Cluster and Aggregate: Face Recognition with Large Probe Set

Anonymous Author(s)
Affiliation
Address
email

Abstract

Feature fusion plays a crucial role in unconstrained face recognition where inputs (probes) comprise of a set of N low quality images whose individual qualities vary. Advances in attention and recurrent modules have led to feature fusion that can model the relationship among the images in the input set. However, attention mechanisms cannot scale to large N due to their quadratic complexity and recurrent modules suffer from input order sensitivity. We propose a two-stage feature fusion paradigm, *Cluster and Aggregate*, that can both scale to large N and maintain the ability to perform sequential inference with order invariance. Specifically, Cluster stage is a linear assignment of N inputs to M global cluster centers, and Aggregation stage is a fusion over M clustered features. The clustered features play an integral role when the inputs are sequential as they can serve as a summarization of past features. By leveraging the order-invariance of incremental averaging operation, we design an update rule that achieves batch-order invariance, which guarantees that the contributions of early image in the sequence do not diminsh as time steps increase. Experiments on IJB-B and IJB-S benchmark datasets show the superiority of the proposed two-stage paradigm in unconstrained face recognition.

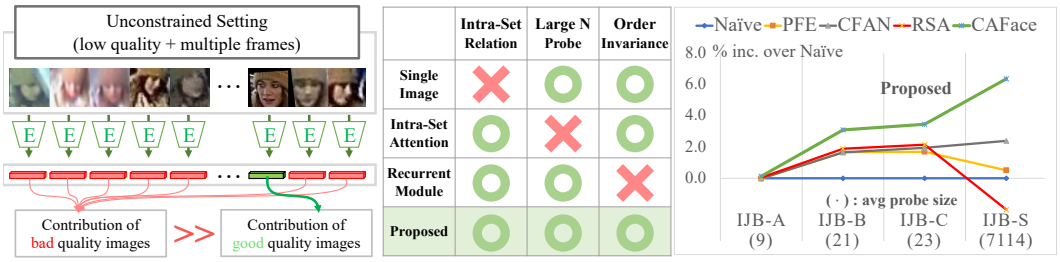
1 Introduction

Face Recognition (FR) is a task of matching a set of input query imagery, known as *probe*, to a pre-registered identity database, known as *gallery*. The FR community has witnessed remarkable progress of FR in semi-constrained settings, where the verification accuracies on datasets such as LFW [22] or CFP-FP [44] have surpassed 99%. However, FR remains challenging in the real-world scenario, and the FR community has shifted its focus to unconstrained benchmark datasets such as IJB-B [53] and IJB-S [24], which are composed of low quality still images and surveillance videos.

Unconstrained FR is a task of identifying a "person of interest", often based on low quality images and videos [5]. It is challenging due to two issues: 1) individual video frames can be of poor quality, causing erroneous FR model prediction and 2) the number of images in a probe can be very large, *e.g.*, a probe video in IJB-S may have $500,000$ frames. Feature fusion across all frames in the probe is especially crucial if frame-based predictions are unreliable. While prior works [26, 39] address the first issue of prediction in low quality images, the large size of probe set was not addressed. Fig. 1 a) illustrates the problem caused by the absence of proper feature fusion. The contribution of good quality image can be made insignificant in the presence of many other poor quality images in the set.

This paper aims to learn a fusion funcion $F : \{\boldsymbol{f}_i\}^N \rightarrow \boldsymbol{f}$ that maps an unordered set of N probe features $\{\boldsymbol{f}_i\}^N$ of the same class (person) to a single fused output $\boldsymbol{f}$, where $\boldsymbol{f}_i = E(\boldsymbol{x}_i)$ is the feature of the i-th sample in the set, using a fixed feature extractor E. The task of fusing multiple features involves : 1) estimating the quality of individual features and 2) modeling the intra-set relationship of the features. Prior feature fusion works utilize either simple average pooling [11, 42], reinforcement learning [32], recurrent models [16] or self-attention [15, 31, 34, 52, 55].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



a) Fusion without intra-set relationship **b) Drawbacks of prior works** **c) Benefit of feature fusion with large N**

Figure 1: a) Illustration of the importance of the intra-set relationship in feature fusion. Without the intra-set relationship, a large weight on a good quality image can still be outweighted by many bad quality images in a probe set. b) We need a framework that can both account for the intra-set relationship of large N probes and handle sequential inputs with order invariance. c) The importance of feature fusion increases with the probe size N. For our proposed method, CAFace, the relative performance gain over Naïve (simple averaging) method, *i.e.*, $\frac{\text{CAFace-Naïve}}{\text{Naïve}} * 100\%$, increases with the probe size acorss four datasets. PFE [46] and CFAN [15] are single-image based and lack intra-set computation. RSA [31] computes intra-set relationship but unusable for large N.

Typically, to compute the intra-set relationship among inputs of an arbitrary size N, one would adopt set-to-set functions such as Multihead Self Attention (MSA) [31, 50, 52], enabling inputs to propagate information among themselves. The downside of this approach is its computational cost of $O(N^2)$ which becomes infeasible when N exceeds a few thousand. Also, when the inputs are sequential as in a live video feed, it is nontrivial to model the intra-set relationship except to compute attention overall past frames at each time step. Recurrent methods [16, 20] are useful in the sequential inference but their drawback is set order inconsistency, *i.e.*, as the number of sequential steps T increases, the contribution of early frames in a set decreases. Fig. 1 b) contrasts various fusion methods.

A feature fusion framework that can consider both 1) intra-set relationship for a large N and 2) efficient sequential inference is necessary in the real-world FR scenarios. Fig. 1 c) shows the average probe sizes of four datasets. IJB-S [24]'s probe size is too large for intra-set attention such as RSA [31] to perform inference with all frames concurrently.

We present a feature fusion framework, Cluster and Aggregate (CAFace), that achieves two abovementioned criteria. It consists of two modules: Cluster Network (CN) and Aggregation Network (AGN). CN makes soft assignments of N features into M fixed number of clusters, *i.e.*, $\{\boldsymbol{f}_i\}^N \rightarrow \{\boldsymbol{f}'_j\}^M$ where $M << N$. While N varies from one set to other, M is fixed. AGN combines M clustered features into a single feature $\boldsymbol{f}$, *i.e.*, $\{\boldsymbol{f}'_j\}^M \rightarrow \boldsymbol{f}$. Conceptually, M intermediate cluster features serve as a summarization of N inputs and AGN models the intra-set relationship among $\{\boldsymbol{f}'_j\}^M$.

The proposed framework depends on learning global cluster assignments $\{\boldsymbol{f}_i\}^N \rightarrow \{\boldsymbol{f}'_j\}^M$ that are consistent across different probes. Thus, we propose learning cluster centers that are *input independent*. These centers govern the clustering assignments. But, it is not obvious which clustering criterion is best for feature fusion. Therefore, we design CN to discover learned clusters with an end-to-end differentiable framework that allows AGN to back-propagate the gradients to CN so that the cluster assignments are learned to maximize the FR performance. We also design an input pipeline, Style Input Maker (SIM) that can perform class (identity) agnostic clustering efficiently.

