SeeABLE: Soft Discrepancies and Bounded Contrastive Learning for Exposing Deepfakes

Nicolas Larue[1,2], Ngoc-Son Vu[1], Vitomir Struc[2], Peter Peer[2], Vassilis Christophides[1]
[1]ETIS - CY Cergy Paris University, ENSEA, CNRS, France
[2]University of Ljubljana, Slovenia

Abstract

Modern deepfake detectors have achieved encouraging results, when training and test images are drawn from the same data collection. However, when these detectors are applied to images produced with unknown deepfake-generation techniques, considerable performance degradations are commonly observed. In this paper, we propose a novel deepfake detector, called SeeABLE, that formalizes the detection problem as a (one-class) out-of-distribution detection task and generalizes better to unseen deepfakes. Specifically, SeeABLE first generates local image perturbations (referred to as soft-discrepancies) and then pushes the perturbed faces towards predefined prototypes using a novel regression-based bounded contrastive loss. To strengthen the generalization performance of SeeABLE to unknown deepfake types, we generate a rich set of soft discrepancies and train the detector: (i) to localize, which part of the face was modified, and (ii) to identify the alteration type. To demonstrate the capabilities of SeeABLE, we perform rigorous experiments on several widely-used deepfake datasets and show that our model convincingly outperforms competing state-of-the-art detectors, while exhibiting highly encouraging generalization capabilities. The source code for SeeABLE is available from: `https://github.com/anonymous-author-sub/seeable`.

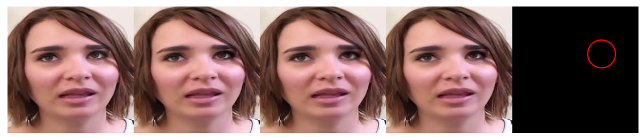


Figure 1: **Examples of faces with soft-discrepancies.** *Can you identify the discrepancy in each image? SeeABLE can. SeeABLE's answer: the perturbed area of the four facial images is within the circle shown on the right.* Note that a different soft discrepancy is used in each image.

1. Introduction

Recent advances in (deep) generative models, such as generative adversarial networks (GAN) [23], diffusion models [32] and generative normalizing flows [14], have made it possible to generate fake images and videos with unprecedented levels of realism. Human faces have been a particularly popular target for such models, enabling the creation of so-called *deepfakes* [12,19,65,66], i.e., manipulated facial images commonly used for malicious purposes. These deepfakes have been shown to constitute a serious psychological and financial treat to individuals, but also society as a whole [8,52]. As a result, the deep learning community is actively working on countermeasures and detection techniques that can help to mitigate this threat.

By having access to datasets with both, real and forged (manipulated) faces [15,16,36,41,47,57,70], existing deepfake detectors [1,7,9,10,33,50,57] essentially learn a binary decision boundary that leads to reasonable detection performance with deepfake-generation techniques seen during training. However, as recent empirical studies [18,39,67] report, the performance of such (discriminatively-trained) detectors degrades significantly when used with unseen face-manipulation methods, which severely limits their use in real-life deployment scenarios [75].

A powerful solution to improve the generalization capabilities of deepfake detectors is to use synthetic data (i.e., *pseudo deepfakes*) during training and encourage the models to learn generalizable decision boundaries. Such strategies are at the core of many of the state-of-the-art (SoTA) detection models [5,45,48,60,71] that either enrich the diversity of available deepfakes by synthesizing novel fake images for training, or rely completely on synthetic deepfakes when learning the detection models. These methods differ in the type of augmentations considered: blending- or adversarial-based techniques, adoption of global or local transformations, and use of single or multiple source/target images, as summarized in Table 1. Once the pseudo-fakes are generated, a classifier is learned to distinguish between real and fake faces. While these methods were observed to lead to highly competitive detection performance, especially in cross-dataset settings, they are still limited by the discriminative nature of the training procedure that tries to differentiate between real faces and the specific artifacts induced by the pseudo-deepfake generation procedure.

In this paper, we propose a novel deepfake detector, called **SeeABLE** (**S**oft discr**e**pancies and bound**e**d contrastive **l**arning for **e**xposing deepfakes), that formulates the detection problem as a (one-class) out-of-distribution detec-

Deefake detection model	Augmentation (blended, adversarial)				Alteration artifact		Problem formulation	
	global		local		global	local	classification	regression
	multiple	single	multiple	single				
Face Xray [45], PCL [71], OST [48]	✓				✓		✓	
SLADD [5]			✓		✓		✓	
SBI [60]		✓			✓		✓	
SeeABLE (proposed)		✓		✓		✓		✓

Table 1: **Comparison of SeeABLE and SoTA detectors that use pseudo-deepfake synthesis during model learning.** The existing techniques differ in terms of augmentation techniques used, the level at which alterations are applied (local vs. global), and the problem formulation. As can be seen, SeeABLE differs significantly from existing techniques.

tion task and generalizes better to unseen deepfakes than discriminatively-learned models. SeeABLE is trained with images of real faces only and differs significantly from existing (pseudo-fake based) detectors, as seen in Table 1. Specifically, the model first generates (subtle) local image perturbations, referred to as **soft discrepancies**, using a rich set of image transformations, as illustrated in Figure 1. Next, the generated soft discrepancies are pushed towards a set of target representations (i.e., *hard prototypes*) using a single **multi-task regressor** learned with a novel **bounded contrastive regression** loss. Here, the objective of the regressor is two-fold: (1) to map the different soft discrepancies into well-separated (and tightly clustered) prototypes that facilitate efficient similarity scoring (akin to prototype matching), and (2) to localize the spatial area of the local image perturbation and, thus, to exploit an auxiliary source of information for the regression task. The subtle image changes introduced by the soft discrepancies force SeeABLE to learn to detect minute image inconsistencies (and in turn a highly robust detector), whereas the local nature of the perturbation allows the model to exploit an additional localization (pretext) task that infuses complementary cues into the learning procedure. Unlike competing one-class detectors that typically rely on (low-level) per-pixel reconstructions to identify deepfakes, e.g. [37], SeeABLE, learns **rich** and **semantically meaningful features** for the detection process that, as we show in the experimental section, lead to highly competitive detection results.

To demonstrate the capabilities of SeeABLE, we evaluate the model in comprehensive cross-dataset and cross-manipulation experiments on multiple datasets, i.e., FF++ [57], CDF-v2 [47], DFDC-p [16] and DFDC [15], and in comparison to twelve SoTa competitors. The results of the experiments show that the proposed model achieves highly competitive results on all considered datasets, while exhibiting encouraging generalization capabilities.

