KeyPoint Relative Position Embedding for Face Recognition

Supplementary Material

1. Training Details

Training code will be released for reproducibility. Our experiments were conducted using the PyTorch deep learning framework. Detailed information pertaining to the training parameters, configurations, and specifics can be referred to in Tab. 6. We employed the Vision Transformer (ViT) model architectures as implemented in the InsightFace GitHub repository, ensuring a well-established and tested model foundation. When measuring the throughput of our KeyPoint Relative Position Embedding (KPRPE), we utilized an NVIDIA RTX3090 GPU.

Table 6. Details for training face recognition models with or without KPRPE.

	Ablation Experiments	**Large Scale Experiments**
Backbone	ViT Small	ViT Large
LR	0.001	0.0001
Batch Size	512	1024
Epoch	34	36
Momentum	0.9	
Weight Decay	0.05	
Scheduler	Cosine	
Optimizer	AdamW	
Warmup	3	
AdaFace Loss Margin	0.4	
AdaFace Loss h	0.333	
Augmentation	Flip, Brightness, Contrast, Scaling, Translation, RandAug [8](magnitude:14/31), Blur, Cutout, Rotation (20°)	
PartialFC	None	sampling rate 0.6
RepeatedAug Prob	0.5	0.1

2. Supplementary Performance Analysis

2.1. Performance Across Various Loss Functions

In our extensive evaluation, we have employed three popular loss functions: AdaFace [30], CosFace [69], and ArcFace [11], to train the Vision Transformer (ViT) in combination with our proposed KeyPoint Relative Position Embedding (KPRPE). As demonstrated by the results in Tab. 7 rows 3-6, our method exhibits consistent performance improvements on lower quality datasets across all three loss functions when compared to the standalone ViT. This signifies the versatility of KPRPE in synergizing with a variety of loss functions to enhance the robustness of face recognition models to less-than-optimal image quality.

Table 7. SoTA comparison on low-quality and high-quality datasets. IJB-C [72] reports TAR@FAR=0.01%

Method	Backbone	Train Data	Low Quality Dataset				High Quality Dataset		
			TinyFace [7]		IJB-S [29]		AgeDB	CFPFP	IJB-C
			Rank-1	Rank-5	Rank-1	Rank-5	Verification Accuracy		0.01%
AdaFace [30]	**ViT**	WebFace4M [91]	74.81	77.58	71.90	77.09	97.48	98.94	97.14
AdaFace [30]	**ViT+KPRPE**	WebFace4M [91]	**75.80**	78.49	72.78	78.20	**97.67**	99.01	97.13
ArcFace [11]	**ViT+KPRPE**	WebFace4M [91]	75.62	**78.57**	**73.04**	**78.62**	97.57	**99.06**	**97.21**
CosFace [69]	**ViT+KPRPE**	WebFace4M [91]	75.48	78.30	72.22	77.67	97.45	98.94	96.98

2.2. Performance with Different Number of Keypoints

We include the impact of the number of keypoints in KP-RPE. We initiated the analysis with 5 keypoints, the maximum available in RetinaFace. And gradually reduce the number of points.

Number of Keypoints	TinyFace Rank1	TinyFace 5	AgeDB	CFPFP
5	**69.88**	**74.25**	**95.92**	96.60
4	69.58	73.63	95.65	96.57
3	69.66	73.95	95.77	**96.80**
2	69.26	73.42	95.73	95.97
No Keypoints (Vanilla ViT)	68.24	72.96	95.57	96.11

For datasets characterized by lower image quality like TinyFace, the performance diminishes as the number of keypoints reduce. But it does not diminish compared to not using the keypoints. It could be that the information about the scale and rotation of an image could still be captured by few points as 2 or 3. Interestingly, in high-resolution datasets, the trend is absent and the performance remains relatively consistent regardless of the number of keypoints used. More keypoints can be adopted with other landmark detectors but they are not trained with low quality images in WiderFace as the dataset only provides 5 points.