The purpose of introducing an intermediate stage $\{\boldsymbol{f}'_j\}^M$ is to facilitate the sequential inference. The key design of CN is to formulate $\{\boldsymbol{f}'_j\}^M$ as a *linear* combination of $\{\boldsymbol{f}_i\}^N$. This guarantees that even when N inputs are divided into T smaller batches, $\{\boldsymbol{f}'_j\}^M$ can be sequentially updated and still maintain batch-order invariance. This is due to our update rule, inspired by the order invariance property of the averaging operation, as in Eq. 8. When the inputs are sequential, we feed only the new features to CN and update the cached $\{\boldsymbol{f}'_j\}^M$. It achieves the same effect as having used all previous features simultaneously. Fig. 2 shows the contrast with previous approaches.

In summary, the contributions of this paper include:

- A novel feature fusion framework for both large N feature fusion and efficient sequential inference. To our knowledge, this is the first approach to utilize linearly combined intermediate clusters to achieve batch-order invariance with intra-set relationship modeling.

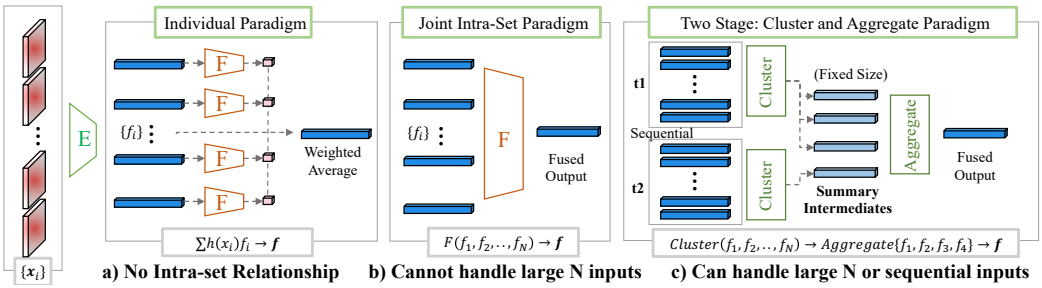


Figure 2: Comparison of feature fusion paradigms. a) In the individual paradigm, the weight of each sample is determined independently of other samples in the probe. b) In the intra-set paradigm, the sample weight is determined w.r.t. all other inputs. Intra-set calculation becomes infeasible when N is large or sequential. c) Our Cluster and Aggregate paradigm enables both large N intra-set modeling and sequential inference by introducing a shared fixed number of intermediate representations that are updated by the batch-order invariant update rule.

- An unsupervised clustering mechanism that can discover latent groups of inputs that are shared across all identities. These groups are learned to maximize the performance of the given task. We propose to use the statistics of intermediate representation to compactly represent the inputs for feature fusion.
- We demonstrate the superiority of our method in unconstrained face recognition on IJB-B and IJB-S datasets.

2 Related Work

Feature Fusion (Unordered Set). The simplest way to perform feature fusion is through an averaging operation over a set of features $\{\boldsymbol{f}_i\}^N$ [11, 42]. In this case, the features with larger norms play a bigger role, and it generally works because easy samples tend to show larger norms [39, 43]. To learn the importance of weights, CFAN and QAN utilize the self-attention mechanism, a learned weighted averaging mechanism [15, 34]. The drawback of these approaches is the lack of an intra-set relationship during the weight calculation process.

Previous works that adopt the intra-set attention mechanism are Non-local Neural network and RSA [31, 52]. These works use intermediate feature maps $\boldsymbol{M}_i$ of size $\mathbb{R}^{C_M \times H \times W}$ during aggregation because feature maps provide rich and complementary information that can be refined by taking the spatial relationship into account. However, the drawback is in the heavy computation in the attention calculation. For a set of N features maps, an attention module involves querying $N \times C_M \times H \times W$ keys and values. Our Cluster Network utilizes a compact style vector from SIM, which greatly increases the computation efficiency in attention computation.

DAC [32] and MARN [16] propose RL-based and RNN based quality estimators, respectively. Yet, they fail to be agnostic to input stream order and are unsuitable for modeling long-range dependencies. Our method can split the N inputs into T smaller batches and still achieve batch-order invariance.

Video Recognition (Ordered Set). The feature fusion for recognition has a resemblance to video-based recognition [33], but set inputs cannot always expect the temporal dependencies to be available. Therefore, most video-based approaches for tasks such as action recognition or quality enhancement [2, 4, 27, 36, 41, 58] focus on exploiting the relationship between nearby frames, whereas feature fusion works do not define them. In video-based FR, the general trend is to focus more on assessing the quality of individual frames as opposed to exploring the connection among nearby frames. Some examples of video-based FR utilize n-order statistics [37], affine hulls [9, 21, 56], SPD matrices [23] and manifolds [18, 51]. Recently, probabilistic representation such as PFE [46] gained popularity [1, 10, 45, 46] since the variance in distribution serves as a quality estimation for individual frames.

Attention Mechanism. Multihead Self Attention (MSA) [50] is a widely adopted set-to-set function that models intra-set relationships via an affinity map. It is also a key component in transformer architectures which outperform CNNs in various vision tasks [8, 12, 14, 30, 35, 49, 57]. The underlying mechanism of MSA which uses the affinity of query and key to update the value is versatile in its application beyond recognition and has led to its usage in memory retrieval and grouping [6, 47, 54]. The unique property of the proposed Cluster Network is in the linear combination of value assignment which enables batch-order invariance using an incremental average update rule.

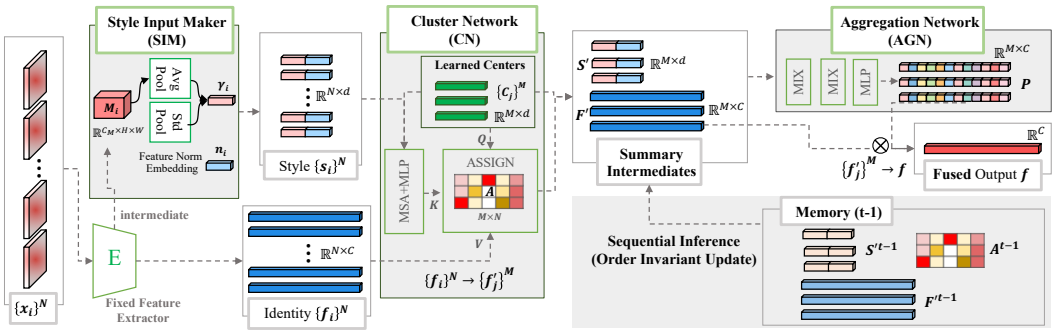


Figure 3: An overview of Cluster and Aggregate (CAFace). The output $\boldsymbol{f}$ is the fused result given the variable sized set of images $\{\boldsymbol{x}_i\}^N$. The main idea is to utilize the style of images $\{\boldsymbol{s}_i\}^N$ from SIM and a set of M learned global cluster centers $\{\boldsymbol{c}_j\}^M$ from CN to first cluster $\{\boldsymbol{f}_i\}^N$ into $\boldsymbol{S}'$ and $\boldsymbol{F}'$ whose number is fixed M. AGN uses MLP-Mixer [48] to consider the intra-set relationship of summary intermediates to estimate the importance of $\boldsymbol{F}'$ for fusion. When the inputs are split into batches, the summary intermediates are stored and used for the update rule that is invariant to the batch order.