In summary, the main contributions of this paper are:

- We propose SeeABLE, a new state-of-the-art deepfake detector trained in one-class self-supervised anomaly detection setting that captures high-level semantic information for the detection task by localizing artificially-generated (spatial and frequency-domain) image perturbations. Unlike (most) competing solutions, SeeABLE learns to provide an anomaly score that allows it to efficiently discriminate between real and fake imagery.
- We introduce a novel Bounded Contrastive Regression (BCR) loss that enables SeeABLE to efficiently push/map the local soft discrepancies to a predefined set of (evenly-distributed) prototypes, and, in turn, to facilitate distance-based prototype matching for deepfake detection.
- Through rigorous (cross-dataset and cross-manipulation) experiments on multiple dataset, we demonstrate the superior generalization capabilities of SeeABLE compared to existing (SoTA) deepfake detectors.

2. Related work

In this section, we review (closely) related prior work needed to provide context for SeeABLE. For a more comprehensive coverage of the relevant topics, the reader is referred to some of the excellent surveys available [51, 54].

Deepfake detection. A considerable amount of deepfake detectors has been introduced in the literature over the years [4, 17, 30, 59, 67, 73]. Early detectors, e.g., [1, 3, 11, 27, 46], relied mostly on the known deficiencies of deepfake-generation techniques and focused on the detection of the corresponding visual artifacts. A notable cross-section of these techniques [20, 49, 56, 57] use frequency-domain representations to discriminate between real and fake images. Liu *et al*. [49], for example, leveraged the phase spectrum to capture the up-sampling artifacts of face manipulation techniques, while Qian *et al*. [56], on the other hand, used a DCT-based model F^3net [69] to extract frequency-domain cues and compute statistical features for forgery detection. Fei *et al*. [20] proposed a weakly supervised second order local anomaly learning module that decomposes local features by different directions and distances to calculate first and second order anomaly maps. Another category of models [21, 26, 72, 74] utilize temporal features for deepfake detection. The local and temporal-aware transformer-based deepfake detector (LTTD) [26], for instance, utilized a local sequence transformer to model temporal consistency on restricted spatial regions to identify deepfakes. In [21], manipulated videos were detected based on the subtle inconsistencies between visual and audio signals by training an autoregressive model that captured the temporal synchronization between video frames and sound. Other recent works [39, 43] also attempted to detect deepfakes in a continual learning setting, with the main goal of avoiding the catastrophic forgetting [55] across different tasks.

Detection through pseudo-deepfake generation. One of the most effective strategies for learning deepfake detectors that generalize well across deepfake generation techniques [5, 25, 45, 60, 71] is to use dedicated augmentation techniques to first synthesize forged images (i.e., pseudo deepfakes) and then train a binary classification model for the detection task. The idea behind Face-Xray [45], for example, is to generate blended images (BI) of two different faces using a global transformation and then learn to discriminate between the real and blended faces. Shiohara *et al*. [60] extended this idea and proposed a synthetic training dataset with self-blended images (SBI) that are generated from a single pristine/real face. SBI has been shown to generalize even better than Face-Xray to unseen deepfakes. SLADD [5] used an adversarial training strategy to find the most difficult BI configuration and trained a classifier to predict the forgeries. In PCL [71], an inconsistency generator was used to synthesize forged data and patch-wise consistencies were later exploited to classify an image as either real or fake. Finally, OST [48] proposed a test-sample-specific auxiliary task, pseudo-training samples, and meta-learning to improve performance on identifying forgeries created by unseen methods.

One-class self-supervised anomaly detection (AD). AD, also referred to as out-of-distribution (OOD) detection, is an established research area, for which many techniques have been presented in the literature, including the one-class support vector machine (OC-SVM) [58], support vector data description (SVDD) [64], deep OC-NN [53] and multiple GAN-based methods [2, 34]. Very recently, self-supervised learning (SSL) has been successfully adopted for one-class AD [31, 35, 62], with highly competitive results [2, 34, 53]. In this setup, a (deep) model is commonly learned to solve an auxiliary (pretext) task in an SSL fashion. Then, to *classify* a given input sample as either anomalous or normal, the result of the auxiliary task is evaluated with the provided test sample. Because the model is trained with normal data only, the assumption is that the model will perform well on normal data but fare poorly on anomalies. While different tasks were considered in the literature as pretexts [31, 44, 62], existing one-class AD models for deepfake detection, such as OC-FakeDect [37], relied exclusively on data reconstruction to facilitate the detection process.

Evenly-distributed prototypes. Evenly-distributed points on a hypersphere maximize the average inter-class distance when being used as class centroids. Formally, in a Euclidean space $\mathbb{R}^n$, a set of $K \in [2, \ldots, n+1]$ vectors $\{\boldsymbol{p}_1 \ldots \boldsymbol{p}_K\}$ vectors are evenly-distributed on a unit hypersphere such that, $\forall\, i \neq j,\ \boldsymbol{p}_i \cdot \boldsymbol{p}_j = -1/(K-1)$. When $K = n+1$, Lange and Wu [42] proposed an analytical expression over the vertices of a regular simplex to generate n points in $\mathbb{R}^n$. Recently, [24] showed that contrastive learning objectives reach their minimum once the representations of each class collapse to the vertices of a regular simplex.

3. Methodology

Unlike competing methods that typically try to discriminate between real faces and (synthetically-generated) fakes, SeeABLE learns low-dimensional representations (hard prototypes) of synthetically-generated local image perturbations (soft discrepancies) and utilizes a prototype-matching procedure during inference to derive a (anomaly) score for the detection task. From a high-level perspective, SeeABLE first uses a diverse set of augmentation techniques to create the soft discrepancies (§3.1) and then trains a deep network to map them to a set of prototypes distributed evenly on a hypersphere, as also illustrated in Figure 2. In the following sections, we detail both, the training (§3.2) and inference stages (§3.3) of SeeABLE.

3.1. Generating soft discrepancies (SD)

We use a parameterized data augmentation technique, denoted as sd, to create synthetic face images with subtle soft discrepancies (SDs). Formally, given a dataset of real face images $\mathcal{D}_{real} = \{\mathbf{I}_i\}_{i=1}^{N}$, we generate a set of faces with synthesized soft discrepancies $\mathcal{D}_{sd}$ as follows:

$$\mathcal{D}_{sd} = \{(\underbrace{sd(\mathbf{I}_i, \tilde{y}_{iloc}, \tilde{y}_{itype})}_{\text{Face with a soft discrepancy}}, \underbrace{lbl(\tilde{y}_{iloc}, \tilde{y}_{itype})}_{\text{self-supervised signal}})\}, \quad (1)$$

where $\mathbf{I}_i$ is the i-th real image, $\tilde{y}_{itype} \in [1 \,..\, N_{type}]$ denotes the type of soft discrepancy to be used for perturbing $\mathbf{I}_i$, and $\tilde{y}_{iloc} \in [1 \,..\, N_{loc}]$ stands for a label encoding the (discrete) location of the soft discrepancy. Additionally, we use the generated ground truth, $\tilde{y}_{iloc}$ and $\tilde{y}_{itype}$, to construct a single label $\tilde{y}_i \in [1 \,..\, N_{loc} \times N_{type}]$ for each image in $\mathcal{D}_{sd}$ in a self-supervised fashion, as follows:

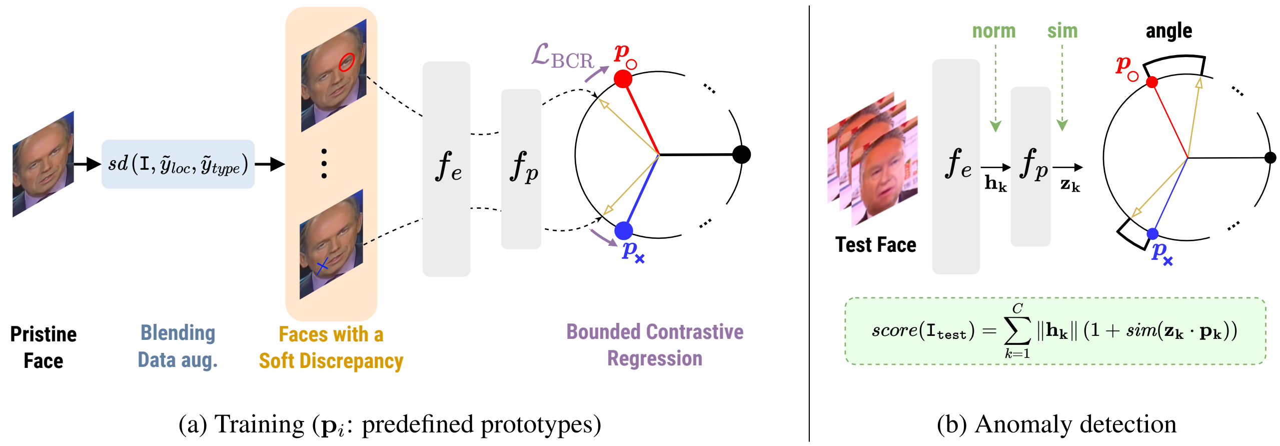
$$\tilde{y}_i = lbl(\tilde{y}_{iloc}, \tilde{y}_{itype}) = \tilde{y}_{iloc} \times N_{type} + \tilde{y}_{itype} \quad (2)$$

and $\tilde{y}_{iloc} = pos(\tilde{y}_i)$ and $\tilde{y}_{itype} = type(\tilde{y}_i)$ to invert the mapping. The presented process serves as a data factory and allows us to generate massive amounts of perturbed faces with corresponding labels for training the proposed model.

a) Soft-discrepancy generation. SeeABLE uses a **blending** operation to generate locally perturbed facial images for training. Given a source and target face image $\mathbf{I}^s, \mathbf{I}^t \in [0, 255]^{W \times H \times 3}$, and a blending mask $\mathbf{M} \in [0,1]^{W \times H}$, the blending operation is defined as follows:

$$blend\left(\mathbf{M}, \mathbf{I}^s, \mathbf{I}^t\right) = \mathbf{M} \odot \mathbf{I}^s + (1 - \mathbf{M}) \odot \mathbf{I}^t, \quad (3)$$

where $\odot$ is the element-wise Hadamard product. Prior work [5, 25, 45, 60] has adopted the above blending procedure to



(a) Training ($\mathbf{p}_i$: predefined prototypes) (b) Anomaly detection

Figure 2: **High-level overview of SeeABLE.** The proposed deepfake detector is trained in a one-class self-supervised learning setting using real face images only. Once trained, SeeABLE is able to provide an anomaly score that can be used for deepfake detection. $p_\circ$ and $p_\times$ are two different prototypes, the color encodes the position and the subscripts the discrepancy type.

produce *globally perturbed faces* that were then used to either fully substitute or to complement the dataset's actual deepfakes. Thus, the blending mask $\mathbf{M}$ was defined in a way that affected the appearance of the entire face [5]. SeeABLE differs from these existing methods in that it generates faces with (subtle) local perturbations that not only encourage the model to learn a robust detector, but also enable using an additional localization task for the training procedure. To generate the locally perturbed images, we utilize Eq. (3) with N_{loc} different masks $\mathbf{M}$ and N_{type} different augmentations, which results in the following augmented dataset for the training of SeeABLE for each input image $\mathbf{I}$:

$$sd(\mathbf{I}, \tilde{y}_{loc}, \tilde{y}_{type}) = blend\left(Loc(\mathbf{I}, \tilde{y}_{loc}), \mathbf{I}, \, Type(\mathbf{I}, \tilde{y}_{type})\right), \tag{4}$$

where Loc is a function that generates a mask at location $\tilde{y}_{loc}$ and $Type$ is a function that modifies the image using a specific augmentation technique defined by $\tilde{y}_{type}$.

b) Discrepancy location. To generate the local image perturbations with Eq. (4), SeeABLE requires suitable blending masks that ensure that only a specific region of the face is altered at the time. To this end, we define a *submask scheme*, that partitions the facial are into a number of subcomponents. We use a simple *grid* strategy for SeeABLE, where the facial region, defined by a set of facial landmarks, is divided into a grid with $N_{loc} = N_{rows} \times N_{cols}$ patches, where N_{rows} and N_{cols} represent the number of rows and columns in the grid, respectively (see Table 7(d) for an illustration). The function Loc, thus, returns a mask that corresponds to one of the N_{loc} patches, defined by $\tilde{y}_{loc}$. We note that this strategy was found to work better than more complex semantics-driven submask schemes, and was, therefore, also used in the design of SeeABLE.

c) Discrepancy type. To detect object-level anomalies or out-of-distribution samples, self-supervised AD methods, such as [61, 62], use strong augmentations like rotations, patch permutations or cutpaste. Because the artifacts introduced by deepfake-generation techniques are typically subtle, we avoid such strong augmentations in SeeABLE and utilize more suitable perturbation techniques that generate soft ("not too pronounced") discrepancies that enable our model to learn rich and robust features for the detection task. Specifically, we consider augmentations that yield discontinuities in both, the spatial and frequency domain. Thus, the number of different discrepancy types returned by the $Type$ function from Eq. (4) is $N_{type} = 2$ and is determined by $\tilde{y}_{type}$. Details of the concrete transformations used for the implementation are presented in Section 4.1.

d) Discrepancy invariant global transformations. In the literature on self-supervised representation learning, special care is typically taken to avoid trivial solutions, *e.g.*, when a network is trained to solve a pretext task to recognize the permutation of image patches and learns only the edge discontinuities instead of useful features. In SeeABLE, we avoid such trivial solutions (and over-fitting) by applying global *discrepancy-invariant* transformations $\mathcal{T}_{inv}$ to $\mathbf{I}$ before generating the local image perturbations. Because a family of global image transformations is used that do not impact the soft discrepancies, the model is encouraged to learn more general features that better capture the characteristics of the synthesized discrepancies.