2.3. Sensitivty to Landmark Error in KPRPE

To test the sensitivity of KPRPE to the landmark prediction error, we take the prediction of the landmark predictor and perturb it by the following equation,

$$\mathbf{L}_{pert} = \mathbf{L} + \alpha\mathbf{L}. \tag{12}$$

α is a parameter that changes the level of noise in the prediction. We change α from 0 to 0.1 after noting that 0.1 makes the NME score to be 0.12 which is far worse than the NME score of 0.05 in WiderFace which is a harder dataset. Therefore, $\alpha = 0.1$ is an extreme scenario where all of the inputs have failed to the level which exceeds the average level of failure in WiderFace by two times.

Note that as we add noise into the landmark prediction, the performance goes down, signaling that KPRPE is dependent on the landmark prediction. However, the amount of performance drop within the range of realistic noise level is not too much (about 1.5%). Fig 11 shows the experiment setting in a diagram.

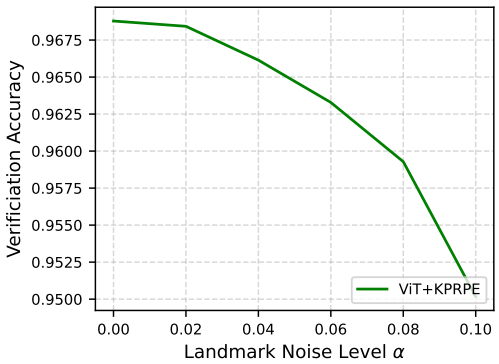


Figure 6. Verification accuracy measured in CFPFP dataset with added noise in landmark predictions.

2.4. Why are noisy keypoints more useful in KP-RPE than in simple alignment?

The short answer is that not all predicted points are noisy in an image while alignment as a result of one or more noisy point impacts all pixels. For a more concrete example, in Fig. 7, we have taken images from WiderFace which contains human-annotated ground truth keypoints and compared them with RetinaFace prediction. Fig. 7 (a) shows a well aligned scenario. (b) and (c) show that when one or two landmarks (red color) deviate from the ground truth (GT), the resulting alignment changes dramatically. For KP-RPE, this is a less severe problem because individual landmarks affect the RPE *indepenently* in the landmark space (0-1). On the other hand, when affine transformation is regressed to align the image to a canonical space, individual landmark error becomes *correlated* and *amplified*.

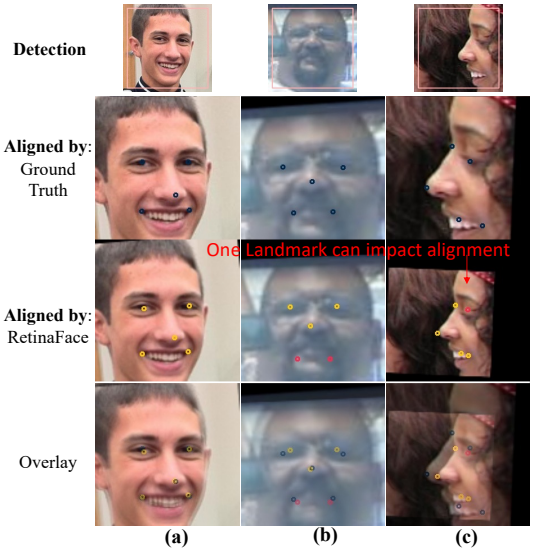


Figure 7. Keypoints in Aligned images. Blue: ground truth keypoints. Yellow/Red: RetinaFace keypoints with less/more than 5% error from GT. Overlay of (b,c) shows how small deviation from one or two points can lead to significant scale, translation change.

3. Training Landmark Detector (MobileNet-RetinaFace)

RetinaFace [10], a single-stage face detector, is built upon Feature Pyramid Network [38] (FPN) and Single Shot MultiBox Detector [42]. It is originally designed for detecting multiple faces using anchor boxes in each location in an image. However, in our case, we assume the presence of one face, and we leverage this constraint to improve the landmark detection performance and efficiency of the model. This assumption is valid if a face detector crops out a face, which is a standard practice in face recognition. With this assumption, we can modify the RetinaFace to predict more accurate landmarks when the input image is cropped. We adopt few training techniques and a faster aggregation technique and name it Differentiable Face Aligner (DFA). The name suggests that with the modifications we propose, the face alignment network is differentiable (unlike RetinaFace because of NMS and CPU based cropping), making it potentially useful for other applications in computer vision.

