Spot the Fake B-cos You Can?

Becker, Nils Braun, Robin-Kiara Fuchs, Toby Neumann, Linus Reffert, Kai Schütz, Tilo

***Abstract*—The rising threat of deepfakes, necessitates detection systems that are not only accurate but also interpretable. In this study, we investigate the application of inherently interpretable B-cos networks to the task of deepfake detection, comparing them to traditional black-box models enhanced with post-hoc explainability techniques. We implement B-cos variants of ResNet34, XceptionNet, and Vision Transformers and evaluate their classification performance and attribution quality using both standard benchmarks and novel explainability metrics. To this end, we introduce the Mask Pointing Game, a fine-grained evaluation framework that leverages deepfake generation masks to assess explanation localization. Our findings reveal that while B-cos models yield competitive attribution maps, they remain sensitive to architecture and hyperparameter tuning. Despite achieving promising results in MPG, B-cos models struggle in generalized settings like the Grid Pointing Game, highlighting their limited robustness. We conclude that B-cos networks show strong potential for interpretable deepfake detection, but require careful engineering to be viable in real-world applications.**
Find our implementation here: https://github.com/nilsbecker0711/Interpretable-Deep-Fake-Detection/.

***Index Terms*—Computer Vision, Deepfake Detection, Explainable AI, B-cos Networks, Attribution Evaluation Metrics**

I. Introduction

The rise of generative AI has made it easier than ever to create deepfakes - realistic fake images, audio, and videos that can closely mimic real people [1]–[6]. These synthetic media are often shared widely on social media, raising serious concerns about privacy, public safety, and political stability [7]. Deepfakes can be used to spread false information, damage reputations, and undermine trust in institutions, from scientific organizations to democratic elections [7]. Deepfake detectors are naturally an essential line of defense; however, research indicates that individuals often cannot reliably detect deepfakes and tend to overestimate their detection abilities, leading them to dismiss alerts or overlook subtle artifacts that signify a forgery [8]. In order to build trust in predictions, Explainable AI (XAI) methods could provide crucial transparent, human-interpretable, fine-grained explanations.

Advances in interpretable deep learning have introduced the B-cos transformation as a novel architectural replacement for standard linear layers in deep neural networks [9]. B-cos models aim to improve interpretability by inducing weight-input alignment during training, allowing the full computation of the network to be summarized as a single, input-dependent linear transformation. This design enables inherently interpretable models, which outperform competitive SOTA XAI methods qualitatively as well as quantitatively [9], [10]. The proposed B-cos framework comes with a range of benefits, most notably, the ability to generate model-faithful and human-interpretable explanations while incurring only a slight performance decrease. However, B-cos models require numerous architectural and training-specific adaptations to achieve competitive performance and stable training.

Our contributions are fourfold:

- We introduce B-cos models to deepfake detection tasks by embedding them within the DeepFake Benchmark [11].
- We apply the B-cos transformation to XceptionNet [12].
- We evaluate SOTA post-hoc XAI methods compared to inherently interpretable B-cosified models qualitatively and quantitatively.
- We propose the Mask Pointing Game, a novel method to evaluate XAI methods in deepfake detection.

First, we discuss related work in Section II to our study. After that, we present our experimental setup in Section IV. Following this, we will discuss our results in Section V and limitations in Section VI, giving a conclusion and outlook at the end in Section VII.

II. Literature Review

XAI Methods: (XAI) methods have emerged as essential tools for making deep learning models more transparent and interpretable, particularly in domains where understanding a model's reasoning is critical. Zhang et al. [13] categorize XAI approaches into two main types: post-hoc and ante-hoc. Post-hoc methods are applied after training and aim to interpret a model's decision-making process without modifying its internal structure. These include methods such as LIME [14] and gradient-based saliency techniques like [15]. Their wide adoption stems from their compatibility with existing models; however, as noted by Kamakshi and Krishnan [16], their explanations can sometimes be misleading, failing to faithfully reflect the model's internal logic. Ante-hoc approaches, on the other hand, embed interpretability directly into the model through design constraints or training objectives. For example, B-cos networks, [9] , [10] introduce alignment-based transparency, while prototype- and concept-based models such as ProtoPNet [17] aim to link predictions to semantically meaningful patterns. While inherently faithful and more interpretable, these methods may require compromises in model flexibility or accuracy.

In the domain of deepfake detection, gradient-based post-hoc explainers have been widely applied to interpret model outputs. Grad-CAM, for instance, identifies class-relevant image regions by combining output gradients with convolutional features [15]. Extensions of this method address

key limitations: Grad-CAM++ [18] incorporates second-order derivatives to improve localization of multiple features, while XGrad-Cam [19], introduced by Fu et al., normalizes gradient contributions and enforces formal attribution axioms for greater faithfulness. A more recent variant, LayerCAM [20], computes saliency at multiple convolutional stages to enhance spatial resolution. In the study by Tsigos et al. [21], these methods were directly compared on deepfake classifiers using a targeted modification framework. By selectively altering the top-k regions highlighted by each explainer, they measured how much classification accuracy dropped, showing that interfering with areas identified by Grad-CAM++ or XGrad-CAM reduced accuracy from 98% to 78%, indicating high attribution relevance. Similarly, Gowrisankar and Thing [22] evaluated these methods through adversarial attacks and found them to outperform alternatives such as RISE [23] and LIME [14] in localizing critical facial features in these specific settings.

LIME [14], which approximates local decision boundaries using modified input samples and interpretable surrogate models, is often used as a flexible, model-agnostic baseline. Despite its adaptability, LIME explanations are highly sensitive to how inputs are segmented and to the randomness introduced during sampling [14]. Schallner et al. [24] highlight that these dependencies can result in unstable or overly coarse explanations. In their evaluation of XAI tools, Tsigos et al. [21] observed that LIME's saliency maps often fail to isolate subtle manipulation cues, such as those involving small regions like the eyes or noselimiting its utility in deepfake detection tasks. By contrast, B-cos networks leverage their inherent alignment with input features to generate more consistent and interpretable attribution maps, a property noted by both Böhle et al. [9] and Kamakshi and Krishnan [16].

While post-hoc methods such as LIME [14] work well in practice they do not necessairly reflect how the model makes decisions. To make models inherently interpretable, architectural changes have to be made as in prototype-based networks like BagNets [25], CoDA Nets [26] and B-cos [9], [10]. Apart from B-cos these architectures are not adaptable to existing neural networks [9]. As mentioned by Böhle et al. [9] there are also other methods that focus on weight-input alignment [27] either by adversarial training or by loss-based model regularization [28].

Deepfake Detection Methods: As deepfakes pose a contemporary thread, many detection methods have been proposed and are an active field of research [29]–[33]. Therefore, Abbasi et al. conducted a comparison between XceptionNet [12], ResNet [34], and VGG16 [35] for deepfake detection. Their study hints that XceptionNet is the optimal choice for accuracy and generalizablity, while VGG16 excels in precision and ResNet promises the fastest inference time. All models failed to identify adversarially altered images, while the study also lacks information on which visual characteristics contributed the most to classification decisions, which obscures interpretability [36].

Vision Transformers (ViTs) [37], although originally developed for high-level vision tasks such as image recognition, show a strong ability to model global representations and analyze intricate visual patterns, which makes them promising for deepfake detection [38].

Another black-box approach is offered by Yan et al. [39] addressing feature integration. They propose a model leveraging both local and global frequency domain cues using Discrete Wavelet- and Fast Fourier Transformations. In comparison to Convolutional Neural Networks, their approach has boosted generalizability across multiple deepfake methods [39].

To provide a common ground for deepfake detection methods Yan et al. [11] provide a benchmark for deep fake detectors, categorized into naive, spatial, and frequency detectors. Naive detectors rely on simple CNN classification, spatial detectors focus on forgery localization and structural artifacts, while frequency detectors exploit anomalies in the frequency domain. In their work they show, that all of their tested models perform really well [40] for within-domain detection, reaching an Area under Curve (AUC) of above 90% for all models. However, they also show a core weakness of all existing deepfake detection methods. In cross-domain evaluations, when detectors are tested on forgery methods not trained on, their performance drops significantly, hinting that models trained on specific forgeries often struggle to adapt to other unseen forgeries. This generalization issue is also mentioned in the works by Patel et al. [41] and remains a potential scope for future work as forgery methods are to adapt and change in the future [42].

The benchmark structure provided by Yan et al. [11], [43] will serve as the basis for our experiments, which will focus on naive detectors with B-cos transformations only. SOTA frequency- and spatial detectors contain complex architectures, which are not yet compatible with the B-cos transformation [11], [9].

XAI Evaluation Methods: The evaluation of saliency maps in XAI has been approached through both objective and subjective methods [44]. Subjective evaluation methods use user studies to assess saliency maps through direct ratings e.g., relevance or consistency [45], or indirect tasks e.g., predicting model outputs [46]. Despite their insights, subjective methods face challenges like high costs and variability in user perceptions. Alternatively, objective metrics quantify properties without human intervention. They typically follow two schools of thought when evaluating saliency maps [47], those measuring faithfulness to the model's decision-making, and those measuring explanation quality via alignment with known important regions or human expectations. Faithfulness is commonly assessed with perturbation-based metrics, in which the most salient pixels are either iteratively removed or added to identify variations in the function value [23], [48], [49]. However, these methods may distort the model behavior [50], especially in the deepfake domain [22]. Furthermore, since B-cos is an inherently designed XAI method, it is faithful by design, shifting our focus to the assessment of explanation quality. A common proxy to assess explanation quality for human-aligned localization is the pointing game [51], which checks if the most salient point lies inside a known region of interest (e.g. an object's bounding box). In multi-class classifi-

cation settings, [52] this is extended to the grid-pointing game (GPG), which is often used to evaluate XAI methods [9], [10], [53]. However, GPG primarly measures whether the correct class can be attributed, without assessing the quality of the explanation for the class itself. To address this, we propose the Mask Pointing Game (MPG) — an application of the pointing game, which utilizes the masks of the deepfake generating process as the region of interest. The MPG is a broader case of the controlled manipulation masks used in [54], in which they manipulated images in specific facial regions enabling direct access to the ground-truth mask of the manipulation. The MPG allows the evaluation of alignment between important image regions and explanations. Unlike GPG, MPG avoids the use of out-of-distribution (OOD) images while providing a direct measure of attribution quality.

III. B-cos Transformation

To build inherently interpretable neural networks, the B-cos transformation [9] replaces standard linear operations by enforcing weight-input alignment. The idea of weight-input alignment is inspired by the behavior of loss gradients in adversarially trained networks, which tend to exhibit greater interpretability [27], [55] by showing better alignment with human perception since adversarial training constrains gradients to lie closer to the image manifold [56]. While there are other methods to achieve weight-input alignment, e.g. regularization via loss function [55], [57], B-cos networks have built-in architectural constraints which inherently enforce weight-input alignment. At the heart of B-cos networks lies the B-cos transform [9]. It replaces the standard dot product between weights w and inputs x of linear layers, defined as:

$$\text{Linear}(x; w) = w^T x = ||w|| \cdot ||x|| \cdot \cos(\angle(x, w)) \quad (1)$$

with $\cos(\angle(x, w))$ representing the cosine of the angle $\angle(x, w)$ between vector x and w. The B-cos transform acts as a drop-in replacement to the standard dot product and is defined as:

$$\text{B-cos}(x;\, w) = \underbrace{||\hat{w}||}_{=1} \cdot ||x|| \cdot |\cos(\angle(x, \hat{w}))|^B \cdot \text{sgn}(\cos(\angle(x, \hat{w}))) \quad (2)$$

where $\hat{w}$ is the scaled unit-norm weight vector, limiting B-cos neurons to learn angular dependencies and bounding their maximum activation to $||x||$, which is only reached if x and w are collinear, i.e. aligned [9]. The parameter B controls how sharply the model favors aligned inputs: larger values of B reduce contributions from misaligned inputs and highlight those that are well aligned. If $B = 1$ Equation 2 reduces to the standard linear transform with $\hat{w}$. Importantly, sequences of B-cos transforms are reliably summarized by a single linear transformation, see [9] for details, enabling the model's output to be faithfully summarized as a dynamic linear map for each input, leading to transparent, localized explanations. In contrast to previous inherently interpretable works [58]–[60] the B-cos transform is applicable with existing model architectures.