3.2. Learning the proposed model

SeeABLE is trained in a multi-task fashion by minimizing the following learning objective:

$$\mathcal{L}_{\text{SeeABLE}} = \mathcal{L}_{\text{BCR}} + \lambda \, \mathcal{L}_{\text{GUI}}, \tag{5}$$

where $\mathcal{L}_{\text{BCR}}$ is the novel *Bounded Contrastive Regression* (BCR) loss that aims to push the soft-discrepancies to the predefined hard prototypes, $\mathcal{L}_{\text{GUI}}$ is the localization-related guidance loss that offers additional cues for the training procedure and infuses geometric constraints into the model optimization process, and λ is a balancing weight.

3.2.1 Supervised contrastive learning

Assume a classification task with K classes and a set of N training samples $\{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^{N}$, where a (deep) encoder f_e first outputs an intermediate representation $\mathbf{h}_i = f_e(\mathbf{x}_i)$ and the projector function f_p, usually an MLP, then computes the final low-dimensional embedding $\mathbf{z}_i = f_p(\mathbf{h}_i) \in \mathbb{R}^D$. Such models are often learned using **contrastive losses** that rely on the normalized temperature-scaled cross entropy (NT-Xent) [6] optimization objective, defined for a positive pair of training examples $(\mathbf{x}_i, \mathbf{x}_p)$ as:

$$\mathcal{L}_{\text{NT-Xent}}\left(\mathbf{z}_i, \mathbf{z}_p\right) = -\log \frac{e^{sim(\mathbf{z}_i, \mathbf{z}_p)/\tau}}{\sum_{j=1;\, j\neq i}^{N} e^{sim(\mathbf{z}_i, \mathbf{z}_j)/\tau}}, \tag{6}$$

where $sim(\mathbf{a}, \mathbf{b}) = \mathbf{a}\cdot\mathbf{b}/(\|\mathbf{a}\|\|\mathbf{b}\|)$ is the cosine similarity and τ is a temperature variable. Recently, a novel loss was proposed in [38] for fully supervised contrastive (SupCon) learning, where the training consists of two stages, where in **Stage 1** the encoder f_e and projector f_p are trained jointly to minimize:

$$\mathcal{L}_{\text{SupCon}} = \sum_{i=1}^{N} \frac{1}{|P(i)|} \sum_{p\in P(i)} \mathcal{L}_{\text{NT-Xent}}\left(\mathbf{z}_i, \mathbf{z}_p\right), \tag{7}$$

where $P(i) = \{p \in [1\ ..\ N]\backslash\{i\}\,|\,\tilde{y_p} = \tilde{y_i}\}$ is a set of indices of the same label as $\tilde{y_i}$, and $|\cdot|$ is the set's cardinality, and in **Stage 2** the encoder f_e is frozen and the projector f_p is replaced by a linear layer f_{lin}, which is trained to solve the classification task using the standard cross-entropy loss.

3.2.2 Bounded contrastive regression

One of the main contributions of this work is a novel loss for bounded contrastive regression (BCR). To introduce BCR, we first define a set $\{\mathbf{p}_1, \ldots, \mathbf{p}_K\}$ of K class prototypes and fix them to be *evenly distributed* on a hypersphere in accordance with the procedure from [42]. Using these predetermined *hard prototypes* as targets for the regression task, we then define the novel BCR loss as follows:

$$\mathcal{L}_{\text{BCR}} = \mathcal{L}_{\text{SupCon}} + \sum_{i=1}^{N} \frac{\mathcal{L}_{\text{NT-Xent}}\left(\mathbf{z}_i, \mathbf{p}_{\tilde{y}_i}\right)}{|P(i)|}, \tag{8}$$

where $\mathcal{L}_{\text{SupCon}}$ and $\mathcal{L}_{\text{NT-Xent}}$ are defined in Eq. (7) and Eq. (6). As can be seen, computing the BCR involves creating positive pairs that combine the embeddings $\mathbf{z}_i$ with their corresponding class prototypes $\mathbf{p}_{\tilde{y}_i}$. Such positive pairs encourage the supervised contrastive loss to generate clusters around the class prototypes. Our intuition here is that this will enable the regression model to generate bounded hyphserspherical cap (evenly distributed and equally sized) clusters for each class within the unit-hypersphere.

BCR properties. Given a sample $(\mathbf{x}_i, \tilde{y}_i)$ and its embedding $\mathbf{z}_i = f_p(f_e(\mathbf{x}_i))$, BCR has the following properties:

- **Optimal representation.** Since we choose the prototypes to be optimal (evenly distributed) for contrastive losses, Eq. (8) tends to its minimum when all embeddings collapse to their corresponding class prototypes:

$$\forall i \in [1\ ..\ N]\,,\ \mathbf{z}_i = \mathbf{p}_{\tilde{y}_i} \tag{9}$$

- **Regression.** By introducing a scalar variable $r_i \in \mathbb{R}^+$, for the cosine similarity between $\mathbf{z}_i$ and $\mathbf{p}_{\tilde{y}_i}$:

$$r_i = d_{sim}(\mathbf{z}_i, \mathbf{p}_{\tilde{y}_i}) = 1 - sim(\mathbf{z}_i, \mathbf{p}_{\tilde{y}_i}) \tag{10}$$

  we observe that Eq. (8) implicitly regress r_i to zero ($r_i \to 0$), with the equality to zero ($r_i = 0$) occurring with the optimal representation.
- **Efficient prediction.** Given an embedding $\mathbf{z}_i$, we can obtain its prediction (i.e., location and discrepancy type) through an efficient prototype-matching procedure, i.e.:

$$y_i = \arg\min_{k}\{sim\left(\mathbf{z}_i, \mathbf{p}_k\right) \,|\, k \in [1\ ..\ K]\} \tag{11}$$

3.2.3 Guidance loss

In order to further enhance the performance of SeeABLE, we introduce an additional guidance loss for the regression task that incorporates task-specific knowledge. To illustrate the idea, let us assume that the model produces a completely erroneous prediction instead of the expected ground truth, i.e., $y \neq \tilde{y_i}$, Since our the prototypes are distributed evenly, the distance to the incorrect class is always the same, regardless of error, i.e., $d_{sim}(\mathbf{z}_i, \mathbf{p}_{\tilde{y}_i}) = K/(K-1)$, resulting in an *equal treatment* of all prediction errors. To address this issue, we propose a loss that explicitly weights the distances r_i based on some prior knowledge encoded in G:

$$\mathcal{L}_{\text{GUI}} = \sum_{i\in[1..N]} G\left(y_i, \tilde{y}_i\right) \times r_i. \tag{12}$$

Model guidance with geometric constraints. In SeeABLE, we consider facial geometry and geometric constraints as sources of prior knowledge for G and incorporate these into the discrepancy localization objective of the proposed detector. The main idea here is to use lower penalties for errors that originate from facial symmetry (e.g., substitutions of the left for the right eye) than other types of localization errors. Similarly, mispredictions of the type of soft-discrepancy within a region should be penalized less than mispredictions of the discrepancy location. Formally, G is defined as follows:

$$G(y_i, \tilde{y}_i) = \begin{cases} 2^{-2} & \text{if } pos(y_i) = pos(\tilde{y}_i) \\ 2^{-1} & \text{else if } sym(pos(y_i)) = pos(\tilde{y}_i) \\ 2^{-0} \times d_{graph}(pos(y_i), pos(\tilde{y}_i)) & \text{otherwise} \end{cases} \tag{13}$$

where $sym(\cdot)$ returns the location of symmetric patch w.r.t. the vertical center axis of the image and d_{graph} is a graph-based distance between the location of two patches.