3 Proposed Approach

In this section, we formally layout assumptions, architectures and loss functions. While our task is face recognition, the feature fusion paradigm can be carried over to other domains as well. Let $\{\boldsymbol{x}_i\}^N$ be a set of N facial images that correspond to the same identity where N is a variable quantity. The task is to produce a fused feature vector $\boldsymbol{f}$ from $\{\boldsymbol{x}_i\}^N$ that is discriminative for the recognition task. We utilize a pretrained feature extractor E, trained with single image inputs that produce $\boldsymbol{f}_i = E(\boldsymbol{x}_i) \in \mathbb{R}^C$, which is often taken as an embedding vector for recognition. C is the dimension of $\boldsymbol{f}_i$ and it is 512 for conventional FR backbones.

The overview of CAFace is shown in Fig. 3. First, for a given input $\{\boldsymbol{x}_i\}^N$, Style Input Maker (SIM) uses the pretrained feature extractor E to prepare style inputs $\{\boldsymbol{s}_i\}^N$ for individual images. Secondly, Cluster Network takes $\{\boldsymbol{s}_i\}^N$ and learnable centers $\boldsymbol{C} = \{\boldsymbol{c}_j\}^M$ to output an $M \times N$ assignment map $\boldsymbol{A}$, which soft-assigns $\{\boldsymbol{f}_i\}^N$ and $\{\boldsymbol{s}_i\}^N$ to M intermediate outputs $\boldsymbol{F}'$ and $\boldsymbol{S}'$. Lastly, Aggregation Network takes $\boldsymbol{C}, \boldsymbol{F}'$ and $\boldsymbol{S}'$ to produce the fused output $\boldsymbol{f}$. The two-stage architecture is designed to learn a global clustering and aggregation mechanism that can 1) model the intra-set relationship of N inputs efficiently and 2) if N is very large or sequential, split N inputs into batches and still carry over the relationship in-between batches using the intermediate stage. In the sequential case, the intermediate $\boldsymbol{F}'$ and $\boldsymbol{S}'$ can be incrementally updated with a rule that is invariant to batch ordering.

3.1 Architecture

Style Input Maker (SIM). SIM aims to efficiently extract the information related to the quality of the prediction w.r.t. the feature extractor E. The prediction quality of E depends on the input image style which encompasses various image traits such as brightness, contrast, quality, pose, or a domain difference from the training data. However, the embedding, $E(\boldsymbol{x}_i) = \boldsymbol{f}_i$ is optimized to be invariant to any characteristics other than the identity. Thus, we utilize a pseudo-style of an image to capture the prediction quality, following the success of using first and second-order statistics as a style of an image [25, 29, 40]. We also use the feature norm due to the correlation between the feature norm and the confidence of the prediction [26, 39]. Let $\boldsymbol{M}_i \in \mathbb{R}^{C_M \times H \times W}$ be the intermediate feature. Then

$$\boldsymbol{\mu}_{\text{sty}} = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}\boldsymbol{M}_{ihw}, \quad \boldsymbol{\sigma}_{\text{sty}} = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(\boldsymbol{M}_{ihw} - \boldsymbol{\mu}_{\text{sty}})^2}, \quad \boldsymbol{\Gamma} = (\boldsymbol{\mu}_{\text{sty}}, \boldsymbol{\sigma}_{\text{sty}}) \tag{1}$$
$$\boldsymbol{\gamma}_i = \text{BatchNorm}(\text{FC}(\text{ReLU}(\text{AvgPool}(\boldsymbol{W}_s \odot \boldsymbol{\Gamma})))), \quad \text{where} \quad \boldsymbol{\Gamma} \in \mathbb{R}^{C_M \times 2},\ \boldsymbol{\gamma}_i \in \mathbb{R}^{64}.$$

A learnable matrix $\boldsymbol{W}_s \in \mathbb{R}^{C_M \times 2}$ controls the importance of $\boldsymbol{\mu}_{\text{sty}}$ or $\boldsymbol{\sigma}_{\text{sty}}$ during average pooling and $\odot$ is element-wise multiplication. The resulting 64-dim vector $\boldsymbol{\gamma}_i$ captures the image style.

We also note that the simplest feature fusion involves averaging unnormalized embedding vectors $\boldsymbol{f}_i$, which implies that the norm $||\boldsymbol{f}_i||_2$ works as a weight for fusion. To test if the feature norm would be an additional clue for fusion, we embed the feature norm $||\boldsymbol{f}_i||_2$ to a 64-dim vector, following the convention of Sinusoidal conversion [50], which is analogous to the position embeddings in ViT [14].

The norm embedding $\boldsymbol{n}_i$ is a 64-dim vector and the details can be found in Supp.B. Finally, the output SIM is the concatenation, $\boldsymbol{s}_i = [\gamma_i, \boldsymbol{n}_i]$ where $\boldsymbol{s}_i \in \mathbb{R}^d$ and $d = 64 + 64 = 128$.

Cluster Network (CN). The role of Clustering Network is to model the intra-set relationship within a set of N inputs to soft-assign each input to M number of clusters, where M is fixed. Cluster Network consists of L alternating MSA (multi-head self attention) and MLP (multi-layer perceptron) layers and one assignment module. Specifically, consider a set of N embeddings $\{\boldsymbol{f}_i\}^N \in \mathbb{R}^{N\times C}$, outputs of SIM $\{\boldsymbol{s}_i\}^N \in \mathbb{R}^{N\times d}$, and a learnable center matrix $\boldsymbol{C} = \{\boldsymbol{c}_j\}^M \in \mathbb{R}^{M\times d}$. We concatenate $\{\boldsymbol{s}_i\}^N$ and $\{\boldsymbol{c}_j\}^M$ to get input $\boldsymbol{I}_0 = [\{\boldsymbol{s}_i\}^N, \{\boldsymbol{c}_j\}^M] \in \mathbb{R}^{(N+M)\times d}$. Then we perform for L times,

$$\begin{aligned} \boldsymbol{I}'_l &= \text{MSA}(\text{LayerNorm}(\boldsymbol{I}_{l-1})) + \boldsymbol{I}_{l-1}, \quad && l = 1, ..., L \\ \boldsymbol{I}_l &= \text{MLP}(\text{LayerNorm}(\boldsymbol{I}'_l)) + \boldsymbol{I}'_l, && l = 1, ..., L. \\ [\{\boldsymbol{s}_i\}^N_L, \{\boldsymbol{c}_j\}^M_L] &:= \boldsymbol{I}_L. \end{aligned} \tag{2}$$

