Iranian Journal of Operations Research
Vol. 17, No. 2, 2026, pp. 31-55

A Neuro-Symbolic Framework for Face Recognition Using a SOAR-Based Cognitive Meta-Controller, Parameter-Efficient Fine-Tuning (PEFT), and Test-Time Adaptation (TTA)

Z.M. Jalali[1], A. Nourbakhsh[2,*], S. Motamed[3]

Face recognition in real-world environments remains challenging due to domain shifts, limited labeled data, decision uncertainty, and constrained computational resources. This paper presents a novel neuro-symbolic framework that combines a SOAR-based cognitive meta-controller, cost-aware reinforcement learning, and domain adaptation to jointly improve recognition accuracy, computational efficiency, and inference latency. A face embedding network is first trained on the source domain and then adapted to the target domain using parameter-efficient fine-tuning (PEFT) and domain-adversarial neural network (DANN). To enhance robustness under limited supervision and distribution shifts, the framework further incorporates few-shot learning and test-time adaptation (TTA). The proposed meta-controller is formulated as a Markov decision process (MDP), enabling dynamic allocation of computational resources based on image quality, uncertainty estimation, and recognition status. Symbolic knowledge encoded in the SOAR architecture guides the reinforcement learning policy through a neuro-symbolic gating mechanism, improving both interpretability and decision consistency. Experiments on the IJB-C and MegaFace benchmarks demonstrate significant improvements in face recognition and open-set recognition performance, while PEFT reduces trainable parameters by more than 95% without increasing inference latency. Overall, the proposed framework offers an accurate, adaptive, interpretable, and computationally efficient solution for real-world face recognition applications.

Keywords: *Face Recognition, Neuro-Symbolic Artificial Intelligence, SOAR Cognitive Architecture, Parameter-Efficient Fine-Tuning (PEFT), Test-Time Adaptation (TTA), Reinforcement Learning.*

1. Introduction

In recent years, face recognition has made significant advances, driven by the rapid development of deep learning techniques, particularly methods that focus on learning effective feature representations. Among these approaches, FaceNet introduced a powerful framework for learning a meaningful similarity space through the triplet loss function, enabling accurate face verification, identification, and clustering based on feature embeddings. Schroff et al. [28]. Later, ArcFace further enhanced the discriminative capability of face recognition models by introducing an additive angular margin loss, becoming one of the most widely used approaches in modern face recognition systems. Deng et al. [3]. These advancements have greatly expanded the use of face recognition technology

* Corresponding Author. (Nourbakhsh@iau.ac.ir)
[1]Department of Computer Engineering, Ra.C., Islamic Azad University, Rasht, Iran, Email: zinat.mehdidoust@iau.ac.ir.
[2]Department of Computer Engineering, La.C., Islamic Azad University, Lahijan, Iran, Email: Nourbakhsh@iau.ac.ir.
[3]Department of Computer Engineering, FSh.C., Islamic Azad University, Fouman, Iran, Email: sara.motamed@iau.ac.ir.

across various real-world applications, including digital authentication, access control, intelligent surveillance, and security systems.

Despite these advances, face recognition systems still encounter significant challenges when deployed in real-world environments. Variations in lighting conditions, head poses, image quality, partial occlusions caused by masks or glasses, as well as the domain gap between the training domain and real-world target domains, can substantially affect the performance of deep learning models. Furthermore, practical applications often require these systems to recognize or reject individuals who were not included in the training data. In such open-world settings, decisions based solely on Softmax scores or fixed thresholds are not sufficient to reliably handle uncertainty, which can lead to incorrect recognition of unknown identities. Bendale and Boult [1], Galvan et al. [5].

For this reason, recent evaluations of face recognition systems have gradually shifted from relatively controlled datasets, such as LFW, toward more challenging benchmarks such as IJB-C. These datasets represent more realistic conditions by incorporating factors such as age variations, differences in image quality, diverse environmental conditions, and combinations of still images and video sequences. Maze et al. [24]. Moreover, with the increasing deployment of face recognition systems in security-sensitive applications, presentation attack detection (PAD) has become an essential component for improving the reliability and trustworthiness of these systems. The ISO/IEC 30107 family of standards also provides a comprehensive framework and evaluation criteria for assessing PAD performance. ISO [11], Ibrahim et al. [10].

Despite significant progress in perceptual modeling, most existing research has primarily focused on improving feature extraction and developing increasingly powerful deep learning architectures. In contrast, the intelligent management of decision-making processes and computational resource allocation has received relatively limited attention. Many existing systems apply a fixed processing pipeline to all inputs, even though individual samples may differ substantially in complexity, uncertainty, and computational requirements. This limitation can result in unnecessary computational overhead, increased latency, and higher energy consumption, particularly in real-time applications and resource-constrained edge computing environments. Laird [15], Falade et al. [4], Yuan et al. [37].

Therefore, a fundamental question arises: Can a face recognition system be designed to intelligently select the most appropriate processing path based on the characteristics of each input, the confidence associated with its decisions, and the available computational resources? In other words, how can a balance be achieved among recognition accuracy, processing speed, and computational cost while maintaining the interpretability of system decisions? A review of previous studies indicates that the integration of a cognitive meta-control layer for the adaptive management of this process has not yet been comprehensively explored in face recognition systems. Russell and Wefald [26], Lieder and Griffiths [17], Rajendran and Gebremedhin [25], Majer et al. [21].

The main contributions of this study are summarized as follows:

1. A cognitive–symbolic framework, named SOARFace-ID, is introduced for adaptive face recognition by integrating deep learning capabilities with cognitive reasoning mechanisms.
2. A reinforcement learning-based meta-controller is designed to dynamically select appropriate processing paths in a cost-aware manner, while considering computational resources and system requirements.
3. Open-set recognition, presentation attack detection (PAD), domain adaptation, parameter-efficient fine-tuning (PEFT), and test-time adaptation (TTA) are integrated into a unified decision-making framework. Coleman et al. [2], Tian et al. [32], Kumar et al. [14].
4. The framework balances recognition accuracy, processing latency, and computational cost while maintaining the interpretability of system decisions.
5. The proposed framework is comprehensively evaluated under both closed-set and open-set scenarios using standard face recognition evaluation metrics.