However, besides the adoption of the B-cos transform, B-cos networks require further architectural changes and retraining, unless pretrained B-cosification [53] is considered.

Böhle et al. introduce a wide set of changes across their two papers [9], [10] that fall into the following categories:

1) Architectural Replacements: Linear layers and convolutional kernels are replaced with B-cos transforms, promoting alignment. Standard non-linearities (like ReLU) are substituted or removed due to inherent non-linearity of B-cos layers (for $B > 1$). MaxOut layers are often used instead of ReLU, as they are more compatible with optimization and maintain alignment structure. A final logit layer has been added with a logit-bias and logit-temperature to prevent exaggerated activation values and stabilize training, while also scaling down output logits.
2) Training Strategy Modifications: Binary Cross-Entropy (BCE) loss is used even for multi-class tasks, combined with tailored logit biases to encourage meaningful output scaling. Fixed scaling (γ) is used to address signal decay and instability, especially for deep networks or high B values. Weight normalization and additional scaling layers are introduced.
3) Normalization and Regularization Techniques: Modified normalization layers (e.g., bias-free BatchNorm) are required to maintain linear explanation completeness. Gradient clipping (with lower magnitudes than conventional models) and extended warm-up schedules are introduced to improve stability. No weight decay is applied, and larger models are particularly sensitive at early training stages.
4) Input and Output Encoding: Input images are augmented with a 6-channel color encoding ([r, g, b, 1–r, 1–g, 1–b]) to stabilize activation norms and improve interpretability.
5) Data Augmentation: Techniques such as random cropping, resizing with bilinear interpolation, random horizontal flipping, CutMix [61], MixUp [62], random Repeated Augmentations, Random Erasing and Trivial Augment [63] are utilized.
6) Task-Specific Adaptations: CIFAR-10 [64] and ImageNet [65] required different bias settings, output temperatures, auxiliary losses and layer norms. This results in additional hyperparameters that have to be tuned. For multi-class settings, negative class alignment is also suppressed by encoding non-target classes as uniform distributions.

These extensive modifications highlight both the flexibility and the fragility of B-cos models in practice. While their interpretability is unmatched [9], [10] on CIFAR-10 [64] and ImageNet [65], their implementation demands a fine balance of architectural choices and training heuristics.

IV. Experimentental Setup

A. Dataset

We used the *FaceForensics++* (FF++) [66] dataset to train and evaluate our models. This 2019 published dataset provides

manipulated images, based on *Deep-Fakes* [67], *Face2Face* [68], *FaceSwap* [69] and *NeuralTextures* [70] on different compression levels. While *Deep-Fakes* and *FaceSwap* perform a facial identity swap, *Face2Face* and *NeuralTextures* alter an image by facial reenactment. Due to the four forgery methods, the dataset has a 80/20 split regarding fake and real images. To test cross-domain performance we also used the following datasets: *FaceShifter* (Fsh) [71], *DeepfakeDetection* (DFD) [72], *Deepfake Detection Challenge Preview* (DFDCP) [73],*CelebDF-v2* (CDF-v2) [74] and *UADFV* [75]. All these datasets are widely used for deepfake detection benchmarking [11]. A detailed overview is given in V. Additionally we used the *FFHQ* dataset [76]. Information on this dataset can be found in the Appendix, Section IV.

B. Model Architectures and Training Setup

We employed ResNet-34, a lightweight Vision Transformer with a shallow convolutional stem (ViT-C), and XceptionNet as backbone architectures. The standard non B-cos ResNet-34 and XceptionNet models were sourced from the Deepfake Benchmark [43], The non B-cos ViT-C model was obtained from the bcos-v2 repository [10], [77]. Specifically, we used the tiny ViT with a patch size of one and a spatial resolution, i.e.feature dimension after the convolutional stem, of 14. The bcos-v2 repository itself obtained these models from Xiao et al. 2021 [78]. For the B-cos variants, we used the B-cosified ResNet-34 and Simple ViT-C model as described in the bcos-v2 repository [10], [77]. The XceptionNet model was manually adapted to incorporate the B-cos framework. We kept the overall model architecture but swapped all convolutional and linear layers with their B-cos alternative [9], dropped all ReLU activation functions, swapped max-pooling with average-pooling, set dropout to 0, added B-cos norm layers throughout the network and added the B-cos logit layer at the end. This architecture change is in line with how the Resnet-34 and other CNN-architecture have been B-cosified by Böhle et al. [10].

We chose the Resnet-34 [79] because of its simplicity, the ViT as it is a state-of-the-art image classification network [80] and the XceptionNet as it showed the best backbone performance on the Deepfake Benchmark [43].

All models were optimized using the Adam optimizer and a weight decay. We performed hyperparameter tuning using RandomSearch [81] and Hyperband [82] algorithms via Weights & Biases [83]. Specific hyperparamter settings for each model can be found in the Appendix, see A. Model selection was based on the Area Under the ROC Curve (AUC) evaluated on a validation split. Each model was trained on 20 epochs from scratch. The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the Max-Planck-Institute. We trained our models on Nvidia A100s, H100s and Quadro RTX 8000s, depending on their availability.

V. Results and Evaluations

A. Deepfake Detection Performance

Across the *FaceForensics++* [66] test split on c23 compression shown in Table I, the vanilla XceptionNet remains the strongest overall detector, achieving the highest in-domain scores (AUC = 0.97, F1 = 0.96). The ResNet-34 baseline follows (AUC = 0.88), while the plain ViT lags behind (AUC = 0.65). It should be noted here, that the pretrained XceptionNet from [11] was used, which was already extensively optimized for deepfake detection, while the other models were trained from scratch. More details on the models hyperparameters and their optimization can be found in the Appendix A and VI. Introducing weight–input alignment through B-cos affects the three backbones very differently. For convolutional nets, increasing B consistently hurts both discrimination and localization: ResNet-34 drops by up to 0.23 in AUC at B=1.25 and its localization falls from GPG = 0.36 to 0.16, with a further collapse to 0.10 at B=2.5. XceptionNet also shows a decline (–0.23 AUC at B=2.5), which is probably due to the fragility of the B-cos architecture and will be looked into in Section VI. In contrast to that, the ViT seems to benefit from the B-cosification, reaching higher AUCs for each of the B-cos implementations, when comparing them to the base ViT. This shows that a higher B value does not necessarily lead to worse performance as Böhle et al. stated in their original paper [9]. However, we cannot be sure, that with more extensive hyperparameter optimization this claim can still hold.

TABLE I
PERFORMANCE METRICS ON FACEFORENSICS++ WITH C23 COMPRESSION. SEE APPENDIX TABLE VII FOR PRECISION RECALL AND ACCURACY.

Model	AUC	F1
Resnet34 (Vanilla)	0.88	0.91
Resnet34 (B=1.25)	0.78	
Resnet34 (B=1.75)	0.82	0.88
Resnet34 (B=2)	0.84	0.89
Resnet34 (B=2.5)	0.84	0.90
Xception (Vanilla)	0.97	0.96
Xception (B=2.5)	0.75	0.89
ViT (Vanilla)	0.65	0.89
ViT (B=1.25)	0.72	0.88
ViT (B=1.75)	0.77	0.87
ViT (B=2)	0.78	0.87
ViT (B=2.5)	0.78	0.88

Out-of-distribution performance shown in Table II follows the trend seen in Table I. Vanilla XceptionNet again offers the best average generalization, topping four of the five sets (e.g., 0.94 on *UADFV* [75]) and only trailing on *FaceShifter* [71]. The ViT variants remain rather poor. Overall, weight–input alignment improves transformer-based detectors yet compromises both the robustness and the forgery-localization ability of CNNs, leaving the unmodified XceptionNet as the most reliable choice when both in-domain accuracy and cross-domain generalization are required.

We see generalization issues as all of our models struggle to adapt to unseen forgeries, dropping by up to 0.23 in AUC. This result does not come surprising, as other studies

TABLE II
OOD AUC ON FF++ WITH C23 COMPRESSION. THE DETAILED RESULTS CAN BE FOUND IN THE APPENDIX, SEE TABLE IX.

Model	CDFv2	DFD	DFDCP	Fsh	UADFV
Resnet34 (Vanilla)	0.65	0.69	0.63	0.66	0.78
Resnet34 (1.25)	0.57	0.57	0.55	0.63	0.9
Resnet34 (1.75)	0.63	0.58	0.61	0.63	0.84
Resnet34 (2)	0.64	0.56	0.66	0.66	0.93
Resnet34 (2.5)	0.63	0.59	0.56	0.62	0.84
Xception (Vanilla)	0.74	0.85	0.73	0.59	0.94
Xception (2.5)	0.61	0.55	0.62	0.67	0.73
ViT (Vanilla)	0.57	0.54	0.59	0.65	0.78
ViT (1.25)	0.58	0.57	0.58	0.63	0.76
ViT (1.75)	0.61	0.58	0.58	0.62	0.84
ViT (2)	0.64	0.58	0.58	0.62	0.74
ViT (2.5)	0.62	0.57	0.58	0.64	0.78

resulted in similar findings [11], [41]. B-Cos models tend to generalize a little better on unseen data compared to their vanilla counterparts when looking at the relative performance drop. This perceived benefit is however, doubtable, as an AUC of below 0.7 already implies poor performance [40]. Interestingly, the ViTs performance drop was the smallest, which might be due to their previous poor performance on the *FF++* [66] dataset.

To complement these results, we conducted studies regarding the c40 compression level of *FF++* [66], which represents a lower resolution compared to c23, as well as assessing ResNets performance while trained and tested on *FFHQ* [76]. These results are listed in the Appendix in Tables VIII (in-domain) and X (cross-domain) evaluations in c40 compression as well as Table XI for *FFHQ*. Looking at the c40 compression, we can see, that even if the models are trained on a higher resolutions, they are still able to output reasonable performance. The slight performance drop can be traced back to information loss within the pictures due to the lower resolution. The outstanding performance of all models on the *FFHQ* dataset hints that this dataset is easier to train on.

B. Qualitative Evaluation of XAI Methods

In order to conduct a holistic evaluation we combine qualitative and quantitative approaches. Figure 1 provides promising qualitative results for B-cos models in the realm of deepfake detection. We see that for a low B-parameter of 1.25 the attribution maps surround the face, but are only loosely focused on specific facial features. When moving to higher B values the attribution maps become more focused, e.g. for $B = 2.5$ the mouth in the first example and nose and eyes in the second example are highlighted. Similar figures are created for the B-cos XceptionNet and ViT models, see Figure 8 and Figure 9 in the Appendix.

