KeyPoint Relative Position Encoding for Face Recognition

Minchul Kim    Yiyang Su    Feng Liu    Anil Jain    Xiaoming Liu
(kimminc2; suyiyan1; liufeng6; jain; liuxm)@msu.edu

Abstract

In this paper, we address the challenge of making ViT models more robust to unseen affine transformations. Such robustness becomes useful in various recognition tasks such as face recognition when image alignment failures occur. We propose a novel method called KP-RPE, which leverages key points (e.g. facial landmarks) to make ViT more resilient to scale, translation, and pose variations. We begin with the observation that Relative Position Encoding (RPE) is a good way to bring affine transform generalization to ViTs. RPE, however, can only inject the model with prior knowledge that nearby pixels are more important than far pixels. Keypoint RPE (KP-RPE) is an extension of this principle, where the significance of pixels is not solely dictated by their proximity but also by their relative positions to specific keypoints within the image. By anchoring the significance of pixels around keypoints, the model can more effectively retain spatial relationships, even when those relationships are disrupted by affine transformations. We show the merit of KP-RPE in face and gait recognition. The experimental results demonstrate the effectiveness in improving face recognition performance from low-quality images, particularly where alignment is prone to failure. Code and pre-trained models are available.

1. Introduction

Geometric alignment has shown to be highly effective for certain computer vision problems, such as face, body and gait recognition [11, 12, 25, 30, 33, 34, 37, 40, 43, 48–50, 59, 69, 72, 82, 83, 87, 88]. Alignment is the process of transforming input images, to a consistent and standardized form, often by scaling, rotating, and translating. This standardization helps recognition models learn the underlying patterns and features more effectively. As a result, many state-of-the-art (SoTA) face recognition models [11, 30, 49, 69] rely on well-aligned datasets [10, 11, 22, 91] to achieve high accuracy.

Fig. 1 shows a toy example with a training dataset MNIST [14] and test set AffNIST [57] which is in unseen affine transformation of MNIST. Using a shallow ViT [15] model, one can easily achieve 98.1% accuracy in the MNIST test set. However, in AffNIST, ViT with the original Absolute Position Embedding obtains only 77.27% accuracy. Such a sharp decrease in performance with unseen affine transform causes problems in applications that rely on accurate input alignments.

In face recognition, alignment can be imperfect, especially in low-quality images where accurate landmark detection is difficult [10, 39]. Thus, images with low resolution or taken in poor lighting may result in misalignment during testing. Given the interplay between alignment and recognition, it becomes crucial to proactively handle potential alignment failures, which often result from, e.g., low-quality images. In other words, there is a need for a recognition model that is robust to scale, rotation, and translation variations.

We revisit the Relative Position Encoding (RPE) concept
used in ViT [15] and find that RPE can be useful for introducing affine transform robustness. RPE [60] enables the model to capture the relative spatial relationships among regions of an image, learning the positional dependencies without relying on absolute coordinates. As shown in Fig. 1, adding RPE to ViT increases the performance in AffNIST. With RPE [60], queries and keys of self-attention [68] at closer distances can be assigned different attention weights compared to those at a greater distance. While RPE allows the model to exploit relative positions, it has a limitation: even if an image changes in terms of scaling, shifting, or orientation, the significance of the key-query position in RPE stays the same. This static behavior is illustrated in Figs. 2 a)-c). Notably, the key-query relationship is the same regardless of the corresponding pixels’ semantic meaning changes.

We hypothesize that an RPE which dynamically adapts based on image keypoints, such as facial landmarks, could improve the model’s comprehension of spatial relationships in the image. By leveraging the spatial relationships with respect to these keypoints, the model can adapt to variations in scale, rotation, and translation, resulting in a more robust recognition system capable of handling both aligned and misaligned data. Fig. 2 d) highlights a keypoint-dependent query-key relationship.

To this end, we introduce KeyPoint RPE (KP-RPE), a method that dynamically adapts the spatial relationship in ViT based on the keypoints present in the image. Our experiments demonstrate that incorporating KP-RPE into ViT significantly enhances the model’s robustness to misaligned test datasets while maintaining or even improving performance on aligned test datasets. We show the usage of KP-RPE in face recognition and gait recognition as the inputs share the same topology (face or body) that allows the keypoints to be defined. Finally, KP-RPE is an order of magnitude faster than iRPE [74], a widely used RPE that depends on the image content.

In summary, the contributions of this paper include:

• The insight that RPE (or its variants) can improve the robustness of ViT to unseen affine transformations.

• The development of Keypoint RPE (KP-RPE), a novel method that dynamically adapts the spatial relationship in Vision Transformers (ViT) based on the keypoints in the image, significantly enhancing the model’s robustness to misaligned test datasets while maintaining or improving performance on aligned test datasets.

• Comprehensive experimental validation demonstrating the effectiveness of our proposed KP-RPE, showcasing its potential for advancing the field of recognition by bringing model’s robustness to geometric transformation. We improve the recognition performance across unconstrained face datasets such as TinyFace [7] and IJB-S [29] and even non-face datasets such as Gait3D [18, 87].