The objective of this research is to develop a cost-aware neuro-symbolic framework based on the SOAR architecture and reinforcement learning that can improve the robustness and accuracy of face recognition while reducing computational costs and deployment time through the integration of PEFT, TTA, and domain adaptation techniques. The following sections describe the proposed methodology, experimental results, and analysis of the findings.

2. Proposed Method

2.1. Overview of the Proposed Framework

In this study, we propose a neuro-symbolic framework for face recognition that aims to achieve a balanced trade-off between recognition accuracy, computational cost, response time, and decision interpretability. The proposed framework integrates a perception pipeline built on an embedding-based deep neural network, a cognitive meta-controller based on the SOAR architecture, and a cost-aware reinforcement learning policy. By combining these components, the framework adaptively selects the most appropriate processing strategy for each input, improving both recognition performance and computational efficiency in real-world scenarios.

Unlike conventional methods that process all input samples through a fixed pipeline, the proposed framework dynamically determines the most suitable processing path for each sample. This decision is based on the characteristics of the input, including image quality, prediction uncertainty, the likelihood of out-of-distribution data, feature-space distances, the available time budget, and energy constraints. Teerapittayanon et al. [31], Wang et al. [35], Wu et al. [36]. By adapting the processing strategy to the requirements of each sample, the framework achieves a more effective balance between recognition accuracy, computational efficiency, and inference time. The overall architecture of the proposed framework is presented in Figure 1.

The processing pipeline begins with face detection followed by geometric alignment to normalize the input images. Facial representations are then extracted using an ArcFace-based feature extraction network. Wang et al. [34], Liu et al. [19]. Depending on the characteristics of the input, additional modules—including domain adaptation, few-shot learning, open-set recognition, and test-time adaptation (TTA)—are activated only when needed. In the final stage, the SOAR-based cognitive meta-controller evaluates the current system state and determines whether the sample can be processed through the lightweight inference path or requires more computationally intensive modules. This adaptive decision-making strategy enables the framework to allocate computational resources efficiently while maintaining high recognition accuracy across diverse operating conditions.

2.2. Facial Feature Extraction

In this study, the training dataset is defined as follows:

$$D = \{(x_i, y_i) \mid i = 1, 2, \ldots, N\} \quad (1)$$

where:

- x_i: the input face image;
- y_i: the corresponding identity label; and
- N: the total number of training examples.

The feature extractor generates a compact face embedding from the input face image:

$$z_i = f_\theta(x_i) \quad (2)$$

where:

- f_θ: the feature extraction network parameterized by θ;
- z_i: the facial feature vector.

The ArcFace loss is employed to improve the separation among different identity classes. Huang et al. [9], Kim et al. [13]:

$$L_{ArcFace} = -(1/N)\ \Sigma_{i=1}^{N} \log [\ \exp (s \cos (\theta_{Y_i} + m)) / (\exp (s \cos (\theta_{Y_i} + m)) + \Sigma_{j\neq Y_i} \exp (s \cos \theta_j))] \quad (3)$$

where:

- s: the scale factor;
- m: the angular margin;
- θ_{Y_i}: the angle between the normalized feature of sample i and the weight vector of its ground-truth class;
- θ_j: the corresponding angle for class j;
- N: the number of training samples;
- C: the number of identity classes; and
- Y_i: the ground-truth class of the i-th sample.

This loss brings samples belonging to the same identity closer together while increasing the angular separation between samples belonging to different identities.

2.3. Domain Adaptation with Parameter-Efficient Fine-Tuning (PEFT)

To avoid the high computational cost of full fine-tuning, the LoRA method is adopted. Houlsby et al. [7], Hu et al. [8]. In this approach, the original network weights are kept frozen, and only the low-rank updates are trained.

$$W = W_0 + \Delta W \quad (4)$$

where:

- W_0: the pretrained weight matrix;
- ΔW: the weight update; and
- W: the adapted weight matrix.

In LoRa, the weight update is expressed using two trainable low-rank matrices:

$$\Delta W = BA \quad (5)$$

where:

- B, A: trainable low-rank matrices;
- r: the rank of the adaptation.

The rank is much smaller than the dimensions of the original weight matrix, which allows the model to adapt to a new domain without updating all of its original parameters. In the proposed framework, the LoRA rank is set to $r = 8$.

This strategy considerably reduces the number of trainable parameters and the computational cost of domain adaptation while preserving the model's ability to learn domain-specific features. During inference, the learned update ΔW is combined with the pretrained weights according to (4).

2.4. Domain Adaptation

To reduce the effect of distribution differences between the source and target domains, an adversarial domain adaptation mechanism based on the domain-adversarial neural network (DANN) is employed. The main objective of this component is to learn representations that are effective for identity recognition while remaining robust to domain variations.

In DANN, a feature extractor, an identity classifier, and a domain discriminator are jointly optimized during training. Sun and Saenko [30], Long et al. [20], Ganin et al. [6]. The overall objective function is defined as follows:

$$L_{total} = L_{identity} + \lambda L_{domain} \tag{6}$$

where:

- $L_{identity}$: the Arc Face identity recognition loss;
- L_{domain}: the domain-adversarial loss; and
- *λ:* a weighting factor that controls the contribution of domain adaptation to the overall objective.

The domain classification loss is defined as follows:

$$L_{domain} = - \Sigma_i \, [d_i \log (D (z_i)) + (1 - d_i) \log (1 - D (z_i))] \tag{7}$$

where:

- $D(z_i)$: the predicted probability that the sample belongs to the target domain;
- d_i: the domain label, where 0 and 1 indicate the source and target domains, respectively; and
- z_i: the feature representation of the *i*-the sample.

During the training phase, the gradient associated with domain classification is reversed through the Gradient Reversal Layer (GRL), As a result, the feature extractor is encouraged to learn representations that are discriminative for identity recognition while being less sensitive to domain-specific variations.

