead of speech and these were discarded); (iii) the identified face is larger than 100px$\times$100px; (iv) the duration of audio-video segment is between 3 and 60 seconds.

D. Baselines

Randomly initialized models. For the randomly initialized AV-HuBERT models, we keep the same architecture as the pre-trained version. We also adopt the same initialization scheme used when training AV-HuBERT from scratch: BatchNorm and LayerNorm layers start from constant parameters (weights set to 1 and biases to 0), while linear and embedding layers follow a BERT-style initialization, with weights sampled from a normal distribution with mean 0 and standard deviation 0.02. Finally, we preserve the same preprocessing pipeline as in the pre-trained setup, using log filterbanks for audio and mouth crops for the visual stream.

AVFF [58] is a multimodal, two stage deepfake detector. In the first self-supervised stage, the model encodes both the input audio and video with two separate encoders, masks the tokens in a complementary way, predicts the masked ones using the remaining, visible tokens and then reconstructs the original input. This stage helps the model extract meaningful information from both streams and better align the features. In the second stage, the features extracted are used as input for a classifier network which is trained on the task of deepfake detection. For our experiments we finetuned the checkpoint pretrained on Kinetics400, available in the unofficial open version.[3]

SpeechForensics [44] is an unsupervised method that detects deepfakes by measuring the alignment between audio and video streams. The alignment is computed as the cosine similarity between audio and visual features extracted from a pretrained AV-HuBERT. To account for desynchronizations,

[3] https://github.com/JoeLeelyf/OpenAVFF

which are common in in-the-wild real videos, the authors propose measuring the best alignment score by shifting the streams within a fixed window. For a fair comparison, we use the same set of AV-HuBERT features, as previously used for linear probing. Since audio and visual features sometimes have different lengths, we align them by trimming at the end. We note, however, that the original implementation uses uniform sampling, which yields slightly better performance. In Sec. E.2, we conduct an ablation study on the pooling operation and window size parameters.

AuViRe [41] is a supervised method trained on the task of temporal forgery localization. The model uses rich self-supervised representations (AV-HuBERT features) as input for a 1D CNN. Firstly, the model reconstructs the input features both within and cross modality (for cross modality, only the video features are reconstructed based on audio ones). Then, a classification head is used to detect the per frame manipulations and a regression head to predict the manipulation segment boundaries. The method was trained on two datasets, LAV-DF and AV-Deepfake 1M. In our experiments, we used the checkpoint trained on AV-Deepfake 1M from the official GitHub repository, together with the default parameters.

AVAD [19] is an unsupervised method which focuses on modeling the distribution over delays in real data and detect fakes at test time based on the deviation from norm. This is achieved by training an autoregressor on the distribution over delays predicted by another component: the audio-visual synchronization model. The authors trained these components only on real data. For our experiments, we used the default parameters and checkpoint specified on the official GitHub repository. The checkpoint used was trained on LRS2 and LRS3 datasets.

RealForensics [22] is a supervised two-step method that leverages self-supervised pretraining to improve robustness. In the first step, the objective is to learn temporally dense video representations using a cross-modal student-teacher framework, where each student predicts the output of the other modality's teacher. The backbones for each modalities consist of convolutional networks with one-block transformer encoder predictors on top. In the second stage, the detector is tasked with predicting the video targets produced by the video teacher from stage 1 for real videos, combined with a cross-entropy minimization loss on both real and fake inputs. This multi-task learning encourages the model to focus on high-level facial dynamics. For our experiments, we use the model trained on FaceForensics++ [67], with additional real samples from LRW [14].

E. Further results

E.1. Average precision results

We complement the results from Tab. 2 and Tab. 4 with average precision (AP) scores reported in Tab. 5 and Tab. 6, respectively.

E.2. Synchronization of pretrained representations

If audio-visual self-supervised features are trained jointly, then they are already aligned to a degree. Here, we investigate how well this pre-existing alignment works for the task of deepfake detection. This contrasts with the additional alignment introduced in Sec. 3.2, which was required when combining representations from different models.

We use audio-only and visual-only features from AV-HuBERT and measure their alignment via cosine similarity. The fakeness score s between the audio features $\mathbf{a}$ and visual features $\mathbf{v}$ is defined as a generalization of [44, 65]:

$$s(\mathbf{a}, \mathbf{v}) = \min_{\delta \in [-\Delta, +\Delta]} \mathrm{pool}_t \left(-\cos(\mathbf{a}_t, \mathbf{v}_{t+\delta}) \right), \qquad (3)$$

where

- δ is a temporal shift that compensates for imperfect synchronization of real video [19]. Allowing moderate shifts can therefore improve discrimination between real and fake samples. We test both strict ($\Delta = 0$) and moderate alignment ($\Delta = 15$), following [44].
- pool aggregates per-frame fakeness scores. We experiment with several pooling strategies: average, min, max, 3rd and 97th percentiles, and a scaled log-sum-exp (with temperature given by the length of the sequence).