2. Related Works

Relative Position Encoding in ViT Relative Position Encoding (RPE) is first introduced by Shaw et al. [60] as a technique for encoding spatial relationships between different elements in a sequence. By adding relative position encodings into the queries and keys, the model can effectively learn positional dependencies without relying on absolute coordinates. Subsequent works, such as those by Dai et al. [9] and Huang et al. [28], refine and expand upon the concept of RPE, demonstrating its effectiveness in natural language processing (NLP) tasks.

The adoption of RPE in Vision Transformers [15] has been explored by several researchers. For instance, Ramachandran et al. [53] propose a 2D RPE method that computes the $x$, $y$ distance in an image plane separately to include directional information. A notable RPE method in ViT is iRPE [74], which considers directional relative distance modeling as well as the interactions between queries and relative position encodings in a self-attention mechanism.
Despite the success of these RPE methods in various vision tasks, they do not specifically address the challenges associated with scale, rotation, and translation variations in face recognition applications. This shortcoming highlights the need for RPE methods that can better handle these variations, which are common in real-world low-quality face recognition scenarios. To address this, we propose KP-RPE, which incorporates keypoint information during the network’s feature extraction, significantly enhancing the model’s ability to generalize across affine transformations.

**Face Recognition with Low-Quality Images** Recent FR models [11, 27, 43, 49, 69] have achieved high performance on datasets with discernable facial attributes, such as LFW [25], CFP-FP [59], CPLFW [88], AgeDB [50], CALFW [89], IJB-B [72], and IJB-C [48]. However, unconstrained scenarios like surveillance or low-quality videos present additional challenges. Datasets in this setting, such as TinyFace [7] and IJB-S [29], contain low-quality images, where detecting facial landmarks and achieving proper alignment becomes increasingly difficult. This adversely affects existing FR models that rely on well-aligned inputs.

Several studies [20, 21, 26, 30, 31, 36, 41, 54, 63, 80, 84, 85] tackle recognition with low-quality imagery. Particularly, AdaFace [30] introduces an image quality adaptive loss function that reduces the influence of low-quality or unidentifiable samples. A-SKD [63] employs teacher-student distillation to focus on similar areas regardless of image resolution. But, these models, which are trained on aligned training sets, do not tackle the challenges associated with misaligned inputs in real-world situations. In contrast, KP-RPE adjusts spatial relationships within ViT based on image keypoints, allowing the model to better generalize even when alignment is unsuccessful in low-quality imagery.

**Keypoints and Spatial Reasoning** Keypoint detection, often associated with landmarks, has been fundamental in various vision tasks such as human pose estimation [5, 51], face landmark detection [4, 35, 65, 81], and object localization [52]. These keypoints serve as representative points that capture the essential structure or layout of an object, facilitating tasks like alignment, recognition, and even animation.

Face landmark detection is commonly carried out alongside face detection. MTCNN [81] is a widely-used method for combined face detection and facial landmark localization, utilizing cascaded CNNs (P-Net, R-Net, and O-Net) that collaborate to detect faces and landmarks in an image. RetinaFace [10], on the other hand, is a single-stage detector [38, 42] based landmark localization algorithm, demonstrating strong performance when trained on the annotated WiderFace [78] dataset. TinaFace [90] further enhances detection capabilities by incorporating SoTA generic object detection algorithms. MTCNN and RetinaFace are often used for aligning face datasets.

Recent advances in keypoint detection techniques, particularly using deep neural networks, have led to using keypoints to improve the performance of recognition tasks [64, 77]. For instance, [23] proposes a keypoint-based pooling mechanism and shows promising results in skeleton-based action recognition and spatio-temporal action localization tasks. Albeit its benefit, many models including ViTs do not have pooling mechanisms. KP-RPE is the first attempt at incorporating keypoints into the RPE which can be easily inserted into ViT models.

**3. Proposed Method**

**3.1. Background**

**Self-Attention** Self-attention is a crucial component of transformers [68], which is a popular choice for a wide range of NLP tasks. ViT [15] applies the same self-attention mechanism to images, treating images as sequences of non-overlapping patches. The self-attention mechanism in Transformers calculates attention weights based on the compatibility between a query and a set of keys. Given a set of input vectors, the Transformer computes query (Q), key (K), and value (V) matrices through linear transformations:

\[
Q_i = x_i W_Q, \quad K_j = x_j W_K, \quad V_j = x_j W_V, \quad (1)
\]

where \(x_i\) is the \(i\)-th input vector, and \(W_Q, W_K,\) and \(W_V\) are learnable weight matrices.

The self-attention mechanism computes attention weights as the dot product between the query and key vectors, followed by a softmax normalization:

\[
e_{ij} = \frac{Q_i K_j^T}{\sqrt{d_k}}, \quad a_{ij} = \frac{\exp(e_{ij})}{\sum_{j=1}^N \exp(e_{ij})}, \quad (2)
\]

where \(d_k\) is the dimension of the key vectors. Finally, the output matrix \(Y\) is computed as the product of the attention weight matrix and the value matrix: \(Y_i = \sum_{j=1}^N a_{ij} V_j\).