2.5. Few-Shot Learning

To handle classes with only a few available training samples, a prototype-based few-shot learning module is introduced. Snell et al. [29], Khosla et al. [12]. The prototype of each class is computed in the feature space as follows:

$$p_k = \left(\frac{1}{|S_k|}\right) \Sigma_{i \in S_k} z_i \tag{8}$$

where:

- p_k: the prototype of class *k;*

- S_k: the set of samples belonging to class *k;*
- $|S_k|$: the number of support samples in that class; and
- z_i: the feature vector of the *i*-th sample.

The similarity between the test sample embedding and class *k* is measured using cosine similarity:

$$Similarity\ (z, p_k) = (z.\ p_k) / (||z||\ ||p_k||) \tag{9}$$

where:

- p_k: the prototype of class *k;*
- z: the feature vector of the test sample;
- $z.\ p_k$: their dot product; and
- $||z||$ *and* $||p_k||$: their Euclidean norms.

The higher the similarity, the more likely the sample is to belong to that class.

2.6. Open-Set Recognition and Uncertainty Estimation

In real-world applications, the system may encounter face identities that are not present in the training set. Wang et al. [33], Schneider et al. [27]. To detect such unknown samples, multiple uncertainty measures are combined.

The energy score is defined as follows:

$$E(x) = -T \log \Sigma_i \exp(f_i(x) / T) \tag{10}$$

where:

- $f_i(x)$: the output corresponding to class *i;*
- T: the temperature scaling parameter, and the summation is performed over all classes.

Additionally, the Mahalanobis distance is computed in the feature space. Liu et al. [18], Lee et al. [16]:

$$M(x) = (z - \mu)^{T} \Sigma^{-1} (z - \mu) \tag{11}$$

where:

- μ: the mean feature vector of the reference class;
- Σ: the corresponding covariance matrix; and
- z: the feature representation of the input sample.

These indicators are combined to generate a final uncertainty score:

$$U(x) = \alpha E(x) + \beta M(x) + \gamma\ (1 - \cos(x)) \tag{12}$$

where:

- α, β, γ: the weighting coefficients
- *cos(x)*: the cosine similarity between the input sample and its nearest class prototype

A higher value of *U(x)* indicates a greater likelihood that the sample is unknown or lies outside the training data distribution.

2.7. Test-Time Adaptation (TTA)

To improve the model's robustness against data distribution shifts during deployment, the TENT method is employed. In this stage, the parameters of the batch normalization layers are selectively updated without requiring labeled samples from the target domain.

The main objective of test-time adaptation (TTA) is to reduce the uncertainty of the model predictions by minimizing the prediction entropy:

$$H(p) = - \Sigma_i \ p_i \log (p_i) \tag{13}$$

where:

- p_i: the probability of predicting class i;
- $H(p)$: the predictive entropy of the model.

The adjustable parameters in TTA are updated by solving the following optimization problem:

$$\theta^* = argmin_\theta \ H(p(y|x;\theta)) \tag{14}$$

where:

- θ^*: the optimized parameters during testing;
- θ: the adjustable parameters of the batch normalization layers;
- x: the unlabeled input sample; and
- $p(y|x;\theta)$: the predicted class-probability distribution for input x under parameters θ.

To prevent model instability, only the scale and shift parameters of the batch normalization layers are updated during test-time adaptation, while the remaining network parameters remain frozen.

2.8. SOAR-Based Cognitive Meta-Controller

The core component of the proposed framework is a cognitive meta-controller based on the SOAR architecture, which dynamically selects the appropriate processing path.

At each time step t, the system state is represented by the following state vector:

$$S_t = \{Q_t, U_t, OOD_t, B_t, C_t\} \tag{15}$$

where:

- Q_t: image quality score;
- U_t: prediction uncertainty level;
- OOD_t: estimated out-of-distribution (OOD) probability;
- B_t: remaining time and energy budget; and
- C_t: current computational cost.

Based on the current state, the meta-controller selects an action from the following action space:

$$A_t = \{Accept, Reject, TTA, Few\text{-}Shot, PAD, Human\} \tag{16}$$

where:

- *Accept*: Directly accepts the sample based on the current prediction;
- *Reject*: Rejects the sample when the prediction is considered unreliable;
- *TTA*: Applies test-time adaptation to address potential distribution shifts;
- *Few-Shot*: Activates few-shot learning when additional target-domain adaptation is required;
- *PAD*: Activates the presentation attack detection module or performs additional processing potentially unreliable samples; and
- *Human*: Refers the sample to a human operator when the system cannot make a sufficiently reliable decision.

The SOAR symbolic rules help guide the decision-making process by limiting the available actions based on the current system state.

2.9. Reinforcement Learning Formulation and Reward Function

The selection of the processing path is formulated as a Markov decision process (MDP).

$$M = (S, A, P, R, \gamma) \tag{17}$$

where:

- S: State space;
- A: Action space;
- P: State transition probability;
- R: Reward function; and
- γ: Discount factor.

The reinforcement learning agent learns a policy π that determines the probability of selecting each possible action:

$$\pi(a|s) = P\,(A_t = a \mid S_t = s) \tag{18}$$

The reward function is designed to improve recognition accuracy while keeping the computational cost under control:

$$R_t = Acc_t - \lambda_1 L_t - \lambda_2 E_t - \lambda_3 Risk_t \tag{19}$$

where:

- Acc_t : Decision accuracy;
- L_t : Processing latency;
- E_t : Energy consumption;
- $Risk_t$: Error risk, such as the false positive identification rate (FPIR); and
- $\lambda_1 . \lambda_2 . and\ \lambda_3$: are weighting coefficients that control the relative importance of latency, energy consumption, and error risk, respectively.

Thus, the agent learns to activate computationally expensive processing paths only when the expected improvement in recognition performance justifies the additional computational cost.

2.10. Unified Training Objective of the Proposed Model

The proposed framework is trained in two stages. In the first stage, the perceptual network, including the feature extractor, domain adaptation module, and auxiliary modules, is optimized. In the second stage, the SOAR-based meta-controller policy is learned through reinforcement learning from the recorded experiences.

