# Towards Robust Face Recognition with Comprehensive Search

Manyuan Zhang[1,2], Guanglu Song[2], Yu Liu[2], and Hongsheng Li[1]

[1] Multimedia Laboratory, The Chinese University of Hong Kong
[2] SenseTime Researsch
zhangmanyuan@link.cuhk.edu.hk,songguanglu@sensetime.com,
liuyuisanai@gmail.com,hsli@ee.cuhk.edu.hk

**Abstract.** Data cleaning, architecture, and loss function design are important factors contributing to high-performance face recognition. Previously, the research community tries to improve the performance of each single aspect but failed to present a unified solution on the joint search of the optimal designs for all three aspects. In this paper, we for the first time identify that these aspects are tightly coupled to each other. Optimizing the design of each aspect actually greatly limits the performance and biases the algorithmic design. Specifically, we find that the optimal model architecture or loss function is closely coupled with the data cleaning. To eliminate the bias of single-aspect research and provide an overall understanding of the face recognition model design, we first carefully design the search space for each aspect, then a comprehensive search method is introduced to jointly search optimal data cleaning, architecture, and loss function design. In our framework, we make the proposed comprehensive search as flexible as possible, by using an innovative reinforcement learning based approach. Extensive experiments on million-level face recognition benchmarks demonstrate the effectiveness of our newly-designed search space for each aspect and the comprehensive search. We outperform expert algorithms developed for each single research track by large margins. More importantly, we analyze the difference between our searched optimal design and the independent design of the single factors. We point out that strong models tend to optimize with more difficult training datasets and loss functions. Our empirical study can provide guidance in future research towards more robust face recognition systems.

**Keywords:** Face Recognition, Comprehensive Search

## 1 Introduction

Large-scale face recognition is a fundamental problem and of great practical value in computer vision. It is challenging to learn robust feature representations from million-level datasets. Recently, the vision community has rapidly improved the performance of face recognition. To a large extent, these advances have been driven by three aspects: large-scale noisy data cleaning strategies [9, 34, 2, 1, 13],

**Loss A**

| | Backbone A | Backbone B | Backbone C |
|---|---|---|---|
| C Clean | 0.84 | 0.72 | 0.83 |
| B Clean | 0.79 | 0.75 | 0.79 |
| A Clean | 0.72 | 0.79 | 0.78 |

**Loss B**

| | Backbone A | Backbone B | Backbone C |
|---|---|---|---|
| C Clean | 0.82 | 0.73 | 0.84 |
| B Clean | 0.81 | 0.79 | 0.78 |
| A Clean | 0.75 | 0.85 | 0.78 |

**Loss C**

| | Backbone A | Backbone B | Backbone C |
|---|---|---|---|
| C Clean | 0.8 | 0.68 | 0.83 |
| B Clean | 0.83 | 0.79 | 0.77 |
| A Clean | 0.73 | 0.84 | 0.76 |

**A Clean**

| | Backbone A | Backbone B | Backbone C |
|---|---|---|---|
| Loss C | 0.76 | 0.89 | 0.82 |
| Loss B | 0.77 | 0.72 | 0.81 |
| Loss A | 0.83 | 0.8 | 0.72 |

**B Clean**

| | Backbone A | Backbone B | Backbone C |
|---|---|---|---|
| Loss C | 0.87 | 0.88 | 0.9 |
| Loss B | 0.81 | 0.74 | 0.85 |
| Loss A | 0.82 | 0.75 | 0.78 |

**C Clean**

| | Backbone A | Backbone B | Backbone C |
|---|---|---|---|
| Loss C | 0.71 | 0.72 | 0.9 |
| Loss B | 0.82 | 0.88 | 0.91 |
| Loss A | 0.8 | 0.86 | 0.89 |

**Backbone A**

| | A Clean | B Clean | C Clean |
|---|---|---|---|
| Loss C | 0.75 | 0.75 | 0.88 |
| Loss B | 0.87 | 0.78 | 0.91 |
| Loss A | 0.86 | 0.81 | 0.85 |

**Backbone B**

| | A Clean | B Clean | C Clean |
|---|---|---|---|
| Loss C | 0.73 | 0.72 | 0.92 |
| Loss B | 0.89 | 0.81 | 0.9 |
| Loss A | 0.85 | 0.8 | 0.86 |

**Backbone C**

| | A Clean | B Clean | C Clean |
|---|---|---|---|
| Loss C | 0.76 | 0.73 | 0.89 |
| Loss B | 0.85 | 0.81 | 0.92 |
| Loss A | 0.86 | 0.76 | 0.88 |

Fig. 1: Given a fixed component (referred to the title of each table), we explore the optimal combinations between the other components (horizontal and vertical axes of each table). The optimal design of each component actually changes following the modifications of the other two, which suggests that it is sub-optimal to consider them separately. A, B, C represent different designs for each component. And the value is the validation accuracy.

margin-based loss function formulations [6, 28, 17, 31], and proper architecture designs [4, 38].

Intuitively, large-scale datasets are important for face representation learning. Along with the development of various large-scale datasets [6, 9, 26, 42, 2, 1], extensive works [26, 5, 30] tried to conduct automatic or semi-automatic data cleaning to alleviate the influence of noisy data. In addition, learning robust feature representation with margin-based loss functions [6, 28, 17, 37, 20, 22] can effectively enhance the representations. The advanced architecture design also plays an import role for robust face recognition, including many common backbones [10, 23, 36, 38] and some specially designed backbones [4, 3].