Note that MSA (Multihead Self Attention) [14] and MLP (Multi Layer Perceptron) are defined as

$$\text{MSA}(\boldsymbol{I}) = \sigma_{\text{row}}\left(\frac{\boldsymbol{I}\boldsymbol{W}_q\,(\boldsymbol{I}\boldsymbol{W}_k)^\intercal}{\sqrt{C}}\right)(\boldsymbol{I}\boldsymbol{W}_v), \quad \text{MLP}(\boldsymbol{I}) = \boldsymbol{W}_1\rho\left(\boldsymbol{W}_2 * \text{LayerNorm}(\boldsymbol{I})\right), \tag{3}$$

for an arbitrary input $\boldsymbol{I} \in \mathbb{R}^{(N+M)\times d}$ (shown here is a single head MSA). Also, σ_{row} is SoftMax normalizing each row of an $(N+M)\times(N+M)$ matrix, ρ is GELU [19] operation, and $\boldsymbol{W}$ is the weight matrix. LayerNorm [3] is inserted for the stability of training.

The output $\{\boldsymbol{s}_i\}^N_L$ is an updated style input which has the chance to exploit the information in the center $\boldsymbol{C}$ and $\{\boldsymbol{s}_i\}^N$. Then we perform the assignment operation with $\boldsymbol{Q} = \{\boldsymbol{c}_j\}^M$ as a query, $\boldsymbol{K} = \{\boldsymbol{s}_i\}^N_L$ as a key and $\boldsymbol{V} = \{\boldsymbol{f}_i\}^N$ or $\{\boldsymbol{s}_i\}^N$ as values. Specifically,

$$\text{ASSIGN}_{N\to M}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{A}\boldsymbol{V}, \quad \boldsymbol{A} = \sigma_{\text{col}}\left(\frac{\boldsymbol{Q}\boldsymbol{W}_q\,(\boldsymbol{K}\boldsymbol{W}_k)^\intercal}{\sqrt{C}}\right). \tag{4}$$

Note that $\boldsymbol{A}$, the assignment map, governs how values will be softly assigned to M different centers. The key difference from MSA is that the normalization in SoftMax, σ_{col}, is for each column of $M\times N$ matrix, assigning each sample $\boldsymbol{f}_i$ to M clusters. And the value $\boldsymbol{f}_i$ is preserved with no projection operation $\boldsymbol{W}_v$. Cluster Network is learning a linear combination of values where the weights are determined by the corresponding style $\boldsymbol{s}_i$. Although ASSIGN is a linear operation, the prediction for the assignment $\boldsymbol{A}$ is a non-linear operation. We denote

$$\boldsymbol{F}' = \text{ASSIGN}(\{\boldsymbol{c}_j\}^M, \{\boldsymbol{s}_i\}^N_L, \{\boldsymbol{f}_i\}^N), \quad \boldsymbol{S}' = \text{ASSIGN}(\{\boldsymbol{c}_j\}^M, \{\boldsymbol{s}_i\}^N_L, \{\boldsymbol{s}_i\}^N). \tag{5}$$

The role of $\boldsymbol{F}' \in \mathbb{R}^{M\times C}$ and $\boldsymbol{S}' \in \mathbb{R}^{M\times d}$ is to provide a summary intermediate representation for the subsequent Aggregation Network. When the set size N is large or inputs are sequential, one would have to divide the inference into multiple steps, and $\boldsymbol{F}'$ and $\boldsymbol{S}'$ serve as a summarization of past inputs that is expressed as M features, one per cluster.

Aggregation Network (AGN). The Aggregation Network is responsible for fusing a fixed number of M inputs, $\boldsymbol{F}'$ and $\boldsymbol{S}'$ into a single fused output $\boldsymbol{f}$ with intra-set relationship. Following the information flow of Cluster Network, we use $\boldsymbol{S}'$ and $\boldsymbol{C}$ to learn the weights $\boldsymbol{P}$ for fusing $\boldsymbol{F}'$, *i.e.*,

$$\begin{aligned} \boldsymbol{P}' &= \underset{d'\to C}{\text{MLP}}\left(\text{MIX}_{M\times d'}\left(\text{MIX}_{M\times d'}\left([\boldsymbol{S}', \boldsymbol{C}]\right)\right)\right), \quad \boldsymbol{P}' \in \mathbb{R}^{M\times C}, \\ \boldsymbol{f} &= \sum_M \boldsymbol{P}\odot\boldsymbol{F}', \quad \boldsymbol{P} = \text{SoftMax}_{\text{row}}(\boldsymbol{P}'), \quad \boldsymbol{f} \in \mathbb{R}^C, \end{aligned} \tag{6}$$

where $d' = 2 * d$, a dimension of concatenated $\boldsymbol{S}'$ and $\boldsymbol{C}$. We adopt MLP-Mixer (MIX) [48], as opposed to MSA, because the information is to be propagated among a fixed number of inputs which can be more efficient with MIX. Specifially, for an arbitrary input $\boldsymbol{O} \in \mathbb{R}^{M\times d'}$ of fixed M, MIX is

$$\text{MIX}_{M\times d'}(\boldsymbol{O}) = \boldsymbol{O}' + \text{MLP}(\boldsymbol{O}'), \quad \boldsymbol{O}' = \boldsymbol{O} + (\text{MLP}(\boldsymbol{O}^\intercal))^\intercal. \tag{7}$$

It can efficiently propagate the information across fixed-sized inputs with channel-wise linear layers. The magnitude of $\boldsymbol{P}$ is an interpretable quantity showing the importance of each cluster during fusion. The final output $\boldsymbol{f}$ is a weighted average of $\boldsymbol{F}'$ whose weight is $\boldsymbol{P}$.

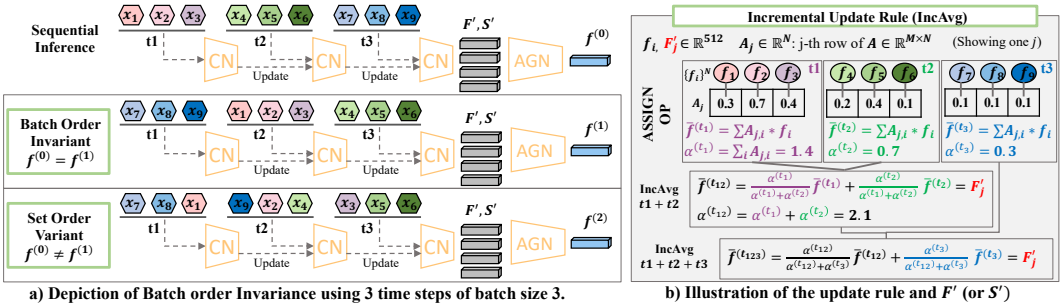


Figure 4: a) An illustration of the batch order invariance property in sequential inference. As long as the elements within the set are the same, the order of batches can be changed. b) An illustration of how $\boldsymbol{F}'$ is updated and remain batch-order invariant. The key is $\boldsymbol{F}'$ being the weighted average of $\{\boldsymbol{f}_i\}$.