The overall objective function of the perceptual module is defined as follows. Mao et al. [23], Manhaeve et al. [22]:

$$L_{perception} = L_{ArcFace} + \lambda_d \, L_{Domain} + \lambda_f \, L_{few-shot} + \lambda_t \, L_{TTA} \quad (20)$$

where:

- $L_{ArcFace}$: identity recognition loss;
- L_{Domain} : domain adaptation loss;
- $L_{few-shot}$: few-shot learning loss;
- L_{TTA} : test-time adaptation regularization loss; and
- λ_d, λ_f, *and* λ_t*:* loss weighting coefficients.

The objective of the meta-controller is formulated as the maximization of the expected cumulative reward:

$$J(\pi) = E\pi[\Sigma_{t=0}^{T} \gamma^t \, R_t] \quad (21)$$

where:

- π: action selection policy;
- γ: discount factor;
- R_t : reward at time step t; and
- T: the total number of decision steps.

The goal of the agent is to learn the optimal policy π^*:

$$\pi^* = argmax\pi \; J(\pi) \quad (22)$$

The learned policy aims to balance three main objectives: improving recognition accuracy, reducing computational cost, and controlling decision risk.

Algorithm 1: Workflow of the Proposed SOAR Face Framework

Input: Face image *x*, feature extraction model f_θ meta-controller policy π, and computational budget *B*.

Output: The final decision y and the corresponding uncertainty estimate *U*.

1. Receive the input face image x.

2. Detect the face and perform geometric alignment.

3. Extract image quality indicators:

Q_t = {$Blur_t$, $Pose_t$, $Illumination_t$, $Occlusion_t$}.

4. Generate the facial feature embedding:

$z_t = f_\theta(x)$.

5. Estimate the uncertainty associated with the extracted features:

U_t = {$Energy_t$, $Mahalanobis_t$, $CosineSimilarity_t$}

6. Estimate the out-of-distribution score:

$OOD_t = g_{OOD}(z_t)$.

7. Form the state representation used by the meta-controller:

S_t = {Q_t, U_t, OOD_t, B_t, C_t}.

8. Define the feasible action set:

A = {Accept, Reject, TTA, Few-Shot, PAD, Human}.

9. Apply the SOAR production rules and safety constraints to determine A_t.

10. Select the action Using the meta-controller policy:

$a_t = \pi(S_t \mid A_t)$.

11. If a_t = Accept, retain the current recognition result.

12. If a_t = Reject, reject the input sample.

13. If a_t = TTA, apply test-time adaptation and recompute the feature representation and prediction.

14. If a_t = Few-Shot, update the class prototypes and recompute the class scores.

15. If a_t = PAD, perform the additional presentation-attack detection procedure and update the recognition state.

16. If a_t = Human, refer the sample to a human operator.

17. Update S_{t+1} and B_{t+1} according to the selected processing path.

18. Determine the final class decision:

$y = argmax_c\ P(c \mid x, S_t)$

19. Record the selected processing path, inference time, confidence scores, and decision statistics.

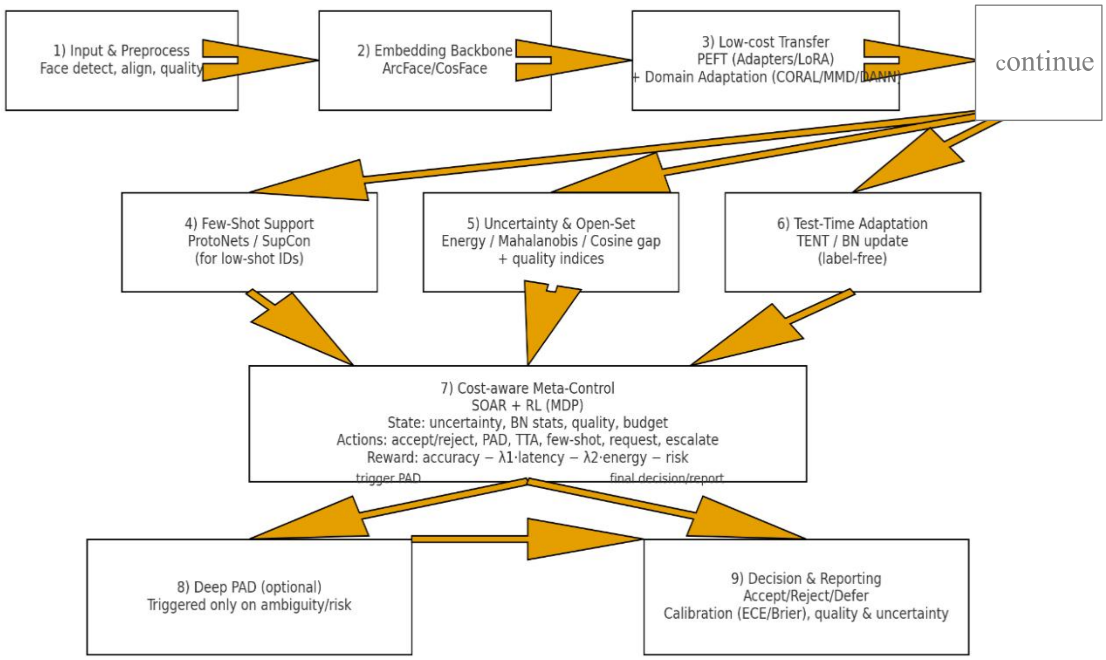

Figure 1. Workflow of the Proposed Model

The experimental evaluation of the proposed framework is presented in the following sections. Before discussing the experimental results, the rationale for selecting the major components of the framework is described below.

2.11. Design Rationale and Component Selection

This section discusses why the key components of the proposed framework were selected. The underlying idea is that face recognition in real-world environments goes beyond simply classifying faces. In practice, the system needs to make adaptive decisions while handling uncertainty, changes in the environment, and limited computational resources.

2.11.1 Selection of the SOAR-Based Neuro-Symbolic Architecture

Deep learning models are highly effective at extracting meaningful features, but their behavior can become less predictable when they encounter domain shifts, unfamiliar samples, or limited computational resources. To address these challenges, the proposed framework incorporates a SOAR-based symbolic layer that acts as a meta-controller. Unlike a conventional neural network, this symbolic layer relies on interpretable decision rules. It can therefore take practical constraints—such as the available time budget, risk level, and safety requirements—into account when selecting the appropriate processing path.