However, there lacks a unified understanding of optimal designs of all the three aspects. Previous methods all delved into one aspect while ignoring the other two, therefore neglecting the coupling between these aspects. For instance, many loss functions [20, 41, 12] rely on hard sample mining to boost face recognition performance. However, this approach would be quite sensitive to noisy data. More specifically, many data cleaning methods [42, 6] leverage the sample-to-class similarity scores to filter out outliers and merge similar classes. They pre-define a confidence threshold and samples with scores lower than the thresh-

old are treated as the noise. Obviously, different thresholds would lead to different loss function designs.

To further evaluate this argument, we conduct extensive experiments to explore the coupling among these individual research aspects. We randomly select some combinations of different cleaning strategies, loss function designs, and backbones. As shown in Figure 1, different combinations lead to significantly different performances. The optimal design of each component actually changes following the modifications of the other two. Separate studies greatly limit the performance and would bias each aspect's algorithmic design.

To eliminate the bias of single-aspect research and provide an overall understanding of the face recognition models, we first propose new design space for each aspect, then deliberately design a comprehensive search method to jointly search data cleaning, architectures, and loss function design. Undoubtedly, the proper design of search space for each aspect will greatly contribute to the comprehensive search result. For the data cleaning, we introduce the innovative sample discriminability-guided data cleaning search strategy to deal with the inter- and inter-class noise. We consider the relationship between a sample with its class centroid and the hardest-negative class to determine whether the sample is a noisy sample or not. It is worth mentioning that the search-based data cleaning approach can effectively alleviate the problem of tangled judgment on whether the edge samples are hard positive or noisy samples. We clean the data simply based on whether the searched cleanup result helps the training model form a more robust representation and whether it can cooperate well with the design of the other two aspects. For the loss function design formulation, we follow the successful margin-based loss function [6, 28, 27], but emphasize the importance of the scales of positive and negative samples, namely the scale-aware margin-based loss function design. As for the backbone design, we optimize the width and depth expansion ratio for the existing base, which have been shown to be the most important two factors in network design [25].

After carefully designing the search space for each aspect, we conduct the comprehensive search to jointly optimize them and explore their collaboration patterns. Obviously, it is hard to simply integrate data cleaning, architectures, and loss function design together. The main barrier is how to effectively design and explore such a complex and huge search space. To solve challenge, given the searchable formulation, we view the comprehensive search as a sequence prediction and generate their hyper-parameters as a sequence of tokens following the order of `data cleaning` $\rightarrow$ `loss function design` $\rightarrow$ `architecture design`. Every prediction is produced by a softmax classifier and then fed into the next time step as input to influence the generation of the next hyper-parameter. We adopt a Recurrent Neural Network (RNN) as the controller to explore the search space. Once the controller RNN finishes generating the data clean strategy, an architecture, and a loss function, a joint solution based on these components is built and trained. The parameters of RNN are then optimized to maximize the expected validation accuracy of the searched components. The proposed

comprehensive search effectively explores the learning of the coupling between different components and enables a unified system.

Although some previous works utilize search-based methods for auto loss function [29, 15] design, to our best knowledge, we are the first to jointly explore all three research fields and try to understand their entanglement. And our proposed scale-aware margin-based loss function can also surpass them substantially for the comparison in loss function design alone. Extensive experiments show the superiority of our newly-proposed search space for each aspect, then the comprehensive search will further outperform former expert algorithms developed for each single research track by large margins. In summary, the main contributions of this paper can be summarized as follows:

- We identify an ignored problem of face recognition that the data cleaning strategy, loss function design, and backbone architecture are coupled to each other. Separate studies on each factor greatly limit their performance.
- We carefully design search space for each aspect and propose the comprehensive search method to jointly explore data cleaning strategy, loss function formulation, and backbone architecture design. We introduce an innovative reinforcement learning based search framework to carry out the research and learns a unified system to achieve robust face recognition.
- We conduct extensive experiments on million-level face recognition tasks and evaluation on various benchmarks, including LFW, SLLFW, CALFW, CPLFW, IJB-B, and IJB-C. It demonstrates the superiority of our search space design for each aspect and the huge performance gained from the integration together by the comprehensive search.

## 2   Related work

### 2.1   Deep Face Recognition

The proposal of large-scale noise-control datasets, strong backbone architectures, and well-designed loss functions have all greatly advanced the face recognition community. From the initial CASIA-Webface [34] to the recent large-scale datasets Glint360K [1] and Webface260M [42], these datasets have been proposed to greatly improve the accuracy of face recognition models. However, these datasets rely on internet search engines and therefore contain a large percentage of noise. How to deal with these noises and learn face recognition with noisy labels has drawn much attention [41, 33, 5, 42, 16]. Zhu [42] introduces the cleaning automatically by self-training (CAST) pipeline, which introduces a self-training process to auto clean inter-class and intra-class noise by an iteration manner and achieves good performance. For the model design, some architectures have been proposed to achieve efficient face recognition, such as PolyNet [38] and MobileFaceNet [4]. For sophisticated face recognition loss function design, efficient margin-based softmax approaches [6, 28, 27, 17, 18] were proposed by modifying the softmax loss function to achieve tighter intra-class compactness and more sparse inter-class separation, which achieve state-of-the-art performance.

Since separate design on each component has achieved good performance, all the above methods ignore the relationship among data cleaning strategy, loss function formulation, and training backbone. Disentangled studies greatly limit performance.