3.3. Inference: anomaly score computation

SeeABLE utilizes a deepfake detection score that is derived from the cosine similarity (Eq. (10)) with the trained prototypes, serving as a direct cue. Moreover, the norm is employed as an indirect cue of the model's confidence.We, therefore, define the anomaly score for a test image $\mathbf{I}_{\text{test}}$ as:

$$score(\mathbf{I}_{\text{test}}) = \sum_{k=1}^{K} \underbrace{\|\mathbf{h}_k\|}_{\text{indirect}} \times \underbrace{(1 + sim\,(\mathbf{z}_k, \mathbf{p}_k))}_{\text{direct}} \tag{14}$$

where $\mathbf{h}_k = f_e(sd(\mathbf{I}_{\text{test}}, pos(k), type(k)))$ and $\mathbf{z}_k = f_p(\mathbf{h}_k)/\|f_p(\mathbf{h}_k)\|$. Note that $1 + sim(\mathbf{z}_k, \mathbf{p}_k) = 2 - d_{sim}(\mathbf{z}_k, \mathbf{p}_k) \geq 0$.

4. Experiments

4.1. Implementation details

Model implementation. As in [60], we use EfficientNet-b4 [63] as the encoder f_e of SeeABLE, and a one-layer MLP (with $D = 128$ outputs) followed by a ℓ_2 normalization for the projection layer f_p. The number of vertices (prototypes) of the regular simplex in $\mathbb{R}^{K-1}$ is set to $K = 33$. SeeABLE is trained for 200 epochs with the SGD optimizer, a batch size of 6 and a learning rate of $1e^{-3}$ that is decayed to $1e^{-5}$ with a cosine scheduler. During training, the balancing parameter λ is first set to 0 and then increased gradually to 0.1 to strengthen the importance of the geometric constraint in later epochs. Once trained, SeeABLE requires around 15 ms to process one frame on a PC with an RTX 3060.

Considered transformations. For the global transformations $\mathcal{T}_{inv}$, we consider the following operations: (1) random translations of up to 3% and 1.5% of the image width and height, respectively, (2) random scaling (followed by center cropping) by up to 5%, and (3) random shifting of HSV channel values by up to 0.1. To generate the *SDs in the spatial domain*, we use: (1) shifting of RGB channel values by up to 20, (2) random shifting of HSV channel values by up to 0.3, and (3) random scaling of the brightness and contrast by a factor of up to 0.1. For the *SDs in the frequency domain*, one of the following operators is used: (1) down-sampling by a factor of 2 or 4, (2) application of a sharpening filter and blending with the original with an α value in the range $[0.2, 0.5]$, and (3) JPEG compression with a quality factor between 30 and 70. These hyperparameter values were selected in a way that resulted in visually subtle soft discrepancies without major artifacts, similarly to [60].

Data preprocessing. We use a pretrained RetinaFace [13] model for face detection and dlib [40] for locating 68 facial landmarks. The detected face regions are resized to 256×256 pixels (using bilinear interpolation) prior to the experiments. No special effort is made to align the faces across frames and the landmarks are used only to define the region-of-interest and the grid for the guidance loss.

Deepfake detection. During the inference phase, we uniformly sample 30 frames per test video. The anomaly score is first computed for each frame separately using Eq. (14), and the anomaly score for the entire video is obtained by averaging the frame-level scores.

4.2. Experimental setup

Experimental datasets. As in [5, 57, 62, 71], we use the *FaceForensics++* (FF++) dataset to train our model. The FF++ dataset [57] contains 1000 videos that are split into three groups: 720 videos for training, 140 for validation and 140 for testing. We utilize only the pristine training videos for learning SeeABLE and sample at most 1 frame per video when constructing batches for the optimization procedure. To evaluate SeeABLE in cross-manipulation settings, we adopt the deepfake (test) part of FF++, where each video is generated using one of the four deepfake-generation techniques: Deepfakes (DF) [12], Face2Face (F2F) [66], NeuralTextures (NT) [65] and FaceSwap (FS) [19].

To demonstrate the performance of SeeABLE in cross-dataset settings, three additional datasets are adopted, i.e., Celeb-DF-v2 (CDF-v2) [47], DeepFake Detection Challenge preview (DFDC-p), and DeepFake Detection Challenge public (DFDC) [15]. The CDF-v2 dataset [47] consists of 590 real and $5,639$ deepfake celebrity videos, generated using a sophisticated deepfake approach. The DFDCp dataset contains over 5000 videos (original and fake) and features two deepfake generation methods, while the DFDC dataset [15] comprises over $128,000$ video sequences with more than $100,000$ deepfakes of different quality.

Performance indicators. In line with standard evaluation methodology [5,60,71], we use the Area Under the Receiver Operating Characteristic Curve (AUC) to evaluate the performance of our detector in a threshold-free manner.

SoTa baselines. We compare SeeABLE to multiple SoTA competitors: (1) *pseudo-deepfake* based detection methods, i.e., DSP-FWA [46], Face X-ray [45], SLADD [5], PCL [71], SBI [60], and OST [48] (2) *video-based techniques*, i.e., Two-branch [50] and LipForensics [28], (3) *transformer-based* methods, i.e., UIA-ViT [74], FTCN-TT [72], and LTTD [26], and (4) two versions of the *one-class* OC-FakeDect model [37]. For a fair comparison, results from the original papers are reported (where available).

4.3. Performance evaluation

Cross-dataset evaluation. Table 2 compares the performance of SeeABLE with SoTA detectors in a cross-dataset scenario. Here, all models are trained on FF++ and tested on datasets not seen during training. As can be observed, SeeABLE yields the best overall (average) performance, while being among the simplest of all considered detectors. Unlike other competitors, the proposed model does not rely on adversarial training schemes or availability of deepfake examples during training, but still leads to highly competitive

Method	Pristine only	Test set - AUC (%)			
		CDFv2	DFDC	DFDCp	Avg.
DSP-FWA [46]	✓	69.3	-	-	69.3
Two-branch [50]		76.6	-	-	76.6
LipForensics [28]		82.4	73.5	-	77.9
Face X-ray [45]		79.5	65.5	-	72.5
SLADD [5]		79.7	-	76.0	77.8
PCL+I2G [71]	✓	**90.0**	67.5	74.4	77.3
SBI† [60]	✓	85.9	69.8	74.9	76.9
OST [48]		74.8	-	83.3	79.1
UIA-ViT [74]	✓	82.4	-	75.8	79.1
FTCN-TT [72]		86.9	74.0	-	80.4
LTTD [26]	✓	89.3	-	80.4	-
SeeABLE (ours)	✓	87.3	**75.9**	**86.3**	**83.2**

†SBI was re-evaluated using the official code with $\text{M}_{\text{ConvexHull}}$.

