xCos: An Explainable Cosine Metric for Face Verification Task

Yu-Sheng Lin[1], Zhe-Yu Liu[1], Yu-An Chen[1], Yu-Siang Wang[2], Hsin-Ying Lee[1], Yi-Rong Chen[1], Ya-Liang Chang[1], and Winston H. Hsu[1]

[1] National Taiwan University, Taipei, Taiwan
[2] University of Toronto, Toronto, Canada

Abstract. We study the XAI (explainable AI) on the face recognition task, particularly the face verification here. Face verification is a crucial task in recent days and it has been deployed to plenty of applications, such as access control, surveillance, and automatic personal log-on for mobile devices. With the increasing amount of data, deep convolutional neural networks can achieve very high accuracy for the face verification task. Beyond exceptional performances, deep face verification models need more interpretability so that we can trust the results they generate. In this paper, we propose a novel similarity metric, called explainable cosine ($xCos$), that comes with a learnable module that can be plugged into most of the verification models to provide meaningful explanations. With the help of $xCos$, we can see which parts of the 2 input faces are similar, where the model pays its attention to, and how the local similarities are weighted to form the output $xCos$ score. We demonstrate the effectiveness of our proposed method on LFW and various competitive benchmarks, resulting in not only providing novel and desiring model interpretability for face verification but also ensuring the accuracy as plugging into existing face recognition models.

Keywords: explainable AI, XAI, interpretable AI, face verification

1 Introduction

Recent years have witnessed rapid development in the area of deep learning and it has been applied to many computer vision tasks, such as image classification [2,13], object detection [25], semantic segmentation [29], and face verification [30], *etc.* In spite of the astonishing success of convolutional neural networks (CNNs), computer vision communities still lack an effective method to understand the working mechanism of deep learning models due to their inborn non-linear structures and complicated decision-making process (so-called "black box"). Moreover, when it comes to security applications (e.g., face verification for mobile screen lock), the false positive results for unknown reasons by deep learning models could lead to serious security and privacy issues. The aforementioned problems will make users insecure about deep learning based systems and also make developers hard to improve them. Therefore, it is crucial to increase transparency during the decision-making process for deep learning models.

A rising field to address this issue is called explainable AI (XAI) [10], which attempts to empower the researcher to understand the decision-making process of neural nets via explainable features or decision processes. With the support of explainable AI, we can understand and trust the neural networks' prediction more. In this work, we focus on building a more explainable face verification framework with our proposed novel $xCos$ module. With $xCos$, we can exactly know how the model determines the similarity score via examining the local similarity map and the attention map.

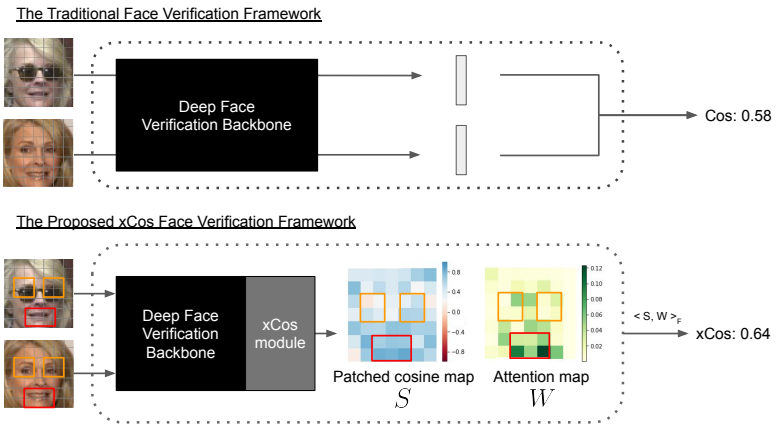


Fig. 1: **Example of $xCos$ framework.** Traditional face verification models provide no spatial clues about why the two images are the same identity or not. The models equipped with our proposed $xCos$ module allow the user to visualize the similarity map between two people for each part of a face and our model cares to produce the final similarity score, $xCos$ (explainable cosine). The $< S, W >_F$ denotes the Frobenius inner product between S and W. We can see that $xCos$ module can be plugged into any existed deep face verification models and the existed face verification models can be more easily interpreted with our proposed $xCos$.

We begin our work with a pivotal question: How to produce more explainable results? To answer this question, we first investigate the pipeline of current face verification models and then introduce the intuition of the human being's decision-making process for face verification. Next, we formulate our definition of interpretability and design the explainable framework that meets our needs.

State-of-the-art face verification models [8,19] extract deep features of a pair of face images and compute the cosine similarity or the L2-distance of the paired features. Two images are said to be from the same person if the similarity is larger than a threshold value. However, with this standard procedure, we can hardly interpret these high dimensional features with our knowledge. Although

there are some previous works attempting to visualize the results [27,7,5] on the input images with saliency map, these saliency map based visualizations are mostly used for the localization of objects in a single image rather the similarity of two faces. Therefore, our framework provides a new verification branch to calculate similarity maps and discriminative location maps based on the features extracted from two faces. (cf. Figure 4) This way, we can strike a balance between verification accuracy and visual interpretability.