## 2.2  Auto-ML for Face Recognition

Auto-ML methods aim to automatically design suitable machine learning systems. Reinforcement learning guided search [43, 24] automate the design process and can easily outperform manually designed components in a given search space. Some former works use the searching methods to optimize the training loss function for face recognition. Li *et al.* [15] proposes the auto loss function search method from a hyper-parameter optimization perspective. Wang *et al.* [29] develop a unified formulation for the prevalent margin-based softmax loss. Then the random and reward-guided methods are designed to search for the best candidate. The above methods only focus on the loss function search. In this work, we consider all three aspects for face recognition. We first re-design the search space for each aspect, then jointly consider them and achieve excellent performance.

## 3  Method

To eliminate the bias of single-aspect research and provide an overall understanding of the face recognition model design, we propose the comprehensive search method to jointly search optimal data cleaning strategy, loss function formulation, and backbone architecture design for robust face recognition. In this section, we will introduce the details of our comprehensive search. The key components of comprehensive search include three aspects, *i.e.* search space, search objective, and search algorithms. In the following, we will first introduce our newly-designed search space for each aspect, then couple them together to form the comprehensive search space. The search objective to evaluate the searched solution and the innovative reinforcement learning-based reward-guided comprehensive approach to identify the optimal combination design are introduced consequently.

### 3.1  Comprehensive Search Space Design

**Discriminabiliy-guided data cleaning search strategy.** There are two kinds of data noise for face recognition datasets, the 'outliers' and 'label flips', corresponding to intra-class and inter-class noise respectively. An 'outlier' noise means an image does not belong to any class of the datasets. A 'label flip' noise refers to an image that is wrongly labeled with the incorrect class label. To clean the noise, we conduct inter-class filtering and intra-class merging. As for the intra-class noise cleaning, we filter the samples by their *discriminability* [35]. The *discriminability* is defined as:

$$\mathcal{D}_i = \frac{dist_{ip}}{\max\left\{dist_{in} \mid n \in [1, K], n \neq p\right\}},  \tag{1}$$

where $p$ is the positive class label for sample $i$ and $n, n \in [1, K], n \neq p$ is the negative class label. $K$ is the number of class. $dist$ represents the feature similarity. We use the cosine similarity as the similarity indicator here. The *discriminability* is the ratio between the feature's similarity with the centroid of its class and the similarity from the hardest-negative class. It can successfully distinguish the outliers and can be used to filter the samples whose *discriminability* is lower than a pre-defined confidence threshold $\tau_{intra}$. As for the inter-class noise, we simply merge a pair of classes whose class center similarity is higher than the threshold $\tau_{inter}$. The $\tau_{intra}$ and $\tau_{inter}$ are the two hyper-parameters that we need to optimize for the data cleaning strategy.

One of the most difficult problems in past manual or semi-automatic data cleaning pipelines is how to classify edge samples as hard positive or noisy samples. Many loss functions [20, 22] rely on strategies such as hard sample mining to further boost performance. However, edge samples are often hard to distinguish. By conducting the search-based data cleaning, we have eliminated this tangle. The result of data cleaning, *i.e.* whether an edge sample is filtered or not, only depends on whether the cleanup result ultimately helps the training model to form a more robust feature representation, or whether it can cooperate well with the design of the other two aspects.

**Scale-aware margin-based loss function design.** Margin-based softmax loss functions [6, 27, 28, 17] have been proposed in recent years to enhance the feature discrimination for face recognition. In summary, they can be defined in a uniform formulation. Suppose $\boldsymbol{w}_k \in \mathbb{R}^d$ is the $k$-th class's weight ($k \in \{1, 2, \dots, K\}$) and $K$ is the number of classes. $\boldsymbol{x} \in \mathbb{R}^d$ denotes the face feature. The formulation for the margin-based loss function can be written as follows:

$$\mathcal{L} = -\log \frac{e^{s_p f\left(m, \theta_{w_y, x}\right)}}{e^{s_p f\left(m, \theta_{w_y, x}\right)} + \sum_{k \neq y}^{K} e^{s_n \cos\left(\theta_{w_k, x}\right)}},  \tag{2}$$

