CS588 Paper Presentation 2

# SeeABLE: Soft Discrepancies and Bounded Contrastive Learning for Exposing Deepfakes

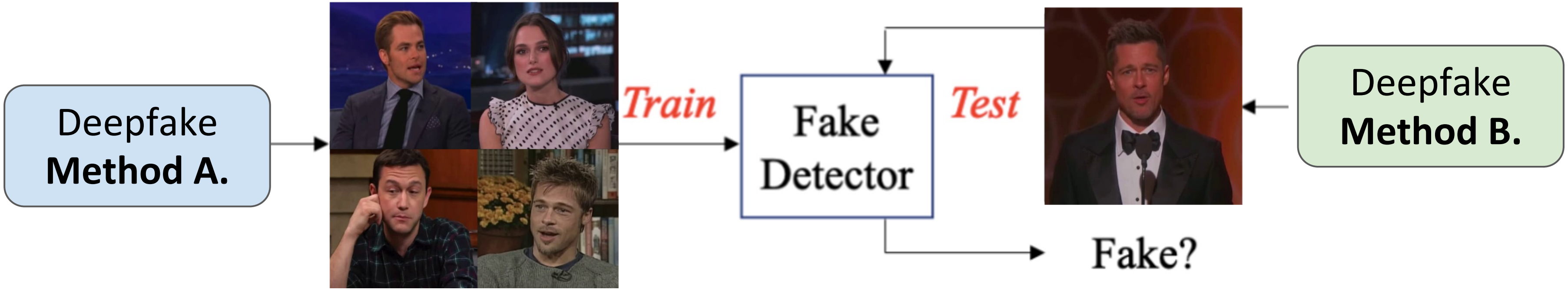
ICCV 2023

Jumin Lee

2024. 05. 22.

SGVR Lab KAIST

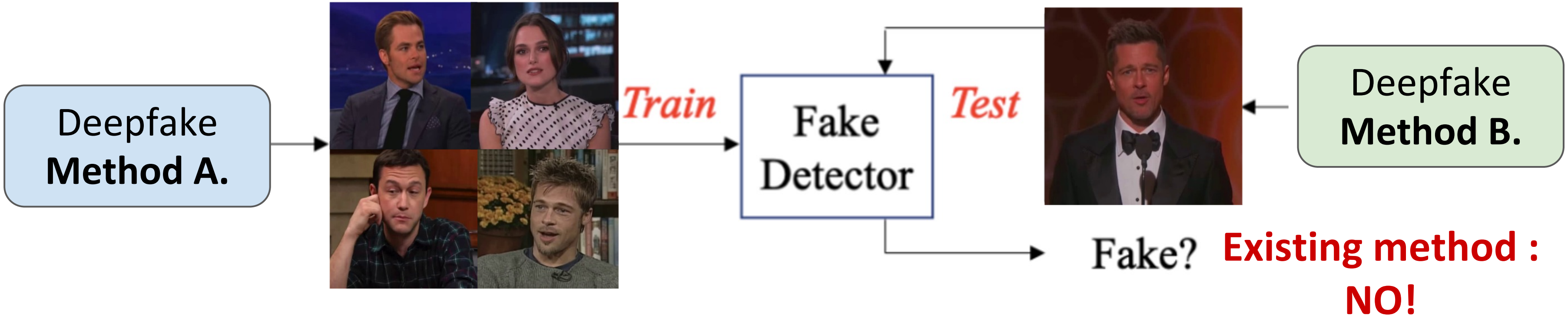
# Goal




- **Deepfakes Detection with Generalizability**

- By training only with real images, the model detects when it gets a fake image as input at test time.
- Achieve high performance on any deepfakes.

# Motivation




- **Generative models** are spreading rapidly, and there is a growing concern about them - <u>human faces have been a particularly target</u> for such models.
- However, the **existing method**(deepfakes classifier) to distinguish between real and fake images **doesn't work with new deepfake models.**