We observe that humans usually decide whether the two face images are from the same identity by comparing their face characteristics. For instance, if two face images are from the same person, then the same parts of the 2 face images should be similar, including the eyes, the nose, etc. Based on this insight, we come up with a novel face verification framework, *xCos*, which behaves closely to our observation.

Illustrated by the observation above, we define the **interpretability** in the face verification that the output similarity metric aims to provide not only the local similarity information but also the spatial attention of the model. Based on our definition of interpretability, we propose a similarity metric, *xCos*, that can be analyzed in an explainable way. As shown in Fig. 1, we can insert our novel *xCos* module into any deep face verification networks and get 2 spatial-interpretable maps. Here we plug the proposed *xCos* module into ArcFace [8] and CosFace [31].The first map displays the cosine similarity of each grid feature pair, and the second one shows what the model pays attention to. With the two visualized maps, we can directly understand which grid feature pair is more similar and important for the decision-making process.

The main contributions of this work are as follows:

- We address the interpretability issue in the face verification task from the perspective of local similarity and model attention, and propose a novel explainable metric, *xCos* **(explainable cosine)**.
- We introduce the using of the convolution feature as the face representation, which preserves location information while remaining good verification performances.
- The proposed *xCos* module can be plugged into various face verification models, such as ArcFace and CosFace (cf. Table 1).

2 Related Work

2.1 Face Verification

Face verification task has come a long way these years. GaussianFace [21] first proposed Discriminative Gaussian Process Latent Variable Model that surpasses human-level face verification accuracy. Due to the emerging of deep learning, DeepFace [24], SphereFace [19], CosFace [31], and ArcFace[8] achieve great performances on face verification task with different loss function designs and deeper backbone architectures. Recently, [12] optimizes the face verification model on contrastive face characteristics. However, this work still does not meet our needs for explainability.

2.2 Explainable AI

With the rising demand for explainable AI, there have been plenty of works related to this topic in recent years. Visualizations of convolution neural networks using saliency maps are the main techniques used in [38,27,7]. However, as we have mentioned, saliency map based visualization is more suitable for the localization of objects. Knowledge Distillation [15] is another path to interpretable machine learning because we can transfer the learned knowledge from the teacher model to the student model. [20] realizes this idea by means of distilling Deep Neural Networks into decision trees. In our work, the current face verification model functions as the teacher model to supervise the $xCos$ module with the cosine similarity values it produces.

As for BagNet [3], the authors combine the bag-of-local-feature concept with convolution neural network models and perform well on the ImageNet by classifying images based on the occurrences of patched local features without considering their spatial ordering. To some extent, the authors provide a straightforward way to quantitatively analyze how exactly each patch of the image impacts on the classification results, with the constraint on local representations.

In [14], the authors mentioned that there are many challenges to provide AI explanations, such as the lack of one satisfying formal definition for effective human-to-human explanations. However, [26] outlines four desirable characteristics for explanation methods, including interpretable, local fidelity, model-agnostic, and global perspective, and our work manages to satisfy these criteria by constructing interpretable maps with local information in the field of face verification.

The most related work is [35]. In this work, the authors applied the spatial activation diversity loss and the feature activation diversity loss to learn more structured face representations and force the interpretable representations to be discriminative. Their definition of interpretability of the face representation is that each dimension of the representation can represent a face structure or a face part. Nevertheless, their method mainly focuses on how to visualize the individual identification result given a single image, while it cannot quantitatively tell people which filters or responses are worthy of notice given a pair of images in the face verification task. Compared to [35], our model can provide both the quantitative and qualitative reasons that explain why 2 face images are from the same person or not. If the 2 face images are viewed as the same person by the model, our proposed method can clearly show which patches on the face are more representative than others via providing local similarity values and the attention weights.

3 Proposed Approach

First, we define the ideal properties of $xCos$ metric. Second, we propose 3 possible $xCos$ formulas.

3.1 Ideal *xCos* Metric

In comparison with the traditional cosine similarity for face verification, the ideal *xCos* (explainable cosine) metric should not only output a single similarity score but also produce **spatial explanations** on it. That is, *xCos* should enable humans to understand why the 2 face images are from the same person (or not) by showing the composition of *xCos* value in terms of **components that make sense to humans (e.g., their noses look similar)**. Besides this explainable property, face verification models using *xCos* as the metric should remain good performance so that it could be used to replace cosine metric in real scenarios.