C. Grid Pointing Game

To take things a step further and quantify how well XAI methods localize the focus of deepfake detectors, we adapt the grid pointing game from Böhle et al. [9]. The grids are

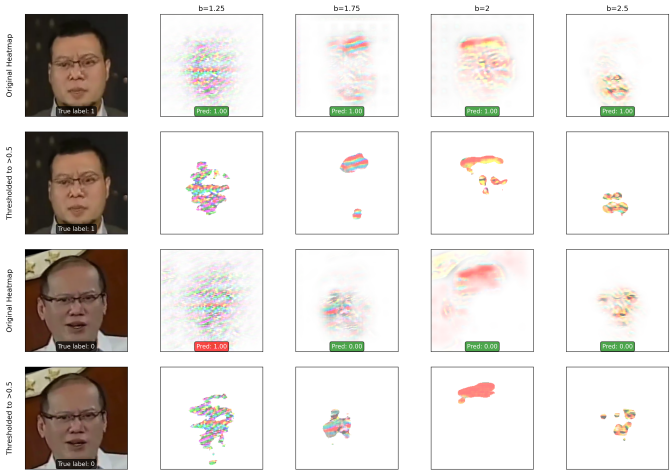

Fig. 1. B-cosified ResNet34 attribution maps for different B-parameters as well as a threshold of 0.5. The attribution maps align with the predicted class instead of the true class. Examples stem from *FF++* [66]

constructed by selecting test images from *FF++* [66] and placing them randomly in a 3×3 grid, see Figure 2. In Figure 2 a B-cos ResNet34 with a b = 2 is evaluating a 3×3 Grid with the true fake position being the middle-bottom grid. The main attribution of the heatmap falls into the correct cell, but seven of the eight real images also show signs of the B-cos heatmap. The visualization is not as precise as in the work by Böhle et al. [9], which comes down to two main factors. Firstly, deepfakes may be present on a frequency level [84], which are much more fine-grained and hidden in the image. Secondly, explaining on a 3×3 grid instead of one image is much more challenging for the XAI method, resulting in more ambiguous predictions.

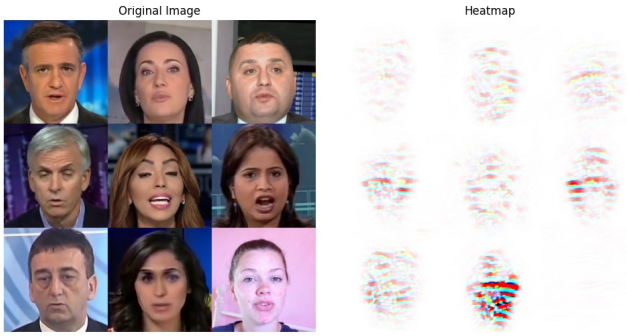

Fig. 2. An example of a grid pointing game 3x3 grid. The true fake position is the middle-bottom grid.

Because the original GPG was designed for multi-class problems [58], we adapated the grids to contain one fake image and eight real images since we are primarily interested in testing the capability of the XAI methods in explaining fakes. Moreover, to ensure that the underlying model is confident enough to detect the fake, we only use fake images that have been classified correctly with high confidence ($p > 0.5$). By doing so, we can evaluate the explanation quality of XAI

methods on an even playing field, as the model has already proven its ability to identify the fake. In order to isolate the model performance from the XAI evaluation, we ensure that the model is as confident as possible by having each model generate grids based on the top-n confident fakes. We test six XAI methods: B-cos, [15], Grad++ [18], LayerCAM [20], XGrad-CAM [85], and LIME [14]. Each method produces heatmaps highlighting the regions most relevant to the model's decision. B-cos and Grad-CAM methods already provide intensity maps with scale [0,1], but LIME's outputs needed to be normalized to the [0,1] scale to better align with the other methods' attribution maps for consistent evaluation. However, an obvious limitation of this scaling is that the absolute values of LIME can no longer be interpreted, but crucially the relative differences between salient regions can now be interpreted on the same scale as the other methods. This is especially important for consistency across our thresholding mechanism. Once heatmaps are generated, we assess how accurately they localize the fake image within each grid using two metrics:

An unweighted localization score, which checks whether the activated region overlaps with the fake image grid cell, and a weighted localization score, which measures how much attribution falls within the correct grid cell. Both metrics can be expressed by the following general formula:

$$\text{GPG Localization Score} = \frac{\sum\limits_{(i,j)\in C_{\text{true}}} w(i,j)}{\sum\limits_{(i,j)} w(i,j)} \tag{3}$$

where

$$w(i,j) = \begin{cases} 1 & \text{if } A_{i,j} > 0 \text{ and unweighted score} \\ A_{i,j} & \text{for the weighted score} \end{cases}$$

Here, C_{true} denotes the set of pixel indices in the grid cell containing the fake image, and $A_{i,j}$ represents the attribution value at pixel position (i, j). In case of thresholding all $A_{i,j} < threshold$ become 0.

Table III shows that for the unweighted localization score B-cos ResNet34 performs generally poorly with a score of 0.111, which is equivalent to uniform attribution across all grids. In terms of the weighted localization scores, B-cos seems to be better at a lower B-parameter. This may seem counter-intuitive as we expect a stronger localization effect with a higher B-parameter [9], but one reason could be that if the model is not finding the fake in the grid setup, it makes sense that a stronger localization effect only worsens the prediction. While the weighted localization score of B-cos is higher in combination with the B-cos XceptionNet (0.2826), Grad-CAM still outperforms B-cos on both models. For the ResNet34 the best out of all XAI models is Grad-CAM $(0.3613, \text{weighted})$ performing slightly better than LayerCAM $(0.3611, \text{weighted})$. For the XceptionNet, LayerCAM is the best XAI method with an weighted localization score of 0.5858 tightly followed by Grad-CAM++ with a weighted score of 0.5819. Grad-CAM predicts a dense center of mass with high confidence and this tapers off as one moves further away from the center mass. This is different to B-cos which attempts to create highly dense and localized attribution maps. Based on the results one could hypothesize that if the model is not sure, Grad-CAM will be better as it doesn't have as much pressure to create dense localizations [15]. Lime has a lot of difficulties handling the experimental set up as its weighted and unweighted localization scores are close to uniform attribution for both models.

TABLE III
FULL GRID POINTING GAME PERFORMANCE (NO THRESHOLD).

Model	**ResNet34**		**XceptionNet**	
	Unweighted	Weighted	Unweighted	Weighted
B = 1.25	0.1111	0.1580	N/A	N/A
B = 1.75	0.1111	0.2624	N/A	N/A
B = 2	0.1111	0.1365	N/A	N/A
B = 2.5	0.1111	0.0958	0.1111	0.2826
Grad-CAM	0.1950	0.3613	0.1749	0.3365
LayerCam	0.1951	0.3611	0.2141	0.5858
XGrad	0.1403	0.2310	0.1493	0.5146
Grad++	0.1682	0.3076	0.1078	0.5819
Lime	0.1158	0.1184	0.1232	0.1621

D. Mask Pointing Game

In addition to the GPG, we introduce the Mask Pointing Game (MPG) to evaluate the localization capabilities of our XAI methods. MPG makes use of the mask annotations provided in the *FF++* dataset. There, the forgeries are applied within a defined mask region, which we adopt as ground truth to assess whether the explanation highlights manipulated areas. This is similar to the approach used in [54]. A key advantage of MPG is that the model has to focus on regions where forgery has happened and not on points in a grid cell where forgery somewhere happened. This leads to a more fine-grained localization evaluation. This is only possible on the FF++ dataset, where mask annotations exist. Nevertheless, the MPG gives us a novel measurement of the degree to which an explanation method is able to align its attribution map with the forgery area. MPG is depicted in Figure 3 where a higher accuracy means that more attribution falls within the mask region.

Formally, we assess the XAI methods using the MPG accuracy, defined as:

$$\text{MPG Localization Score} = \frac{\sum\limits_{(i,j)\in M_{\text{true}}} w(i,j)}{\sum\limits_{(i,j)} w(i,j)} \tag{4}$$

where

$$w(i,j) = \begin{cases} 1 & \text{if } A_{i,j} > 0 \text{ and unweighted score} \\ A_{i,j} & \text{for the weighted score} \end{cases}$$

Here, M_{true} denotes the set of pixel indices within the mask, and $A_{i,j}$ represents the attribution value of the explanation algorithm at pixel position (i, j). In case of thresholding all $A_{i,j} < threshold$ become 0. Similar to the GPG, Equation 4 can be applied to both intensity-based and thresholded heatmaps.

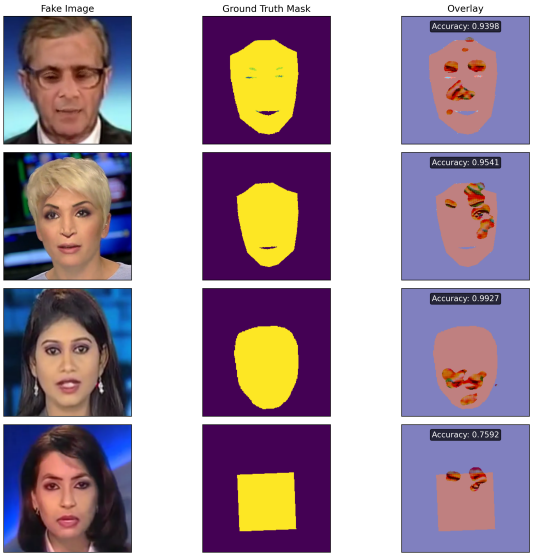

Fig. 3. Illustration of the MPG. Left: original deepfake images of *FF++* [66] in c23 compression. Middle: mask that points to the the region were the original image was altered. Right: B-cos heatmap overlayed with the mask. The heatmap is created from our B-cos ResNet34 with $B = 2.5$ and threshold = 0.5.

The MPG results are shown in Table IV. Regarding the weighted accuracy, B-cos models tend to have better performance, whereas the other XAI methods tend to perform better with respect to the unweighted setting. Especially at higher B values, B-cos is designed to produce highly fine grained and localized class attributions [9], whereas Grad-CAM predicts a center of high intensity that slowly dissipates as one moves away from the center [15]. Therefore, it makes sense that we will outperform Grad-CAM in the MPG with a decent B-cos model, as it should assign the large majority of mass to the forgery region and very little elsewhere. This logic seems to fall apart when looking at XceptionNet, however this may be taken with a grain of salt since the model parameters and weights for the normal XceptionNet are taken from the best DeepfakeBench configuration [43] and the XceptionNet for B=2.5 didn't complete extensive HPO runs. This is also reflected in the model performance gap between the two XceptionNets as seen in Table I. Once again this highlights a key issue with B-cos, reflected in the level of engineering required to produce a good model. Interestingly, the Grad-CAM variants lie very close to another.

E. Thresholding

The performance of XAI methods can often be improved by removing low intensity signals from the attribution maps. This can be achieved with pooling, smoothing [77] [15], or, as we propose, with thresholding. We favor thresholding because it adjusts each pixel deterministically and independently, as it avoids the contextual blending introduced by other methods like smoothing. This results in a pixel-level reproducibility and interpretability of the resulting heatmaps.

TABLE IV
FULL MASK POINTING GAME PERFORMANCE (NO THRESHOLD).

Model	**ResNet34**		**Vit**		**XceptionNet**	
	Unweighted	Weighted	Unweighted	Weighted	Unweighted	Weighted
B = 1.25	0.2666	0.7497	0.2599	0.8545	N/A	N/A
B = 1.75	0.2652	0.7940	0.2591	0.7398	N/A	N/A
B = 2	0.2650	0.3525	0.2618	0.8517	N/A	N/A
B = 2.5	0.2639	0.8408	0.2589	0.8675	0.2639	0.6092
Grad-CAM	0.3533	0.5518	0.2724	0.2076	0.5061	0.8573
LayerCam	0.3510	0.5520	0.2729	0.2128	0.5138	0.8626
XGrad	0.3511	0.5540	0.2715	0.2087	0.5136	0.8620
Grad++	0.3523	0.5552	0.2717	0.2090	0.5119	0.8587
Lime	0.2815	0.4933	0.4184	0.6742	0.3451	0.5955

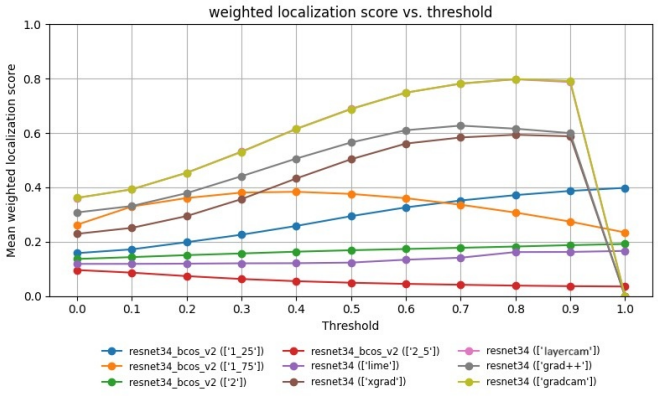

Fig. 4. Weighted localization scores from our Grid Pointing Game across different XAI methods on ResNet34 plotted over different threshold values.