Sequential Inference. The key characteristic of the Cluster and Aggregate paradigm is its ability to divide the inputs into T-step sequential inference with incremental averaging. Consider sequential inputs $x_1, x_2, ..., x_T$, with weights $\alpha_1, \alpha_2, ..., \alpha_T$. We can rewrite weighted average, $\overline{x}_T = \frac{1}{T}\sum \alpha_t x_t$ as a recursive function of the past $T-1$ samples' mean and the current value,

$$\overline{x}_t = \text{IncAvg}(\hat{\alpha}_{t-1}, \overline{x}_{t-1}, \alpha_t, x_t) = (\hat{\alpha}_{t-1}\overline{x}_{t-1} + \alpha_t x_t)/(\alpha_{t-1} + \alpha_t), \quad \hat{\alpha}_{t-1} = \sum\nolimits_{l=1}^{t-1} \alpha_l, \quad (8)$$

which requires storing the past samples' mean and the sum of weights. Note that this operation is invariant to sample orders as the final result will always be the mean, $\overline{x}_T$. We can apply the IncAvg operation to Cluster Network (CN) as each cluster assignment is a weighted average of inputs. We define the update rule of two sequential non-overlapping batches $\{x_{i1}\}^{T_1}$ and $\{x_{i2}\}^{T_2}$ as

$$\begin{aligned} \boldsymbol{S}'_{T_1}, \boldsymbol{F}'_{T_1} &= \text{CN}(\{x_{i1}\}^{T_1}), \quad \boldsymbol{S}'_{T_2}, \boldsymbol{F}'_{T_2} = \text{CN}(\{x_{i2}\}^{T_2}) \\ \boldsymbol{S}'_{T_1+T_2} &= \text{IncAvg}\left(\boldsymbol{\alpha}_{T_1}, \boldsymbol{S}'_{T_1}, \boldsymbol{\alpha}_{T_2}, \boldsymbol{S}'_{T_2}\right), \quad \boldsymbol{F}'_{T_1+T_2} = \text{IncAvg}\left(\boldsymbol{\alpha}_{T_1}, \boldsymbol{F}'_{T_1}, \boldsymbol{\alpha}_{T_2}, \boldsymbol{F}'_{T_2}\right), \\ \text{Update}(\text{CN}(\{x_{i1}\}^{T_1}), \text{CN}(\{x_{i2}\}^{T_2})) &= \text{AGN}(\boldsymbol{S}'_{T_1+T_2}, \boldsymbol{F}'_{T_1+T_2}, \boldsymbol{C}), \end{aligned} \quad (9)$$

where $\boldsymbol{\alpha}_{T_1} = \sum_i \boldsymbol{A}_{T_1}$ and $\boldsymbol{\alpha}_{T_2} = \sum_i \boldsymbol{A}_{T_2}$ are the M-dim vectors from the row-sum of assignment map $\boldsymbol{A}$. The assignment map carries the information of how much contribution each $\boldsymbol{f}_i$ is making to each j-th cluster. Then by the property of IncAvg operation, we can obtain the invariance to the order of batch $T_1 \rightarrow T_2$ or $T_2 \rightarrow T_1$, *i.e.*,

$$\text{Update}(\text{CN}(x_1, x_2, x_3), \text{CN}(x_4, x_5)) = \text{Update}(\text{CN}(x_4, x_5), \text{CN}(x_1, x_2, x_3)). \quad (10)$$

Fig. 4 illustrates the batch-order invariance of CN. This is a strong property that allows the influence of inputs to stay the same regardless of the order. Note that we do not achieve order invariance across set element permutation, *i.e.*,

$$\text{Update}(\text{CN}(x_1, x_2, x_3), \text{CN}(x_4, x_5)) \neq \text{Update}(\text{CN}(x_1, x_2), \text{CN}(x_3, x_4, x_5)). \quad (11)$$

This is because Clustering Network (CN) contains MSA layers that propagate information among the given input set. Thus, we further enforce this property with an additional loss term.

3.2 Loss Function

Template Loss. If we let p and q be the index of the identity, the objective of FR is to learn a discriminative identity feature $\boldsymbol{f}^{(p)}$ whose angular displacement with another identity $\boldsymbol{f}^{(q)}$ is large. This problem has been proposed as maximizing the negative pair distance with triplet loss [15]. We pose the problem as learning the class template, *i.e.*, reducing the angular displacement of the fused $\boldsymbol{f}^{(p)}$ to the template feature $\boldsymbol{f}^{(p)}_{GT}$, because the feature extractor E is fixed and it is easier to optimize towards the known target. The task would be viewed as correctly finding the true class center in the presence of low quality features $\{\boldsymbol{f}_i\}^N$. During training, we heavily augment our input data to create low quality features. Thus, the loss term is to increase the cosine similarity between $\boldsymbol{f}^{(p)}$ and $\boldsymbol{f}^{(p)}_{GT}$, *i.e.*, $\mathcal{L}_t = \sum_{p=1}^{B} 1 - \text{CosSim}(\boldsymbol{f}^{(p)}, \boldsymbol{f}^{(p)}_{GT})$. The class centers $\boldsymbol{f}^{(p)}_{GT}$ can be taken from the last FC layer of the pretrained model E or pre-calculated by averaging the embeddings $\boldsymbol{f}_i$ of labeled samples in the training data. In the latter case, we assume that with a sufficient number of good quality training images per identity, the average of $\boldsymbol{f}_i$ will closely approximate the center.

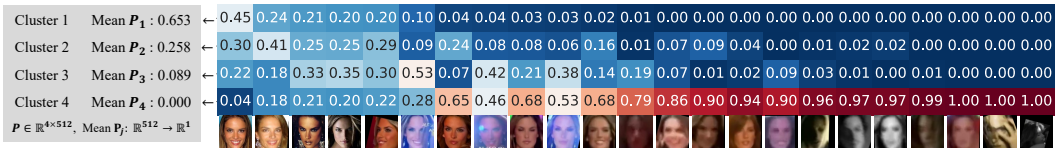


Figure 5: A plot of assignment map $\boldsymbol{A} \in \mathbb{R}^{4\times23}$ (right) and the mean of cluster weights $\boldsymbol{P}$ (left) for samples in IJB-B [53]. For each column in $\boldsymbol{A}$, the values sum up to 1.0. $\boldsymbol{A}$ shows that 1) high quality images are assigned to clusters $1, 2, 3$, with large mean cluster weights $\boldsymbol{P}$; low quality images are assigned to cluster 4 with near 0.0 weight. 2) There are variations among clusters $1, 2, 3$ as to which images have more influence, *e.g.*, cluster 3 focuses on relatively blurred or occluded images.