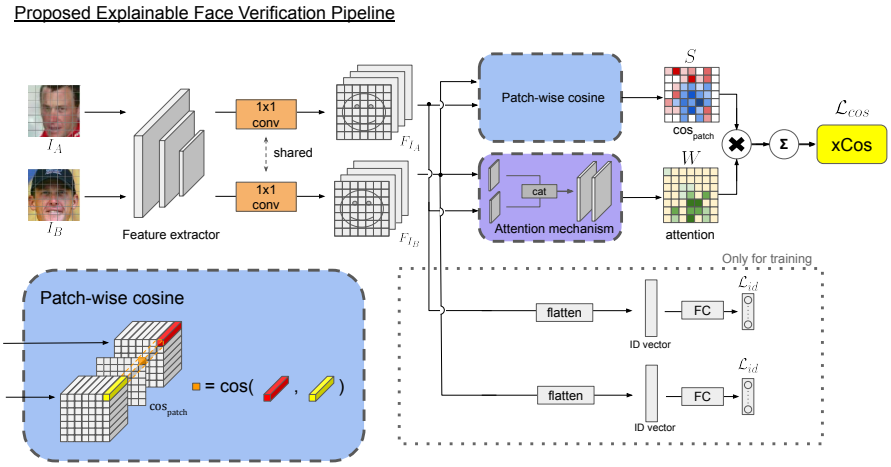


Fig. 2: **Proposed Architecture.** Our proposed architecture contains one modified CNN backbone and two branches for *xCos* and identification. The CNN backbone is responsible for extracting face feature for each identity. To preserve the position information of each feature point, the final flatten and fully-connected layers of the backbone (e.g., ArcFace or CosFace) are replaced with an 1 by 1 convolution. On the *xCos* branch, we compute one patched cosine map **S** (i.e. cos_{patch} in the figure) by measuring the cosine similarity element-wisely between the two feature maps of compared images. Meanwhile, an attention weight map **W** is generated by our attention mechanism based on the two feature maps. The patched cosine map **S** is then weighted summed according to the attention weight map **W** to get the final *xCos* similarity value. The *xCos* is supervised under the cosine similarity generated by another face recognition model like ArcFace. The identification branch flattens the extracted feature and passes it into another fully connected layer for ID prediction. The loss L_{id} is used to stabilize the training process and can be any common face recognition loss like the one in ArcFace.

3.2 *xCos* Candidates.

We propose 3 candidates for the $xCos$ metric. Given an face image I and a CNN feature extractor C, we can get the grid features F_I of size (h_F, w_F, c_F):

$$F_I = C(I) \in \mathbf{R}^{h_F, w_F, c_F} \tag{1}$$

In order to concisely demonstrate the core idea of each $xCos$ candidate, we first formulate $xCos$ metric as a general function of F_A, F_B, and $\mathbf{W}$:

$$xCos(F_{I_A}, F_{I_B}, \mathbf{W}) = \sum_{i=1}^{h_F} \sum_{j=1}^{w_F} w_{i,j} * cos(F_{I_A}^{i,j}, F_{I_B}^{i,j}) \tag{2}$$

where $F_I^{i,j}$ is the feature of position (i, j), $\mathbf{W} \in \mathbf{R}^{h_F, w_F}$ is the matrix of the attention weights, $w_{i,j} \in \mathbf{W}$ is the attention weight for the grid of position (i, j), and I_A, I_B refer to2 different face images A and B.

Patched *xCos* This $xCos$ candidate simply realizes the idea that every pair of the grids on faces should be similar if the 2 faces are from the same person. Thus, we let **unit attention U**:

$$\mathbf{U} = \frac{1}{h_F * w_F} \mathbf{J}_{h_F, w_F} \tag{3}$$

where $\mathbf{J}_{h_F, w_F}$ is the all-ones matrix of size (h_F, w_F), and the patched $xCos$ can be calculated in this way:

$$xCos_{patched} = xCos(F_{I_A}, F_{I_B}, \mathbf{U}) \tag{4}$$

Correlated-patched *xCos* Inspired by the idea that each patch on the face may contribute different weights to $xCos$, we can change the unit attention to correlated-attention $\mathbf{P}$, with the features f_{I_C}, f_{I_D} extracted from any other deep face verification models:

$$\mathbf{P} \in \mathbf{R}^{h_F, w_F} \tag{5}$$

where the element $p^{i,j}$ in $\mathbf{P}$ is the Pearson-correlation of the set

$$\left\{ (cos(F_{I_C}^{i,j}, F_{I_D}^{i,j}), cos(f_{I_C}, f_{I_D})) \right\} \tag{6}$$

over all the image pairs (I_C, I_D) in the training dataset (C, D are arbitrary identity indices in the dataset). As a result, we get the formula of correlated-patched $xCos$:

$$xCos_{corr} = xCos(F_{I_A}, F_{I_B}, \mathbf{P}) \tag{7}$$

Attention-patched *xCos* We propose another kind of $xCos$ metric which learns the attention $\mathbf{L}$, i.e.

$$\mathbf{L} = M_{attention}(F_{I_A}, F_{I_B}) \in \mathbf{R}^{h_F, w_F} \tag{8}$$

, where $M_{attention}$ is a CNN module. The learned attention $\mathbf{L}$ is supervised by the cosine similarity of f_{I_A} and f_{I_B} that is generated with any other target face verification model. With this module, we can formulate the attention-patched $xCos$ as follows:

$$xCos_{attention} = xCos(F_{I_A}, F_{I_B}, \mathbf{L}) \tag{9}$$

3.3 Network Architecture

For current face verification models, the main obstacle to interpretability is that the fully connected layer removes spatial information, so it is hard for humans to understand how the convolution features before the FC are combined in a human sense. To address this problem, we propose a 2-streamed network with a slightly different backbone and one plug-in $xCos$ module, as described in the following sections:

Backbone Modification We target on learning the face representation which is not only discriminative but also spatially informative. To achieve this goal, we choose a current face recognition backbone, called $f(C'(I))$, delete its fully-connected part $f(x)$ for face feature extraction, and then append the 1 by 1 convolutional layer C_{1x1} after the original convolutional layers $C'(I)$, i.e. the $C(I)$ in the previous subsection is equal to $C_{1x1}(C'(I))$. The resulting feature F_I plays two roles:

1. When it is flattened, F_I can represent the entire face.
2. When it is viewed as the grid features, we can make use of the local information of every grid $F_I^{i,j}$ so as to provide $xCos$ with regard to the inputs.