Figure 4 illustrates the mean weighted-localization score as the attribution threshold is moved from 0.0 (no pruning) to 1.0 in 0.1 steps. Four B-cos ResNet34 variants b = 1.25, b=1.75, b = 2.0, and b = 2.5 are shown in blue, orange, green and red alongside four ResNet-34 explainers: LIME (purple), XGrad-CAM (brown), Grad-CAM++ (grey), LayerCAM (pink) and Grad-CAM (light green). At threshold 0, LayerCam achieves the highest score at approximately 0.36, followed by Grad-CAM++ with a score of 0.31. The B-cos model with b=1.75b comes next, scoring 0.26, just ahead of XGrad-CAM at 0.23. Another B-cos variant with b=1.25 ranks sixth with a score of 0.16, while the remaining methods are closely grouped: B-cos with b=2 scores 0.14, LIME scores 0.12, and B-cos with b=2.5 rounds out the list at 0.10. As the threshold increases, low-value pixels are pruned, sharpening the attribution maps. While B-cos 1.75's weighted localization score has the fastest increase for the first threshold of 0.1, its already reached its peak at an threshold of 0.3 with a score of about 0.38. This indicates that B-cos 1.75 is benefiting of the first three thresholds to get rid of noise and sharpening its prediction while it also quickly reaches the methods peak performance. Although xGrad and B-cos 1.25 both do not have much of an increase at the first threshold, they surpass B-cos 1.75 at a threshold of 0.4 (XGrad-CAM) and 0.7 (B-cos 1.25). B-cos 1.25 climbs steadily from roughly 0.16 at threshold 0.0 to

around 0.40 at a threshold of 1. Both B-cos 1.25 and 1.75 show that the threshold for B-cos models can effectively remove noise and concentrates the signal on the true fake patch. In contrast, B-cos with b = 2.0 remains relatively flat between 0.12 and 0.19 across thresholds, while b = 2.5 actually declines toward zero. Among the gradient-based methods, vanilla and LayerCam benefit the most from thresholding, rising from approximately 0.36 at threshold 0 to about 0.80 by threshold 0.8. Grad-CAM++ and XGrad-CAM follow a similar upward trajectory but XGrad-CAM peaks at lower values of around 0.61 at a threshold of 0.9, while Grad-CAM++ peaks at about 0.63 but already at threshold of 0.7. All Grad-CAM related models fall to a localization score of 0 at an threshold of 1 visualizing the absence of pixels with an attribution of 1. LIME remains largely unchanged, fluctuating only slightly between 0.12 and 0.17. Besides B-cos 2.5 all XAI methods seem to massively benefit from thresholding but not in the same way or same extent, therefore no universal threshold can be found.

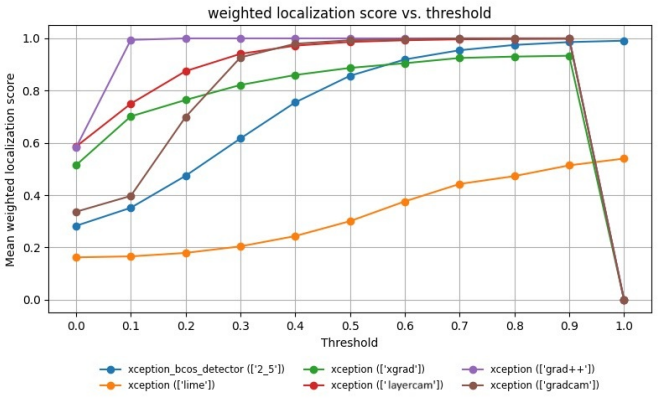

Fig. 5. Weighted localization scores from our Grid Pointing Game across different XAI methods on XceptionNet plotted over different threshold values.

This trend becomes more pronounced when examining the performance of the XceptionNet, see Figure 5. Although for all XAI methods the weighted localization scores are substantially higher compared to the ResNet34 variants, the initial benefiting of thresholding is still visible. The performance of LayerCAM and Grad-CAM++ is particularly good, as both reach near perfect scores (0.98) at relatively low thresholds. Grad-CAM++ is already close to its peak at a threshold of 0.1. This suggest that the attributions of those XAI methods are already well-localized and benefit quickly from minor thresholding. Grad-CAM and XGrad-CAM also show steep gains, reaching values around 0.94 and 0.91 respectively with a threshold of 0.9. The B-cos 2.5 variant improves consistently across the threshold range. It ultimately matches the gradient-based methods, peaking just below 1.0 and demonstrating that B-cos explanations in this architecture are highly effective once noise is pruned. In contrast, LIME lags behind again, improving only modestly from about 0.16 to approximately 0.53 while displaying less sensitivity to thresholding. Although the performance remains relatively low, it still represents an improvement over ResNet34, whose scores hovered near a uniform distribution baseline. Notably, all gradient-based methods sharply drop to zero at a threshold of 1.0, confirming that none of them assigns a full attribution score to any single pixel, highlighting the smoothness of these methods' saliency distributions. The plots also drive the point home that XAI methods tend to reach their performance peak as soon as no further low-relevance noise can be pruned. At this point, the curve either flattens, indicating that the remaining attributions are too strong to be removed by further thresholding, or begins to decline in some cases. The latter occurs when correctly attributed pixels have attribution scores that are too low to survive aggressive pruning, leading to a drop in localization performance, as observed in Figure 4. We also observe that for Grad-CAM and its variants, XceptionNet reaches its peak performance with minimal pruning, whereas ResNet-34 continues to benefit from pruning for a longer duration. This contrast may be attributed to the strong baseline performance of the vanilla XceptionNet model in deepfake detection (as shown in Table I), which likely contributes to a more precise focus on relevant image regions from the outset

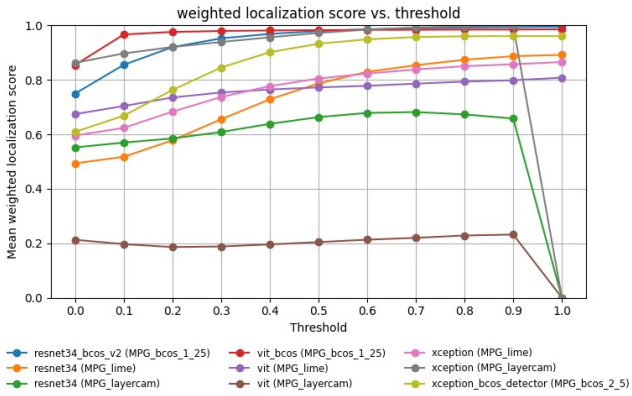

Fig. 6. Weighted localization scores from our Mask Pointing Game across different XAI methods on XceptionNet, ResNet34, and ViT plotted over different threshold values.

Building on these observations, the Mask Pointing Game evaluation offers additional insight into how different models and XAI methods respond to thresholding. Figure 6 shows how the ViT with B-cos explanation, stand out with high localization scores, reaching near-perfect performance after minimal thresholding. Similarly, to the B-cosified ViT, the other B-cos models show strong and steady improvements, saturating with near perfect scores around thresholds of 0.3–0.5. This is supported by strong performances, even at low thresholds, where removing low-intensity noise still leaves high-density attribution in the relevant region. Notably, LayerCAM performs inconsistently. Although it supports high scores in ResNet34 and XceptionNet, it underperforms significantly with the ViT backbone, suggesting potential compatibility issues. The performance of Lime is better across all models in the MPG while increasing steadily to a maximum unweighted localization score of about 0.9 at a threshold of 1. The results

lead to the assumption, that the MPG provides a good accurate and intuitive assessment tool for the localization quality. See supplementary Figures 14,15 and 16 in the Appendix, which show all available XAI methods applied to all models.

A more nuanced picture emerges when comparing the different XAI methods. B-cos strongly outperforms Grad-CAM variants on our MPG (weighted accuracy) and Grad-CAM performs better on the Grid Pointing Game metric. It seems like B-cos models struggle more with generating explanations on images that are not similar to its training data, such as the grid structure in GPG. In subsequent research, it would be beneficial to further analyze and potentially quantify how B-cos performance changes as inputs deviate further from the distribution.

VI. Limitations and Future Research

The B-cos implementations provided by Böhle et al. [9], [10], [77] turned out to be very fragile during training. The hyperparameter settings used for CIFAR-10 [64] and ImageNet [65] only yielded very poor performances, in which our imbalanced dataset further led to the models exclusively predicting the majority class.

Despite the best of our efforts the VGG16 [35], InceptionNetv3 [86], ConvNeXt [87] and DenseNet [88] models did not produce notable results. The main challenges were exploding and vanishing gradients, as well as heavy class imbalance towards fake images on the *FF++* dataset [66]. To tackle vanishing and exploding gradients we kept track of gradient values and applied gradient clipping, gradient clipping by norm, layer normalization, logit clamping and adjusted the logit bias and temperature. To address the class imbalance we tried a weighted loss function with different ratios and Sharpness-Aware Minimization (SAM) [89]. Neither of these approaches could yield meaningful results for the mentioned models. Furthermore, the B-cos authors stated that the second model implementations are more stable and easier to train [77]. This is in line with our findings since the Resnet34 and ViT from the second implementation were the only working models, whereas all models from the first publication [9] failed. This also included more robust models such as ResNet34. Still, even the models from the second version [77] remained hard to train in general. During hyperparameter optimization we observed that a plethora of combinations did not yield promising results. An illustration of this observation can be found in Figure 7 in the Appendix. Even in *FFHQ* [76], a balanced data set, the models turned out to be hard to train as only few HP combinations worked. This indicates that the class imbalance in *FF++* [66] is likely not the reason for difficult training.

Unfortunately, due to computational limitations, we were not able to train all Xception B-cos variants as originally intended. Similarly, the GPG for the ViTs remains a scope for future research. Notably, applying ViTs in the GPG context is particularly challenging as Böhle et al. already pointed out [90]. Due to the global attention mechanism in ViTs, synthetic image grids are much further out of distribution at every layer compared to CNNs, where locally applied convolutional kernels remain largely unaffected by the artificial nature of these inputs. Since the GPG, relies on the assumption that the model extracts comparable features in the synthetic setting and can locally classify subimages correctly, ViTs must apply a sliding window over the synthetic grids, specifically using 224×224 patches with a stride of 112 [90]. However, this significantly increases computational cost.

Given these findings, several avenues for future research emerge. First, exploring the B-cosification of pre-trained models as discussed in Arya et al. [53] appears to be a promising direction. This approach could bypass the extensive training iterations required for training from scratch, making the process more efficient and stable. Second, including larger and more diverse baseline models would help strengthen the evaluation and generalizability of results. Lastly, integrating more sophisticated deepfake detection techniques, such as multimodal inputs, may help improve performance, particularly in cross-domain settings.