2.11.2 Rationale for Using Cost-Aware Reinforcement Learning

In many face recognition systems, keeping computationally intensive modules active at all times can lead to unnecessary increases in processing time and energy consumption. In the proposed framework, reinforcement learning serves as a dynamic decision-maker, determining when the potential benefit of additional processing is worth the extra cost. This allows the system to handle straightforward samples through a fast path, while activating more resource-intensive processing only when the input is uncertain or carries a higher risk of error.

2.11.3 Why PEFT Was Chosen Over Full Fine-Tuning

Adapting a large model to a new domain can be costly if all of its parameters have to be retrained. In the proposed framework, LoRA is used to learn only small, low-rank updates to the existing weights rather than fine-tuning the entire model. This makes domain adaptation more efficient in terms of computational resources while maintaining most of the model's original performance.

2.11.4 The Role of TTA in Real-World Conditions

Changes in lighting, face angle, imaging conditions, and sensor quality can affect the performance of a face recognition model. To address this issue, the TTA module allows the model to gradually adapt its feature representations during testing, without relying on labeled data. By reducing uncertainty in its predictions, the model can adjust to changing conditions and maintain more reliable recognition performance in real-world settings.

3. Experimental Setup and Evaluation Protocol

To assess the effectiveness of the proposed meta-controller and its integration with parameter-efficient transfer learning, domain adaptation, few-shot learning (FSL), and test-time adaptation (TTA), we conduct a comprehensive set of experiments under both closed-set and open-set scenarios.

The evaluation is carried out using single-image settings across benchmark datasets and a low-resource target domain.

The evaluation protocol is designed to provide a comprehensive analysis by considering not only recognition accuracy but also the trade-off between accuracy, inference latency, and computational cost.

This analysis highlights the role of the proposed meta-controller in dynamically selecting suitable processing paths and improving the overall efficiency of the recognition system.

3.1. Datasets and Evaluation Scenarios

The embedding network is pre-trained using a large-scale, carefully curated collection of web images. To thoroughly evaluate the effectiveness of the proposed approach, three experimental benchmarks are considered. The first benchmark is the IJB-C dataset, which serves as a challenging evaluation platform for real-world face recognition scenarios. It provides both verification and identification protocols under diverse variations in pose, illumination, image quality, and other environmental conditions, enabling a comprehensive assessment of the robustness and generalization capability of the learned representations.

Second, the Mega Face dataset is used to evaluate the performance of the proposed method in large-scale identification scenarios. This evaluation examines the robustness of the approach against an increasing number of distractor identities while maintaining recognition accuracy at large scales. Third, a low-resource target domain with distributional heterogeneity is considered, including surveillance images captured under varying image qualities, viewing angles, and illumination conditions. This scenario is designed to evaluate the capability of the proposed method for knowledge transfer, few-shot learning (FSL), and test-time adaptation (TTA) under conditions that closely resemble real-world applications. Furthermore, for subset-based evaluation scenarios, a set of unknown identities is introduced alongside known identities from the same or similar sources. This enables a realistic assessment of the model's ability to perform open-set recognition (OSR) and out-of-distribution (OOD) detection under distribution shifts that better reflect practical deployment environments.

3.2. Implementation Details and Training Procedure

The feature extraction network was built on the ResNet-IR100 architecture and optimized using the ArcFace loss function. During domain adaptation, LoRA (Low-Rank Adaptation) is employed instead of full fine-tuning to significantly reduce the number of trainable parameters while maintaining competitive performance. The LoRA rank is denoted by r and is set to $r = 8$, where r represents the rank of the low-rank adaptation matrices. To alleviate the distribution discrepancy between the source and target domains, a domain-adversarial neural network (DANN) module is incorporated with an adaptation coefficient $\lambda = 0.1$, where λ controls the contribution of the domain-adaptation loss to the overall training objective. The model was trained using the Adam optimizer. The learning rate is set to 1×10^{-4} for the LoRA parameters and 1×10^{-5} for the trainable DANN and classification layers. The batch size was 32, the model is trained for 50 epochs, and the weight decay is fixed at 1×10^{-4}.

The reinforcement learning agent in the meta-controller is trained using experiences stored in a replay buffer. The state representation and action space used by the agent is formally defined in Section 3.3. Based on the observed state, the agent dynamically selects the most appropriate processing strategy while balancing recognition performance and computational efficiency.

3.3. Meta-Controller Configuration and RL Policy Design

In the proposed framework, the system state is characterized by a set of informative indicators, including image quality, the cosine similarity gap between the top two matching candidates, energy scores or Mahalanobis distance, the deviation of batch normalization (BN) statistics from their reference values, and the available time and energy budget. Together, these indicators provide a comprehensive representation of the current operating conditions and guide the selection of the most appropriate processing strategy.

$$s_t = [q_t, g_t, u_t, d_t, b_t] \tag{23}$$

where:

- q_t: the image-quality score;
- g_t: the cosine-similarity gap between the top two candidates;
- u_t: the prediction uncertainty;
- d_t: the distribution-shift indicator; and
- b_t: the remaining computational budget at time step t.

Based on the observed state, the meta-controller selects one of the available processing paths:

$$a_t \in \{a_{fast}. a_{TTA}, a_{FSL}, a_{PAD}. a_{Accept}. a_{Reject}. a_{Human}\} \tag{24}$$

where:

- a_{fast}: the fast-processing path;
- a_{TTA}: activates test-time adaptation;
- a_{FSL}: activates few-shot adaptation; and

- a_{PAD}: activates the presentation-attack detection module.

The selected action is evaluated using a reward function that jointly considers recognition performance, inference latency, and computational cost:

$$r_t = \alpha A_t - \beta L_t - \gamma C_t \tag{25}$$

where:

- A_t: the recognition performance;
- L_t: inference latency;
- C_t: computational cost; and
- α, β, and γ: weighting coefficients. In our experiments, α = 1.0, β = 0.3, and γ = 0.2.