Set Permutation Consistency Loss. The Cluster And Aggregate paradigm achieves batch-wise order invariance by formulation, but it does not achieve element-wise permutation invariance, as noted in Sec. 3.1. Therefore, we explore the element-wise permutation's added benefit using an additional loss function. Thus, our final loss is $\mathcal{L} = \lambda_c \mathcal{L}_t + \lambda_p \mathcal{L}_p$, and the set permutation consistency loss $\mathcal{L}_p$ is

$$\mathcal{L}_p = 1 - \text{CosSim}\left(\text{AGN}(\text{Update}(\text{CN}(\{x_i\}^{T_1}), \text{CN}(\{x_i\}^{T_2}))), \text{AGN}(\text{CN}(\{x_i\}^{T_1+T_2}))\right). \quad (12)$$

4 Experiments

4.1 Datasets and Implementation Details

We use WebFace4M [59] as our training dataset. It is a large-scale dataset with 4.2M facial images from $205,990$ identities. The fixed feature extractor E has been trained with the whole training dataset. To train the aggregation module, we use its randomly sampled subset, consisting of $813,482$ images from $10,000$ identities. We do not use VGG-2 [7] or MS1MV2 [13, 17] as they were withheld by their distributors due to privacy and other issues.

For the feature extractor E, we use the IResNet-101, trained with ArcFace loss [13]. Since the performance of the aggregation depends on the quality of E, we set E to be the same for *all* experiments. E produces an embedding vector $\boldsymbol{f}_i \in \mathbb{R}^{512}$ for each of N images. To offer variations in the training data features, we randomly augment the dataset with cropping, blurring, and photometric augmentations. The details for training hyper-parameters such as optimizers can be found in Supp.A.

We take IJB-B [53], IJB-C [38] and IJB-S [24] datasets for testing. IJB-B is a widely used FR test set containing both high-quality images and low-quality videos of celebrities (see Fig. 5 for examples). IJB-C is an updated version of IJB-B with more complex motions in the video. IJB-S is a surveillance video dataset, benchmarking extremely low-quality image/video face recognition. The probe and gallery set size can exceed $500,000$. Within the set, there are many low-quality images, making IJB-S very challenging and suitable for measuring the feature fusion framework (see Fig. 6 for examples).

For IJB-S, we use protocols, Surv.-to-Single, Surv.-to-Booking and Surv.-to-Surv. The first/second word in the protocol refers to the probe/gallery image source. Surv. is the surveillance video, 'Single' is the frontal high-quality enrollment image and 'Booking' refers to the 7 high-quality enrollment images. For the ablation study, Sec. 4.2, we report the average of all 9 metrics listed in Tab. 3.

4.2 Ablation and Analysis

Effect of Different Style Input. In Tab. 1, we ablate the efficacy of various components in SIM which prepares the input for CN. The table shows that using $\boldsymbol{f}_i$ for clustering is harmful to performance and increases the no. of parameters for CN. It also shows that using the additional norm embedding $\boldsymbol{n}_i$ along with $\boldsymbol{s}_i$ produces the best results for IJB-B and IJB-S datasets. However, the margin is small, so simply using $\boldsymbol{s}_i$ would suffice for the setting that requires faster computation.

Effect of $\mathcal{L}_p$. As noted in Sec. 3.2, we further propose to constrain the set permutation consistency with the additional loss $\mathcal{L}_p$. The ablation between $\lambda_p = 0$ and $\lambda_p = 1$ is shown in Tab. 1.

Effect of Number of Clusters. In Tab. 1, the effect of the number of clusters M is shown. As we vary M from 1 to 8, the performance on IJB-S peaks when $M = 4$. When $M = 2$, the summary representations $\boldsymbol{F}'$ and $\boldsymbol{S}'$ have only two options for the assignment where one cluster takes the poor quality images with low weight estimation. The behavior could be interpreted as performing an outlier detection, which is powerful enough to give high performance. When $M > 2$, $\boldsymbol{F}'$ and $\boldsymbol{S}'$ have the capacity to store richer history of previous

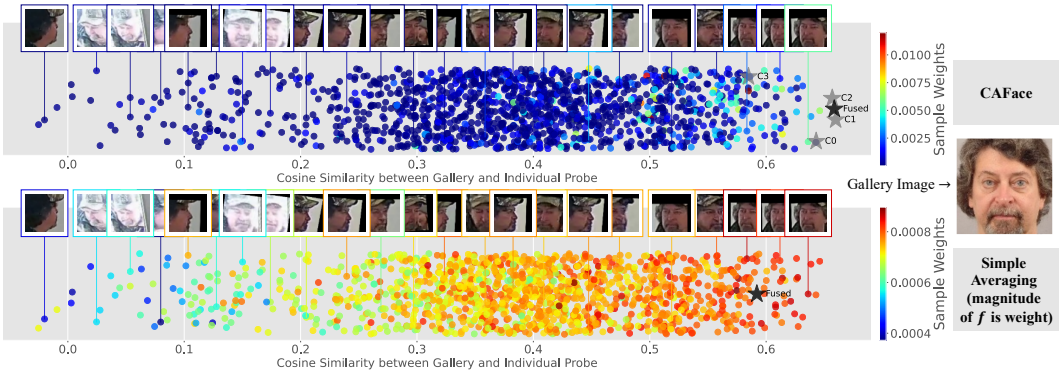


Figure 6: A plot of similarity of fused probe vs gallery. The circles represent individual probe images in IJB-S. The colors represent the weight of each image computed by CAFace (top) and simple averaging (bottom). The black star represents the fused feature. The x-axis is the cosine similarity with the gallery feature. The closer it is to 1, the better the match. Note that with simple averaging, lots of samples far from the gallery are averaged together. In CAFace, only a select few are contributing the most to fusion. The fused output is also closer to the gallery than any other individual probe features.

Table 1: Ablation of varied inputs, loss functions and the number of centers.

$\boldsymbol{f}_i$	$\boldsymbol{s}_i$	$\boldsymbol{n}_i$	# of Centers (M)	$\mathcal{L}_p$	# of Params	IJB-B (TAR@FAR=1e-3)	IJB-B (TAR@FAR=1e-4)	IJB-S (AVG)
✓	✓	✓	4	✓	16.28M	96.11	94.38	53.83
	✓				0.25M	96.81	95.53	57.42
	✓	✓			0.79M	**96.91**	**95.53**	**57.55**
	✓	✓	4	✓	0.79M	**96.91**	**95.53**	**57.55**
	✓	✓				96.86	95.52	57.36
	✓	✓	1	✓	0.7860M	96.10	94.31	53.87
	✓	✓	2		0.7862M	96.88	95.52	57.11
	✓	✓	4		0.7865M	**96.91**	95.53	**57.55**
	✓	✓	8		0.7874M	96.90	**95.61**	57.33
Naive Average Fusion					0	96.10	94.30	54.12

frames which would be beneficial in sequential inference. IJB-S has large N probes which require dividing the inference into batches. Higher IJB-S performance when $M = 4$ indicates that the freedom to assign samples to different clusters is important in the sequential setting. A similar phenomenon is observed in Fig. 7. The performance gap widens for $M = 2$ as we reduce the batch size to make more sequential steps in inference.