Table 2: **Comparison of SeeABLE and SoTA methods in the cross-dataset scenario.** For a fair comparison, the reported results are cited directly from the original papers.

Method	Test set - AUC (%)				
	DF	F2F	FS	NT	Avg.
OC-FD1† [37]	86.2	70.7	84.8	95.3	84.2
OC-FD2† [37]	88.4	71.2	86.1	97.5	85.8
Face X-ray [45]	-	-	-	-	87.3
SBI [60]	97.5	89.0	96.4	82.8	91.4
OST [48]	-	-	-	-	98.2
SLADD [5]	-	-	-	-	98.4
SeeABLE (ours)	99.2	98.8	99.1	96.9	**98.5**

†OC-FD1 and OC-FD1 refers to two versions of OC-FakeDect

Table 3: **Cross-manipulation evaluation on FF++ HQ.** SeeABLE achieves SoTA results on all subsets of FF++. SLADD, OST, Face Xray report only the average result.

results. SeeABLE convincingly outperforms all methods on the DFDC and DFDCp datasets, including all pseudo-deepfake based detectors, video-based models, one-shot techniques and transformer-based models, while being rivaled on CDFv2 only by the video transformer based detector LTTD [26] and the PCL-I2G technique [71]. For a fair interpretation of the results, it should be noted that SBI was evaluated using the official code associated with $\text{M}_{\text{ConvexHull}}$ (see Table 7(a)) since the results reported in the original paper were obtained with a more sophisticated masking scheme and a Sharpness Aware Minimization (SAM) [22] procedure, which commonly improves performance at the cost of doubling the training time. We also note that some of the competitors do not provide results for all test datasets, but SeeABLE is still the top performer even if the average AUC scores are computed only across the available results. For instance, the average AUC of SeeABLE and LTTD on CDFv2 and DFDCp are 86.8% and 84.8%, respectively.

Cross-manipulation evaluation. A key aspect of deepfake detectors is their generalization to different manipulation techniques. Following the evaluation protocol from [71],

Encoder (f_e)	Test set - AUC (%)				
	DF	F2F	FS	NT	Avg.
ResNet50 [29]	96.2	94.7	95.2	92.6	94.7
Xception [7]	94.5	95.0	96.4	94.2	95.0
EfficientNet-b4 [63]	**99.2**	**98.8**	**99.1**	**96.9**	**98.5**

Table 4: **Performance of SeeABLE with different backbones.** Results are shown in terms of AUC (in %) on FF++.

we evaluated SeeABLE on the four manipulation methods of FF++, i.e., DF, F2F, FS, and NT. As in all experiments presented in this paper, the raw version of FF++ is used for training and the HQ version is considered for testing. As can be seen from Table 3, SeeABLE outperforms all competing detectors on all four manipulation types with an average AUC of 98.5%. Especially interesting here is the comparison to SeeABLE's closest competitors, the one-class OC-FakeDect detectors, which the proposed model outperforms by a margin of more than 12%.

4.4. Ablation study

Backbone impact. In Table 4, we evaluate the effect of different (backbone) encoder architectures f_e on the performance of SeeABLE, i.e., ResNet-50 [29], Xception [7] and EfficientNet-b4. We observe that the best overall performance is obtained with EfficientNet-b4, which outperforms the runner-up, Xception [7], by 3.5%, and the weakest model (ResNet-50) by 3.8%. In general, larger and more powerful encoders lead to better generalization, but even with the weaker backbones, SeeABLE still outperforms many of the SoTA competitors from Table 3. These results suggest that SeeABLE is applicable to different backbone models and is expected to further benefit from future developments in model topologies.

Regression vs. classification. We explore in Table 5 the impact of learning SeeABLE within the proposed regression task (using $\mathcal{L}_{\text{BCR}}$), as opposed to classification tasks (learned with cross-entropy $\mathcal{L}_{\text{CE}}$ and supervised-contrastive $\mathcal{L}_{\text{SupCon}}$ losses). For a fair comparison, we compare all strategies with and without the geometric constraint ($\mathcal{L}_{\text{GUI}}$) and the best performing hyperparameters.

As can be seen, $\mathcal{L}_{\text{BCR}}$ convincingly outperforms $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{SupCon}}$ with a performance difference of 7.3% and 4.3%. When adding the geometric constraint $\mathcal{L}_{\text{GUI}}$, an additional improvement of 2.2% for $\mathcal{L}_{\text{BCR}}$, 1.7% for $\mathcal{L}_{\text{CE}}$, and 1.9% for $\mathcal{L}_{\text{SupCon}}$ can be observed. This illustrates the strength of the proposed regression-based learning objective. We conjecture that the drop in detection performance when changing from the regression loss to the classification losses is due to the fact that: (1) the representation learned through $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{SupCon}}$ is not optimized for the cosine-similarity-based scoring; (2) some useful characteristics, such as the hardness-aware property [68] of the contrastive loss, were lost, ultimately leading to suboptimal detection results.

Method	Test set - AUC (%)				
	DF	F2F	FS	NT	Avg.
$\mathcal{L}_{\text{CE}}$	96.8	91.7	86.7	80.8	89.0
$\mathcal{L}_{\text{CE}}$ + $\mathcal{L}_{\text{GUI}}$	96.3	92.6	89.8	84.0	90.7
$\mathcal{L}_{\text{SupCon}}$	96.9	94.4	91.3	85.4	92.0
$\mathcal{L}_{\text{SupCon}}$ + $\mathcal{L}_{\text{GUI}}$	97.9	97.1	93.1	87.4	93.9
$\mathcal{L}_{\text{BCR}}$	97.4	96.1	96.4	95.5	96.3
$\mathcal{L}_{\text{BCR}}$ + $\mathcal{L}_{\text{GUI}}$	**99.2**	**98.8**	**99.1**	**96.9**	**98.5**

Table 5: **Learning in a regression vs. classification settings.** Shown are AUC scores (in %) on FF++.