The decent performance on the MPG and the poor performance on the GPG could indicate a lack of robustness of the XAI methods at hand w.r.t. localization. This is supported by Figure 1, where the localization works on a single image, and Figure 2, where the localization performance is poor and the quality of the explanation looks more pebbly. In our work the best model was selected based on AUC. Future work could investigate if selecting models on localization performance makes them more robust for the given task while it might also benefit classification performance in general, as the model focuses more on the relevant parts of the image. This might also be done via a unified loss function.

Our proposed MPG also has its limitations. As depicted in Figure 3 the mask is not always finely aligned with the forgery area, meaning that the MPG will likely be forgiving and therefore more influenced by noisy pixels. Albeit, the noise issue can be counteracted via the weighted accuracy metric and our thresholding scheme. Also, not every part within the mask has been altered to the same extend. E.g. a swapped mouth movement represents a stronger forgery than a slight change in skin or hair color and texture. The strength of the forgery cannot be weighted in the MPG.

VII. Conclusion

Our study highlights the promises and challenges of using inherently interpretable models for deepfake detection. The B-cos framework allows fine-grained, faithful explanations of model decisions and shows strong performance in attribution quality when evaluated via the MPG and stable AUC results for increasing B values. However, our results also show that B-cos networks are fragile and highly sensitive to hyperparameters and architecture-specific quirks, which often require extensive tuning to achieve competitive performance.

Moreover, B-cos explanations performed poorly in the GPG which turned out to be mitigable to a certain point by applying a threshold to the attribution maps.

REFERENCES

[1] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 852–863. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/076ccd93ad68be51f23707988e934906-Paper.pdf

[2] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 674–10 685.

[3] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis, T. Karras, and G. Wetzstein, "Efficient geometry-aware 3d generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 16 123–16 133.

[4] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 8633–8646. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/39235c56aef13fb05a6adc95eb9d8d66-Paper-Conference.pdf

[5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, p. 99–106, Dec. 2021. [Online]. Available: https://doi.org/10.1145/3503250

[6] S. Yang, L. Leng, C.-C. Chang, and C.-C. Chang, "Reversible adversarial examples with minimalist evolution for recognition control in computer vision," *Applied Sciences*, vol. 15, no. 3, 2025. [Online]. Available: https://www.mdpi.com/2076-3417/15/3/1142

[7] R. Chesney and D. Citron, "Deepfakes and the new disinformation war: The coming age of post-truth geopolitics," *Foreign Affairs*, vol. 98, no. 1, pp. 147–155, 2019. [Online]. Available: https://www.foreignaffairs.com/articles/2018-12-11/deepfakes-and-new-disinformation-war

[8] N. C. Köbis, J.-F. Bonnefon, A. Shariff, and I. Rahwan, "Fooled by deepfakes: People overestimate their detection abilities but fail to detect manipulated videos," *iScience*, vol. 24, no. 2, p. 102129, 2021.

[9] M. Böhle, M. Fritz, and B. Schiele, "B-cos networks: Alignment is all we need for interpretability," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 329–10 338.

[10] M. Böhle, N. Singh, M. Fritz, and B. Schiele, "B-cos alignment for inherently interpretable cnns and vision transformers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 6, pp. 4504–4518, 2024.

[11] Z. Yan, Y. Zhang, X. Yuan, S. Lyu, and B. Wu, "Deepfakebench: A comprehensive benchmark of deepfake detection," in *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 4534–4565. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/0e735e4b4f07de483cbe250130992726-Paper-Datasets_and_Benchmarks.pdf

[12] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[13] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726–742, 2021.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, p. 618–626.

[16] V. Kamakshi and N. C. Krishnan, "Explainable image classification: The journey so far and the road ahead," *AI*, vol. 4, no. 3, pp. 620–651, 2023. [Online]. Available: https://www.mdpi.com/2673-2688/4/3/33

[17] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, and C. Rudin, *This looks like that: deep learning for interpretable image recognition*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[18] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 839–847.

[19] R. Fu, Q. Hu, X. Dong, Y. Guo, Y. Gao, and B. Li, "Axiom-based grad-cam: Towards accurate visualization and explanation of cnns," 2020.

[20] P.-T. Jiang, C.-B. Zhang, Q. Hou, and Y. Wei, "Layercam: Exploring hierarchical class activation maps," *IEEE Transactions on Image Processing*, vol. PP, pp. 1–1, 06 2021.

[21] K. Tsigos, E. Apostolidis, S. Baxevanakis, S. Papadopoulos, and V. Mezaris, "Towards quantitative evaluation of explainable ai methods for deepfake detection," in *Proceedings of the 3rd ACM International Workshop on Multimedia AI against Disinformation*, ser. MAD '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 37–45. [Online]. Available: https://doi.org/10.1145/3643491.3660292

[22] B. Gowrisankar and V. L. Thing, "An adversarial attack approach for explainable ai evaluation on deepfake detection models," *Computers Security*, vol. 139, p. 103684, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404823005941

[23] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," in *British Machine Vision Conference (BMVC)*, 2018. [Online]. Available: http://bmvc2018.org/contents/papers/1064.pdf

[24] L. Schallner, J. Rabold, O. Scholz, and U. Schmid, "Effect of superpixel aggregation on explanations in lime – a case study with biological data," in *Machine Learning and Knowledge Discovery in Databases*, P. Cellier and K. Driessens, Eds. Cham: Springer International Publishing, 2020, pp. 147–158.

[25] Y. Lei, S. Yan, J. Zhang, X. Li, P. Wang, X. Gao, and H. Cao, "Bag-net: A novel architecture for enhanced medical image segmentation with global context attention and boundary self-attention," *Symmetry*, vol. 17, no. 4, 2025. [Online]. Available: https://www.mdpi.com/2073-8994/17/4/531

[26] M. Böhle, M. Fritz, and B. Schiele, "Convolutional Dynamic Alignment Networks for Interpretable Classifications," 6 2021. [Online]. Available: https://publications.cispa.de/articles/conference_contribution/Convolutional_Dynamic_Alignment_Networks_for_Interpretable_Classifications/24613644

[27] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *International Conference on Learning Representations*, 2018.

[28] S. Srinivas and F. Fleuret, "Rethinking the role of gradient-based attribution methods for model interpretability," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=dYeAHXnpWJ4

[29] N. Ur Rehman Ahmed, A. Badshah, H. Adeel, A. Tajammul, A. Daud, and T. Alsahfi, "Visual deepfake detection: Review of techniques, tools, limitations, and future prospects," *IEEE Access*, vol. 13, pp. 1923–1961, 2025.

[30] R. Sunil, P. Mer, A. Diwan, R. Mahadeva, and A. Sharma, "Exploring autonomous methods for deepfake detection: A detailed survey on techniques and evaluation," *Heliyon*, vol. 11, 2025.

[31] J. Xu, X. Liu, W. Lin, W. Shang, and Y. Wang, "Localization and detection of deepfake videos based on self-blending method," *Scientific Reports*, vol. 15, 2025.

[32] M. A. Talib, Q. Nasir, A. B. Nassif, N. B. Fadhl, and O. Gouda, "Chrominance and luminance: a study to detect deepfakes," *Multimedia Tools and Applications*, 2025.

[33] G. Bendiab, H. Haiouni, I. Moulas, and S. Shiaeles, "Deepfakes in digital media forensics: Generation, ai-based detection and challenges," *J. Inf. Secur. Appl.*, vol. 88, p. 103935, 2025.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[36] M. Abbasi, P. Váz, J. Silva, and P. Martins, "Comprehensive evaluation of deepfake detection models: Accuracy, generalization, and resilience to adversarial attacks," *Applied Sciences*, vol. 15, no. 3, p. 1225, 2025. [Online]. Available: https://doi.org/10.3390/app15031225

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[38] Z. Wang, Z. Cheng, J. Xiong, X. Xu, T. Li, B. Veeravalli, and X. Yang, "A timely survey on vision transformer for deepfake detection," *CoRR*, vol. abs/2405.08463, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2405.08463

[39] J. Yan, Z. Li, Z. He, and Z. Fu, "Generalizable deepfake detection via effective local-global feature extraction," 2025. [Online]. Available: https://doi.org/10.48550/arxiv.2501.15253

[40] T. C. F. Polo and H. A. Miot, "Use of roc curves in clinical and experimental studies," *Jornal Vascular Brasileiro*, vol. 19, p. e20200186, 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8218006/

[41] Y. Patel, S. Tanwar, P. Bhattacharya, R. Gupta, T. Alsuwian, I. E. Davidson, and T. F. Mazibuko, "An improved dense cnn architecture for deepfake image detection," *IEEE Access*, vol. 11, pp. 22 081–22 095, 2023.

[42] Z. Yan, T. Yao, S. Chen, Y. Zhao, X. Fu, J. Zhu, D. Luo, C. Wang, S. Ding, Y. Wu, and L. Yuan, "Df40: Toward next-generation deepfake detection," 2024. [Online]. Available: https://arxiv.org/abs/2406.13495

[43] SCLBD, "Deepfakebench: A comprehensive benchmark suite for deepfake detection," https://github.com/SCLBD/DeepfakeBench, 2024, accessed: 2025-04-28.

[44] T. Gomez and H. Mouchère, "Computing and evaluating saliency maps for image classification: a tutorial," *Journal of Electronic Imaging*, vol. 32, 2023.

[45] S. Z. S. Samuel, V. Kamakshi, N. Lodhi, and N. C. Krishnan, "Evaluation of Saliency-based Explainability Method," Jun. 2021, arXiv:2106.12773 [cs]. [Online]. Available: http://arxiv.org/abs/2106.12773

[46] A. Alqaraawi, M. Schuessler, P. Weiß, E. Costanza, and N. Berthouze, "Evaluating saliency map explanations for convolutional neural networks: a user study," in *Proceedings of the 25th International Conference on Intelligent User Interfaces*, ser. IUI '20. New York, NY, USA: Association for Computing Machinery, Mar. 2020, pp. 275–285. [Online]. Available: https://doi.org/10.1145/3377325.3377519

[47] Y. Zhang, J. Song, S. Gu, T. Jiang, B. Pan, G. Bai, and L. Zhao, "Saliency-bench: A comprehensive benchmark for evaluating visual explanations," 2025. [Online]. Available: https://arxiv.org/abs/2310.08537

[48] S. Srinivas and F. Fleuret, "Full-gradient representation for neural network visualization," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., Dec. 2019, no. 371, pp. 4124–4133.

[49] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Muller, "Evaluating the Visualization of What a Deep Neural Network Has Learned," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2660–2673, Nov. 2017.

[50] T. Gomez, T. Fréour, and H. Mouchère, "Metrics for Saliency Map Evaluation of Deep Learning Explanation Methods," in *Pattern Recognition and Artificial Intelligence: Third International Conference, ICPRAI 2022, Paris, France, June 1–3, 2022, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, Jun. 2022, pp. 84–95. [Online]. Available: https://doi.org/10.1007/978-3-031-09037-0_8

[51] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-Down Neural Attention by Excitation Backprop," *International Journal of Computer Vision*, vol. 126, no. 10, pp. 1084–1102, Oct. 2018. [Online]. Available: https://doi.org/10.1007/s11263-017-1059-x

[52] M. Böhle, M. Fritz, and B. Schiele, "Convolutional dynamic alignment networks for interpretable classifications," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[53] S. Arya, S. Rao, M. Boehle, and B. Schiele, "B-cosification: Transforming deep neural networks to be inherently interpretable," in *38th Conference on Neural Information Processing Systems*, 2024.

[54] F. Baldassarre, Q. Debard, G. F. Pontiveros, and T. K. Wijaya, "Quantitative metrics for evaluating explanations of video deepfake detectors," in *33rd British Machine Vision Conference (BMVC)*, 2022.