Eq. (3) subsumes prior methods as special cases: FACTOR [65] corresponds to $\Delta = 0$ and 97th-percentile pooling; SpeechForensics [44] corresponds to $\Delta = 15$ with average pooling.

Tab. 7 shows results on the four datasets introduced in Sec. 5.1. At a high level, all variants behave similarly when compared to a supervised baseline (linear probing on AV-HuBERT multimodal features): they underperform on AV1M and perform better on AVLips and DFE-2024. If we look closer, and at the two axes, we first see that average pooling, as used in SpeechForensics [44], is rarely optimal. This is especially evident for AV1M, where the max-based pooling variants perform best. This behavior aligns with the nature of the dataset: since AV1M contains local manipulations, max pooling is able to capture these brief artifacts. Interestingly, for AVLips and DFE-2024, it is the minimum pooling that yields consistent (albeit small) gains. Compared to max pooling, which searches for "evidence of fakeness somewhere," min pooling searches for "evidence of realness anywhere." This is particularly suitable for real in-the-wild videos, whose synchronization is imperfect and which contain only a few strongly aligned segments.

	Model	Modality	A Test on FAVC FAVC ↓ FAVC	B Test on FAVC AV1M ↓ FAVC	C Test on AV1M AV1M ↓ AV1M	D Test on AV1M FAVC ↓ AV1M	E Test on AVLips FAVC ↓ AVLips	F Test on AVLips AV1M ↓ AVLips	G Test on DFE-2024 DFE ↓ DFE	H Test on DFE-2024 FAVC ↓ DFE	I Test on DFE-2024 AV1M ↓ DFE	J mean OOD
1	AV-HuBERT (A) random	audio	100	99.9	99.1	91.1	55.7	60.2	89.2	85.2	87.1	79.9
2	AV-HuBERT (V) random	visual	99.0	96.2	52.3	51.3	63.2	59.5	88.4	91.6	88.8	75.1
3	AV-HuBERT (A)	audio	**100**	**100**	**100**	**99.2**	55.6	62.9	93.4	89.1	88.9	82.6
4	Auto-AVSR (ASR)	audio	100	98.4	96.8	51.2	58.3	55.3	93.4	88.6	88.8	73.4
5	Wav2Vec2	audio	**100**	**100**	**100**	96.5	**61.9**	**64.3**	92.5	**92.5**	**91.7**	**84.5**
6	BRAVEn (A)	audio	**100**	**100**	99.9	88.7	51.2	53.9	**94.0**	92.4	90.4	79.4
7	AV-HuBERT (V)	visual	**100**	99.6	94.7	62.1	98.4	88.6	95.1	92.9	92.8	89.1
8	Auto-AVSR (VSR)	visual	99.9	98.7	60.3	52.6	86.2	74.3	92.9	88.8	90.3	81.8
9	FSFM	visual	99.9	95.2	95.5	53.6	83.6	47.1	93.8	**94.5**	84.3	76.4
10	CLIP VIT-L/14	visual	**100**	99.8	96.8	**71.5**	65.3	60.9	93.8	90.7	85.2	78.9
11	Video-MAE-large	visual	**100**	98.1	**99.7**	61.3	73.9	53.2	89.3	86.9	84.3	76.3
12	BRAVEn (V)	visual	**100**	**99.9**	94.1	63.7	**98.8**	**96.9**	**95.3**	93.2	**95.5**	**91.3**
13	AV-HuBERT	audio-visual	**100**	**100**	**99.9**	**94.9**	**76.5**	**82.1**	**94.7**	**90.8**	**88.7**	**88.8**
14	Auto-AVSR	audio-visual	99.7	97.7	92.8	54.4	63.7	58.4	92.7	85.9	89.1	74.9

Table 5. Average precision (AP, %) performance of linear probes trained on multiple self-supervised representations.