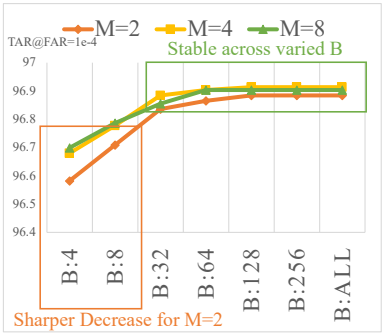


Figure 7: A plot of IJB-B performance of CAFace with varied temporal batch size B for models with different numbers of clusters M.

Weight Visualizations. An example of clustering assignments when $M = 4$ could be viewed in Figs. 5 and 6. Fig. 5 shows how samples are soft-assigned to different clusters along with the weight estimation. The cluster weight is calculated by averaging along 512 dimensions of $\boldsymbol{P} \in \mathbb{R}^{M \times 512}$. Note that each column sums to 1 but $\boldsymbol{F}'$ and $\boldsymbol{S}'$ are calculated by averaging each row. Thus, the relative contribution of samples in each row is important. Fig. 6 shows the actual contribution of each sample during fusion. The contribution can be calculated by multiplying the magnitudes of $\boldsymbol{A}$ and $\boldsymbol{P}$. Note that in the presence of many poor quality images, selecting a few good ones is very important and the sample weight of our method can effectively select a subset of samples during fusion.

4.3 Comparison with SoTA methods

To compare with prior feature aggregation methods, we use the same feature extractor E as in Sec. 4.1, for a fair comparison. Average is the conventional embedding $\boldsymbol{f}_i$ averaging scheme that is adopted in the absence of a learned aggregation model. It is equivalent to the stand-alone ArcFace model performance. The rest of the methods learn an additional network for fusing the set of features.

In Tab. 2, we show the performance of various feature fusion methods on the IJB-B dataset. The table shows that our method can achieve a large gap performance improvement in all TAR@FAR metrics. CFAN [15] and PFE [46] do not use any intra-set relationship, as they are learning to predict

Table 2: A performance comparison of recent methods on the IJB-B [53] dataset.

Method	# of Params	Intra-set Att	Sequential Inference	TAR@FAR=1e-3	TAR@FAR=1e-4	TAR@FAR=1e-5
Average	0	F	T	96.10	94.30	89.53
PFE [46]	13.37M	F	T	96.37	94.82	91.02
CFAN [15]	12.85M	F	T	96.43	94.83	91.10
RSA [31]	2.62M	T	F	96.41	95.00	91.22
CAFace	0.79M	T	T	**96.91**	**95.53**	**92.29**

Table 3: A performance comparison of recent methods on the IJB-S [24] dataset.

Method	Surveillance-to-Single			Surveillance-to-Booking			Surveillance-to-Surveillance		
	Rank-1	Rank-5	1%	Rank-1	Rank-5	1%	Rank-1	Rank-5	1%
Naive Average	69.26	74.31	57.06	70.32	75.16	56.89	32.13	46.67	5.32
PFE [46]	69.50	74.39	57.51	70.53	75.29	57.98	32.27	46.70	5.41
CFAN [15]	70.00	74.58	57.93	70.90	75.58	58.09	31.66	45.59	5.79
RSA [31]	63.04	67.33	51.62	63.54	68.23	51.89	16.82	31.80	0.75
CAFace	**71.61**	**76.43**	**62.21**	**72.72**	**77.41**	**62.68**	**36.51**	**49.59**	**8.78**
CAFace (Random Order)	71.65 ±0.05	76.37 ±0.04	62.27 ±0.11	72.77 ±0.04	77.37 ±0.03	62.70 ±0.06	36.43 ±0.08	49.40 ±0.05	8.89 ±0.03

the confidence of a single image. RSA [31] calculates intra-set attention of feature maps, which is computationally costly and incapable of sequential inference. Proposed CAFace obtains the best results with the least number of parameters. The results for IJB-C show a similar trend and are included in Supp. We also include the comparison of required computation resources in Supp.

In Tab. 3, we compare feature aggregation models in the IJB-S dataset that has large N low quality images/videos in probes. RSA [31] cannot load all images in the probes concurrently for large N. As an alternative, we divide the probes into a manageable batch size of $B = 256$ and average the results. Since RSA does not have a sequential update mechanism, dividing large N probes reduces the performance, which shows why the sequential capacity is important. CAFace also divides the probes image into batch size of $B = 256$ images yet achieves a large margin improvement in IJB-S. It shows that our two-stage mechanism is very effective in the large N setting. Particularly, the performance gain in the hardest protocol, Surveillance to Surveillance, is the largest. We also randomly shuffle images within the probes 5 times and measure the mean and std. of the performance in the last row. The result shows that our model is robust to input ordering. We also include an experiment on a high quality image dataset, IJB-A [28], in Supp, and note that the performance gain with feature fusion is negligible. As noted in Fig. 1 c), the improvement over the baseline (averaging) goes up with the increased number of images in the probe, which highlights the importance of large N scalability.

5 Conclusions

We address the two problems arising from the feature fusion of large N inputs, a common scenario in unconstrained FR. With large N features, modeling intra-set relationships with attention mechanisms are prohibitive due to computational constraints while sequential inference suffers from the reduced contribution of early frames. In this work, we explore the possibility of dividing N inputs into T smaller unordered batches whose inference result is the same as concurrent N inference. To this end, we introduce a two-stage cluster and aggregate paradigm. The clustering stage, inspired by order-invariance of incremental mean operation, is designed to linearly combine N inputs to M global cluster centers, whose assignment is invariant to the batch-order. The aggregation stage efficiently produces a fused output from M clustered features, while utilizing the intra-set relationship. We show our proposed CAFace outperforms baselines on unconstrained face datasets such as IJB-B and IJB-S.

Limitations. Cluster and Aggregate is a feature fusion framework that learns the weights of individual inputs, given a fixed feature extractor E. Weight estimation, in other words, is an *interpolation* among the given set of features which is a double-edged sword, as it gives the interpretability, whilst not being capable of *extrapolation*. Therefore, when the given feature extractor E is sub-optimal, it could be favorable to relax the constraint and let the model extrapolate for better performance.