where $\theta_{\boldsymbol{w}_k, \boldsymbol{x}}$ is the angle between $\boldsymbol{w}_k$ and $\boldsymbol{x}$. Suppose $\cos\left(\theta_{\boldsymbol{w}_k, \boldsymbol{x}}\right) = \boldsymbol{w}_k^T \boldsymbol{x}$ is the cosine similarity of the $k$-th class weight and feature, $f\left(m, \theta_{\boldsymbol{w}_y, \boldsymbol{x}}\right) \leq \cos\left(\theta_{\boldsymbol{w}_y, \boldsymbol{x}}\right)$ is the carefully designed margin function and can be summarized into the combined version $f\left(m, \theta_{\boldsymbol{w}_y, \boldsymbol{x}}\right) = \cos\left(m_1 \theta_{\boldsymbol{w}_y, \boldsymbol{x}} + m_2\right) - m_3$. $m_1$ and $m_2$ are the multiplicative and additive angular margin, and $m_3$ is the additive cosine margin. Previous work [6, 28, 27] has mainly focused on the margin optimization, but ignored the importance of tuning the positive sample scale $s_p$ and the negative sample scale $s_n$, which control the optimization difficulty of positive and negative sample. We experimentally prove that face recognition accuracy can be further hugely improved by optimizating the $s_p$ and $s_n$ jointly. Overall, $m_1, m_2, m_3, s_p$ and $s_n$ are the hyper-parameters we need to search for the loss function.
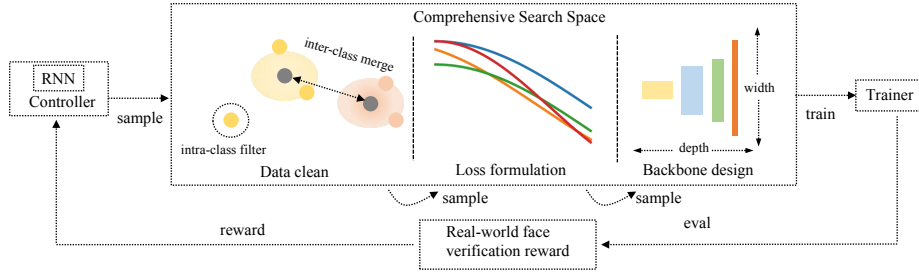
Fig. 2: The overall pipeline for our innovative reinforcement learning-based comprehensive search for robust face recognition. We design it for the sample-eval-update loop. Firstly, the RL agent samples a batch of combinations that contains the hyper-parameters for data cleaning strategy, loss function formulation, and backbone architecture design. For each sampled combination, we train it on the target face recognition task to get its accuracy and reward. After that, the agent will be updated by maximizing the expected reward until converged.

**Effective depth and width search for the backbone.** Following [25], the width expansion ratio $\mathcal{W}$ and the depth expansion ratio $\mathcal{D}$ are the two most important factors for neural network architecture design, so we utilize those two factors to search backbone architecture for face recognition. The base model we selected is modified MobileNet [11], which has been proved successful for face recognition and its search cost is affordable.

### 3.2    Comprehensive Search Objective and Algorithm

Joint search with data cleaning strategy, loss function design, and architecture is not easy to perform due to the complex and huge search space. To enable fast and flexible comprehensive search, we view the comprehensive search as a sequence prediction and generate their hyper-parameters as a sequence of tokens following the order of `data cleaning` $\rightarrow$ `loss function design` $\rightarrow$ `architecture design`. Every prediction of each step is produced by a softmax classifier and then fed into the next time step as input to influence the generation of the next hyper-parameter. We adopt the Recurrent Neural Network (RNN) as the controller to generate the search parameters. Once the controller has completed generation of the data clean strategy, the loss function, and the architecture, a joint solution based on these components is built and trained. The parameters of RNN are optimized to maximize the expected validation accuracy of the searched components. The overall searching process can be optimized with the sample-eval-update loop as shown in Alg. 1 and Figure 2.

To solve the real-world face recognition problem, we test all candidate joint solutions and use the performance as the search criterion. Another object that needs to be considered is the computational cost, since we search for the depth

---

**Algorithm 1** Comprehensive Search

---

**Input:** Training set $\mathcal{S}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, validation set $\mathcal{S}_v$, total searching epochs $T$, agent policy $\pi_\theta$ and its initialized parameter $\theta_0$.

   **for** $t = 1$ to $T$ do **do**

      Sample $B$ combinations of hyper-parameters $\boldsymbol{C}_1, \dots \boldsymbol{C}_B$ for data cleaning strategy $\tau_{intra}$, $\tau_{inter}$, loss function formulation $m_1, m_2, m_3, s_p, s_n$ and backbone architecture design $\mathcal{W}$ and $\mathcal{D}$ via RNN sequencely.

      Train the combinations $\boldsymbol{C}_1, \dots \boldsymbol{C}_B$ for one epoch separately on the training set $\mathcal{S}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with the sampled hyper-parameters.

      Evaluate the trained combinations $\boldsymbol{C}_1, \dots \boldsymbol{C}_B$ on the validation set $\mathcal{S}_v$ to get reward $\mathrm{R}(\boldsymbol{C}_1), \dots \mathrm{R}(\boldsymbol{C}_B)$ via Eq. 3.

      Update $\theta_t$ by $\max \mathbb{E}_P \mathrm{R}(\boldsymbol{C})$

      Update $\pi_{\theta_{t+1}} = \pi_{\theta_t}$

   **end for**

**Output:** Final policy $\pi_\theta$

---

expansion ratio and width expansion ratio for the backbone, we expect the new backbone to have a similar computational budget with the original backbone. To this end, we use a weighted product method to approximate Pareto optimal solutions. The final reward function can be formulated as:

$$\mathrm{R}(\boldsymbol{C}) = \mathrm{ACC}(\boldsymbol{C}) \times \left[ \frac{\mathrm{COST}(\boldsymbol{C})}{\mathrm{TAR}} \right]^\alpha \tag{3}$$

where $\boldsymbol{C}$ is the sampled combination of designs of the three aspects, $\mathrm{ACC}(\boldsymbol{C})$ is the accuracy on real-world face verification task of the sampled combination, $\mathrm{COST}(\boldsymbol{C})$ is the computation cost (FLOPs) of the combination, TAR is the target computation cost which we set it to the original backbone, and $\alpha$ is the weight factor.