Training Data Adaptation We adapt the training data WiderFace [78] for our Differentiable Face Aligner (DFA) by cropping out facial images using the ground truth bounding boxes. And we resize the input to be 160x160. This change in data size and distribution allows the model to specialize in localizing landmarks for single faces, ultimately improving its performance.

Aggregation Network The motivation for the aggregation network is to eliminate the Non-Maximum Suppression (NMS) and output a single landmark prediction from multiple anchor boxes. We design a network that takes in the output of FPN and aggregates it to a single prediction. The architecture of the aggregation network consists of MixerMLP [66]. Specifically, let $\mathbf{X}$ be an image, and let $\mathbf{F}_{bbox}$, $\mathbf{F}_{score}$ and $\mathbf{F}_{ldmk}$ be the set of the output of FPN followed by the corresponding multitask head (bounding box, face score and landmark prediction) for each anchor box. For example, when an image is sized 160×160, there are 1050 anchor boxes. Based on these outputs, we predict the weights for fusing the outputs. Specifically,

$$\mathbf{O} = \text{Concat}(\mathbf{F}_{bbox}, \mathbf{F}_{score}, \mathbf{F}_{ldmk}) \in \mathbb{R}^{1050 \times (C_{bbox}+C_{score}+C_{ldmk})} = \mathbb{R}^{1050 \times (4+1+10)}, \tag{13}$$

$$\mathbf{w} = \text{Softmax}(\text{MixerMLP}(\mathbf{O})) \in \mathbb{R}^{1050}, \tag{14}$$

$$\mathbf{L} = \mathbf{w}^T \mathbf{F}_{ldmk}. \tag{15}$$

The final output $\mathbf{L}$ is the weighted average of the landmarks in all anchor boxes. The aggregation network is trained end to end with the rest of the detection model with the smooth L2 Loss [56] between $\mathbf{L}$ and the ground truth landmark $\mathbf{L}^{GT}$.

By incorporating these modifications, we show in Sec. 3.1 that our DFA achieves superior landmark detection performance compared to the RetinaFace while using a more efficient backbone architecture.

Training Details For the training of our Differentiable Face Aligner (DFA), we incorporated specific training settings to optimize the performance. We used an input image size of 160 pixels, with a batch size of 320. Training was conducted for 750 epochs, ensuring that the model had adequate exposure to learn and generalize from the dataset. Training was performed using the WiderFace training dataset, with images cropped using the ground truth bounding boxes and a padding of 0.1.

3.1. Landmark Detection Performance

In this section, we evaluate the performance of our proposed Differentiable Face Aligner (DFA) in terms of landmark detection. We use the Normalized Mean Error (NME) as the metric and evaluate on WiderFace validation set [78] as in RetinaFace [10].

Fig. 8 shows an improvement in NME when using DFA compared to the baseline RetinaFace. The RetinaFace with MobileNet backbone achieves an NME of 0.077, while the one with ResNet50 achieves 0.0553. In contrast, our DFA achieves 0.0518, demonstrating its superiority in landmark detection.

Moreover, the DFA model benefits from the introduction of the aggregation network, which eliminates the need for the NMS stage. The improvement in NME due to the aggregation network is from 0.0527 to 0.0518. This not only simplifies the overall pipeline but also contributes to the enhanced performance of the DFA model in the landmark detection. With a straightforward modification in the training data and an aggregation stage that assumes a single-face image, a lightweight backbone with better performance can be trained.

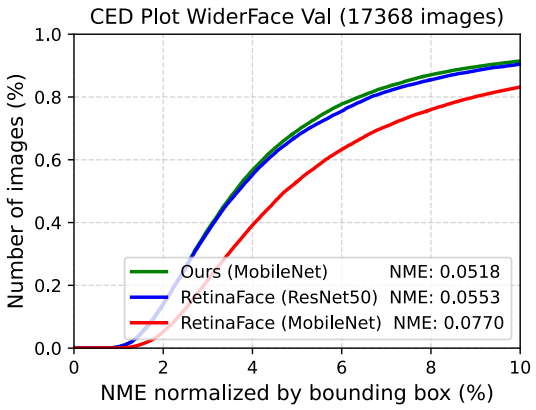


Figure 8. Cumulative Error Distribution curve and the corresponding NME for models evaluated on WiderFace [78] validation set.