[55] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[56] B. Kim, J. Seo, and T. Jeon, "Bridging adversarial robustness and gradient interpretability," *Safe Machine Learning workshop at ICLR*, 2019.

[57] H. Shah, P. Jain, and P. Netrapalli, "Do input gradients highlight discriminative features?" in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 2046–2059. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/0fe6a94848e5c68a54010b61b3e94b0e-Paper.pdf

[58] M. Bohle, M. Fritz, and B. Schiele, "Convolutional dynamic alignment networks for interpretable classifications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 10 029–10 038.

[59] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, "This looks like that: Deep learning for interpretable image recognition," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/adf7ee2dcf142b0e11888e72b43fcb75-Paper.pdf

[60] W. Brendel and M. Bethge, "Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=SkfMWhAqYQ

[61] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "Cutmix: Regularization strategy to train strong classifiers with localizable features," 10 2019, pp. 6022–6031.

[62] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[63] S. G. Muller and F. Hutter, " TrivialAugment: Tuning-free Yet State-of-the-Art Data Augmentation ," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2021, pp. 754–762. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00081

[64] A. Krizhevsky, "Learning multiple layers of features from tiny images," https://www.cs.toronto.edu/~kriz/cifar.html, 2009, technical report, University of Toronto.

[65] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[66] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[67] FaceSwap Contributors, "FaceSwap: Deepfakes Software For All," https://github.com/deepfakes/faceswap, 2025, accessed: 2025-01-14.

[68] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2387–2395.

[69] M. Kowalski, "Faceswap: 3d face swapping implemented in python," https://github.com/MarekKowalski/FaceSwap, 2025, accessed: 2025-01-14.

[70] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: image synthesis using neural textures," *ACM Trans. Graph.*, vol. 38,

no. 4, Jul. 2019. [Online]. Available: https://doi.org/10.1145/3306346.3323035

[71] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "Advancing high fidelity identity swapping for forgery detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[72] N. Dufour and A. Gully, "Contributing data to deepfake detection research," https://research.google/blog/contributing-data-to-deepfake-detection-research/, September 2019, accessed: 2025-04-28.

[73] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. Ferrer, "The deepfake detection challenge (dfdc) preview dataset," 10 2019.

[74] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3204–3213.

[75] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking," 2018. [Online]. Available: https://arxiv.org/abs/1806.02877

[76] T. Karras, S. Laine, and T. Aila, "flickr-faces-hq dataset (ffhq)," 2019, accessed: 2025-04-30. [Online]. Available: https://github.com/NVlabs/ffhq-dataset

[77] B-cos, "B-cos-v2," https://github.com/B-cos/B-cos-v2, 2025, accessed: 2025-04-29.

[78] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollar, and R. Girshick, "Early convolutions help transformers see better," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 30 392–30 400. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/ff1418e8cc993fe8abcfe3ce2003e5c5-Paper.pdf

[79] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[80] J. Maurício, I. Domingues, and J. Bernardino, "Comparing vision transformers and convolutional neural networks for image classification: A literature review," *Applied Sciences*, vol. 13, p. 5521, 04 2023.

[81] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, 2012. [Online]. Available: http://jmlr.org/papers/v13/bergstra12a.html

[82] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: a novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, no. 1, p. 6765–6816, Jan. 2017.

[83] L. Biewald, "Experiment tracking with weights and biases," 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/

[84] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[85] L. Guan, D. Li, Y. Shi, and J. Meng, "Xgrad: Boosting gradient-based optimizers with weight prediction," 2024.

[86] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 2818–2826.

[87] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 11 966–11 976.

[88] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 4700–4708.

[89] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=6Tm1mposlrM

[90] M. Böhle, M. Fritz, and B. Schiele, "Holistically Explainable Vision Transformers," Jan. 2023, arXiv:2301.08669 [cs]. [Online]. Available: http://arxiv.org/abs/2301.08669

[91] Flickr, "Flickr: Image and video hosting platform," 2025, accessed: 2025-04-30. [Online]. Available: https://www.flickr.com/

[92] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The deepfake detection challenge (dfdc) dataset," 2020. [Online]. Available: https://arxiv.org/abs/2006.07397

APPENDIX

DATASETS

The descriptive statistics of the datasets are shown in Table V. They are sourced from the DeepfakeBench repository, see [11], [43]. On top of that, we also considered the *Flickr-Faces-HQ (FFHQ)* dataset, published by NVIDIA [76]. It is a high-quality image dataset designed for generative models and facial analysis tasks. It consists of 70000 high-resolution human face images at 1024x1024, collected from Flickr [91] and processed to be aligned, centered, and standardized. FFHQ features a wide variation in age, ethnicity, and image background, and includes accessories like eyeglasses, hats, and hairstyles, making it promising for robust model training and evaluation.

TABLE V
OVERVIEW OF DEEPFAKE DATASETS TAKEN FROM [11]. THE IMAGES ARE CREATED BY SAMPLING EVERY 32ND FRAME.

Dataset	Real Videos	Fake Videos	Total Videos	Rights Cleared	Total Subjects	Synthesis Methods	Perturbations
FaceForensics++ [66]	1000	4000	5000	NO	N/A	4	2
FaceShifter [71]	1000	1000	2000	NO	N/A	1	-
DeepfakeDetection [72]	363	3000	3363	YES	28	5	-
DFDC (Preview) [73]	1131	4119	5250	YES	66	2	3
DFDC [92]	23654	104500	128154	YES	960	8	19
CelebDF-v2 [74]	590	5639	6229	NO	59	1	-
UADFV [75]	49	49	98	NO	49	1	-

HYPERPARAMETER SPACE

The hyperparameter optimization search space was mostly shared across all models.

- Global hyperparameters included:
  - Learning rate (LR) $\in [2.5\text{e}{-5}, 2.5\text{e}{-3}]$
  - Weight decay $\in [1\text{e}{-7}, 1\text{e}{-3}]$
  - Adam betas:
    - $\beta_1 \in [0.85, 0.95]$
    - $\beta_2 \in [0.8, 0.999]$
  - $\epsilon \in [1\text{e}{-8}, 1\text{e}{-5}]$
  - `amsgrad` $\in$ {True, False}
  - Data `augmentation` $\in$ {True, False}
  - `norm_bias` $\in$ {True, False}
- Model-specific hyperparameters included:
  - For ResNet:
    - `norm` $\in$ {AllNormUncentered2d, BatchNormUncentered2d, GroupNormUncentered2d, GNInstanceNormUncentered2d, GNLayerNormUncentered2d, PositionNormUncentered2d, AllNorm2d, BatchNorm2d, DetachableGroupNorm2d, DetachableGNInstanceNorm2d, DetachableGNLayerNorm2d, DetachableLayerNorm, DetachablePositionNorm2d}
  - For ViT:
    - `norm_2d` $\in$ {BatchNorm2d, BatchNormUncentered2d}
    - `norm_2d_bias` $\in$ {True, False}

Table VI shows the combinations that yielded the best performance in terms of validation AUC. We used `Weights and Biases` [83] and `hyperband` [82] for convenient hyperparameter optimization.

TABLE VI

BEST HPO CONFIGURATIONS TRAINED ON FF++ C23. TRAINED FOR A MAXIMUM OF 20 EPOCHS. [1] = DEFAULT HYPERPARAMETERS FROM [11] USED. [2] = TRAINED FOR 5 EPOCHS ONLY DUE TO COMPUTATIONAL LIMITATIONS

		Adam parameters							model parameters		
Model	**BatchSize**	**LR**	**Decay**	β_1	β_2	**eps**	**amsgrad**	**augmentation**			
Resnet34 (Vanilla)	128	2.7×10^{-4}	8.6×10^{-4}	0.86	0.86	4.2×10^{-6}	True	False	–	–	–
									norm	norm_bias	–
Resnet34 (B=1.25)	32	4.3×10^{-4}	3.7×10^{-4}	0.92	0.82	5.0×10^{-6}	True	False	BN2d	False	–
Resnet34 (B=1.75)	32	8.7×10^{-4}	1.0×10^{-4}	0.94	0.86	6.0×10^{-6}	True	False	GNINU2d	False	–
Resnet34 (B=2)	32	$8.6 \times 10{-4}$	1.0×10^{-4}	0.95	0.86	6.1×10^{-6}	True	False	GNINU2d	False	
Resnet34 (B=2.5)	16	8.9×10^{-4}	4.0×10^{-4}	0.86	0.88	8.0×10^{-6}	True	False	BNU2d	False	–
Xception (Vanilla)[1]	64	2.0×10^{-4}	5.0×10^{-4}	0.90*	0.99	1.0×10^{-8}	False	False	-	-	
Xception (B=2.5)[1,2]	64	2.0×10^{-4}	5.0×10^{-4}	0.90*	0.99	1.0×10^{-8}	False	False	-	-	
ViT (Vanilla)	64	7.2×10^{-5}	3.5×10^{-4}	0.85	0.86	6.5×10^{-5}	False	False	–	–	–
									norm_bias	norm_2d	norm_2d_bias
ViT (B=1.25)	16	1.6×10^{-4}	2.1×10^{-4}	0.93	0.83	8.8×10^{-6}	True	False	True	BN2d	False
ViT (B=1.75)	16	5.5×10^{-5}	5.3×10^{-4}	0.89	0.94	4.4×10^{-6}	False	False	False	BN2d	True
ViT (B=2)	128	3.2×10^{-4}	3.5×10^{-4}	0.88	0.92	4.7×10^{-6}	False	False	True	BN2d	False
ViT (B=2.5)	64	1.2×10^{-4}	7.6×10^{-4}	0.92	0.95	2.6×10^{-7}	False	False	False	BN2d	False

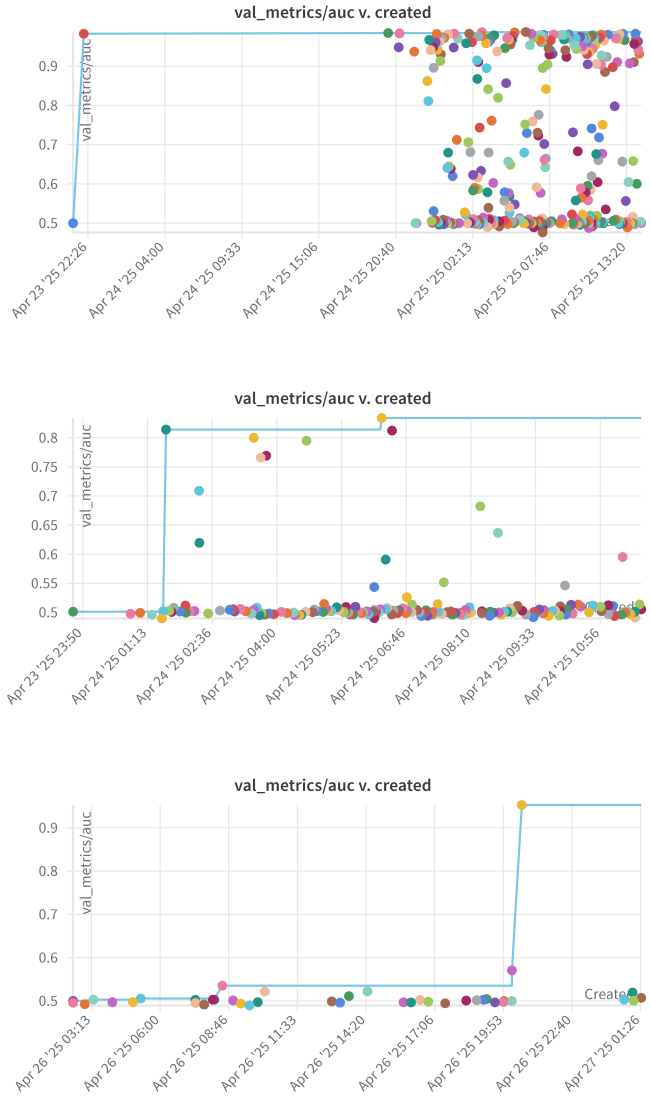

Fig. 7. Runs for hyperparameter optimization on *FaceForensics++* [66] for a vanilla ResNet34 (Top) and a B-cos ResNet34 (b=1.25) (Middle) plus a B-cos ResNet34 (b=1.25) on *FFHQ* [76] (Bottom), showcasing that good hyperparameter combinations are much harder to find for B-cosified networks. Each dot represents a hyperparameter combination, while the y-axis represents the AUC scored with this combintaion. Illustrations are taken from Weights and Biases [83]

ADDITIONAL DEEPFAKE DETECTION RESULTS

In the following, we first display detailed results across FF++ with c23 and c40 compression levels. Table VII extends Table I by also providing average precision, recall and accuracy scores. It includes results for ResNet34, Xception, and ViT models in both their vanilla and B-cosified variants across different B-parameter values. Xception (Vanilla) is also the leading model with respect to accuracy, which was to be expected since it was not trained with our limited resources, but in a more sophisticated way. In addition to this, Table VIII shows the performance of the models on c40, where we followed [11] by training the models on c23 compression while testing them on c40 compression. The lower c40 compression level is seemingly a more challenging task, since it comes with a general decrease in AUC compared to c23 performances. While ResNet34 models show consistent robustness, the performance of vanilla Xception drops more significantly in AUC. ViT models continue to demonstrate very high recall, even under low AUC conditions, highlighting their tendency toward over-detection of fakes.