Following [24], we use Proximal Policy Optimization (PPO) [21] to optimize the RL agent to find Pareto optimal solutions for our comprehensive search problem. For each sampled combination in the search space, we map it to a list of tokens, which are determined by a sequence of action $a_{1:T}$ from the RL agent with policy $\pi_\theta$. The overall objective is to maximize the expected reward:

$$\mathbb{J} = \mathbb{E}_{P(a_{1:T};m)} \mathrm{R}(\boldsymbol{C}) \tag{4}$$

For each sample $\boldsymbol{C}$, we train it on the target task to get its accuracy $\mathrm{ACC}(\boldsymbol{C})$ and its computation cost $\mathrm{COST}(\boldsymbol{C})$. We then calculate the reward $\mathrm{R}(\boldsymbol{C})$ using Eq. 3. After that, the agent with policy $\pi_\theta$ will be updated by maximizing the expected reward in Eq. 4 until the parameter $\theta$ converges. Then we will re-train the top combinations $\boldsymbol{C}$ with the highest $\mathrm{R}(\boldsymbol{C})$ by full train and select the best combination as our searched results.

## 4   Experiment

### 4.1   Datasets

**Training Data.** To conduct the comprehensive search for robust face recognition, we use the MS1MV2 [6] as our training set, which contains 5.8 million face images from 85K identities. To prove the good generation ability of our comprehensive search, we also experiment on the recently introduced large-scale dataset Glint360K [1], which contains 17 million images of 360K individuals. Note that the MS1MV2 and Glint360K are the two largest publicly available face recognition datasets.

**Test Data.** During the sample-eval-update loop, we evaluate our combination on the re-organised MegaFace [14] verification benchmark. After retraining, we test the final model at most popular face benchmarks including LFW [13], CALFW [40], SLLFW [8], CPLFW [39], IJB-B [32] and IJB-C [19]. We follow the unrestricted with labelled outside data protocol [13].

### 4.2   Implementation Details

**Data Processing.** All the faces in the training images are detected by RetinaFace [7]. Alignment by five landmarks is conducted and the face is cropped to $112\times112$. Images are normalized by subtracting 127.5 and dividing by 128. For the data cleaning strategy, we use the ResNet-50 [10] model trained on MS1MV2 to extract feature to conduct intra-class and inter-class cleaning.

**Searching.** During the searching process, each sampled solution is trained on the training set and evaluated on the validation set. The weighted accuracy of TAR@FAR at $10^{-3}$, $10^{-4}$, $10^{-5}$ by 0.5, 0.25 and 0.25 are used as the ACC. After that, we re-train the top reward solution. We choose the top 20 solutions under the computation budget and fully train them on the target task and then report the final performance on popular face benchmarks. A recurrent neural network (RNN) is utilized as the controller to generate the parameter combination. We use Adam optimizer with learning rate of $5\times10^{-4}$ and momentum 0.9 to update the controller. For the sampled solution's training, we only train each solution for one epoch. The search process needs around 1,000 samples to converge and costs around 37 GPU days (NVIDIA A100, FP16 training).

**Retraining.** After choosing the top 20 combinations, we fully retrain them. We use SGD optimizer with weight decay $5\times10^{-4}$ and momentum 0.9. We train models on 8 NVIDIA GPUs with batch size 1024 for 100K iterations. The initial learning rate is set to 0.1 and decays by 0.1 at iterations 40K, 60K, and 80K for MS1MV2. As for Glint360K, we train for batch size 1024 with 150k iterations totally and learning rate decay at 60k, 90k, and 13k.

### 4.3   Results

Table 1: Verification performance (%) of different search space combinations on the LFW, SLLFW, CALFW, CPLFW, SLLFW, IJB-B and IJB-C benchmarks. The last row represents the baseline performance, which combines the best previous hand-crafted design for each track. The training set is MS1MV2.

| Search Space | | | LFW | SLLFW | CALFW | CPLFW | IJB-B | | IJB-C | |
| Data | Loss | Backbone | | | | | $10^{-5}$ | $10^{-4}$ | $10^{-5}$ | $10^{-4}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | ✓ | ✓ | 99.55 | **98.78** | **94.75** | **84.80** | **85.66** | **91.46** | **90.45** | **93.58** |
| ✗ | ✓ | ✓ | **99.58** | 98.43 | 94.15 | 84.33 | 83.67 | 91.20 | 89.55 | 93.24 |
| ✓ | ✗ | ✗ | 99.43 | 98.15 | 94.00 | 82.48 | 82.35 | 90.72 | 88.47 | 92.84 |
| ✗ | ✓ | ✗ | 99.52 | 98.27 | 94.03 | 84.30 | 83.26 | 90.80 | 88.88 | 92.91 |
| ✗ | ✗ | ✓ | 99.35 | 98.22 | 94.23 | 82.78 | 80.62 | 90.40 | 87.92 | 92.70 |
| ✗ | ✗ | ✗ | 99.43 | 98.12 | 94.03 | 82.30 | 79.71 | 90.44 | 87.29 | 92.61 |

We test our comprehensive search method on widely-used face verification benchmarks including LFW, SLLFW, CALFW, CPLFW, IJB-B, and IJB-C. The results are shown in Table 1. The bold numbers in each column represent the best results. In Table 1, the last line represents the baseline results without any search-based design component. For the baseline, the training dataset is semi-automatically cleaned by [6], the loss func-

| Best combination | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\tau_{intra}$ | $\tau_{inter}$ | $m_1$ | $m_2$ | $m_3$ | $s_p$ | $s_n$ | $\mathcal{D}$ | $\mathcal{W}$ |
| 0.3 | 0.62 | 1.15 | 0.22 | 0 | 40 | 48 | 1.47 | 0.84 |
| 0.22 | 0.76 | 1.0 | 0.32 | 0 | 40 | 40 | 1.22 | 0.91 |
| 0.26 | 0.62 | 1.2 | 0.36 | 0 | 32 | 32 | 1.47 | 0.84 |