# Proposed Method

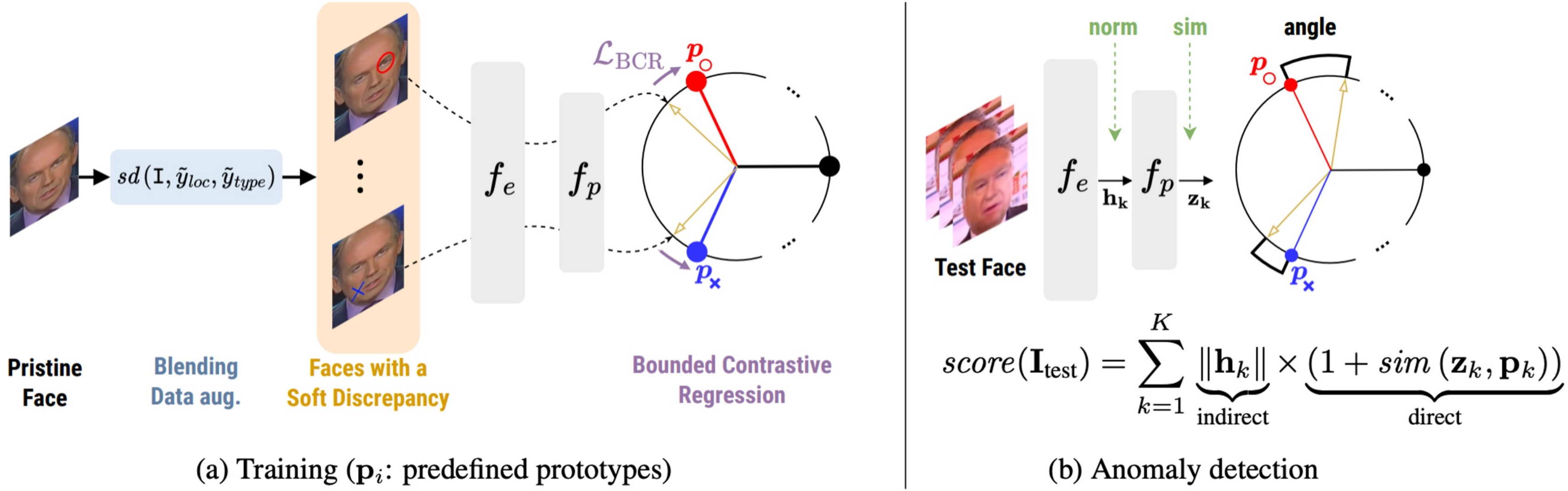
- **One-class self-supervised learning** using real face images only.

- **Soft discrepancy** : Different local perturbations introduced into real images.
- **Pretext Task**: Through the localization of the soft discrepancy region and the detection of different augmentation methods.

< Examples of faces with soft-discrepancies >

The perturbed area of the four images is within the circle with different augmentation.

# Proposed Method

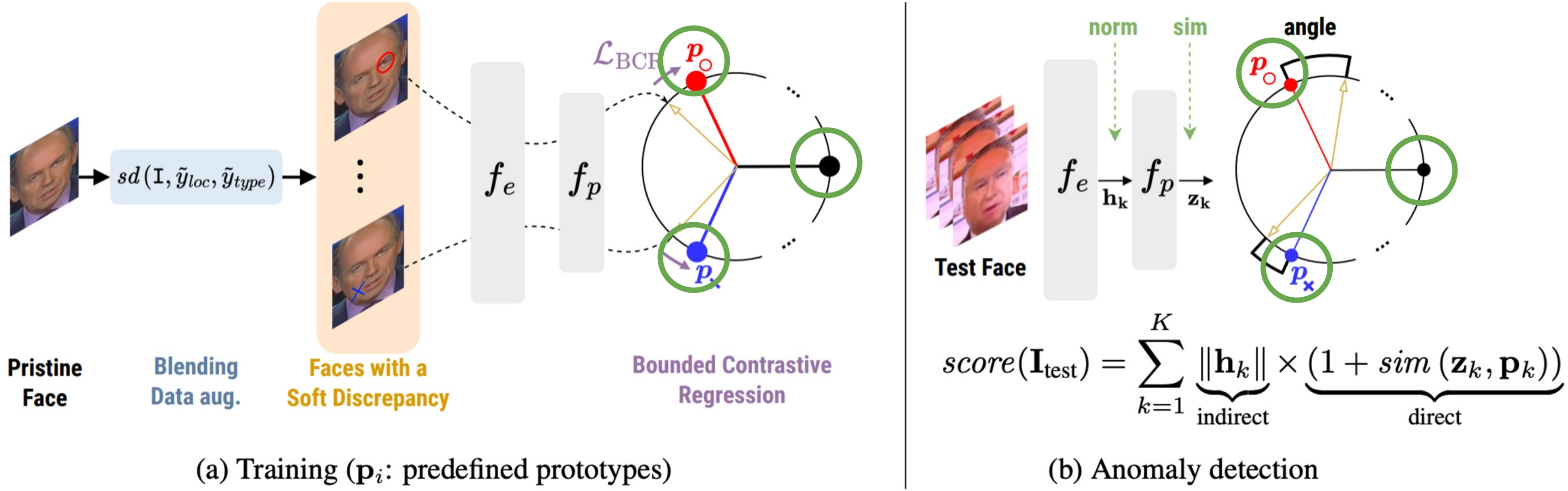



(a) Training ($\mathbf{p}_i$: predefined prototypes)