$\mathcal{L}_{\text{BCR}}$	$\mathcal{L}_{\text{GUI}}$	λ	DF	F2F	FS	NT	Avg.
✓	-	-	97.3	96.1	96.3	95.4	96.3
-	✓	-	90.3	87.0	87.9	82.0	86.8
✓	✓	const.	97.8	96.0	97.1	94.4	96.4
✓	✓	$\searrow$	98.7	98.7	95.4	97.8	97.6
✓	✓	$\nearrow$	**99.2**	**98.8**	**99.1**	**96.9**	**98.5**

Table 6: **Impact of loss terms ($\mathcal{L}_{\text{BCR}}$, $\mathcal{L}_{\text{GUI}}$) and balancing strategies (λ) on performance (AUC in %) on FF++.**

Contribution of different losses. In Table 6, we investigate the impact of the two loss terms utilized to learn SeeABLE, i.e., $\mathcal{L}_{\text{BCR}}$ and $\mathcal{L}_{\text{GUI}}$, as well as the strategy used to balance the two during training. Several cases are considered: (1) a fixed trade-off with λ = const. = 0.1, (2) increasing λ linearly from 0 to 0.1 during training - marked $\lambda = \nearrow$, (3) decreasing λ linearly from 0.1 to 0 during training - marked $\lambda = \searrow$. In general, we see that $\mathcal{L}_{\text{BCR}}$ and $\mathcal{L}_{\text{GUI}}$ complement each other and better results are obtained when both are considered jointly, as opposed to either one alone. Additionally, we observe that the best overall performance is achieved when the importance of the geometric constraint is gradually increased during training.

Effect of submask generation strategies. In Table 7, we evaluate the impact of different submask-generation strategies on the performance of SeeABLE on the DFDC dataset. Specifically, we consider: (1) the single global mask strategy $\text{M}_{\text{ConvexHull}}$ used in [45, 60], (2) the semantics-guided strategy SM_{SLADD} used in SLADD [5], a baseline mesh-grid strategy $\text{SM}_{\text{Meshgrid}}$ and the 4×4 patch-based strategy SM_{Grid} used with SeeABLE. The four strategies are illustrated in Table 7(a)-(d). As can be seen, the best performance is obtained with SM_{grid} with 4 rows and 4 columns. In $\text{SM}_{\text{Grid 4x4}}$, 33 hard prototypes are used ($C = 1 + 2 \times 4 \times 4$). We note that the similar results are obtained with $\text{SM}_{\text{Grid 3x3}}$ and $\text{SM}_{\text{Grid 5x5}}$. In SeeABLE, a good submask-generation strategy should have the following properties:

- $\mathcal{A}_1$ *(full-coverage)*: the (sub)masks should cover the whole face to avoid blind spots and loss of information.
- $\mathcal{A}_2$ *(no-overlap)*: submasks should not overlap with others, as this introduces to uncertainty with respect to the prototype to be regressed to.
- $\mathcal{A}_3$ *(balanced)*: submasks should have similar sizes.

	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$	Avg.
(a) $\text{M}_{\text{ConvexHull}}$	✓	✓		58.5
(b) SM_{SLADD}			✓	68.3
(c) $\text{SM}_{\text{Meshgrid}}$	✓	✓		63.8
(d) $\text{SM}_{\text{Grid 4x4}}$	✓	✓	✓	**75.9**

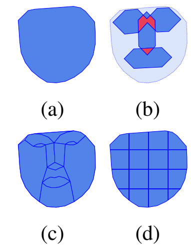


Table 7: **Performance of different submask schemes.** Results are presented in terms of AUC scores (in %). The considered submask schemes are shown on the right.

Figure 3: **Visual examples of real (in green) and fake (in red) faces with different anomaly scores.**

As can be seen from Table 7, SM_{SLADD} has overlapping submasks (marked red) and does not cover the entire face, whereas the mesh-grid strategy has submasks of unequal size. The grid strategy is the only one satisfying all the above properties, which explains its advantage over the competing submask-generation schemes.

5. Qualitative analysis

In Figure 3, we show a cross-section of visual examples of real and fake samples with different anomaly scores to analyze the **strengths and limitations** of SeeABLE. As can be seen, the model performs well overall and generates expected anomaly scores. However, in a small number of cases it also produces low scores for real images that come with deepfake-like artifacts (first image) and large scores for high-quality deepfakes, such as the one on the far right.

6. Conclusion

In this paper, we presented a powerful new deepfake detector, called SeeABLE, that successfully formalizes deepfake detection as a one-class self-supervised anomaly detection task. The key idea behind the model is to push soft discrepancies synthesized from real faces towards predefined evenly-distributed prototypes using novel learning objectives. The results of our experiments in cross-dataset and cross-manipulation scenarios point to superior generalizability of SeeABLE over current SoTA methods. Future work involves improving SeeABLE, *e.g.*, by considering additional losses and pretext tasks.

Ethic Statement. We did not identify any potential negative societal impacts of the proposed research. All face images used in this paper were obtained from public datasets.

Acknowledgement. Supported by ARRS P2-0250 and P2-0214 and the Eutopia PhD funding scheme.

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. MesoNet: a compact facial video forgery detection network. In *IEEE International Workshop on Information Forensics and Security*, dec 2018. 1, 2

[2] Samet Akcay, Amir Atapour-Abarghouei, and Toby P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *ACCV*, 2018. 3

[3] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. Deepfake video detection through optical flow based cnn. In *CVPR-W*, 2019. 2

[4] Shivangi Aneja and Matthias Nießner. Generalized zero and few-shot transfer for facial forgery detection. *arXiv preprint arXiv:2006.11863*, 2020. 2

[5] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In *CVPR*, 2022. 1, 2, 3, 4, 6, 7, 8

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 5

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 1, 7

[8] S M Abrar Kabir Chowdhury and Jahanara Islam Lubna. Review on deep fake: A looming technological threat. In *2020 11th International Conference on Computing, Communication and Networking Technologies*, pages 1–7, 2020. 1

[9] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection. In *5th ACM Workshop on Information Hiding and Multimedia Security*, page 159–164, 2017. 1

[10] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil Jain. On the detection of digital face manipulation. In *CVPR*, 2020. 1

[11] Sowmen Das, Selim Seferbekov, Arup Datta, Md Islam, Md Amin, et al. Towards solving the deepfake problem: An analysis on improving deepfake detection using dynamic face augmentation. In *ICCV*, pages 3776–3785, 2021. 2

[12] DeepFakes. www.github.com/deepfakes/faceswap Accessed 2022-11-11. 1, 6

[13] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *CVPR*, pages 5202–5211, 2020. 6

[14] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *ICLR*, 2017. 1

[15] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2011.02578*, 2020. 1, 2, 6

[16] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019. 1, 2

[17] Shichao Dong, Jin Wang, Jiajun Liang, Haoqiang Fan, and Renhe Ji. Explaining deepfake detection by analysing image matching. In *ECCV*, 2022. 2

[18] Mengnan Du, Shiva Pentyala, Yuening Li, and Xia Hu. Towards generalizable deepfake detection with locality-aware autoencoder. In *CIKM*. ACM, 2020. 1