During decision-making, the SOAR symbolic rules first assess conditions such as degraded image quality, high prediction uncertainty, or limited computational resources. When these conditions are detected, the SOAR rules restrict the candidate action set according to the available computational budget and predefined risk constraints. As a result, the reinforcement learning agent selects the next action only from a set of safe and feasible alternatives. This mechanism not only reduces unnecessary computational overhead but also improves the robustness and reliability of the overall system.

The reinforcement learning policy is trained using the Soft Actor–Critic (SAC) algorithm. By leveraging experiences stored in the replay buffer together with feedback obtained from actual decision outcomes, the agent gradually learns a policy that effectively balances recognition accuracy and computational cost. The reward function is designed to maximize recognition performance while simultaneously minimizing processing latency and energy consumption. When required, this constraint is further formulated as a probabilistic safety constraint, ensuring that the system consistently operates within an acceptable level of risk under different deployment conditions.

The reinforcement learning agent uses a discount factor of 0.99 and a learning rate of 1×10^{-4}. A replay buffer with a capacity of 10,000 experiences is employed. The target network is updated every 500 training steps. The exploration rate ε is initialized at 1.0 and gradually decreased to 0.05.

3.4. Evaluation Metrics and Analysis Methods

The performance of the proposed framework is evaluated using established metrics for face verification, identification, and open-set recognition, including EER, ROC/DET curves, TPIR@FPIR, and AUROC. In addition, reliability and computational efficiency are assessed through prediction calibration metrics, runtime analysis, and resource consumption measurements to investigate the trade-off between recognition performance and computational cost. Statistical analyses are also conducted to determine the significance of performance differences between the proposed approach and the baseline methods.

3.5. Baselines and Ablation Studies

To investigate the contribution of each individual component to the overall performance of the proposed framework, a comprehensive set of baseline comparisons and ablation experiments is conducted. First, the base face-embedding network without knowledge-transfer or adaptive-processing mechanisms is used as the primary baseline. In addition, a fully fine-tuned version of the same network is compared with the proposed parameter-efficient fine-tuning (PEFT) strategy to analyze the trade-off between performance improvement, the number of trainable parameters, and training computational cost. Next, the impact of Domain Adaptation is evaluated both independently and in combination with PEFT. This analysis aims to quantify the contribution of distribution alignment between the source and target domains and to examine whether combining parameter-efficient transfer with domain adaptation provides additional performance gains.

To evaluate the robustness of the proposed framework under limited-data conditions and distribution shifts, two additional variants are examined: one without few-shot learning and another without test-time adaptation (TTA). Comparing these variants with the complete model allows us to measure the individual contribution of each adaptation mechanism in improving generalization capability. Furthermore, the role of the proposed SOAR-based meta-controller is investigated through three different configurations: (1) a model without meta-control, where the processing pipeline remains fixed; (2) a rule-based symbolic controller, where decisions are made using predefined expert rules; and (3) the complete hybrid controller combining symbolic reasoning with a reinforcement learning policy. This comparison highlights the contribution of the RL-based decision-making mechanism over static control strategies.

The effect of the presentation attack detection (PAD) module is also studied under two settings: always-active PAD and selectively activated PAD guided by the meta-controller policy. This experiment evaluates whether adaptive activation can maintain security performance while reducing unnecessary computational overhead. To ensure fair and unbiased comparisons, all experimental variants are evaluated under identical conditions, including the same datasets, backbone architecture, hyperparameter settings, and computational budget. The reported results are obtained from multiple independent runs and are presented together with the corresponding variation measures to assess the consistency and reliability of the observed performance differences.

3.6. Computational Budget and Reproducibility Details

All experiments were performed using the same hardware configuration to provide a fair and consistent comparison of computational efficiency, including inference latency and energy consumption. The experimental platform consisted of a server equipped with an NVIDIA RTX 3090 GPU (24GB GDDR6X), an Intel Xeon Gold 6226R CPU, and 128GB of system memory. The software environment was based on Ubuntu 22.04 LTS, PyTorch 2.0, and the required supporting libraries. To enhance reproducibility, all experimental factors, including input image size, preprocessing procedures, model initialization, training settings, learning rate, and other relevant hyperparameters, were maintained consistently across all experiments. The key components of the proposed framework, including the meta-controller, adaptation modules, and decision-making strategy, were developed in a modular architecture, allowing each component to be evaluated independently and the experiments to be reproduced reliably. Beyond quantitative evaluation using recognition accuracy, inference latency, and energy consumption, representative failure cases were also analyzed qualitatively. These analyses help identify the conditions that trigger computationally intensive processing paths and demonstrate how SOAR-based rules contribute to guiding the adaptive decision-making process.

3.7. Hypotheses and Expected Outcomes

The proposed method is expected to outperform the baseline approaches in terms of key evaluation metrics, including EER and TPIR@FPIR, across both verification and identification scenarios. In open-set recognition scenarios, the method is anticipated to achieve improved AUROC performance while maintaining the average inference latency within the target computational budget. This is attributed to the proposed meta-control mechanism, which activates more computationally expensive processing paths only for uncertain or high-risk samples. It is hypothesized that the performance improvement is partly attributable to the interaction between the meta-controller, test-time adaptation (TTA), and adaptive PAD selection. Furthermore, the use of parameter-efficient fine-tuning (PEFT) is expected to substantially reduce the number of trainable parameters compared with full model fine-tuning, while preserving competitive recognition performance. Finally, the use of parameter-efficient fine-tuning (PEFT) is expected to substantially reduce the number of trainable parameters compared with full model fine-tuning, while preserving competitive recognition performance.

4. Results and Discussion

In this section, the proposed SOARFace framework is evaluated in terms of three key aspects: recognition performance, model reliability, and computational efficiency. Recognition performance is evaluated using EER for face verification, TPIR@FPIR for identification, and AUROC for open-set recognition. Model reliability is assessed using the Expected Calibration Error (ECE) and the Brier Score, while computational efficiency is evaluated based on inference latency and energy consumption. Unless otherwise specified, all results are reported as the mean ± standard deviation across five independent runs with different random seeds. When appropriate, 95% confidence intervals and McNemar's statistical test are also provided to examine the statistical significance of the observed performance differences. Verification and identification results are obtained following the standard IJB-C evaluation protocol, whereas open-set recognition performance is assessed using unknown samples that are distributionally close to the target domain.