**Absolute Position Encoding** Transformers are inherently order invariant, as their self-attention mechanism does not consider input token positions. To address this, absolute position encoding is introduced [19, 68], which adds fixed, learnable positional embeddings to input tokens:

\[
x_i' = x_i + \text{PE}(i), \quad (3)
\]

where \(x_i'\) is the updated input token with positional information, \(x_i\) is the original input token, and \(\text{PE}(i)\) is the positional encoding for the \(i\)-th position. These embeddings, generated using sinusoidal functions or learned directly, enable the model to capture the absolute positions of elements.
Relative Position Encoding (RPE)  RPE, introduced by Shaw et al. [60] and refined by Dai et al. [9] and Huang et al. [28], encodes relative position information, essential for tasks focusing on input element relationships. Unlike absolute position encoding, RPE considers query-key interactions based on sequence-relative distances. The modified self-attention calculation for RPE is:

\[
\epsilon'_{ij} = \frac{(Q_i + R^{Q}_{ij})(K_j + R^{K}_{ij})^T}{\sqrt{d_k}}, \quad Y_i = \sum_{j=1}^{n} a_{ij}(V_j + R^{V}_{ij}).
\]  

(4)

Here, \(R^{Q}_{ij}, R^{K}_{ij}, \) and \(R^{V}_{ij}\) are relative position encoding between the \(i\)-th query and \(j\)-th key with shape \(\mathbb{R}^{d_k}\). Each \(R\) is a learnable matrix of \(\mathbb{R}^{K \times d_k}\), where \(R_{i,j}\) corresponds to the relative position encoding for distance \(d(i,j) = k\) and \(K\) is the maximum possible value of \(d(i,j)\). To obtain relative position encoding, we index the \(R\) matrix using the computed distance \(R[d(i,j)]\). Common choices for \(d\) are quantized Euclidean distance, separate \(x, y\) cross distance [53]. [74] uses a quantized \(x, y\) product distance, which encodes direction information. Note, query location \(i\) is a 2D point \((i_x, i_y)\). Fig. 3 a) and b) illustrate the distance between \(i\) and all possible \(j\) with different distance functions. For KP-RPE, we modify [74] and allow the RPE to be keypoint dependent.

3.2. Keypoint Relative Position Encoding

Building upon the general formulation of [74], we begin with the following RPE formulation:

\[
\epsilon'_{ij} = \frac{Q_iK_j^T + B_{ij}}{\sqrt{d_k}}.
\]  

(5)

Here, \(B_{ij}\) is a scalar that adjusts the attention matrix based on the query and key indices \(i, j\). Assuming a set of keypoints \(\mathbf{P} \in \mathbb{R}^{N \times 2\times 2}\) is available for each \(x\), our goal is to make \(B_{ij}\) dependent on \(\mathbf{P}\). For face recognition, \(\mathbf{P}\) is the five facial landmarks (two eyes, nose, mouth tips). For gait recognition, \(\mathbf{P}\) is 17 points from the joint locations of skeleton predictions. For the MNIST toy example, \(\mathbf{P}\) is five keypoints from the four corners and the center of the minimum cover box of a foreground image. As such \(\mathbf{P}\) can be defined for objects with shared topology.

The novelty of KP-RPE lies in the design of \(B_{ij}\). Since

\[
B_{ij} = W[d(i,j)] \in \mathbb{R}^{1},
\]  

(6)

comprises of a learnable table \(W\) and a distance function \(d(i,j)\), we can make \(W\) or \(d(i,j)\) depend on the keypoints. At a first glance, \(d(i,j, \mathbf{P})\), conditioning the distance on \(\mathbf{P}\) seems plausible. However, we find that it leads to inefficiencies, as distance caching, which is precomputing \(d(i,j)\) for a given input size, is only feasible when \(d(i,j)\) is independent of the input. Therefore, we propose an alternative where the bias matrix itself, \(W\), is a function of \(\mathbf{P}\):

\[
B_{ij} = f(\mathbf{P})[d(i,j)].
\]  

(7)

We propose three variants of \(f(\mathbf{P})\) building up from the simplest solution.

Absolute \(f(\mathbf{P})\). Let \(\mathbf{P} \in \mathbb{R}^{N L \times 2}\) be the normalized keypoints between 0 and 1. First, the simplest way to model the indexing table is to linearly map \(\mathbf{P}\) to the desired shape, \(f(\mathbf{P}) = \mathbf{P}' \cdot \mathbf{W}_L\), where \(\mathbf{P}' \in \mathbb{R}^{1 \times (2NL)}\) is reshaped keypoints \(\mathbf{P}\) and \(\mathbf{W}_L \in \mathbb{R}^{(2NL \times K)}\) is a learnable matrix. \(K\) is the maximum distance value in \(d(i,j)\). For each distance between \(i\) and \(j\), we learn a keypoint adaptive offset value. However, this \(f(\mathbf{P})\) only works with the absolute position information of \(\mathbf{P}\) and the relative distance between \(i\) and \(j\). It is missing the relative distance between \(\mathbf{P}\) and \((i, j)\).