[19] FaceSwap. www.github.com/deepfakes/faceswap Accessed 2022-11-11. 1, 6

[20] Jianwei Fei, Yunshu Dai, Peipeng Yu, Tianrun Shen, Zhihua Xia, and Jian Weng. Learning second order local anomaly for general face forgery detection. In *CVPR*, 2022. 2

[21] Chao Feng, Ziyang Chen, and Andrew Owens. Self-supervised video forensics by audio-visual anomaly detection. *arXiv preprint arXiv:2301.01767*, 2023. 2

[22] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021. 7

[23] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, 2014. 1

[24] Florian Graf, Christoph D. Hofer, Marc Niethammer, and Roland Kwitt. Dissecting supervised contrastive learning. In *ICML*, 2021. 3

[25] Jiazhi Guan, Hang Zhou, Mingming Gong, Youjian Zhao, Errui Ding, and Jingdong Wang. Detecting deepfake by creating spatio-temporal regularity disruption. *arXiv preprint arXiv:2207.10402*, 2022. 3

[26] Jiazhi Guan, Hang Zhou, Zhibin Hong, Errui Ding, Jingdong Wang, Chengbin Quan, and Youjian Zhao. Delving into sequential patches for deepfake detection. In *NeurIPS*, 2022. 2, 6, 7

[27] Hui Guo, Shu Hu, Xin Wang, Ming-Ching Chang, and Siwei Lyu. Eyes tell all: Irregular pupil shapes reveal gan-generated faces. In *ICASSP*, pages 2904–2908, 2022. 2

[28] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. In *CVPR*, 2021. 6, 7

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 7

[30] Yang He, Ning Yu, Margret Keuper, and Mario Fritz. Beyond the spectrum: Detecting deepfakes via re-synthesis. In *IJCAI*, 2021. 2

[31] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, 2019. 3

[32] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1

[33] Chih-Chung Hsu, Yi-Xiu Zhuang, and Chia-Yen Lee. Deep fake image detection based on pairwise learning. *Applied Sciences*, 10(1), 2020. 1

[34] Marija Ivanovska and Vitomir Štruc. Y-gan: Learning dual data representations for efficient anomaly detection. *arXiv preprint arXiv:2109.14020*, 2021. 3

[35] Loic Jezequel, Ngoc-Son Vu, Jean Beaudet, and Aymeric Histace. Efficient anomaly detection using self-supervised multi-cue tasks. *IEEE Trans. Image Processing*, 2023. 3

[36] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. DeeperForensics-1.0: A large-scale

dataset for real-world face forgery detection. In *CVPR*, 2020. 1

[37] Hasam Khalid and Simon S. Woo. Oc-fakedect: Classifying deepfakes using one-class variational autoencoder. In *CVPRW*, pages 2794–2803, 2020. 2, 3, 6, 7

[38] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020. 5

[39] Minha Kim, Shahroz Tariq, and Simon S. Woo. Fretal: Generalizing deepfake detection using knowledge distillation and representation learning. In *CVPRW*, 2021. 1, 3

[40] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 6

[41] Pavel Korshunov and Sebastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018. 1

[42] Kenneth Lange and Tong Tong Wu. An mm algorithm for multicategory vertex discriminant analysis. *Journal of Computational and Graphical Statistics*, 17(3):527–544, 2008. 3, 5

[43] Chuqiao Li, Zhiwu Huang, Danda Pani Paudel, Yabin Wang, Mohamad Shahbazi, Xiaopeng Hong, and Luc Van Gool. A continual deepfake detection benchmark: Dataset, methods, and essentials. In *WACV*, 2023. 3

[44] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *CVPR*, 2021. 3

[45] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *CVPR*, 2020. 1, 2, 3, 6, 7, 8

[46] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. In *CVPRW*, 2018. 2, 6, 7

[47] Yuezun Li, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics. In *CVPR*, 2020. 1, 2, 6

[48] Chen Liang, Zhang Yong, Song Yibing, Jue Wang, and Lingqiao Liu. Ost: Improving generalization of deepfake detection via one-shot test-time training. In *NeurIPS*, 2022. 1, 2, 3, 6, 7

[49] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain. In *CVPR*. IEEE, 2021. 2

[50] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed. Two-branch recurrent network for isolating deepfakes in videos. In *ECCV*, 2020. 1, 6, 7

[51] Yisroel Mirsky and Wenke Lee. The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)*, 54(1):1–41, 2021. 2

[52] Philip Oltermann. European politicians duped into deepfake video calls with mayor of kyiv. *The Guardian*, 2022. 1

[53] Poojan Oza and Vishal M. Patel. One-class convolutional neural network. *IEEE Signal Processing Letters*, 26(2):277–281, 2019. 3

[54] Pramuditha Perera, Poojan Oza, and Vishal M Patel. One-class classification: A survey. *arXiv preprint arXiv:2101.03064*, 2021. 2

[55] Julien Pourcel, Ngoc-Son Vu, and Robert M. French. Online task-free continual learning with dynamic sparse distributed memory. In *ECCV*, 2022. 3

[56] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *ECCV*, 2020. 2

[57] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, 2019. 1, 2, 6

[58] Bernhard Scholkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *NIPS*, 2000. 3

[59] Rui Shao, Tianxing Wu, and Ziwei Liu. Detecting and recovering sequential deepfake manipulation. In *ECCV*, 2022. 2

[60] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *CVPR*, 2022. 1, 2, 3, 6, 7, 8

[61] Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. Learning and evaluating representations for deep one-class classification. In *ICLR*, 2021. 4

[62] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. CSI: Novelty detection via contrastive learning on distributionally shifted instances. In *NeurIPS*, 2020. 3, 4, 6

[63] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 6, 7

[64] David M.J. Tax and Robert P.W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004. 3

[65] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics*, 2019. 1, 6

[66] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. *Communications of ACM*, 2020. 1, 6

[67] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 64:131–148, 2020. 1, 2

[68] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *CVPR*, 2021. 7

[69] Jun Wei, Shuhui Wang, and Qingming Huang. F3net: Fusion, feedback and focus for salient object detection. In *ECCV*, 2020. 2

[70] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP*, 2019. 1

[71] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *ICCV*, 2021. 1, 2, 3, 6, 7

[72] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. Exploring temporal coherence for more general video face forgery detection. In *ICCV*, 2021. 2, 6, 7

[73] Gu Zhihao, Yao Taiping, Chen Yang, Ding Shouhong, and Ma Lizhuang. Hierarchical contrastive inconsistency learning for deepfake video detection. In *ECCV*, 2022. 2

[74] Wanyi Zhuang, Qi Chu, Zhentao Tan, Qiankun Liu, Haojie Yuan, Changtao Miao, Zixiang Luo, and Nenghai Yu. Uia-vit: Unsupervised inconsistency-aware method based on vision transformer for face forgery detection. In *ECCV*, 2022. 2, 6, 7

[75] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. Wild deep fake: A challenging real-world dataset for deepfake detection. In *ACM Int. Conference on Multimedia*, 2021. 1