TABLE VII
DETAILED PERFORMANCE METRICS ON FACEFORENSICS++ WITH C23 COMPRESSION.

Model	AUC	F1	Avg. Precision	Recall	Accuracy
Resnet34 (Vanilla)	0.88	0.91	0.96	0.94	0.86
Resnet34 (B=1.25)	0.78	0.89	0.93	0.95	0.81
Resnet34 (B=1.75)	0.82	0.88	0.94	0.88	0.81
Resnet34 (B=2)	0.84	0.89	0.95	0.89	0.82
Resnet34 (B=2.5)	0.84	0.90	0.95	0.93	0.84
Xception (Vanilla)	0.98	0.96	0.99	0.97	0.93
Xception (B=2.5)	0.75	0.89	0.91	0.95	0.81
ViT (Vanilla)	0.65	0.89	0.87	1.0	0.80
ViT (B=1.25)	0.72	0.88	0.9	0.94	0.8
ViT (B=1.75)	0.77	0.87	0.93	0.88	0.8
ViT (B=2)	0.78	0.87	0.93	0.86	0.79
ViT (B=2.5)	0.78	0.88	0.93	0.92	0.81

TABLE VIII
DETAILED PERFORMANCE METRICS ON FACEFORENSICS++ WITH C40 COMPRESSION.

Model	AUC	F1	Avg. Precision	Recall	Accuracy
Resnet34 (Vanilla)	0.79	0.89	0.93	0.94	0.82
Resnet34 (B=1.25)	0.74	0.89	0.92	0.98	0.81
Resnet34 (B=1.75)	0.78	0.89	0.93	0.93	0.81
Resnet34 (B=2)	0.79	0.89	0.93	0.92	0.81
Resnet34 (B=2.5)	0.77	0.89	0.93	0.92	0.81
Xception (Vanilla)	0.80	0.83	0.94	0.76	0.75
Xception (B=2.5)	0.73	0.89	0.91	0.98	0.81
ViT (Vanilla)	0.63	0.89	0.85	1.0	0.8
ViT (B=1.25)	0.67	0.88	0.88	0.97	0.79
ViT (B=1.75)	0.72	0.88	0.9	0.94	0.88
ViT (B=2)	0.73	0.88	0.91	0.93	0.80
ViT (B=2.5)	0.73	0.89	0.91	0.97	0.81

After that, we investigate the out-of-distribution performance in more detail. First, Table IX and Table VIII present the out-of-distribution performance across several datasets in c23 and c40 compression respectively, including in-domain FF subtypes as well as out-of-domain CDFv2, DFD, DFDCP, Fsh, and UADFV datasets. In the FaceForensics++ datasets, labelled with FF, models generally perform well under both c23 and c40 compression levels. The vanilla Xception achieves the highest scores on c23. However, when evaluated on out-of-distribution (OOD) datasets (Table IX), all models suffer a notable performance drop. For example, ResNet34 (Vanilla) drops from 0.92 AUC on FF-DF to 0.65 on CDFv2. This highlights a key challenge in deepfake detection: poor generalization to unseen manipulations due to overfitting to dataset-specific artifacts. B-cosified models tend to slightly generalize better across OOD datasets, with ResNet34 and ViT variants showing improved or more consistent relative AUC and F1 scores compared to their vanilla counterparts. This suggests B-cosification may help models learn more transferable features, mitigating overfitting and improving real-world robustness, possibly due to their input alignment properties. Nevertheless, the real impact of this may be doubted as well, since the AUC are relatively low [40]. As expected, most models experience a performance drop due to increased compression in Table X. Importantly, CDFv2, DFDCP, and UADFV do not provide c40-compressed data, hence their entries are omitted. Nevertheless, trends from the c23 setting remain largely consistent, with B-cosified models maintaining competitive AUC and F1 values while achieving similar OOD performance.

To further assess model robustness, we trained and evaluated ResNet34 variants on the FFHQ dataset, a more balanced data set than FF++. Table XI shows that even with only 10 epochs of training, all models perform very well.

TABLE IX
DETAILED OUT OF DISTRIBUTION PERFORMANCE ON C23 COMPRESSION.

Model	**FF-F2F**		**FF-DF**		**FF-FS**		**FF-NT**		**CDFv2**		**DFD**		**DFDCP**		**Fsh**		**UADFV**	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
Resnet34 (Vanilla)	0.9	0.79	0.92	0.8	0.9	0.79	0.8	0.75	0.65	0.8	0.69	0.87	0.63	0.79	0.66	0.64	0.78	0.77
Resnet34 (1.25)	0.83	0.72	0.85	0.73	0.78	0.71	0.65	0.67	0.57	0.78	0.57	0.91	0.55	0.79	0.63	0.67	0.9	0.74
Resnet34 (1.75)	0.83	0.76	0.87	0.78	0.83	0.76	0.76	0.72	0.63	0.78	0.58	0.77	0.61	0.76	0.63	0.61	0.84	0.77
Resnet34 (2)	0.85	0.77	0.89	0.78	0.87	0.77	0.75	0.71	0.64	0.75	0.56	0.81	0.66	0.78	0.66	0.64	0.93	0.80
Resnet34 (2.5)	0.87	0.77	0.87	0.77	0.87	0.77	0.77	0.73	0.63	0.77	0.59	0.85	0.56	0.75	0.62	0.62	0.84	0.77
Xception (Vanilla)	0.99	0.89	0.99	0.89	0.98	0.88	0.95	0.86	0.74	0.76	0.85	0.88	0.73	0.74	0.59	0.40	0.94	0.82
Xception (2.5)	0.76	0.71	0.86	0.73	0.75	0.71	0.64	0.68	0.61	0.80	0.55	0.88	0.62	0.81	0.67	0.70	0.73	0.72
ViT (Vanilla)	0.65	0.67	0.75	0.67	0.61	0.67	0.60	0.67	0.57	0.79	0.54	0.94	0.59	0.79	0.65	0.79	0.78	0.67
ViT (1.25)	0.78	0.7	0.79	0.7	0.66	0.67	0.65	0.68	0.58	0.79	0.57	0.82	0.58	0.78	0.63	0.67	0.76	0.71
ViT (1.75)	0.79	0.73	0.84	0.76	0.77	0.73	0.69	0.68	0.61	0.79	0.58	0.83	0.58	0.77	0.62	0.62	0.84	0.77
ViT (2)	0.80	0.74	0.85	0.77	0.78	0.73	0.68	0.67	0.64	0.79	0.58	0.77	0.58	0.75	0.62	0.62	0.74	0.73
ViT (2.5)	0.81	0.73	0.86	0.75	0.79	0.72	0.67	0.68	0.62	0.8	0.57	0.85	0.58	0.78	0.64	0.65	0.78	0.75

TABLE X
DETAILED OUT OF DISTRIBUTION PERFORMANCE ON C40 COMPRESSION. NOTE THAT THE CDFV2, DFDCP AND UADFV DATA SETS ONLY POSSESS C23 COMPRESED IMAGES.

Model	**FF-F2F**		**FF-DF**		**FF-FS**		**FF-NT**		**CDFv2**		**DFD**		**DFDCP**		**Fsh**		**UADFV**	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
Resnet34 (Vanilla)	0.8	0.72	0.82	0.74	0.84	0.74	0.69	0.69	–	–	0.59	0.85	–	–	0.63	0.67	–	–
Resnet34 (1.25)	0.79	0.69	0.82	0.69	0.75	0.69	0.62	0.67	–	–	0.55	0.93	–	–	0.62	0.67	–	–
Resnet34 (1.75)	0.78	0.92	0.83	0.73	0.81	0.73	0.70	0.69	–	–	0.56	0.81	–	–	0.64	0.66	–	–
Resnet34 (2)	0.80	0.73	0.85	0.74	0.84	0.74	0.69	0.69	–	–	0.54	0.82	–	–	0.67	0.67	–	–
Resnet34 (2.5)	0.8	0.73	0.81	0.73	0.82	0.74	0.66	0.67	–	–	0.57	0.84	–	–	0.63	0.66	–	–
Xception (Vanilla)	0.81	0.74	0.87	0.79	0.87	0.79	0.66	0.58	–	–	0.58	0.53	–	–	0.62	0.53	–	–
Xception (2.5)	0.73	0.68	0.83	0.70	0.76	0.69	0.61	0.67	–	–	0.54	0.90	–	–	0.65	0.68	–	–
ViT (Vanilla)	0.63	0.67	0.71	0.67	0.61	0.67	0.58	0.67	–	–	0.63	0.67	–	–	0.63	0.67	–	–
ViT (1.25)	0.72	0.68	0.73	0.68	0.64	0.67	0.6	0.67	–	–	0.54	0.86	–	–	0.61	0.67	–	–
ViT (1.75)	0.73	0.7	0.77	0.71	0.74	0.7	0.65	0.68	–	–	0.55	0.87	–	–	0.61	0.67	–	–
ViT (2)	0.75	0.70	0.78	0.72	0.73	0.70	0.65	0.68	–	–	0.55	0.79	–	–	0.62	0.66	–	–
ViT (2.5)	0.76	0.69	0.79	0.7	0.74	0.69	0.64	0.68	–	–	0.55	0.87	–	–	0.62	0.67	–	–

TABLE XI
DETAILED PERFORMANCE METRICS FOR RESNET TRAINED AND TESTED ON FFHQ. [1] = TRAINED FOR ONLY 10 EPOCHS

Model	**AUC**	**F1**	**Avg. Precision**	**Recall**	**Accuracy**
Resnet34 (Vanilla)[1]	0.99	0.93	0.88	0.97	0.92
Resnet34 (B=1.25)	0.96	0.89	0.87	0.90	0.88
Resnet34 (B=1.75)	0.99	0.96	0.99	0.93	0.96
Resnet34 (B=2)	0.99	0.95	0.95	0.95	0.95
Resnet34 (B=2.5)	0.99	0.95	0.97	0.93	0.95

ADDITIONAL XAI RESULTS

We further analyze qualitative attribution maps for B-cos models. Figure 8 shows attribution maps for the B-cos ViT across varying B values. As with the ResNet34 results in Figure 1, increasing B sharpens and localizes the heatmaps. However, the ViT explanations are less interpretable: the attribution colors are not as well aligned with the input image, and even at higher B values, the facial shapes remain only partially visible. In contrast, the B-cos XceptionNet model (Figure 9) produces more coherent and interpretable attribution maps. Key facial regions are more distinctly highlighted, suggesting better visual grounding. However, since only one B-cos XceptionNet variant was available, we could not examine how different B values affect its attributions.