4. IJB-S Evaluation Method

IJB-S [29] is a video-based dataset that defines probe and gallery templates according to its predefined video clip of arbitrary length. This naturally implies one must perform feature aggregation (fusion) when frame-level features are predicted. Since the backbone predicts a unit-norm feature vector for one image, the simplest method would be to average all the features within the template. The most popular method is to utilize norm-weighted average, where the features are averaged before normalization [30]. This only works if the norm is a good proxy for the prediction quality. However, in certain cases, depending on various factors such as dataset, learning rate, backbone, optimizer, etc., that go into the training of a model, this may not be the case. Also, in our experience, ViT+KPRPE was not the case.

Therefore, we propose a proxy that could easily replace the norm with another quantity that can be found within a model. Since DFA predicts the landmarks $\mathbf{L}$ and a face score $\mathbf{F}_{score}$, we derive a fusion score using those quantities. First, let us review the conventional norm-weighted feature fusion equation for a set of N number of feature vectors $\{f_i\}_N$ where $f_i = ||f_i||_2 \cdot \bar{f}_i$ decomposes f_i into the norm and the unit length feature.

$$f_{\text{norm weighted}} = \frac{\sum_{i=1}^{N} ||f_i||_2 \cdot \bar{f}_i}{N}. \tag{16}$$

In the equation above, f_i represents the i-th frame-level feature, and N is the total number of frames. Now, for KPRPE, we propose a new feature fusion method, incorporating the face score and the Euclidean distance between predicted landmarks $\mathbf{L}$ and the canonical landmark $\hat{\mathbf{L}}$, which is a known set of landmarks that the training images are aligned to. This distance score, d, is computed as:

$$d_i = \frac{h - \min(|\mathbf{L}_i - \hat{\mathbf{L}}|_2, h)}{h}, \tag{17}$$

where $h = 0.2$ is a fixed constant that allows the score to be bounded between 0 and 1. The face score $\mathbf{F}^i_{score}$ represents the quality of the image, and d_i assigns more weight to well-aligned images. Proposed feature fusion equation, hence, becomes:

$$f_{\text{KPRPE}} = \frac{\sum_{i=1}^{N} (d_i \cdot \mathbf{F}^i_{score}) \cdot \bar{f}_i}{N}. \tag{18}$$

This method allows for the aggregation of features even when the feature norm does not serve as a good proxy for the quality of an image. In computing IJB-S result for ViT+KPRPE, we use this fusion method.

However, for a fair comparison in IJB-S, it is important to apply this fusion method to previous methods. Therefore, we include the breakdown of with and without landmark score based fusion. For single image based datasets such as TinyFace, AgeDB or CFPFP, feature fusion is not needed.

Training Data: MS1MV3	Feature Fusion Method	IJBS Rank1	IJBS Rank5	TinyFace Rank1
ViT Base+IRPE	Average	62.49	70.50	69.05
ViT Base+IRPE	Landmark based	63.81	71.30	69.05
ViT Base+KPRPE	Average	63.44	72.04	69.88
ViT Base+KPRPE	Landmark based	64.68	72.33	69.88
Training Data: WebFace4M	Feature Fusion Method	IJBS Rank1	IJBS Rank5	TinyFace Rank1
ViT Large+IRPE	Average	71.32	76.22	74.92
ViT Large+IRPE	Landmark based	71.93	77.14	74.92
ViT Large+KPRPE	Average	65.95	71.64	75.80
ViT Large+KPRPE	Landmark based	72.78	78.20	75.80

Table 8. Breakdown of with and without fusion method in various backbones and datasets.

The performance of ViT+KP-RPE consistently surpasses ViT+iRPE, both in scenarios using Averaging or Landmark-based fusion. This affirms the efficacy of KP-RPE in enhancing performance, even in single image contexts like TinyFace. Importantly, while the keypoint detection step is integral to KP-RPE, it isn't incorporated within iRPE, making a direct comparison based on this score less straightforward for iRPE.

Interestingly, average fusion does not synergize well with ViT+KP-RPE. Contrary to typical observations where feature magnitude positively correlates with image quality [30], with ViT+KP-RPE, a higher feature magnitude actually suggests reduced image quality. It remains unclear why this inverse relation emerges in our model. Through empirical observations, the relationship between feature magnitude and image quality appears contingent on the chosen training dataset and model architecture. For instance, models based on the ResNet architecture consistently exhibit a positive correlation between feature magnitude and image quality.