Relative \(f(\mathbf{P})\). To improve, \(f(\mathbf{P})\) can be adjusted to work with the position of keys and queries relative to the keypoints. In other words, so that the query-key relationship in \(B_{ij}\) depends on the query-landmark relationship. To achieve this, we generate a mesh grid \(M \in \mathbb{R}^{N \times 2}\) of patch containing all possible combinations of \(i_x\) and \(i_y\). \(N\) represents the number of patches. We then compute the element-wise difference between the normalized grid and keypoints \(\mathbf{P}\) to
obtain a grid of $i, j$ relative to the keypoints:

$$D = \text{Expand}(M, \text{dim} = 1) - \text{Expand}(P, \text{dim} = 0),$$  \hspace{1cm} (8)

where $D$ is the broadcasted tensor difference of shape $R^{N \times L \times 2}$. Finally, we reshape $D$ and linearly project it with $W_L$. Specifically,

$$D' = \text{Reshape}(D) \in R^{N \times (2N_L)}$$  \hspace{1cm} (9)

$$f(P) = D'W_L \in R^{N \times K}$$  \hspace{1cm} (10)

$$B_{ij} = f(P)[i, d(i, j)] \in R^1.$$  \hspace{1cm} (11)

In other words, the offset value $B_{ij}$ is determined with respect to the positions of the keypoints and is unique for each query location. This approach allows for more expressive control of the query-keypoint relationships with the keypoint locations. An illustration of this is shown in Fig. 4.

**Multihead Relative $f(P)$**. Lastly, we can further enhance our method by tailoring the query-keypoint relationship for each head in the attention mechanism. When there are $H$ heads, we simply expand the dimension of $W_L$ to $W_L \in R^{(2N_L) \times HK}$. By reshaping $f(P)$, we obtain $f(P)^h$ for each head. Furthermore, considering the multiple self-attentions in ViT which entails multiple RPEs, we can individualize $f(P)$ for each self-attention by additionally increasing the dimension of $W_L$ to $W_L \in R^{(2N_L) \times N_h HK}$, where $N_h$ represents the transformer’s depth. Since $f(P)$ is computed only once per forward pass, this modification reduces negligible computational overhead compared to other operations. In Sec. 4.2, we evaluate and compare the various KP-RPE versions (basic, relative keypoint, multiple relative keypoint), demonstrating the superior performance of the multiple relative keypoint approaches.

### 4. Face Recognition Experiments

#### 4.1. Datasets and Implementation Details

To validate the efficacy of KP-RPE, we train our model using aligned face training data and evaluate on three distinct types of datasets: 1) aligned face data, 2) intentionally unaligned face data, and 3) low-quality face data containing misaligned images. For the evaluation, aligned face datasets include CFPFP [59], AgeDB [50], and IJB-C [48]. For unaligned face data, we intentionally use the raw CFPFP [59] and IJB-C [48] datasets without aligning them. Raw images, as provided by their respective creators, are equivalent to images cropped based on face detection bounding boxes. Lastly, we assess the model’s robustness on low-quality face datasets, specifically TinyFace [7] and IJB-S [29], which are prone to alignment failures. This comprehensive setup enables us to examine the effectiveness of our proposed method across diverse data conditions.

The training datasets MS1MV2 [11] MS1MV3 [13] and WebFace4M [91] are released as aligned and resized to $112 \times 112$ by RetinaFace [10] whose backbone is ResNet50 model trained on WiderFace [78]. For keypoint detection in KP-RPE, we use RetinaFace [10] but with lighter backbone MobileNetV2 for faster inference. Given the sensitivity of ViTs to hyperparameters, we report the exact settings for learning rate, weight decay, and other parameters in the supplementary material. For ablation dataset, we take the MS1MV2 subset dataset as used in [30].

Following the training conventions of [30, 67], we adopt RandAug [8], repeated augmentation [24], random resized crop, and blurring. We utilize the AdaFace [30] loss function to train all models. For ablation, we employ ViT-small, while for SoTA comparisons, we use ViT-base models. The AdamW [46] optimizer and Cosine Learning Rate sched-
Table 1. Ablation of RPE on ViT-small. Aligned is the standard protocol with raw face images (detector bounding box) aligned by RetinaFace [10] and resized to $112 \times 112$. Unaligned takes the raw face images and simply resizes it to $112 \times 112$. Aligned setting always shows better performances and Unaligned is for simulating alignment failure. Low Quality Aligned dataset may have alignment failures.