Potential Negative Societal Impacts. We believe that the machine learning community as a whole should strive to minimize the negative societal impacts. Large scale face recognition training datasets inevitably comprise web-crawled images which are without formal consent, or IRB review. We refrained from using any dataset withdrawn by its creators such as VGG-2 [7] or MS1MV2 [17] to avoid any known copyright issues. We hope that the FR community can collectively move toward collecting datasets with informed consent, fostering R&D without societal concern.

References

[1] Ognjen Arandjelovic, Gregory Shakhnarovich, John Fisher, Roberto Cipolla, and Trevor Darrell. Face recognition with image sets using manifold density divergence. In *CVPR*, 2005.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding. *arXiv preprint arXiv:2102.05095*, 2(3):4, 2021.

[5] Lacey Best-Rowden, Hu Han, Charles Otto, Brendan F Klare, and Anil K Jain. Unconstrained face recognition: Identifying a person of interest from a media collection. *TIFS*, 2014.

[6] Mikhail S Burtsev, Yuri Kuratov, Anton Peganov, and Grigory V Sapunov. Memory transformer. *arXiv preprint arXiv:2006.11527*, 2020.

[7] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. VGGFace2: A dataset for recognising faces across pose and age. In *FG*, 2018.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[9] Hakan Cevikalp and Bill Triggs. Face recognition based on image sets. In *CVPR*, 2010.

[10] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei. Data uncertainty learning in face recognition. In *CVPR*, 2020.

[11] Jun-Cheng Chen, Rajeev Ranjan, Amit Kumar, Ching-Hui Chen, Vishal M Patel, and Rama Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *ICCVW*, 2015.

[12] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

[13] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[15] Sixue Gong, Yichu Shi, Nathan D Kalka, and Anil K Jain. Video face recognition: Component-wise feature aggregation network (C-Fan). In *ICB*, 2019.

[16] Sixue Gong, Yichun Shi, and Anil Jain. Low quality video face recognition: Multi-mode aggregation recurrent network (MARN). In *ICCVW*, 2019.

[17] 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.

[18] Mehrtash T Harandi, Conrad Sanderson, Sareh Shirazi, and Brian C Lovell. Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching. In *CVPR*, 2011.

[19] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GeLUs). *arXiv preprint arXiv:1606.08415*, 2016.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[21] Yiqun Hu, Ajmal S Mian, and Robyn Owens. Sparse approximated nearest points for image set classification. In *CVPR*, 2011.

[22] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled Faces in the Wild: A database forstudying face recognition in unconstrained environments. In *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008.

[23] Zhiwu Huang, Ruiping Wang, Shiguang Shan, Xianqiu Li, and Xilin Chen. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *ICML*, 2015.

[24] Nathan D Kalka, Brianna Maze, James A Duncan, Kevin O'Connor, Stephen Elliott, Kaleb Hebert, Julia Bryan, and Anil K Jain. IJB–S: IARPA Janus Surveillance Video Benchmark. In *BTAS*, 2018.

[25] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[26] Minchul Kim, Anil K Jain, and Xiaoming Liu. AdaFace: Quality adaptive margin for face recognition. In *CVPR*, 2022.

[27] Tae Hyun Kim, Mehdi SM Sajjadi, Michael Hirsch, and Bernhard Scholkopf. Spatio-temporal transformer network for video restoration. In *ECCV*, 2018.

[28] Brendan F Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah, and Anil K Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark-A. In *CVPR*, 2015.

[29] HyunJae Lee, Hyo-Eun Kim, and Hyeonseob Nam. Srm: A style-based recalibration module for convolutional neural networks. In *ICCV*, 2019.

[30] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV*, 2021.

[31] Xiaofeng Liu, Zhenhua Guo, Site Li, Lingsheng Kong, Ping Jia, Jane You, and BVK Kumar. Permutation-invariant feature restructuring for correlation-aware image set-based recognition. In *ICCV*, 2019.

[32] Xiaofeng Liu, BVK Kumar, Chao Yang, Qingming Tang, and Jane You. Dependency-aware attention control for unconstrained face recognition with image sets. In *ECCV*, 2018.

[33] Xiaoming Liu and Tsuhan Chen. Video-based face recognition using adaptive hidden markov models. In *CVPR*, 2003.

[34] Yu Liu, Junjie Yan, and Wanli Ouyang. Quality aware network for set to set recognition. In *CVPR*, 2017.

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[36] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

[37] Jiwen Lu, Gang Wang, and Pierre Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *ICCV*, 2013.

[38] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, and Patrick Grother. IARPA Janus Benchmark-C: Face dataset and protocol. In *ICB*, 2018.

[39] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. MagFace: A universal representation for face recognition and quality assessment. In *CVPR*, 2021.

[40] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *CVPR*, 2021.

[41] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *ICCV*, 2021.

[42] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, 2015.

[43] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

[44] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *WACV*, 2016.

[45] Gregory Shakhnarovich, John W Fisher, and Trevor Darrell. Face recognition from long-term observations. In *ECCV*, 2002.

[46] Yichun Shi and Anil K Jain. Probabilistic face embeddings. In *ICCV*, 2019.

[47] Yi Tay, Vinh Q Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. Transformer memory as a differentiable search index. *arXiv preprint arXiv:2202.06991*, 2022.

[48] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. In *NeurIPS*, 2021.

[49] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[51] Ruiping Wang, Shiguang Shan, Xilin Chen, and Wen Gao. Manifold-manifold distance with application to face recognition based on image set. In *CVPR*, 2008.

[52] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[53] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. IARPA Janus Benchmark-B face dataset. In *CVPRW*, 2017.

[54] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. *arXiv preprint arXiv:2202.11094*, 2022.

[55] Jiaolong Yang, Peiran Ren, Dongqing Zhang, Dong Chen, Fang Wen, Hongdong Li, and Gang Hua. Neural aggregation network for video face recognition. In *CVPR*, 2017.

[56] Meng Yang, Pengfei Zhu, Luc Van Gool, and Lei Zhang. Face recognition based on regularized nearest points between image sets. In *FG*, 2013.

[57] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, 2021.

[58] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *ICCV*, 2021.

[59] Zheng Zhu, Guan Huang, Jiankang Deng, Yun Ye, Junjie Huang, Xinze Chen, Jiagang Zhu, Tian Yang, Jiwen Lu, Dalong Du, et al. WebFace260M: A benchmark unveiling the power of million-scale deep face recognition. In *CVPR*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Sec. 5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Sec. 5
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] The proof for the batch order invariance property is in Sec. 3.1
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Tab. 3
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We discuss the computation efficency and resources needed for running our model in Supp.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] When we used the datasets, we cited the corresponding papers.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] We do not provide any new dataset.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We discuss the ethical issues related to the training data collected from the web in Sec. 5.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] We discuss the copy-right issues in Sec. 5.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We do not use crowdsourcing in our paper.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [No] We dicsuss the difficulty of obtaining consents for large scale face datasets in Sec. 5.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We do not collect any dataset.