4.1. Comparison with Related Work

Table 1 presents a comprehensive comparison between the baseline face recognition methods and the proposed SOARFace framework. For EER and ECE, lower values indicate better performance, whereas higher values are preferred for TPIR@FPIR and AUROC. The “Trainable Params vs. Full FT” column reports the percentage of trainable parameters relative to the full fine-tuning setting. As shown in Table 1, SOARFace achieves the lowest EER among the compared methods while maintaining a competitive TPIR@FPIR of 0.90. This improvement is mainly attributed to the SOAR + RL meta-controller, which activates computationally intensive modules, such as TTA and few-shot learning, only when they are needed. Moreover, PEFT reduces the number of trainable parameters to less than 5% of those required for full fine-tuning, substantially reducing the training cost while preserving the overall recognition performance.

Table 1. Performance Comparison of Different Methods on IJB-C and OSR

Method	Backbone/ Loss	Transfer (PEFT/DA)	Few-Shot	TTA	Meta-control	EER (IJB-C)	TPIR@FPIR=1%	AUROC (Open-Set)	ECE	Trainable Params vs. Full FT
FaceNet [28]	Inception + Triplet	No	No	No	No	0.038	0.76	0.9	0.06	100%
SphereFace [19]	ResNet-IR + SphereFace	No	No	No	No	0.03	0.8	0.92	0.055	100%
CosFace [34]	ResNet-IR + CosFace	No	No	No	No	0.026	0.83	0.94	0.048	100%
ArcFace baseline [3]	ResNet-IR + ArcFace	No	No	No	No	0.024	0.84	0.94	0.045	100%
CurricularFace [9]	ResNet-IR + CurricularFace	No	No	No	No	0.022	0.86	0.946	0.042	100%
AdaFace [13]	ResNet-IR + AdaFace	No	No	No	No	0.02	0.87	0.948	0.04	100%

ArcFace + CORAL [30]	ResNet-IR + ArcFace	CORAL	No	No	No	0.022	0.86	0.946	0.043	100%
ArcFace + MMD [20]	ResNet-IR + ArcFace	MMD	No	No	No	0.022	0.86	0.946	0.043	100%
ArcFace + DANN [6]	ResNet-IR + ArcFace	DANN	No	No	No	0.021	0.86	0.945	0.043	100%
ArcFace + OpenMax [1]	ResNet-IR + ArcFace + OpenMax	No	No	No	No	0.022	0.85	0.955	0.042	100%
ArcFace + Energy OOD [18]	ResNet-IR + ArcFace + Energy	No	No	No	No	0.021	0.86	0.958	0.041	100%
ArcFace + TENT [33]	ResNet-IR + ArcFace	No	No	TENT	No	0.018	0.88	0.952	0.034	100%
PEFT (Adapters) + DA	ResNet-IR + ArcFace	Adapters + DA	No	No	No	0.019	0.88	0.955	0.038	≈5%
PEFT + DA + Few-Shot	ResNet-IR + ArcFace + Proto/SupCon	Adapters + DA	Yes	No	No	0.0175	0.9	0.958	0.036	≈5%
SOARFace (ours)	ResNet-IR/ViT + ArcFace	Adapters/LoRA + DA	Yes	TENT/BN	SOAR + RL	0.015	0.9	0.95	0.028	<5%

As shown in Table 1, SOARFace achieves the lowest EER and ECE among the compared methods, while maintaining competitive TPIR@FPIR and AUROC performance.

4.2. Verification

We evaluated the proposed SOARFace framework on the IJB-C benchmark for identity verification. Unless otherwise noted, all experimental results are presented as the mean ± standard deviation over five independent runs using different random seeds. The evaluation primarily focuses on the Equal Error Rate (EER) and the ROC and DET curves. To provide a more comprehensive assessment, we also report the Area Under the ROC Curve (AUROC) and the True Positive Rate (TPR) at the standard operating points of FPR = 10^{-3} and 10^{-2}. The statistical significance of the performance differences was assessed using bootstrap resampling with 1,000 iterations together with McNemar's test.

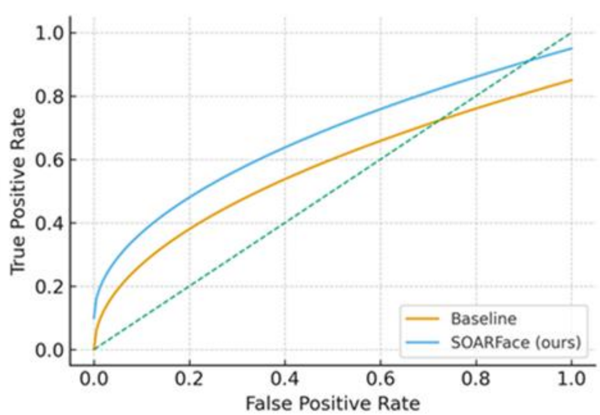

Figure 2. ROC curve comparing the proposed SOARFace framework with the baseline method for identity verification on the IJB-C dataset.

As illustrated in Fig. 2, the ROC curve of the proposed SOARFace framework consistently outperforms the baseline method across all operating points, demonstrating its superior ability to distinguish between genuine and impostor samples. This improvement is more pronounced at low false positive rate (FPR) values, which are particularly important in security-critical applications. Specifically, at FPR = 10^{-3}, the true positive rate (TPR) increases from 0.82 for the baseline method to 0.89 for SOARFace, achieving an improvement of $\Delta = +0.07$. These findings indicate that the proposed framework enhances recognition accuracy while maintaining effective control over false positive errors.