Model	AV1M Sup.	AV1M NTP	AV1M Sync.	FAVC Sup.	FAVC NTP	FAVC Sync.
Single features						
AV-HuBERT (A)	**99.2**	**91.8**	N/A	**100**	**98.8**	N/A
Wav2Vec2	96.5	56.4	N/A	**100**	96.6	N/A
AV-HuBERT (V)	62.1	49.2	N/A	99.6	97.1	N/A
CLIP	71.5	49.7	N/A	99.8	96.8	N/A
Combination of features						
AV-H (A + V) (rand.)	78.3	64.4	50.8	99.8	98.1	96.0
AV-H (A + V)	97.2	83.4	**85.7**	**100**	**99.5**	**99.7**
AV-H (A) + CLIP	**99.2**	**88.0**	50.0	**100**	98.6	96.1
W2V2 + AV-H (V)	95.7	59.4	85.1	**100**	98.6	99.6
W2V2 + CLIP	97.0	56.6	50.5	**100**	97.6	90.7

Table 6. Average precision (AP, %) performance for deepfake detection when training for the two anomaly detection proxy tasks: next-token prediction (NTP) and audio-video synchronization (sync.). Supervised models (sup.) are trained cross-domain (FAVC→AV1M and AV1M→FACV, respectively). Anomaly detection models are trained on real data only (a subset of VoxCeleb).

In terms of feature alignment by shifting, we see that it does help in certain setups, most notably for DFE-2024.

Layer-wise analysis. We further analyze the representational capabilities of AV-HuBERT features across all 24 transformer layers. This analysis is conducted on the AV1M test set and, using SpeechForensics [44] with its default hyperparameters (average pooling and a shift window of 15). As shown in Fig. 6, performance roughly stabilizes from layer 7, with a small drop at layer 22. Interestingly, the best performance is obtained at layer 9 (68.6%), but the improvement over the last layer is not substantial. Note that the last-layer performance in Fig. 6 (67.1%) is slightly lower than the corresponding value in Tab. 7 (68.1%; for average pooing and $\Delta = 15$ on AV1M); this discrepancy is due to the absence of LayerNorm in the layer-wise analysis.

E.3. Classification head analysis

We compare the linear classifier described in Sec. 3 with a more powerful head: a transformer. The transformer has width 768, 4 layers, 8 attention heads. Representations are projected using a linear layer to the input 768 dimension. To obtain a classification prediction, we use the [CLS] token and project it through a linear layer. The results are displayed in Tab. 8. We observe that, in most cases, the performance obtained by the transformer model is on par with or slightly below that of linear probing. The single notable exception observed occurs on FSFM features when trained on AV1M and tested on FAVC. In this scenario, performance increases significantly from 40.9% AUC to 72.3% AUC.

E.4. Combinations of features

In Fig. 7 we show the results for combinations of features when testing is done on DFE-2024. Compared to Fig. 5 (testing done on FAVC), certain trends are much better highlighted: first, the performance of every model greatly increases when combined with Wav2Vec2 or AV-HuBERT (V); second, the correlations between models' results are weaker, suggesting that the models capture more distinct patterns.

Pooling function	FAVC		AV1M		AVLips		DFE-2024	
	$\Delta = 0$	$\Delta = 15$	$\Delta = 0$	$\Delta = 15$	$\Delta = 0$	$\Delta = 15$	$\Delta = 0$	$\Delta = 15$
average	99.3	**100**	65.3	68.1	94.8	92.7	62.7	75.6
Maximum variants								
max	99.3	99.1	**76.8**	76.9	94.8	93.7	61.0	64.9
log-sum-exp	99.4	99.6	**76.8**	76.9	95.3	94.2	60.8	64.4
percentile-97	**99.5**	99.4	75.0	**80.0**	**95.4**	93.0	61.4	68.8
Minimum variants								
min	98.2	99.3	58.1	56.1	94.0	96.1	66.3	74.4
percentile-3	99.1	99.5	59.9	58.6	**95.4**	**96.4**	**63.9**	**75.7**
Supervised baseline								
AV-H + linear on FAVC (row 14 Tab. 2)	100		94.5		78.5		58.2	

Table 7. AUC performance (%) when directly measuring the alignment of pretrained AV-HuBERT audio and visual features. We evaluate over pooling functions and temporal shifts (Δ), to compensate desynchronizations of real in-the-wild videos. Bold indicates best results in each column except for the supervised baseline (multimodal AV-HuBERT features with a linear classifier trained on the FAVC dataset).