Patched Cosine Calculation Given a pair of face convolutional features, F_{I_A}, F_{I_B}, each of size (h_F, w_F, c_F), we can compute the cosine similarity in each grid pair and generate a patched cosine map $\mathbf{S} \in \mathbf{R}^{h_F, w_F}$. Each element of this map $\mathbf{S}$ represents the similarity of each corresponding grid. With this cosine map $\mathbf{S}$, we can inspect which parts of the face images are considered similar by the model.

***xCos* Calculation** Given 2 convolutional feature maps, F_{I_A}, F_{I_B}, we can first compute the patched cosine map $\mathbf{S}$ and generate the attention map $\mathbf{W} \in \{\mathbf{U}, \mathbf{P}, \mathbf{L}\}$. Then, we perform the Frobenius inner product $< \mathbf{S}, \mathbf{W} >_F$ to get the value of $xCos$. Specifically, we sum over the result of element-wise multiplication on the attention map $\mathbf{W}$ and the patched cosine map $\mathbf{S}$, and then obtain the $xCos$ value defined in 3.2)

Attention on Patched Cosine Map Given 2 face images, I_A and I_B, we can compute their cosine similarity with any face verification model, i.e. let $c' = cos(f_{I_A}, f_{I_B})$. Then, with 2 feature maps F_{I_A}, F_{I_B} and the supervising cosine scores c', we can learn the attention module $M_{attention}$.

Inside $M_{attention}(F_{I_A}, F_{I_B})$, we use convolution layers to perform dimensionality reduction for the two face features F_{I_A}, F_{I_B}, and then fuse the 2 deduced features by concatenation along the channel dimension. Next, we feed the fused feature into two convolution layers, normalize the output feature map, and get the attention map $\mathbf{L} \in \mathbf{R}^{h,w}$.

After getting $\mathbf{L}$, we apply element-wise multiplication on the attention map $\mathbf{L}$ and the patched cosine map $\mathbf{S}$, sum the result to get the $xCos_{attention}$ with value c, and calculate the L2-Loss of c and c' so that $\mathbf{L}$ is trainable.

Multitasking for Two-branched Training As shown in Fig. 2, the identification branch is trained with the flattened 1 by 1 convolution feature F_{I_A}, F_{I_B}, and the loss function for the identification task, $\mathcal{L}_{id}$, can be the one from ArcFace [8], CosFace [31], or any other deep face recognition model. In addition, the $xCos$ branch performs the task of regressing the $xCos$ value c to the cosine value c' calculated from the target model, and $\mathcal{L}_{cos}$, the loss of regressing $xCos$ to cosine value, is L2-Loss, i.e.

$$\mathcal{L}_{cos} = \frac{1}{N'} \sum_{n=1}^{N'} (c_n - c'_n)^2 \tag{10}$$

where the N' refers to the number of image pairs in each batch, and n denotes the n-th pair in one batch.

4 Experiments

4.1 Implementation Details

Datasets We use publicly available MS1M-ArcFace [11] [8] as training data, and use LFW [16], AgeDB-30 [23] [9], CFP [28], CALFW [37], VGG2-FP [4], AR database [22], and YTF [33] as our testing datasets.

Date Preprocessing We follow the data preprocessing pipeline that is similar to [8] [31] [19]. We first use MTCNN [36] to detect faces. Then we apply similarity transform with 5 facial landmark points on each face to get aligned images. Next, we randomly horizontal-flip the face image, resize it into 112 x 112 pixels, and follow the convention [32] [31] to normalize each pixel (in [0, 255] for each channel) in the RGB image by subtracting 127.5 then dividing by 128.

CNN Setup We mainly apply the same backbone as the one in ArcFace [8]. However, we replace the last fully connected layer and the flatten layer before it with 1 by 1 conv layer (input channel size = 512; output channel size = 32), and called the output of it as grid features F_I. The size of a grid feature F_I for an (112, 112, 3) RGB image I will be (7, 7, 32). When training the face identification branch, we flatten the grid feature F_I into an 1-D vector with dimension 1568.