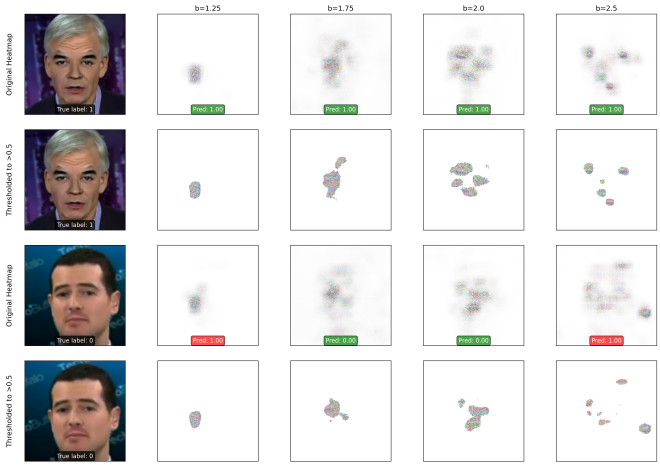

Fig. 8. B-cosified ViT attribution maps with B-parameters 1.25, 1.75, 2 and 2.5 and a threshold of 0.5. The attribution maps align with the predicted class instead of the true class. Examples stem from FF++.

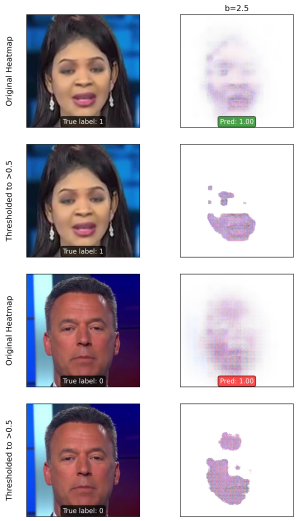

Fig. 9. B-cosified Xception attribution maps with B-parameter 2.5 and a threshold of 0.5. The attribution maps align with the predicted class instead of the true class. Examples stem from FF++.

Fig. 10. Direct comparison between the GPG attribution maps of different versions of Grad-CAM, Lime on XeptionNet and our B-cosified XeptionNet.

Figure 10 shows two different GPG plots for the XceptionNet and its XAI methods. One can clearly see that Grad++ and LayerCAM localize the fake (bottom left) well. While XceptionNet B-cos also shows decent performance, the visualization looks pebbly. LIME seems confused and highlights regions throughout the grid. In general all models performs worse on the GPG at the top.

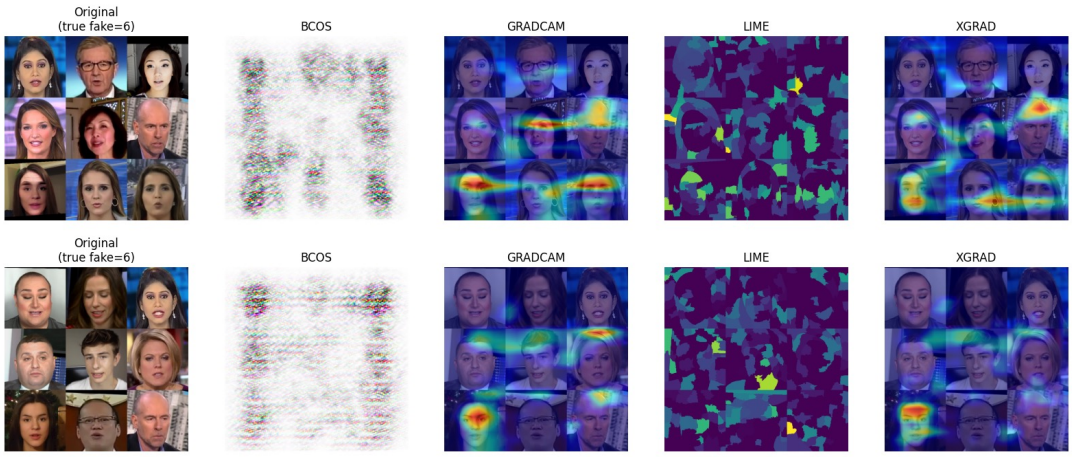

Fig. 11. Direct comparison between the GPG attribution maps of different versions of Grad-CAM, Lime on ResNet34 and our B-cosified ResNet34.

Figure 11 shows more GPG examples, to also compare Grad-CAM and XGrad-CAM. All models focus on multiple parts of the grid. Grad-CAM and XGrad-CAM may performs slighter better than their competitors. Resnet-34 B-cos looks a little less pebbly than XceptionNet B-cos, but still is not visually appealing.

Table XII compares the interpretability performance of the B-cos ResNet34 model ($B = 1.25$) against post-hoc explainability methods applied on top of the same B-cos model. Results are reported on the FaceForensics++ dataset with c23 compression for both the GPG and the MPG, using both unweighted and weighted variants. Overall, the explanation performance across all methods is quite similar in both GPG and unweighted MPG, suggesting that B-cos explanations, while inherently faithful by design, are not necessarily superior. However, B-cos stands out significantly in the weighted MPG score, outperforming all other methods and indicating stronger localization around relevant image regions. Lime results are not included, as it has not yet been integrated for B-cos models in this setting.

TABLE XII
DETAILED GRID POINTING GAME AND MASK POINTING GAME PERFORMANCES WITH RESNET34-BCOS B = 1.25 (NO THRESHOLD) ON FF++ WITH C23 COMPRESSION.

Model	**GridPointingGame**		**MaskPointingGame**	
	Unweighted	Weighted	Unweighted	Weighted
B = 1.25	0.1111	0.1580	0.2666	0.7497
Grad-CAM	0.1237	0.2982	0.2747	0.3348
LayerCam	0.1348	0.3548	0.2761	0.3381
XGrad	0.1237	0.2982	0.2733	0.3334
Grad++	0.1237	0.2982	0.2756	0.3355
Lime	N/A	N/A	N/A	N/A

The distribution of weighted GPG localization scores are shown in Figure 12 and 13 for the B-cos ResNet and Xception models respectively. The performances of XAI methods are relatively evenly spread in Figure 12. Nevertheless, the best performing B-cos ResNet (with $B = 1.75$) is clearly worse than GradCAM, Grad++ and LayerCAM. The B-cos ResNet with $B = 2.5$ performs the worst. In contrast, XAI methods generally produce better with the Xception model, possibly due to the better underlying model. Moreover, this particularly holds for the post-hoc XAI methods. The B-cos Xception model, while performing on par with the best B-cos ResNet, is again outperformed by post-hoc methods. Grad-CAM++ and LayerCAM are again among the best performing methods, while XGrad exhibits a very wide distribution. LIME performs poorly regardless of the underlying model.

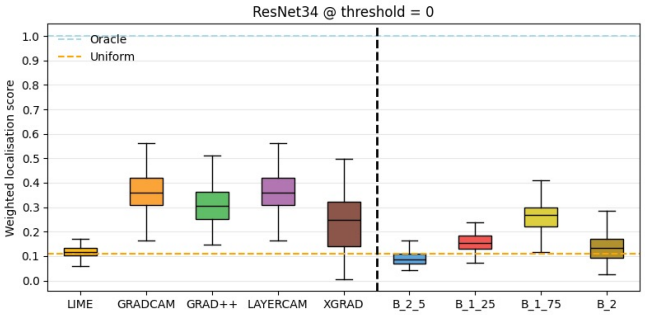

Fig. 12. Box plots of weighted localization scores from our Grid Pointing Game across different XAI methods for ResNet34.

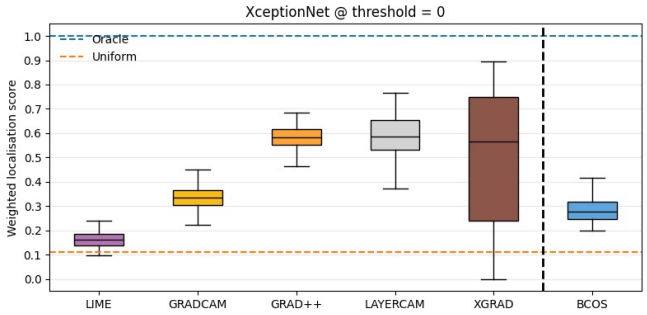

Fig. 13. Box plots of weighted localization scores from our Grid Pointing Game across different XAI methods for XceptionNet.

Figures 14,15, 16 are complementary visualizations to Section V-E.

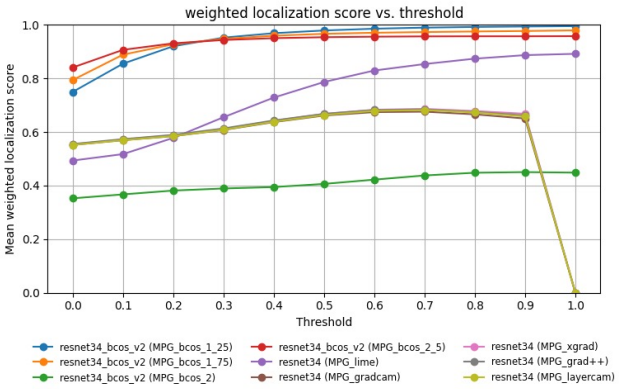

Fig. 14. Weighted localization scores from our Mask Pointing Game across different XAI methods for ResNet34 plotted over different threshold values.

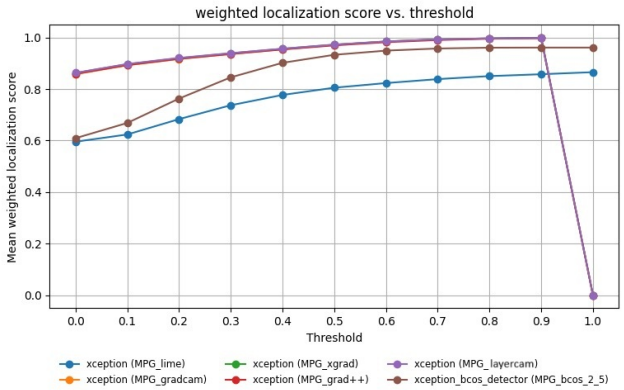

Fig. 15. Weighted localization scores from our Mask Pointing Game across different XAI methods for XceptionNet plotted over different threshold values.

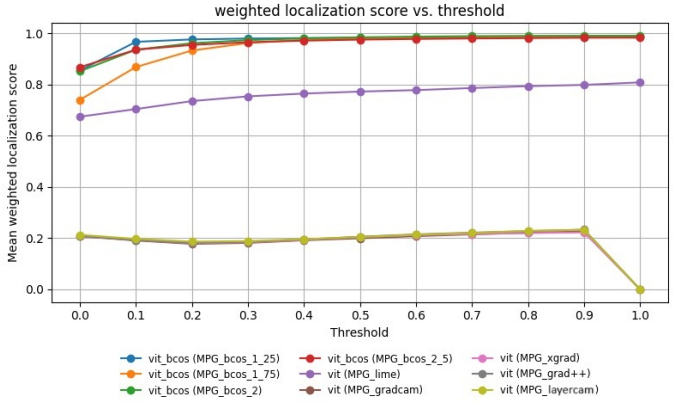

Fig. 16. Weighted localization scores from our Mask Pointing Game across different XAI methods for ViT plotted over different threshold values.