(b) Anomaly detection

- By training these 2 pretext tasks with a single multi-task regressor, the generated soft discrepancies are **pushed towards a set of target prototypes**.
- Once trained, it is able to provide an **anomaly score** for deepfake detection.

# Proposed Method




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

(b) Anomaly detection

- **Target prototypes** (NOT learnable)

  : Generated as **evenly distributed points** on a unit hypersphere, with the number of prototypes determined by the combinations of discrepancy locations and types.

# Contribution

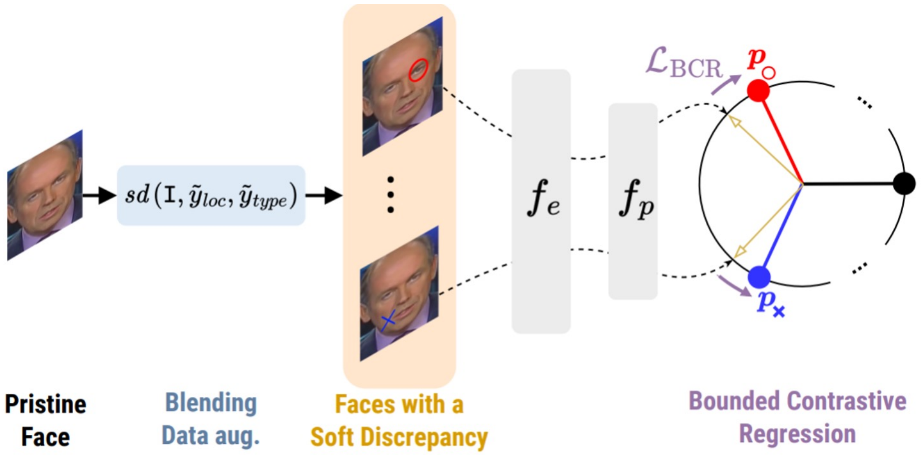
- Treat deepfake detection as an **out-of-distribution (OOD) detection** task.
- Introduce **Bounded Contrastive Regression (BCR) loss** and Guthat to train 2 pretext tasks
  - <u>localization</u> of the soft discrepancy region
  - <u>detection</u> of different augmentation methods.
- Demonstrate the superior **generalization capabilities** compared to existing (SoTA) deepfake detectors.

# Method

# Method

1. Each image is transformed with a **soft discrepancy.**
2. Passed through the $f_e$ and $f_p$ to generate its **embedding**.
3. **Bounded Contrastive Regression** and **Guidance loss** map these embeddings to the corresponding hard prototypes on the hypersphere.

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




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

# Method #1. Soft Discrepancy

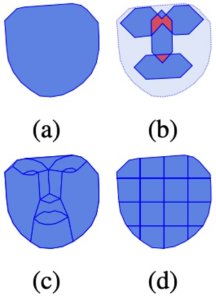
$$blend\left(\mathbf{M}, \mathbf{I}^s, \mathbf{I}^t\right) = \mathbf{M} \odot \mathbf{I}^s + (1 - \mathbf{M}) \odot \mathbf{I}^t$$

- Soft discrepancies
  : **Unique location and type combinations of perturbation** <u>for 2 pretext task.</u>