Table 2: The best combinations searched with our comprehensive search on MS1MV2 dataset.

tion is finely designed margin-based loss function ArcFace [6], and the backbone architecture is modified MobileNet [11] that is specially designed for face recognition. From the results, we can see that our comprehensive search outperforms the baseline by a large margin on all test benchmarks, *i.e.* 0.97% performance gain for IJB-C at 1e-4 and 3.16% at 1e-5. The huge improvement demonstrates the great potential of comprehensive search compared to manually design components separately. What's more, the search for each component also boosts the performance greatly, which shows the excellent and flexible design of our newly-designed search space for each component and the limitation of hand-crafted design. From the result of row 1 and rows 3-5 in Table 1, we can also observe that searching for each component separately may not achieve the best performance. If we comprehensively search for all three aspects jointly, we can further boost the performance significantly. Comparing rows 1 and 2 in Table 1, the addition of search for data cleaning strategy improves the search result dramatically, which has been ignored in previous searching methods [15, 29]. We show some combinations of top verification accuracy we searched on MS1MV2 in Table 2. The results show a very different design preference of searching from the previous manual design for each component separately.
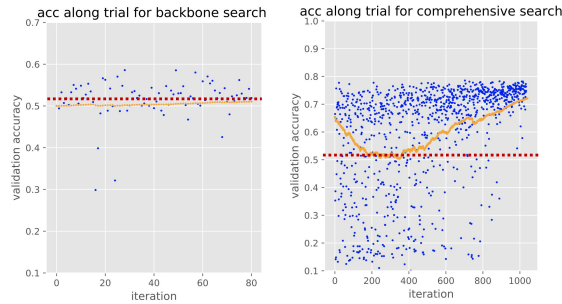
Fig. 3: Visualization of the accuracy on validation set along the search process. **Left**: The accuracy for searching backbone architecture only. **Right**: The accuracy for comprehensive search. The yellow line represents the moving average of all sampled combinations, and the red dot line indicates the baseline.

### 4.4    Searching Process

We visualize the searching process in Figure 3. The yellow line represents the moving average of the accuracy of the sampled solution. From the figure, we can observe that the validation accuracy has been improved gradually through the searching, which suggests that the agent has learned a robust policy that can sample high-quality hyper-parameter combinations. We also compare the searching process for comprehensive search and backbone architecture search in Figure 3. From the figure, we can also see that by enhancing with flexible data cleaning strategy and loss function formulation searching, the validation accuracy of sampled combination has been improved greatly, which confirms the validity of our comprehensive search. In addition, all the rewards of comprehensive and backbone search outperform the baseline combination (represented by the red dot line), which shows the effectiveness of searching.

### 4.5    Ablation study

**Effect of *discriminabiliy*-guided data cleaning strategy.** For the data cleaning strategy, we propose the *discriminabiliy*-guided data intra-class filtering and inter-class merging according to class center similarity to determine 'outlier' and 'label flip' samples respectively. From the results of rows 3 and 6 in Table 1, we can observe that only searching for data cleaning strategy can outperform baseline whose dataset is semi-automatically cleaned hugely, and achieve performance gains of 1.18% at $10^{-5}$ on IJB-C and 2.64% at $10^{-5}$ on IJB-B benchmark. Note that the result is customized, as we have stated before, the best candidate for cleaning hyper-parameters is highly related to the loss function formulation and backbone architecture design.

Table 4: Verification performance (%) of different search space combinations on LFW, SLLFW, CALFW, CPLFW, SLLFW, IJB-B and IJB-C benchmarks. The last row represents the baseline performance, which combines the best previous hand-crafted design for each track. The training set is Glint360K.

| Search Space | | | LFW | SLLFW | CALFW | CPLFW | IJB-B | | IJB-C | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data | Loss | Backbone | | | | | $10^{-5}$ | $10^{-4}$ | $10^{-5}$ | $10^{-4}$ |
| ✓ | ✓ | ✓ | **99.63** | **98.75** | **94.28** | **86.20** | **86.06** | **92.55** | **90.06** | **94.36** |
| × | × | ✓ | 99.38 | 98.11 | 94.07 | 85.10 | 80.96 | 91.13 | 86.68 | 93.25 |
| × | × | × | 99.34 | 98.05 | 93.83 | 84.35 | 79.83 | 90.92 | 86.33 | 93.16 |

**Effect of cleaning feature extraction model.** In the cleaning process, we rely on the feature extracted by a pre-trained model to calculate *discriminabiliy* and inter-class similarity. Intuitively, the more discriminate features would lead to better cleaning accuracy. We conduct an ablation study to study the influence. We use the ResNet-50 backbone trained on MS1MV2 and Glint360K to extract features and the results are shown in Table 3. The better features would lead to better cleaning accuracy, but the impact is quite limited.

| Train Dataset | IJB-C | |
|---|---|---|
| | $10^{-5}$ | $10^{-4}$ |
| MS1MV2 | 88.80 | 92.88 |
| Glint360K | **88.84** | **92.92** |

Table 3: Results of different feature extracted model.

**Effect of scale-aware margin-based loss function.** For the loss function design, we formulate it as the scale-aware margin-based loss function. The results of rows 4 and 6 in Table 1 demonstrate that the search of loss function can improve the performance significantly compared to the widely used hand-crafted design loss ArcFace, *i.e.* 1.59% gain at IJB-C $10^{-5}$. Furthermore, compared to the other two components data and backbone, the separate searching of loss achieves the largest improvement, which demonstrates that sophisticated loss function design is crucial for learning discriminate feature representation.