***xCos* Module Setup** Given 2 grid features, F_{I_A}, F_{I_B}, of size (7, 7, 32), our goal is to produce one attention map $\mathbf{L}$ and one patched cosine map $\mathbf{S}$. To get the attention map $\mathbf{L}$, we first use a convolution layer with kernel size = 3 and padding = 1 to perform dimension reduction on F_I with the output channel dimension = 16. The 2 reduced convolution features of size (7, 7, 16) are then concatenated into a new fused grid feature of size (7, 7, 32). Second, we feed the fused grid feature into another 2 convolution layers to get the output $\mathbf{L}$, of size (7, 7), as the attention map $\mathbf{W}$ for $xCos_{attention}$. Finally, we normalize the attention map with a softmax function in order to make sure the sum of all the 49 grid attention weights is 1. The patch-cosine map $\mathbf{S} \in \mathbf{R}^{7,7}$ is obtained by computing the grid-wise cosine similarity between any grid pair features from F_A and F_B. Once finishing computing the attention map $\mathbf{L}$ and the patch-cosine map $\mathbf{S}$, the model will calculate the $xCos$ output value by performing the Frobenius inner product between $\mathbf{L}$ and $\mathbf{S}$. All models are trained with learning rate 1e-3 which will be divided by 10 after 12, 15, 18 epochs.

4.2 Quantitative Results

Table 1: **Face verification accuracy on LFW dataset.** Compared to other face verification models, the proposed $xCos$ module significantly improves explainability with a minimal drop of performances.

Method	Accuracy
Human performance[1]	97.53%
GaussianFace [21] (non-Deep)	97.79%
CosFace [31]	99.33%
ArcFace [8]	99.83%
attention-patched $xCos$ (Ours, CosFace)	99.67 %
attention-patched $xCos$ (Ours, ArcFace)	99.36 %

Face Verification Performance To demonstrate the effectiveness of our proposed method, we show the performance of $xCos$ in the Table 1. From Table 1, we can observe that the $xCos$ module not only provides explainability with the trade-off of little drop of accuracy on the LFW dataset but also produces promising performance gain over the human performance and some earlier non-deep face verification models like GaussianFace [21].

Table 2: **Ablation Studies.** The patch., corr., atten. refer to patched $xCos$, correlated-patched $xCos$, and attention-patched $xCos$ mentioned in Section 3.2, respectively; ArcFace [8] and CosFace [31] represent common backbone models used in face identification. From this table, we can observe that (1) $xCos$ brings explainability without degrading the performance; (2) The plug-in $xCos$ attention module can perform well in different face verification backbones. Note (*): We train the baseline with the same training setting for $xCos$ and turn off the testing time augmentation to have a fair comparison.

BackBone	ArcFace [8]				CosFace [31]			
Methods	baseline*	patch.	corr.	atten.	baseline*	patch.	corr.	atten.
Feature Layer	FC	1x1			FC	1x1		
Attention Type	-	**U**	**P**	**L**	-	**U**	**P**	**L**
LFW [16] (%)	99.45	**99.47**	99.46	99.36	99.28	99.63	99.60	**99.67**
YTF [33] (%)	95.06	96.68	96.68	**96.68**	96.24	96.78	96.76	**96.80**
VGG2-FP [4] (%)	89.94	**95.68**	92.34	90.06	91.86	**93.66**	93.66	93.38
AgeDB-30 [23] [9] (%)	91.60	95.68	**95.73**	94.87	89.60	95.20	95.28	**95.93**
CALFW [37] (%)	92.55	94.82	94.77	**94.83**	91.30	94.83	94.77	**95.10**
CFP-FF [28] (%)	87.56	**92.66**	92.46	90.71	98.80	99.44	99.44	**99.44**
CFP-FP [28] (%)	85.65	**88.22**	88.18	86.76	90.61	93.07	93.16	**93.54**

Ablation Studies As shown in Table 2, we use the face recognition models without the backbone modification as baseline, and then observe the effectiveness of $xCos$ via applying different attention weights $\mathbf{W} \in \{\mathbf{U}, \mathbf{P}, \mathbf{L}\}$. In pursue of a fair comparison, we train the baseline with the same setting of $xCos$ except the feature extraction layer, and turn off the testing time augmentation for the baseline because it will apply averaging operation over features, which leads to the mix of spatial information for our convolutional features. Among most of the testing datasets, attention-patched $xCos$ achieves the best performances, which suggests that our attention module takes effect. However, in few datasets like VGG2-FP, it seems that the patched $xCos$ and the correlated-patched $xCos$ get a better result than the attention-patched $xCos$. We hypothesize that our proposed models, which are trained on aligned face images, do not perform as expected due to the huge pose difference and pose variations in these 2 datasets. Also, both baseline model and our purposed $xCos$ models have obvious performance drop between the pose-varying datasets and datasets without pose variations. Therefore, we believe this is a general issue for all the face verification models which do not handle pose variations by design. We discuss how to optimize both the explainability and the model performance, and provide some possible solutions to this advanced issue in Section 5.2.

Computational Cost Although in our system, there are some additional costs to calculate the pairwise cosine similarity and attention map, the feature extraction process is still the computational bottleneck. When ignoring all disk reading and writing time and running on an i7-3770 CPU with a 1080ti GPU,

the inference for a pair of faces takes 6.1 ms and 6.7 ms for the original model and our $xCos$ model, respectively. Considering the explainability gain over the original model, this efficiency drop is negligible.