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

- Location : $N_{loc} = N_{rows} \times N_{cols}$

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



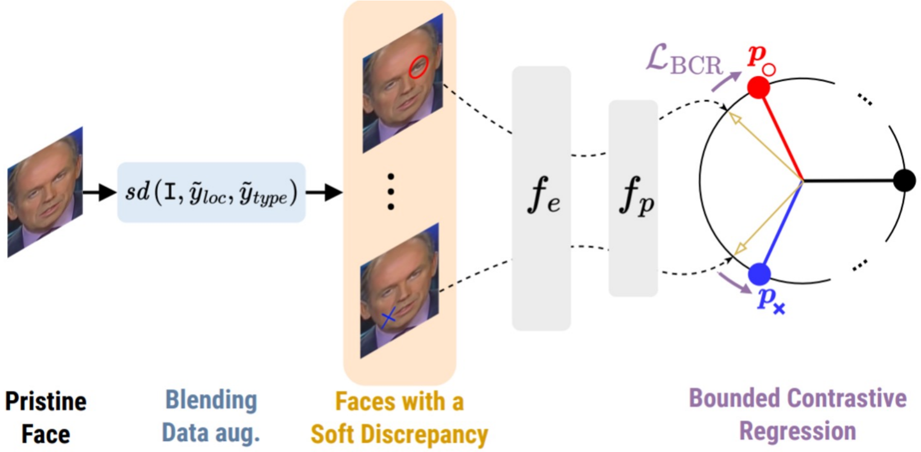

- Type : $N_{type} = 2$
- Spatial and frequency domain perturbations.

- Construct a single label : $\tilde{y}_i = lbl(\tilde{y}_{i_{loc}}, \tilde{y}_{i_{type}}) = \tilde{y}_{i_{loc}} \times N_{type} + \tilde{y}_{i_{type}}$

# Method

1. Each image is transformed with a **soft discrepancy.**
2. Passed through the $f_e$ and $f_p$ to generate its **embedding**.
3. **Bounded Contrastive Regression** and **Guidance loss** map these embeddings to the corresponding hard prototypes on the hypersphere.

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




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

# Method #2. Bounded contrastive regression

- Train embeddings that are not only **distinguishable** but also **well-clustered around predefined target prototypes.**

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

**Supervised Contrastive Loss** **Regressive Loss**

- Supervised Contrastive Loss : Encourages embeddings with the same label to be close together.
- Regressive Loss : Ensures that the embeddings are well-clustered around their respective prototypes.

# Method #2. Bounded contrastive regression

- **Supervised Contrastive Loss**

: Train embeddings with the same label to be close together.

$$\mathcal{L}_{\text{SupCon}} = \sum_{i=1}^{N} \frac{1}{|P(i)|} \sum_{p \in P(i)} -\log \frac{\exp(\text{sim}(z_i, z_p)/\tau)}{\sum_{a \in A(i)} \exp(\text{sim}(z_i, \boxed{z_a})/\tau)}$$

- Normalized temperature-scaled cross entropy loss **between the embeddings $z_i$ and $z_p$.**
  - P(i) : the set of indices of all samples in the batch that have the same label as the i-th sample.
  - A(i) : the set of all embeddings in the batch excluding.

# Method #2. Bounded contrastive regression

- **Regressive Loss**

  : Train embeddings are well-clustered around their respective prototypes.

$$\mathcal{L}_{\text{Regressive}} = \sum_{i=1}^{N} \frac{1}{|P(i)|} \sum_{p \in P(i)} -\log \frac{\exp(\text{sim}(z_i, p_{\tilde{y}_i})/\tau)}{\sum_{j=1}^{K} \exp(\text{sim}(z_i, \boxed{p_j})/\tau)}$$

- Normalized temperature-scaled cross entropy loss **between the embedding $z_i$ and the prototype $p_i$ .**
  - P(i) : the set of indices of all samples in the batch that have the same label as the i-th sample.

# Method #2. Bounded contrastive regression

- Train embeddings that are not only **distinguishable** but also **well-clustered around predefined target prototypes.**

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

**Supervised Contrastive Loss** **Regressive Loss**

- Limitation

- Because the **prototypes are evenly distributed,** the distance from any given embedding to an **incorrect prototype is roughly the same**.
- Therefore, <u>the error is similar regardless of which incorrect prototype.</u>

# Method #2. Guidance Loss

- **Guidance Loss**

- To address this issue, use explicitly weights the distances based on **prior knowledge about facial geometry and symmetry.**