**Effect of depth and width search.** For backbone architecture design, we only search for the width expansion ratio $\mathcal{W}$ and the depth expansion ratio $\mathcal{D}$. We constrain the new backbone to share the same FLOPs with the base model. From Table 1 rows 5 and 6, the simple width and depth search can still boost performance, which indicates the limitation of hand-crafted backbone design.

### 4.6   Search on Glint360K

To show the good generalizability of our comprehensive search, we also conduct experiments on the newly introduced large-scale dataset Glint360K [1]. The results are shown in Table 4. From the table, we can see that the comprehensive search improves the performance significantly on all face benchmarks, which demonstrates good generalizability. Our solution improves 3.73% at $10^{-5}$ for IJB-C and 6.23% at $10^{-5}$ for IJB-B respectively. The main reason is that our

Table 5: Verification performance (%) of dataset transferability experiment on IJB-B and IJB-C.

| Search on | Train on | IJB-B | | IJB-C | |
|---|---|---|---|---|---|
| | | $10^{-5}$ | $10^{-4}$ | $10^{-5}$ | $10^{-4}$ |
| Glint360K | MS1MV2 | **84.26** | **91.12** | **89.41** | **93.18** |
| MS1MV2 | MS1MV2 | 85.66 | 91.46 | 90.45 | 93.58 |
| - | MS1MV2 | 79.71 | 90.44 | 87.29 | 92.61 |

Table 6: Verification performance (%) of comprehensive search for ResNet50 on IJB-B and IJB-C benchmarks with MS1MV2. The last row represents the baseline performance.

| Transfer | | IJB-B | | IJB-C | |
|---|---|---|---|---|---|
| Data | Loss | $10^{-5}$ | $10^{-4}$ | $10^{-5}$ | $10^{-4}$ |
| ✓ | ✓ | **90.11** | **94.55** | **94.11** | **96.04** |
| × | × | 89.11 | 94.24 | 93.68 | 95.76 |

proposed comprehensive search can effectively capture the intrinsic connections among the different aspects and find the best combinations. Moreover, the best combinations in Glint360K of the three aspects are different from the results searched on MS1MV2, which validates our observation that the three aspects are coupled to each other.

### 4.7 Transferability

Since the three aspects are highly related and the best combination would be changed for different training datasets, we explore the transferability of our comprehensive search in this subsection. We use the best combination searched on Glint360K to train models on MS1MV2. Results are shown in Table 5. The results show that even the transfer learning reduces performance gains, we still achieve significant performance improvements compared to the baseline. The result also suggests that it is better to search and test with the same dataset and the best candidate of one component would be changed along with the other two.

### 4.8 Search with larger backbone

In the above experiments, we have performed the search based on the modified MobileNet whose FLOPs is around 0.33G. In this section, we perform a joint search based on ResNet50 whose FLOPs is around 6G. We search for data cleaning strategy and loss function, and the results are shown in Table 6. The performance improvement verifies the generalization of our joint search.

### 4.9 Discussion

In order to explore the intrinsic relationship among these three aspects, we analyse the results of the search to find the best combination pattern. To facilitate the analysis, we defined the difficulty of the data and the difficulty of the loss function separately. For data cleaning, a lower $\tau_{intra}$ as well as a higher $\tau_{inter}$
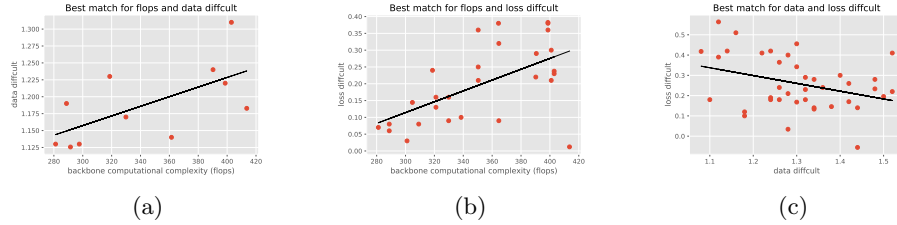
Fig. 4: Visualization of best matches among data difficult, loss difficult and backbone computational complexity (FLOPs).

implies greater training difficulty. For loss function, stricter margins and larger $S_n/S_p$ imply more difficult optimizations. They are defined as follows,

$$\text{Difficulty}_{\text{data}} = 1 - \tau_{\text{intra}} + \tau_{\text{inter}}$$
$$\text{Difficulty}_{\text{loss}} = \frac{s_n * (m_1 - 1 + m_2 + m_3)}{s_p} \tag{5}$$

We visualize the relationship between data difficulty, loss difficulty, and backbone architecture in Figure 4. In Figure (4a) and (4b), larger models (higher FLOPs) tend to train with greater data difficulty and loss difficulty. A larger model implies a stronger fitting ability and is, therefore, able to handle more complex optimization problems. So it can learn more from difficult samples and strict optimization objectives. For the best matching of loss function and data cleaning, as Figure (4c) suggests, models tend to choose easier optimization loss functions under severe data. These findings provide a fresh perspective for the design of face recognition pipeline.