5. Alignment Visualizations

TinyFace [7] and IJBS [29], which are prone to alignment failures. In Fig 9 we show some success and failure caes in alignment. These images are taken from the released aligned dataset itself.

Figure 9. Actual examples of aligned and mis-aligned images from TinyFace [7] (row1,3) and IJB-S [29] (row2,4) datasets. These are shown as processed and used by [30]. Lines are placed on the eyes for a visual guide for an alignment.

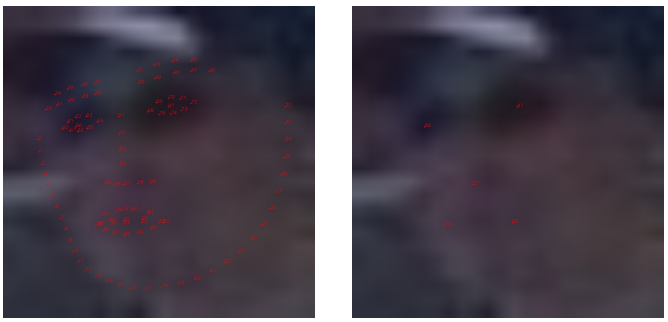
6. Comparison with SoTA Off-the-Shelf Landmark Detector

We evaluate the off-the-shelf landmark detector SLPT [76] (CVPR2022), which delivers strong performance on the high-quality WFLW [75] dataset. However, its performance dips significantly on the WiderFace dataset, populated with lower-quality images, as demonstrated in Tab. 9. This evaluation is not aimed at drawing a direct comparison between SLPT and DFA, as DFA is trained specifically on WiderFace. Instead, it serves to underline the performance variations of landmark detectors when trained on diverse datasets, stressing the importance of training dataset selection. Additionally, DFA boasts a magnitude faster speed than SLPT.

Since SLPT predicts 98 landmarks compared to 5 landmarks in DFA, we convert the SLPT landmarks by selecting indices that represent the left eye, right eye, nose, left mouth, and right mouth. An example is shown in Fig. 10.

Table 9. Comparison of DFA to SoTA Landmark detector. Note that NME is evaluated on on WiderFace Validation set. DFA is trained on WiderFace training set. SLPT is trained on WFLW. Direct NME comparison is not fair as the training dataset is different.

Models	Train Data	NME	FLOP	Params
DFA MobileNet	WiderFace [78]	0.0518	0.14 GFLOP	0.49M
SLPT [76] 6 Layer	WFLW [75]	0.1104	8.40 GFLOP	13.19M



SLPT 98 Landmarks Converted to 5 Landmarks

Figure 10. For converting 98 points landmarks from SLPT output, we choose indices 96, 97, 57, 76, 82.

7. Pipeline Detail

In this section, we elaborate on the inference scenarios involved in evaluation pipelines. A face recognition pipeline could be simplified to the following diagram. For a given raw image (a), the face detector crops out an image containing a face region (b). Then a conventional alignment algorithm (MTCNN, RetinaFace, DFA) simultaneously predicts the landmarks (c) from (b). The least-square minimization algorithm is used to align (b) into the aligned image (d) using keypoints (c) and a reference landmark. This reference landmark is arbitrarily chosen, but the FR community usually adopts one popular setting.

When one trains or evaluates face recognition models, most of the time, it is using aligned images (d), highlighted by the blue path. In our main paper, Tables 1,2, and 4, the aligned dataset and low-quality dataset are evaluated this way. The unaligned dataset in Tables 1 and 2 refers to the orange path. Whenever KPRPE is used, the keypoints are predicted using the inputs (b) or (d) depending on the path.