<table>
<thead>
<tr>
<th>Method</th>
<th>Low Quality Aligned Dataset</th>
<th>High Quality Aligned Dataset</th>
<th>High Quality Unaligned Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViT</td>
<td>68.24 72.96 59.60 68.31</td>
<td>96.11 92.22 72.81</td>
<td>21.62</td>
</tr>
<tr>
<td>ViT + iRPE</td>
<td>69.05 73.10 62.49 70.50</td>
<td>97.01 92.72 77.91</td>
<td>34.73</td>
</tr>
<tr>
<td>ViT + KP-RPE</td>
<td>69.88 74.25 63.44 72.04</td>
<td>96.60 94.20 93.56</td>
<td>91.85</td>
</tr>
</tbody>
</table>

Table 2. Ablation of KP-RPE with three different formulations of keypoint dependent RPE tables $f(P)$. The sharp increase in Unaligned setting shows the robustness to unseen affine transform manifests with Relative $f(P)$. MultiHead $f(P)$ further improves the performance.

<table>
<thead>
<tr>
<th>Method</th>
<th>Low Quality Aligned Dataset</th>
<th>High Quality Aligned Dataset</th>
<th>High Quality Unaligned Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>KP-RPE Absolute $f(P)$</td>
<td>68.11 72.42 69.13</td>
<td>96.61 90.96</td>
<td>68.99 14.91</td>
</tr>
<tr>
<td>KP-RPE Relative $f(P)$</td>
<td>69.42 73.71 70.77</td>
<td>96.74 94.28</td>
<td>89.70 85.22</td>
</tr>
<tr>
<td>KP-RPE MultiHead $f(P)$</td>
<td>69.88 74.25 63.44</td>
<td>96.60 94.20</td>
<td>93.56 91.85</td>
</tr>
</tbody>
</table>

4.2. Ablation Analysis

Row 1 in Tab. 1 shows results on the baseline ViT. Row 2 and 3 show results on the baseline ViT with iRPE and our proposed KP-RPE. KP-RPE demonstrates a substantial performance improvement on unaligned and low-quality datasets, without compromising performance on aligned datasets. Last row highlights the difference between ViT and ViT+KP-RPE. Also, Fig. 5 shows the sensitivity to the affine transformation, i.e., how the performance changes when one interpolates the affine transformation from the face detection images to the aligned images in CFPFP dataset.

Tab. 2 further investigates the effect of modifications to KP-RPE. By making KP-RPE dependent on the difference between the query and keypoints (row 2), we observe a significant improvement in unaligned dataset performance. Also, by allowing unique mapping for each head and module in ViT (row 3), we achieve a further improvement. In other words, more expressive KP-RPE is beneficial for learning complex RPE that depends on the keypoints of an image. Overall, the ablation study highlights the necessity of each component in KP-RPE and the effectiveness of KP-RPE in enhancing the robustness of face recognition models, particularly with unaligned and low-quality datasets.

4.3. Computation Analysis

In this section, we analyze the computational efficiency of our proposed KP-RPE in terms of FLOPs, throughput, and the number of parameters. Tab. 3 shows that KP-RPE is highly efficient, with only a small increase in the computational cost (FLOPs) compared to the backbone: 0.02 GFLOP increase for ViT Small and 0.07 GFLOP increase for ViT Base (ViT vs ViT+KP-RPE). Notably, KP-RPE is considerably more efficient than iRPE, which incurs an increase of 0.71 GFLOP for ViT Small and 1.42 GFLOP for ViT Base.

Considering training throughput, which factors in computation time during training (with backpropagation), KP-RPE’s efficiency is more pronounced. It only reduces throughput by 9.15% for ViT Small and 16.44% for ViT Base, as opposed to iRPE’s larger decrease. Also, we show the GLOP and throughput with the landmark detection time included. Landmark detection time is negligible compared to the total feature extraction time.

Also, our method introduces a negligible increase in the number of parameters: just 0.05M for ViT Small and 0.21M for ViT Base. Hence, incorporating KP-RPE into the model achieves enhancements performance without a substantial rise in computational cost or model complexity.
The results, as shown in Tab. 4, underscore the strengths of Table 3.

vanilla RetinaFace on face alignment, ours is fully differentiable during inference and name it Differentiable Face

the face recognition performances are also similar. Unlike

improvements on low-quality ones, marking it a valuable

mance performance on low-quality datasets. In summary, our

findings of our ablation study, the introduction of KP-RPE

leads to an appreciable improvement in these challenging

Scenarios. This supports our thesis that face recognition

methods with large-scale datasets and high-quality datasets. We undertake a comprehensive evaluation, covering both high-quality and low-quality image datasets. The results, as shown in Tab. 4, underscore the strengths of KP-RPE. Notably, the inclusion of KP-RPE does not impair the performance on high-quality datasets, a testament to its applicability to both low and high-quality datasets.

This becomes particularly compelling when we observe the performance on low-quality datasets. Consistent with the findings of our ablation study, the introduction of KP-RPE leads to an appreciable improvement in these challenging scenarios. This supports our thesis that face recognition models with robust alignment capabilities can indeed enhance performance on low-quality datasets. In summary, our model with KP-RPE not only maintains competitive performance on high-quality datasets but also brings significant improvements on low-quality ones, marking it a valuable contribution to the field of face recognition.