4.3 Qualitative Results

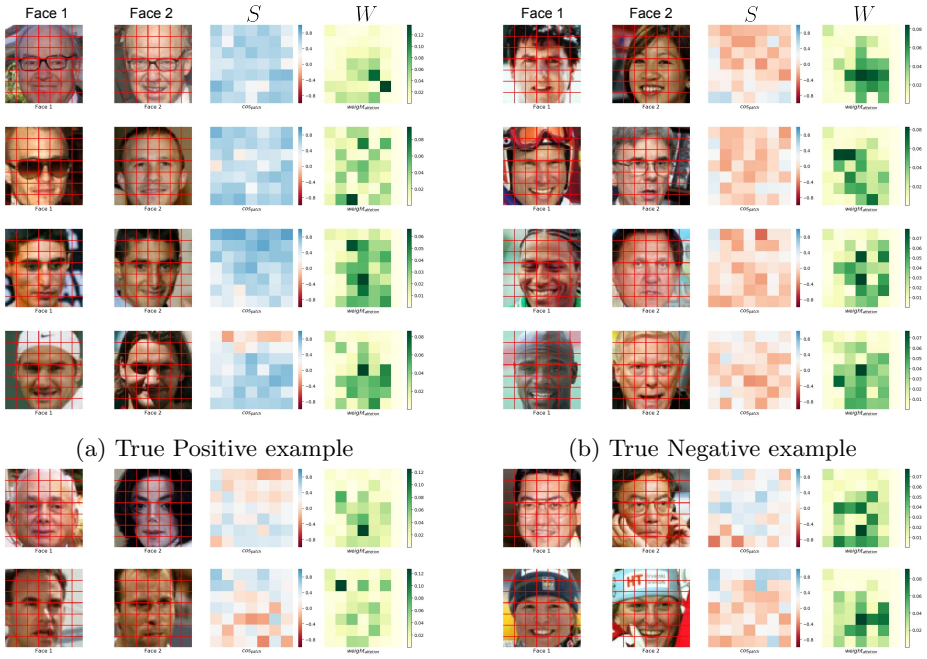
Visualizations of *xCos* As shown in Fig 3, there are two interesting phenomena worth mentioning:

1. The area around central columns is of great interest to the $xCos$ model. By observing the weight distributions on the attention maps, we can conclude that the central convolution feature is influential for the model to verify the identity.
2. The area near mouths and chins is of greater importance than the upper parts of faces. People may wear hats, change hairstyles, or become bald as growing older, so the model pays less attention to the area on the top of faces. On the contrary, the variations of the shape of mouths and chins are constrained to the color of lips or facial expression like smiling. For instance, the fourth row in Fig 3a and the second row in Fig 3d both contain faces with hats while the model pays less attention to those facial parts which occupied with hats.

Table 3: **Differences between $xCos$ and saliency methods. S** and **W** are the interpretable maps defined in the paper. Note (*): saliency methods are methods whose outputs are 2 individual heat maps for 1 verification pair.

	local importance	local similarity	verification metric
$xCos$	V	V	$xCos$ + **S** + **W**
saliency methods*	V	X	cosine value

Comparison with saliency methods. Saliency methods like Grad-CAM[27] can provide attention-like heat maps. However, it is mainly for identification tasks but not verification tasks. Fig.4 shows four qualitative results of Grad-CAM. It is hard for us to interpret why the 2 face images are verified as the same person or not. Several previous works have dealt with finding the pixels that contribute the most. However, those works, even the most relevant one [35], (1) provide no **local similarity** information in their saliency maps and (2) hardly focus on the face verification task. (See Table 3.) Contrarily, $xCos$ not only highlights important regions but tells users which grids are (dis)similar. Revealing local similarity can help users debug the verification system, for example, by showing the local dissimilarities caused by hand occlusion (e.g., the first row of 3d).



(a) True Positive example

(b) True Negative example

(c) False Positive example

(d) False Negative example

Fig. 3: **Qualitative Results.** The third and the fourth columns of each example represent the patched cosine similarity map and the attention weights map. The blue color in the patched cosine similarity map indicates the corresponding grids are similar. The model generally shows interest in the area around the mouth among all examples. In the first row of (a), it is clear that the nasolabial folds of the target person attract the model more. And in the third row of (b), our model pays attention to the different shapes of the 2 noses, rather than the different hairstyles between the two faces. In the first row of (c), our model mismatches the mouths while ignoring the different shapes of noses. In the first row of (d), it is clear that the appearance of hands distracts the model. With the visualizations, we can alarm users to put their hands away in order to avoid verification failure. With the aid of our proposed cosine similarity map **S** and attention map **W**, we can easily interpret the visualized results in the confusion matrix. Thus, users can be more confident to know when models go right (or wrong), and *xCos* can play a role in helping optimize the design of face verification model.