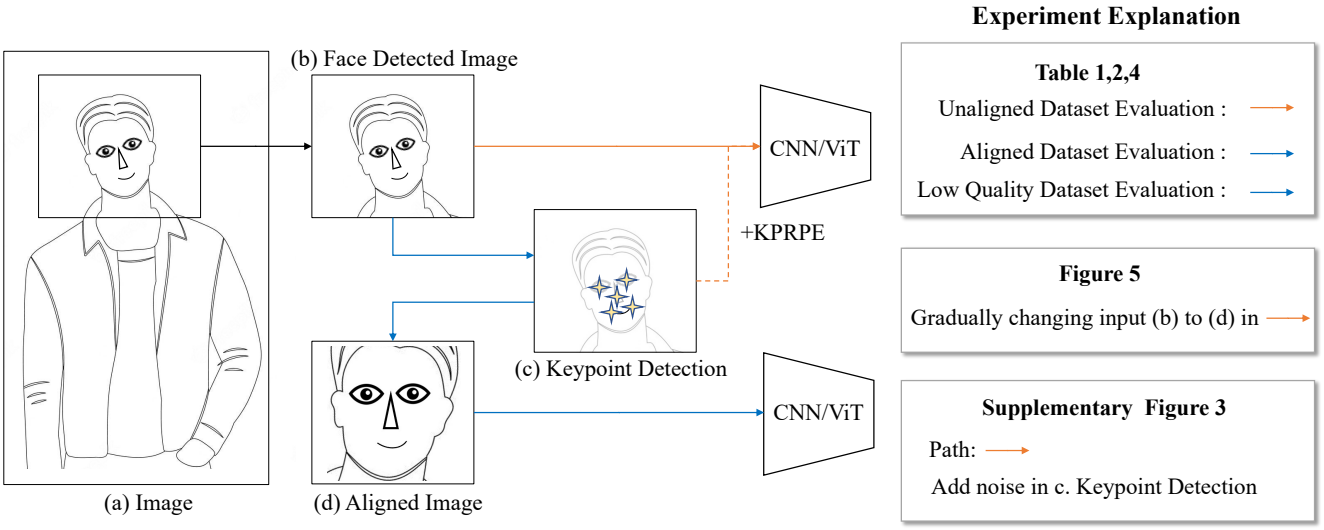


Figure 11. An illustration of face recognition pipeline from the raw image (a) to the aligned image (d).

8. KPRPE Visualization

We show the learned attention offset values in KPRPE. The red star denotes the query location and the blue circles represent the predicted landmarks. We pick head index 0 and plot the Transformer depth 0,1,3,5,7. Figs. 12 show different patterns of learned offset based on depth and query locations. Note that the higher values are denoted by a stronger blue color. Some attention offsets are 1) far from the query location, 2) horizontal pattern, etc. But there is an inherent bias toward attending nearby pixels.