4.5. Note on the Landmark Predictor

KP-RPE in all experiments uses our own MobileNet [58] based RetinaFace [10] to predict landmarks for KP-RPE. We train MobileNet version for computation efficiency. However, the original landmark predictor used for aligning the test datasets is ResNet50-RetinaFace [10]. We also report the KP-RPE performance with the officially released ResNet50-RetinaFace. We report this to compare KP-RPE on the same ground with other models by using the same landmark used to pre-align the testset. The face recognition performance of KP-RPE+Official is similar to KP-RPE+Ours (75.80 vs 75.80 in TinyFace Rank1). Our MobileNet-RetinaFace is improved to perform similarly to ResNet50 in landmark prediction by applying additional tricks while training. Therefore, the face recognition performances are also similar. Unlike vanilla RetinaFace on face alignment, ours is fully differentiable during inference and name it Differentiable Face Aligner. Details and analysis can be found in Supp.2 and 3.

4.6. Scalability on Larger Training Datasets

We train the ViT+KP-RPE model on a larger WebFace12M [91] dataset to demonstrate the potential of KP-RPE in its scalability and applicability in real-world, data-rich scenarios. Tab. 4's last row shows that the performance continues to increase with WebFace12M dataset.

Discussion. 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 our attempt at a more detailed answer, please refer to Supp.2.4.

5. Gait Recognition Experiments

KP-RPE is a generic method that can generalize beyond face recognition to any task with keypoints. We apply KP-RPE to gait recognition using body joints as the keypoints.

Dataset. We train and evaluate on Gait3D [87], an in-the-wild gait video dataset. In our experiments, we use silhouettes and 2D keypoints preprocessed and released by the authors directly. Following SMPLGait [79, 87], we use rank-n accuracy (n = 1, 5, 10), mean Average Precision (mAP), and mean Inverse Negative Penalty (mINP) for evaluation.

Implementation Details. We implement SwinGait-2D [17] as the baseline in our experiments. SwinGait-2D is chosen over SwinGait-3D [17] because we focus on exploiting the geometric information in gait recognition. SwinTransformer [44] uses vanilla relative positional encoding for each windowed self-attention. To incorporate KP-RPE into the SwinTransformer, we modify the 2D grid M to be the size of the window as opposed to the image size. Following the default configuration of [87], we use an AdamW [46] optimizer with a learning rate $3 \times 10^{-4}$ and weight decay $2 \times 10^{-2}$, accompanied by an SGDR [45] scheduler. We train our models for 60,000 iterations, sampling 32 subjects and 4 sequences per subject in a batch.
Table 4. SoTA comparison on low-quality and high-quality datasets. ViT models are ViT-Base sized.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Train Data</th>
<th>Low Quality Dataset</th>
<th></th>
<th></th>
<th></th>
<th>High Quality Dataset</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rank-1</td>
<td>Rank-5</td>
<td></td>
<td></td>
<td>Rank-1</td>
<td>Rank-5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TinyFace [7]</td>
<td></td>
<td></td>
<td></td>
<td>IJB-S [29]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PFE [61]</td>
<td>CNN64</td>
<td>MS1MV2 [11]</td>
<td>-</td>
<td>-</td>
<td>50.16</td>
<td>58.33</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>URL [62]</td>
<td>ResNet101</td>
<td>MS1MV2 [11]</td>
<td>63.89</td>
<td>68.67</td>
<td>59.79</td>
<td>65.78</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CurricularFace [27]</td>
<td>ResNet101</td>
<td>MS1MV2 [11]</td>
<td>63.68</td>
<td>67.65</td>
<td>62.43</td>
<td>68.68</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AdaFace  [30]</td>
<td>ViT+KP-RPE</td>
<td>MS1MV1 [13]</td>
<td>72.05</td>
<td>74.84</td>
<td>68.95</td>
<td>71.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AdaFace  [30]</td>
<td>ViT+KP-RPE</td>
<td>MS1MV1 [13]</td>
<td>72.02</td>
<td>74.52</td>
<td>70.42</td>
<td>75.29</td>
<td>97.90</td>
<td>99.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AdaFace  [30]</td>
<td>ViT+KP-RPE</td>
<td>MS1MV1 [13]</td>
<td>74.92</td>
<td>77.98</td>
<td>71.93</td>
<td>77.14</td>
<td>97.15</td>
<td>99.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AdaFace  [30]</td>
<td>ViT+KP-RPE</td>
<td>MS1MV1 [13]</td>
<td>75.80</td>
<td>78.49</td>
<td>72.78</td>
<td>78.20</td>
<td>97.67</td>
<td>99.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AdaFace  [30]</td>
<td>ViT+KP-RPE</td>
<td>MS1MV1 [13]</td>
<td>76.18</td>
<td>78.97</td>
<td>72.94</td>
<td>77.46</td>
<td>98.07</td>
<td>99.30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5. KP-RPE performance on Gait3D [87] compared with the baseline. KP-RPE boosts all metrics by a large margin.