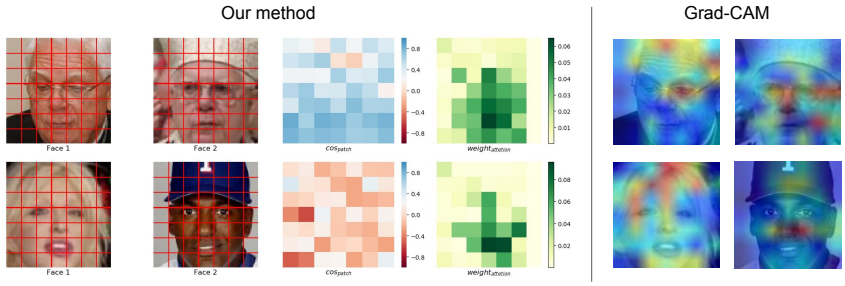


Fig. 4: **Comparison with saliency methods** (1)The first row shows one true positive pair. It is interpretable with the proposed *xCos* that the forehead area is not similar and not important for the verification result, while it is hard for a human to interpret how the 2 individual heat maps around the forehead contribute to the result by applying saliency methods like Grad-CAM[27] on the ArcFace[8] model. (2) The second row is one true negative pair. The saliency method just puts the most significant pixels side by side while our method reveals that the dissimilarity caused by the cap is not important for *xCos* model. Both pairs are from the LFW[16] dataset.

5 Discussions

5.1 Additional Robustness to Occlusion

Since our method considers patch features independently and the attention module will decide where to focus on, our method should be more robust than the original model when faces are partially occluded. AR face is a natural occlusion face database with around 4K faces of 126 subjects and therefore a good test set for this experiment. We select the faces with scarfs or glasses and exclude those which can not be detected by MTCNN. After the selection, 1488 images are used to randomly generate 6000 positive pairs and 6000 negative pairs. As shown in Table 5(a), our proposed methods outperform the original ArcFace model even without the attention module. Besides, we also use the free-form masks in [6] to create synthetic CASIA[34] and LFW[16] occlusion datasets for fine-tuning and testing, respectively. There is one mask that occupies about x% out of the total area for each image in the training or testing dataset (See Table 5(c) for examples.) From Table 5(b), it can be concluded that the proposed *xCos* method has less performance drop than the original face verification model.

5.2 How to Adapt *xCos* From Frontal Images to Profile Ones

In this work, we open a new avenue for explainability for face recognition. As the pilot study for the emerging problem, we have to take 2 steps to make our research more convincing: (1) verify that the existence of our explainable module

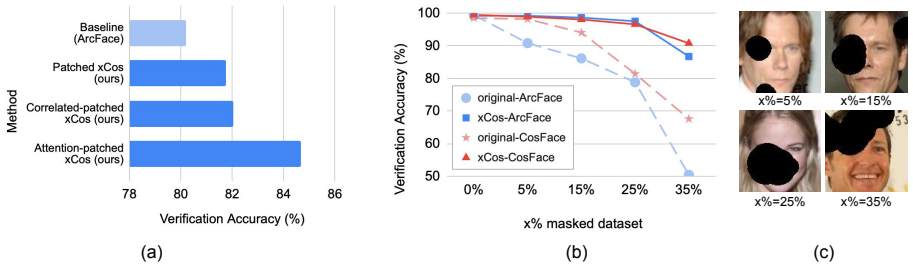


Fig. 5: **(a)Face verification accuracy on the occlusion subset of AR database[22].** The proposed *xCos* method provides not only explainability but also additional robustness to partially occluded faces. **(b)Face verification accuracy on the x% masked LFW dataset.** Free-form masks[6] is applied on the images of LFW dataset. **(c)Examples of x% synthetic occlusion dataset.** The proposed *xCos* has less performance drop than common face recognition models, including ArcFace and CosFace.

does not degrade the overall verification performance on the ideal test setting (e.g. test on the aligned LFW dataset) for SoTA face recognition models; (2) find out how to extend its usage to other rigorous experiment settings, like face images with large pose variations or extreme illuminations. We are optimistic to see that our work, which realizes the main idea in stage (1), is going to inspire more future research on face applications with critical conditions.

There are plenty of papers embarked on tackling various challenging conditions, including low light/resolution settings or large pose variations, cross-age, etc. Following our successful attempt in the first stage, we believe the research communities can extend the adaptability of *xCos* module for many other face recognition problems. For example, some previous works have explored the possibility of recovering the canonical view of face images from non-frontal images using SAE[18], CNN[39] and GAN[17] models, so we can extend the usage of *xCos* to the cross-pose scenario by performing these preprocessing method first.

6 Conclusions

We propose a novel metric for face verification task, called *xCos* (explainable cosine). With this metric, we can not only view it as one similarity score to quantitatively verify 2 face images but also decompose it into patched cosines and one attention map so that humans can intuitively understand which parts of the face are similar and how important each grid on the face is. We believe that *xCos* can be used to inspect the face verification model behavior and bridge the gap between the model complexity and humans' understanding in an explainable way.

7 Acknowledgement

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 109-2634-F-002-032. We benefit from the NVIDIA grants and the DGX-1 AI Supercomputer and are also grateful to the National Center for High-performance Computing.