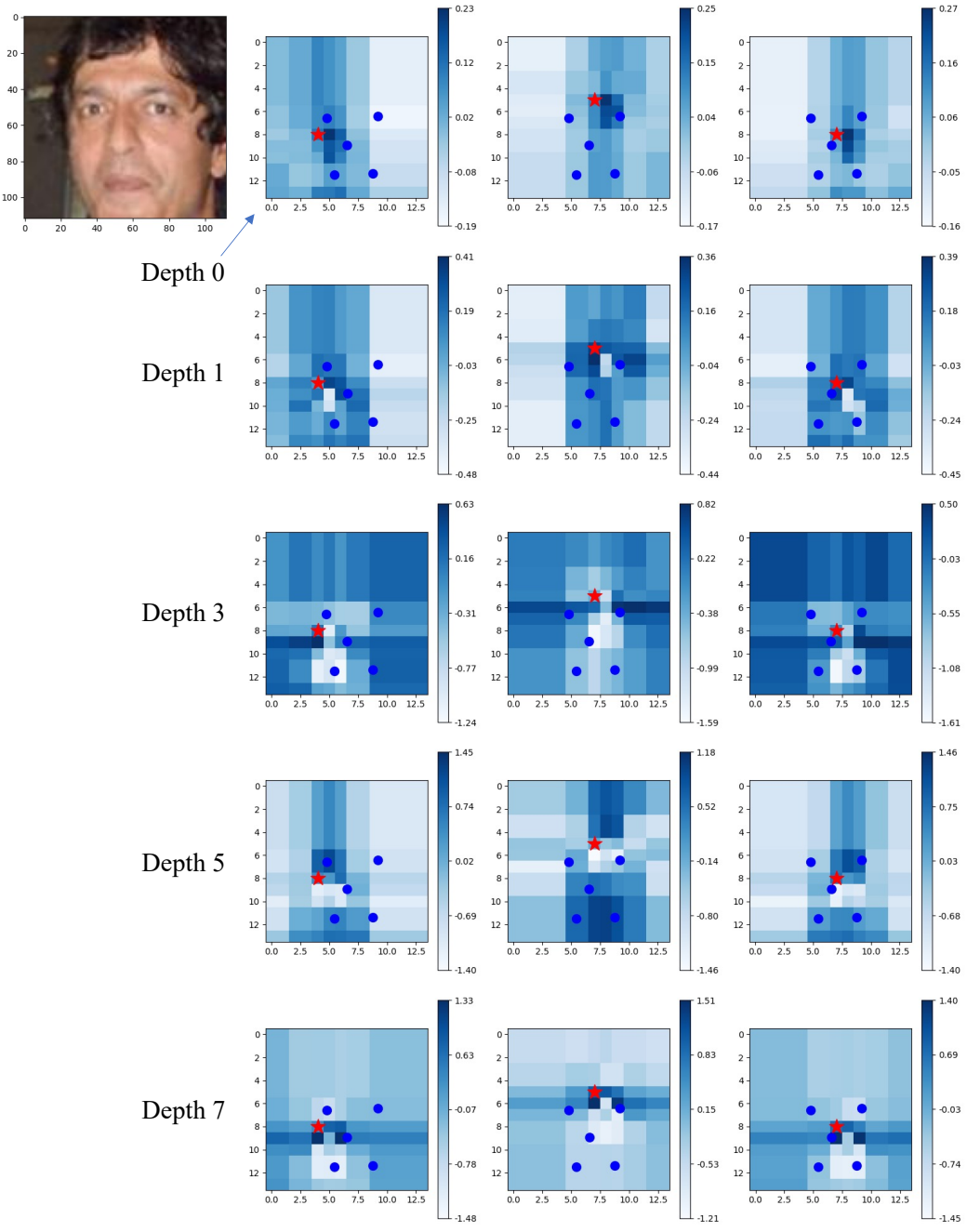


Figure 12. KPRPE Learned Offset $\mathbf{B}_{ij}$ visualization for different Transformer depths.

Also, we show in Fig. 13, an image with different images, therefore different landmark patterns. The changes in attention are not as dramatic as the changes across different head or depth. However, these changes observed in Fig .13 account for the spatial variations in the image once they accumulate over all of the attention modules in the model.

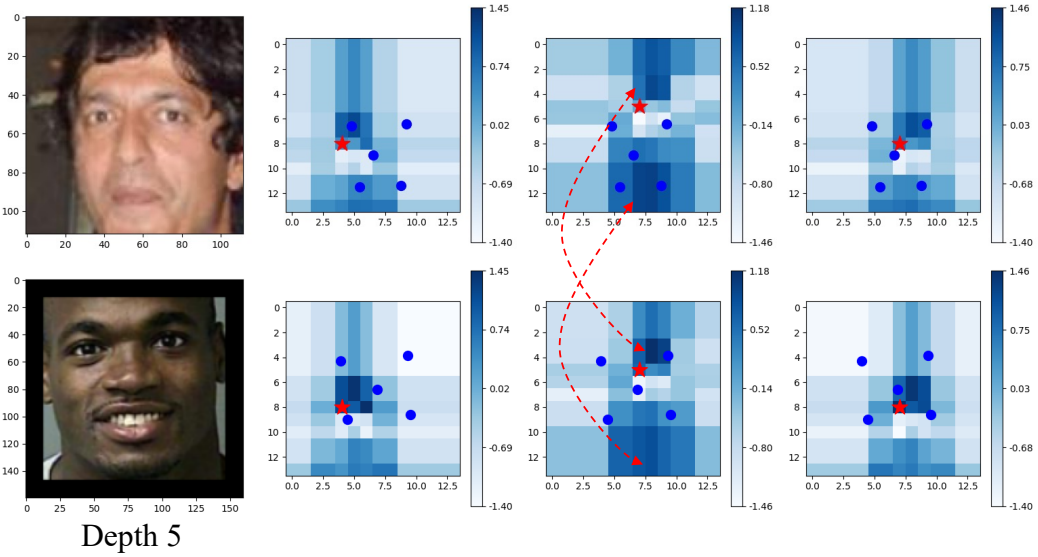


Figure 13. Cross image learned KPRPE visualization. We show the depth 5, and head index 0 of the same model.