<table>
<thead>
<tr>
<th>Model</th>
<th>Rank-1</th>
<th>Rank-5</th>
<th>mAP</th>
<th>mINP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GaitSet   [6]</td>
<td>36.7</td>
<td>58.3</td>
<td>30.01</td>
<td>17.30</td>
</tr>
<tr>
<td>MTSGaT [86]</td>
<td>48.7</td>
<td>67.1</td>
<td>37.63</td>
<td>21.92</td>
</tr>
<tr>
<td>DAnet [47]</td>
<td>48.0</td>
<td>69.7</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>GaitGCI [16]</td>
<td>50.3</td>
<td>68.5</td>
<td>39.5</td>
<td>24.3</td>
</tr>
<tr>
<td>GaitBase [18]</td>
<td>64.6</td>
<td>81.5</td>
<td>55.31</td>
<td>31.63</td>
</tr>
<tr>
<td>HSTL [70]</td>
<td>61.3</td>
<td>76.3</td>
<td>55.48</td>
<td>34.77</td>
</tr>
<tr>
<td>DyGait [71]</td>
<td>66.3</td>
<td>89.8</td>
<td>56.40</td>
<td>37.30</td>
</tr>
<tr>
<td>SwinGait-2D [17]</td>
<td>67.1</td>
<td>83.7</td>
<td>58.76</td>
<td>34.36</td>
</tr>
<tr>
<td>+ KP-RPE</td>
<td>68.2</td>
<td>84.4</td>
<td>60.81</td>
<td>36.19</td>
</tr>
</tbody>
</table>

Results and Analyses. In Tab. 5, we compare to SoTA approaches, including SwinGait-2D [17], with and without KP-RPE. We can see that the KP-RPE shows a significant improvement over SwinGait-2D, with 1.1% and 0.7% improvement on rank-1 and -5 accuracies, respectively. mAP has improved by 2.05% and mINP by 1.23% of the baseline) compared to SwinGait-2D. We believe that a great portion of the improvement comes from KP-RPE exploiting the gait information contained in 2D skeletons. Gait skeletons contain identity-related information, such as body shape and walking posture. This demonstrates that KP-RPE is both effective and generalizable to the gait recognition.

6. Conclusion

In this work, we introduce Keypoint-based Relative Position Encoding (KP-RPE), a method designed to enhance the robustness of recognition models to alignment errors. Our method uniquely establishes key-query relationships in self-attention based on their distance to the keypoints, improving its performance across a variety of datasets, including those with low-quality or misaligned images. KP-RPE demonstrates superior efficiency in terms of computational cost, throughput and recognition performance, especially when affine transform robustness is beneficial. We believe that KP-RPE opens a new avenue in recognition research, paving the way for the development of more robust models.

Limitations. While KP-RPE shows impressive face recognition capabilities, it does require keypoint supervision, which may not always be readily available and can constrain its application, particularly when the dataset does not comprises of images with a consistent topology. Future work should consider the self-discovery of keypoints to lessen this dependence, thereby boosting the model’s flexibility.

Potential Societal Impacts. Within the CV/ML community, we must strive to mitigate any negative societal impacts. This study uses the MS1MV* dataset, derived from the discontinued MS-Celeb, to allow a fair comparison with SoTA methods. However, we encourage a shift towards newer datasets, showcasing results using the recent WebFace4M dataset. Data collection ethics are paramount, often requiring IRB approval for human data collection. Most face recognition datasets likely lack IRB approval due to their collection methods. We support the community in gathering large, consent-based datasets or fully synthetic datasets [3, 32], enabling research without societal backlash.

Acknowledgments. This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 2022-21102100004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.
References

[4] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3d facial landmarks). In ICCV, 2017. 3
[22] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A dataset and benchmark for large-scale face recognition. In ECCV, 2016. 1
[26] Yuge Huang, Pengcheng Shen, Ying Tai, Shaoxin Li, Xiaoming Liu, Jilin Li, Feiyue Huang, and Rongrong Ji. Improving face recognition from hard samples via distribution distillation loss. In ECCV, 2020. 3
[28] Zhiheng Huang, Davis Liang, Peng Xu, and Bing Xiang. Improve transformer models with better relative position embeddings. In EMNLP, pages 3327–3335, Online, November 2020. 1, 2, 4

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.