			A	B	C	D	E	F	G	H	I	J
			Test on FAVC		Test on AV1M		Test on AVLips		Test on DFE-2024			
	Model	Head	FAVC ↓ FAVC	AV1M ↓ FAVC	AV1M ↓ AV1M	FAVC ↓ AV1M	FAVC ↓ AVLips	AV1M ↓ AVLips	DFE ↓ DFE	FAVC ↓ DFE	AV1M ↓ DFE	mean OOD
	Audio features											
3	AV-HuBERT (A)	L	100.0	100.0	100.0	99.0	50.0	57.2	65.8	49.1	48.3	67.3
		T	100.0	99.9	99.9	88.8	47.7	53.4	70.2	48.2	55.0	65.5
4	Auto-AVSR (ASR)	L	99.7	76.0	96.4	50.3	52.9	49.6	63.5	49.4	47.5	54.3
		T	100.0	80.1	97.3	51.3	46.4	49.4	70.3	51.6	50.4	54.9
5	Wav2Vec2	L	100.0	99.9	100.0	96.6	51.3	56.3	58.7	62.3	58.6	70.8
		T	100.0	99.9	100.0	94.2	47.3	56.5	50.6	52.5	58.2	68.1
	Visual features											
6	AV-HuBERT (V)	L	100.0	95.5	93.7	64.1	98.3	90.5	72.1	63.7	67.7	80.0
		T	100.0	92.9	93.7	58.6	98.6	88.3	73.8	70.3	61.8	78.4
7	Auto-AVSR (VSR)	L	97.8	77.5	59.0	51.3	83.3	70.1	64.3	48.7	56.1	64.5
		T	98.5	75.5	56.9	51.4	83.4	63.4	62.4	46.9	56.2	62.8
8	FSFM	L	97.1	40.9	95.3	52.7	84.3	36.8	71.7	71.8	43.5	55.0
		T	98.8	72.3	99.0	47.9	79.8	42.5	70.7	68.5	48.5	59.9
9	CLIP ViT-L/14	L	99.8	95.2	96.5	71.1	60.3	53.3	73.9	55.6	43.5	63.2
		T	99.7	96.4	99.2	62.3	57.0	59.1	73.3	55.8	50.2	63.4
10	Video-MAE-large	L	100.0	70.4	99.8	60.0	71.3	47.2	54.5	45.6	39.3	55.6
		T	100.0	60.8	100.0	53.0	81.3	56.5	52.9	58.9	41.2	58.6
	Audio-visual features											
11	AV-HuBERT	L	100.0	99.5	99.9	94.5	78.5	84.4	70.4	58.2	54.3	78.2
		T	100.0	99.6	99.9	78.6	82.5	76.8	68.9	59.4	55.9	75.5
12	Auto-AVSR	L	94.7	68.3	91.6	53.2	59.6	54.6	61.2	43.0	49.2	54.7
		T	99.7	65.5	93.4	51.1	51.8	51.7	66.5	51.3	51.9	53.9

Table 8. AUC performance (%) when training a linear layer (L) vs. a transformer (T) (indicated in blue) classifier head on top of features.

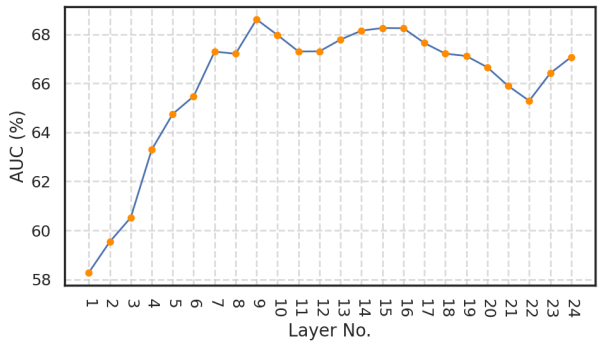

Figure 6. Performance on the AV1M dataset using the pretrained AV-HuBERT representations extracted from different layers. We use the SpeechForensics approach that directly compares AV-HuBERT features over a optimal shifting window.

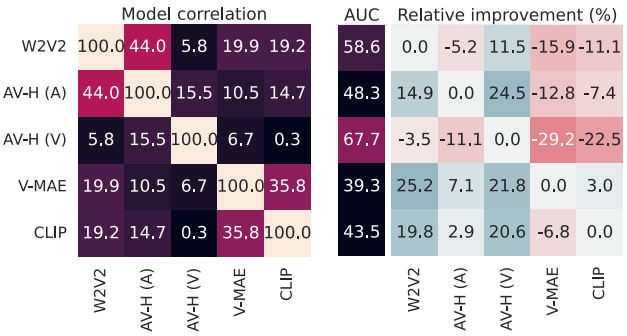

Figure 7. Correlations between models (left) and downstream performance (right). Training was done on AV1M, testing on DFEval-2024.