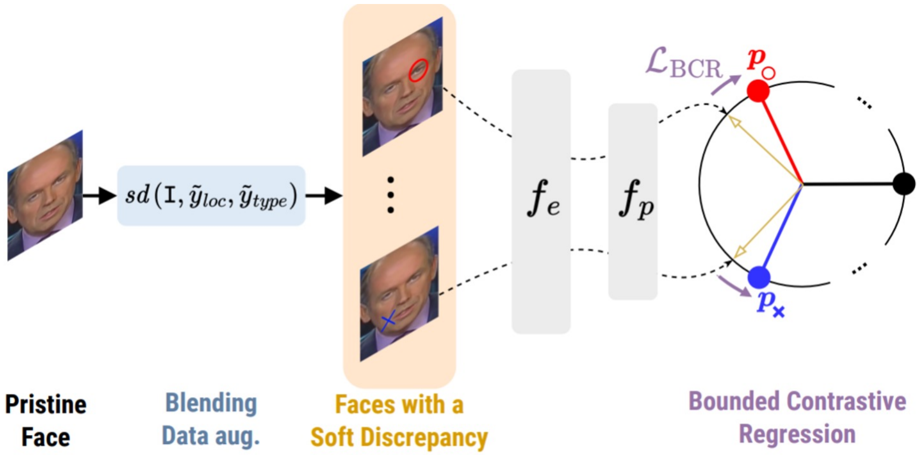
$$\mathcal{L}_{\mathrm{GUI}} = \sum_{i \in [1..N]} G\left(y_i, \tilde{y}_i\right) \times \{ 1 - \mathrm{sim}(z_i, p_{\tilde{y}_i}) \}$$

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

# Method

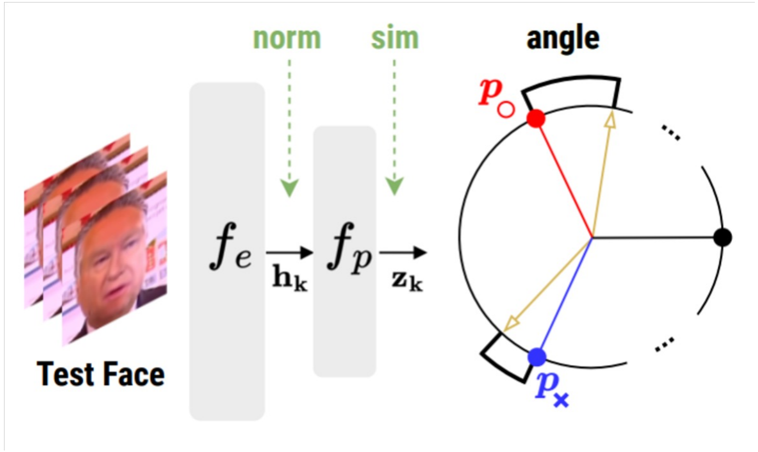
1. Each image is transformed with a **soft discrepancy.**
2. Passed through the <u>$f_e$</u> and <u>$f_p$</u> to generate its **embedding**.
3. **Bounded Contrastive Regression** and **Guidance loss** map these embeddings to the corresponding <u>hard prototypes</u> on the hypersphere.

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




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

# Deepfake Detection




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

- **Anomaly score** for deepfake detection.

- Apply **K soft discrepancy** transformation for test image.
- Classifies test image as such if the <u>anomaly score exceeds a certain threshold.</u>

# Results

# Results

- All models are trained on FF++ and tested on **datasets not seen during training.**

| 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$^{\dagger}$ [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** |

$^{\dagger}$SBI was re-evaluated using the official code with $M_{\text{ConvexHull}}$.

< Generalizability Across Datasets >

# Results

- All models are tested on deepfake videos created using **four different manipulation techniques**
  : Deepfakes (DF), Face2Face (F2F), FaceSwap (FS), and NeuralTextures (NT).

| Method | Test set - AUC (%) | | | | |
|---|---|---|---|---|---|
| | DF | F2F | FS | NT | Avg. |
| OC-FD1$^\dagger$ [37] | 86.2 | 70.7 | 84.8 | 95.3 | 84.2 |
| OC-FD2$^\dagger$ [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** |

$^\dagger$OC-FD1 and OC-FD1 refers to two versions of OC-FakeDect

< Cross-manipulation Evaluation>

# Conclusion

# Conclusion

- Treat deepfake detection as an **out-of-distribution (OOD) detection** task.
- Introduce **Bounded Contrastive Regression and Guidance Loss** that aims to push the soft-discrepancies to the predefined hard prototypes.
- Demonstrate the superior **generalization capabilities** compared to existing (SoTA) deepfake detectors.

# Limitation

- A small number of cases show wrong answer.
- Real images that come with deepfake-like artifacts

- High-quality deepfakes

# Thank you