References

1. Attribute and Simile Classifiers for Face Verification. In: IEEE International Conference on Computer Vision (ICCV) (Oct 2009)
2. Bartlett, P.L., Pereira, F.C.N., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.): Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States (2012), http://papers.nips.cc/book/advances-in-neural-information-processing-systems-25-2012
3. Brendel, W., Bethge, M.: Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. International Conference on Learning Representations (2019), https://openreview.net/pdf?id=SkfMWhAqYQ
4. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: A dataset for recognising faces across pose and age. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). pp. 67–74. IEEE (2018)
5. Castañón, G., Byrne, J.: Visualizing and quantifying discriminative features for face recognition. 2018 13th IEEE International Conference on Automatic Face —& Gesture Recognition (FG 2018) pp. 16–23 (2018)
6. Chang, Y.L., Liu, Z.Y., Lee, K.Y., Hsu, W.: Free-form video inpainting with 3d gated convolution and temporal patchgan. In Proceedings of the International Conference on Computer Vision (ICCV) (2019)
7. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 839–847 (March 2018). https://doi.org/10.1109/WACV.2018.00097
8. Deng, J., Guo, J., Niannan, X., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR (2019)
9. Deng, J., Zhou, Y., Zafeiriou, S.P.: Marginal loss for deep face recognition. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) pp. 2006–2014 (2017)
10. Gunning, D.: Explainable artificial intelligence (xai). Defense Advanced Research Projects Agency (DARPA), nd Web **2** (2017)
11. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: ECCV (2016)
12. Han, C., Shan, S., Kan, M., Wu, S., Chen, X.: Face recognition with contrastive convolution. In: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IX. pp. 120–135 (2018). https://doi.org/10.1007/978-3-030-01240-3_8, https://doi.org/10.1007/978-3-030-01240-3_8
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90, https://doi.org/10.1109/CVPR.2016.90
14. Hind, M., Wei, D., Campbell, M., Codella, N.C.F., Dhurandhar, A., Mojsilović, A., Natesan Ramamurthy, K., Varshney, K.R.: Ted: Teaching ai to explain its decisions. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. pp. 123–129. AIES '19, ACM, New York, NY, USA (2019). https://doi.org/10.1145/3306618.3314273, http://doi.acm.org/10.1145/3306618.3314273

15. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NIPS Deep Learning and Representation Learning Workshop (2015), http://arxiv.org/abs/1503.02531
16. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst (October 2007)
17. Huang, R., Zhang, S., Li, T., He, R.: Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
18. Kan, M., Shan, S., Chang, H., Chen, X.: Stacked progressive auto-encoders (spae) for face recognition across poses. 2014 IEEE Conference on Computer Vision and Pattern Recognition pp. 1883–1890 (2014)
19. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
20. Liu, X., Wang, X., Matwin, S.: Improving the interpretability of deep neural networks with knowledge distillation. 2018 IEEE International Conference on Data Mining Workshops (ICDMW) pp. 905–912 (2018)
21. Lu, C., Tang, X.: Surpassing human-level face verification performance on lfw with gaussian face. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. pp. 3811–3819. AAAI'15, AAAI Press (2015), http://dl.acm.org/citation.cfm?id=2888116.2888245
22. M. Martinez, A., Benavente, R.: The ar face database. Tech. Rep. 24 CVC Technical Report (01 1998)
23. Moschoglou, S., Papaioannou, A., Sagonas, C., Deng, J., Kotsia, I., Zafeiriou, S.: Agedb: The first manually collected, in-the-wild age database. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1997–2005 (July 2017). https://doi.org/10.1109/CVPRW.2017.250
24. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: BMVC (2015)
25. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017). https://doi.org/10.1109/TPAMI.2016.2577031, https://doi.org/10.1109/TPAMI.2016.2577031
26. Ribeiro, M.T., Singh, S., Guestrin, C.: "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. pp. 1135–1144. KDD '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939778, http://doi.acm.org/10.1145/2939672.2939778
27. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 618–626 (Oct 2017). https://doi.org/10.1109/ICCV.2017.74
28. Sengupta, S., Chen, J., Castillo, C., Patel, V.M., Chellappa, R., Jacobs, D.W.: Frontal to profile face verification in the wild. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1–9 (March 2016). https://doi.org/10.1109/WACV.2016.7477558
29. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (2017). https://doi.org/10.1109/TPAMI.2016.2572683, https://doi.org/10.1109/TPAMI.2016.2572683

30. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. pp. 1988–1996 (2014), http://papers.nips.cc/paper/5416-deep-learning-face-representation-by-joint-identification-verification
31. Wang, H.J., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 5265–5274 (2018)
32. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: ECCV (2016)
33. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. CVPR 2011 pp. 529–534 (2011)
34. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. ArXiv **abs/1411.7923** (2014)
35. Yin*, B., Tran*, L., Li, H., Shen, X., Liu, X.: Towards interpretable face recognition. In: In Proceeding of International Conference on Computer Vision. Seoul, South Korea (October 2019)
36. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters **23**(10), 1499–1503 (Oct 2016). https://doi.org/10.1109/LSP.2016.2603342
37. Zheng, T., Deng, W., Hu, J.: Cross-age LFW: A database for studying cross-age face recognition in unconstrained environments. CoRR **abs/1708.08197** (2017), http://arxiv.org/abs/1708.08197
38. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning Deep Features for Discriminative Localization. arXiv e-prints arXiv:1512.04150 (Dec 2015)
39. Zhu, Z., Luo, P., Wang, X., Tang, X.: Deep learning identity-preserving face space. pp. 113–120 (12 2013). https://doi.org/10.1109/ICCV.2013.21