## 5    Conclusion

In this work, we explore the relationship among the data cleaning strategy, loss function formulation, and backbone architecture design for robust face recognition. Previously, people optimize them separately but fail to present a unified understanding of integrating them. We provide a fresh perspective, coupling them and optimizing jointly. We propose an innovative reinforcement learning-based comprehensive search for the best combination. Extensive experiments have proven the excellence of our method for robust face recognition.

## Acknowledgement

# References

1. An, X., Zhu, X., Gao, Y., Xiao, Y., Zhao, Y., Feng, Z., Wu, L., Qin, B., Zhang, M., Zhang, D., et al.: Partial fc: Training 10 million identities on a single machine. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1445–1449 (2021)
2. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: A dataset for recognising faces across pose and age. In: 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018). pp. 67–74. IEEE (2018)
3. Chang, W.Y., Tsai, M.Y., Lo, S.C.: Ressanet: A hybrid backbone of residual block and self-attention module for masked face recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1468–1476 (2021)
4. Chen, S., Liu, Y., Gao, X., Han, Z.: Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In: Chinese Conference on Biometric Recognition. pp. 428–438. Springer (2018)
5. Deng, J., Guo, J., Liu, T., Gong, M., Zafeiriou, S.: Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In: European Conference on Computer Vision. pp. 741–757. Springer (2020)
6. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2019)
7. Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I., Zafeiriou, S.: Retinaface: Single-stage dense face localisation in the wild. arXiv preprint arXiv:1905.00641 (2019)
8. Deng, W., Hu, J., Zhang, N., Chen, B., Guo, J.: Fine-grained face verification: Fglfw database, baselines, and human-dcmn partnership. Pattern Recognition **66**, 63–73 (2017)
9. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: European conference on computer vision. pp. 87–102. Springer (2016)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
12. Hu, W., Huang, Y., Zhang, F., Li, R.: Noise-tolerant paradigm for training face recognition cnns. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11887–11896 (2019)
13. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: 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)
14. Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E.: The megaface benchmark: 1 million faces for recognition at scale. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4873–4882 (2016)
15. Li, C., Yuan, X., Lin, C., Guo, M., Wu, W., Yan, J., Ouyang, W.: Am-lfs: Automl for loss function search. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8410–8419 (2019)
16. Liu, B., Song, G., Zhang, M., You, H., Liu, Y.: Switchable k-class hyperplanes for noise-robust representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3019–3028 (2021)

17. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 212–220 (2017)
18. Liu, Y., et al.: Towards flops-constrained face recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
19. Maze, B., Adams, J., Duncan, J.A., Kalka, N., Miller, T., Otto, C., Jain, A.K., Niggel, W.T., Anderson, J., Cheney, J., et al.: Iarpa janus benchmark-c: Face dataset and protocol. In: 2018 International Conference on Biometrics (ICB). pp. 158–165. IEEE (2018)
20. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
21. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
22. Sun, Y., Wang, X., Tang, X.: Deeply learned face representations are sparse, selective, and robust. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2892–2900 (2015)
23. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
24. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2820–2828 (2019)
25. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. pp. 6105–6114. PMLR (2019)
26. Wang, F., Chen, L., Li, C., Huang, S., Chen, Y., Qian, C., Loy, C.C.: The devil of face recognition is in the noise. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 765–780 (2018)
27. Wang, F., Cheng, J., Liu, W., Liu, H.: Additive margin softmax for face verification. IEEE Signal Processing Letters **25**(7), 926–930 (2018)
28. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5265–5274 (2018)
29. Wang, X., Wang, S., Chi, C., Zhang, S., Mei, T.: Loss function search for face recognition. In: International Conference on Machine Learning. pp. 10029–10038. PMLR (2020)
30. Wang, X., Wang, S., Wang, J., Shi, H., Mei, T.: Co-mining: Deep face recognition with noisy labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9358–9367 (2019)
31. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: European conference on computer vision. pp. 499–515. Springer (2016)
32. Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., Kalka, N., Jain, A.K., Duncan, J.A., Allen, K., et al.: Iarpa janus benchmark-b face dataset. In: proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 90–98 (2017)

33. Wu, X., He, R., Sun, Z., Tan, T.: A light cnn for deep face representation with noisy labels. IEEE Transactions on Information Forensics and Security **13**(11), 2884–2896 (2018)
34. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv preprint arXiv:1411.7923 (2014)
35. Zhang, M., Song, G., Zhou, H., Liu, Y.: Discriminability distillation in group representation learning. In: European Conference on Computer Vision. pp. 1–19. Springer (2020)
36. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6848–6856 (2018)
37. Zhang, X., Zhao, R., Qiao, Y., Wang, X., Li, H.: Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10823–10832 (2019)
38. Zhang, X., Li, Z., Change Loy, C., Lin, D.: Polynet: A pursuit of structural diversity in very deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 718–726 (2017)
39. Zheng, T., Deng, W.: Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. Beijing University of Posts and Telecommunications, Tech. Rep **5**, 7 (2018)
40. Zheng, T., Deng, W., Hu, J.: Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. arXiv preprint arXiv:1708.08197 (2017)
41. Zhong, Y., Deng, W., Wang, M., Hu, J., Peng, J., Tao, X., Huang, Y.: Unequal-training for deep face recognition with long-tailed noisy data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7812–7821 (2019)
42. Zhu, Z., Huang, G., Deng, J., Ye, Y., Huang, J., Chen, X., Zhu, J., Yang, T., Lu, J., Du, D., et al.: Webface260m: A benchmark unveiling the power of million-scale deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10492–10502 (2021)
43. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 (2016)