<table>
<thead>
<tr>
<th></th>
<th>Ablation Experiments</th>
<th>Large Scale Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backbone</td>
<td>ViT Small</td>
<td>ViT Large</td>
</tr>
<tr>
<td>LR</td>
<td>0.001</td>
<td>0.0001</td>
</tr>
<tr>
<td>Batch Size</td>
<td>512</td>
<td>1024</td>
</tr>
<tr>
<td>Epoch</td>
<td>34</td>
<td>36</td>
</tr>
<tr>
<td>Momentum</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>Weight Decay</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>Scheduler</td>
<td>Cosine</td>
<td></td>
</tr>
<tr>
<td>Optimizer</td>
<td>AdamW</td>
<td></td>
</tr>
<tr>
<td>Warmup</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>AdaFace Loss Margin</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>AdaFace Loss $h$</td>
<td>0.333</td>
<td></td>
</tr>
<tr>
<td>Augmentation</td>
<td>Flip, Brightness, Contrast, Scaling, Translation, RandAug <a href="">8</a>, Blur, Cutout, Rotation ($20^\circ$)</td>
<td></td>
</tr>
<tr>
<td>PartialFC</td>
<td>None</td>
<td>sampling rate 0.6</td>
</tr>
<tr>
<td>RepeatedAug Prob</td>
<td>0.5</td>
<td>0.1</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Train Data</th>
<th>Low Quality Dataset</th>
<th>High Quality Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>TinyFace [7]</td>
<td>IJB-S [29]</td>
</tr>
<tr>
<td>AdaFace [30]</td>
<td>ViT</td>
<td>WebFace4M [91]</td>
<td>74.81</td>
<td>77.58</td>
</tr>
<tr>
<td>AdaFace [30]</td>
<td>ViT+KPRPE</td>
<td>WebFace4M [91]</td>
<td><strong>75.80</strong></td>
<td>78.49</td>
</tr>
<tr>
<td>ArcFace [11]</td>
<td>ViT+KPRPE</td>
<td>WebFace4M [91]</td>
<td>75.62</td>
<td><strong>78.57</strong></td>
</tr>
<tr>
<td>CosFace [69]</td>
<td>ViT+KPRPE</td>
<td>WebFace4M [91]</td>
<td>75.48</td>
<td>78.30</td>
</tr>
</tbody>
</table>

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.

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. Sensitivity 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,

\[ L_{\text{pert}} = L + \alpha L. \]  

\( \alpha \) is a parameter that changes the level of noise in the prediction. We change \( \alpha \) 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.](image)
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 independently 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 $X$ be an image, and let $F_{bbox}, F_{score}$ and $F_{ldmk}$ 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 $160 \times 160$, there are 1050 anchor boxes. Based on these outputs, we predict the weights for fusing the outputs. Specifically,

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

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

$$L = w^T F_{ldmk}. \quad (15)$$

The final output $L$ 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 $L$ and the ground truth landmark $L^{GT}$.

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.](image-url)
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 $\hat{L}$ and a face score $F_{\text{score}}$, 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.

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

(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 $\hat{L}$ and the canonical landmark $L$, which is a known set of landmarks that the training images are aligned to. This distance score, $d_i$, is computed as:

$$d_i = \frac{h - \min(|L_i - \hat{L}|_2, h)}{h},$$  

(17)

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

$$f_{\text{KPRPE}} = \frac{\sum_{i=1}^{N} (d_i \cdot F_{\text{score}}^i) \cdot \bar{f}_i}{N}.$$  

(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.

<table>
<thead>
<tr>
<th>Training Data: MS1MV3</th>
<th>Feature Fusion Method</th>
<th>IJB-S Rank1</th>
<th>IJB-S Rank5</th>
<th>TinyFace Rank1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViT Base+IRPE</td>
<td>Average</td>
<td>62.49</td>
<td>70.50</td>
<td>69.05</td>
</tr>
<tr>
<td>ViT Base+IRPE</td>
<td>Landmark based</td>
<td>63.81</td>
<td>71.30</td>
<td>69.05</td>
</tr>
<tr>
<td>ViT Base+KPRPE</td>
<td>Average</td>
<td>63.44</td>
<td>72.04</td>
<td>69.88</td>
</tr>
<tr>
<td>ViT Base+KPRPE</td>
<td>Landmark based</td>
<td>64.68</td>
<td>72.33</td>
<td>69.88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training Data: WebFace4M</th>
<th>Feature Fusion Method</th>
<th>IJB-S Rank1</th>
<th>IJB-S Rank5</th>
<th>TinyFace Rank1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViT Large+IRPE</td>
<td>Average</td>
<td>71.32</td>
<td>76.22</td>
<td>74.92</td>
</tr>
<tr>
<td>ViT Large+IRPE</td>
<td>Landmark based</td>
<td>71.93</td>
<td>77.14</td>
<td>74.92</td>
</tr>
<tr>
<td>ViT Large+KPRPE</td>
<td>Average</td>
<td>65.95</td>
<td>71.64</td>
<td>75.80</td>
</tr>
<tr>
<td>ViT Large+KPRPE</td>
<td>Landmark based</td>
<td>72.78</td>
<td>78.20</td>
<td>75.80</td>
</tr>
</tbody>
</table>

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 cases in alignment. These images are taken from the released aligned dataset itself.

![Alignment Visualizations](image_url)

**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.

<table>
<thead>
<tr>
<th>Models</th>
<th>Train Data</th>
<th>NME</th>
<th>FLOP</th>
<th>Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFA MobileNet</td>
<td>WiderFace [78]</td>
<td>0.0518</td>
<td>0.14 GFLOP</td>
<td>0.49M</td>
</tr>
<tr>
<td>SLPT [76] 6 Layer</td>
<td>WFLW [75]</td>
<td>0.1104</td>
<td>8.40 GFLOP</td>
<td>13.19M</td>
</tr>
</tbody>
</table>

![Comparison with SoTA Off-the-Shelf Landmark Detector](image_url)

**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.

![Diagram of face recognition pipeline](image)

**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 $B_{ij}$ 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.