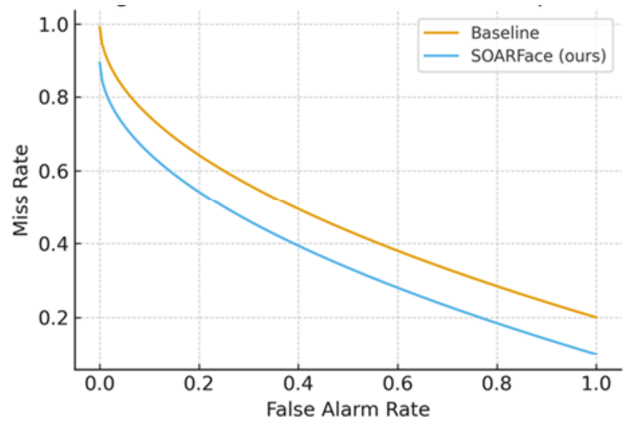

Figure 3. Detection Error Tradeoff (DET) curve illustrating the relationship between the miss rate and the false alarm rate.

Figure 3 illustrates the performance of the proposed method by comparing the miss rate against the false alarm rate. As can be seen, SOARFace consistently outperforms the baseline method by achieving a lower miss rate across all operating points. In addition, the curve's closer position to the lower-left corner of the plot reflects a simultaneous reduction in error rates. These results confirm that the proposed method delivers more robust and reliable performance across a wide range of operating conditions.

4.3. Identification

The 1: N identification performance is evaluated following the standard IJB-C evaluation protocol. The primary evaluation metric is the True Positive Identification Rate (TPIR) at different False Positive Identification Rate (FPIR) levels (TPIR@FPIR), including $\{10^{-4}, 10^{-3}, 10^{-2}, \text{and } 10^{-1}\}$. These operating points cover a wide range of practical scenarios and, in particular, enable a comprehensive assessment of system performance in security-critical applications where extremely low false acceptance rates are required.

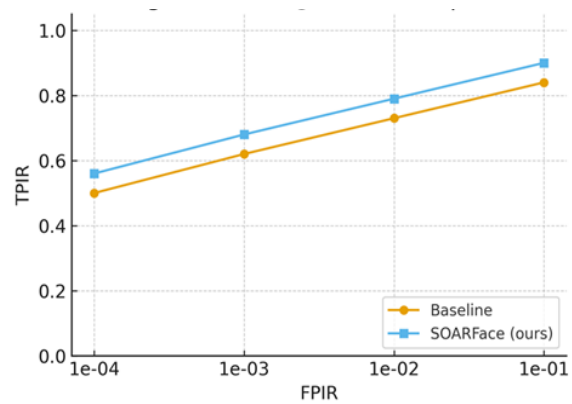

Figure 4. Comparison of the proposed SOARFace framework and the baseline method in terms of TPIR across different FPIR operating points on the IJB-C dataset.

As illustrated in Fig. 4, SOARFace achieves consistently better performance than the baseline approach across all FPIR levels. The performance gain becomes more evident at lower FPIR operating points, which are especially important for security-critical recognition systems. For instance, at FPIR = 10^{-2}, the TPIR improves from 0.73 to 0.79, and at FPIR = 10^{-1}, it increases from 0.84 to 0.90. These findings indicate that the proposed meta-control mechanism can effectively select appropriate processing paths, including few-shot adaptation, TTA, and PAD, leading to higher correct identification rates while maintaining a low false positive rate. As a result, SOARFace provides a more reliable and efficient solution for security-oriented face recognition applications.

4.4. Open-Set and Out-of-Distribution (OOD) Evaluation

To evaluate the performance of the proposed model under open-set recognition and out-of-distribution (OOD) scenarios, the AUROC and AUPR metrics are employed. In addition, the Unknown FPR@TPR=95% metric is reported to provide a more comprehensive assessment of the model's ability to identify and reject unknown samples at a specific and comparable operating point.

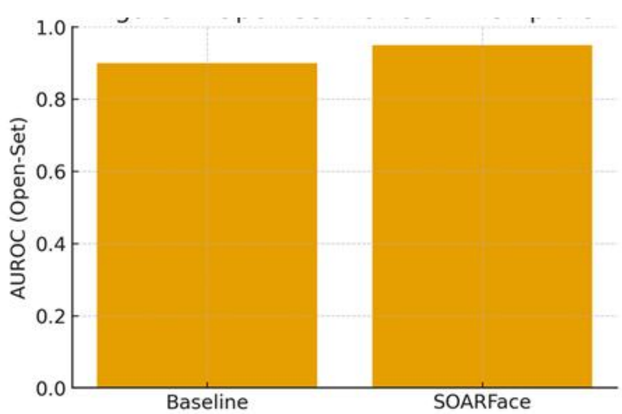

Figure 5. Performance evaluation of the proposed SOARFace framework for Open-Set Recognition (OSR) using the AUROC metric

Figure 5 illustrates the performance of the proposed method in the open-set recognition (OSR) task. As observed, SOARFace provides a stronger capability for distinguishing between known and unknown samples compared with the baseline method, improving the AUROC score from 0.90 to

0.95, corresponding to an absolute improvement of 0.05. In addition, as reported in Table 2, the AUPR score increases from 0.88 to 0.93, while the Unknown FPR@TPR=95% decreases from 0.18 to 0.12. These results demonstrate that the proposed framework not only improves the recognition accuracy of known identities but also enhances the ability to correctly reject unknown samples, thereby reducing the risk of false acceptance. This improvement is attributed to the adaptive decision-making mechanism of the proposed SOARFace framework, which dynamically selects appropriate processing paths according to the characteristics of the input samples.

Table 2. Open-Set Recognition Performance

Method	AUROC (OSR)	AUPR (OSR)	Unknown FPR@TPR=95%
Baseline	0.9	0.88	0.18
SOARFace (ours)	0.95	0.93	0.12

As shown in Fig. 5 and Table 2, the AUROC increased from 0.90 to 0.95, while the false-positive rate for unknown samples at TPR = 95% decreased from 0.18 to 0.12. These results indicate that SOARFace improves the separation between known and unknown samples and reduces the risk of false acceptance in open-set recognition scenarios.

4.5. Calibration and Reliability Evaluation

The reliability of the proposed model was assessed using the Expected Calibration Error (ECE) and Brier Score metrics. All experiments were conducted after calibrating the model's output scores. To achieve this, two complementary calibration approaches were applied: (1) test-time adaptation (TTA) based on controlled BatchNorm updates and entropy minimization with TENT, enabling the model to better adapt to the test-time data distribution; and (2) single-parameter temperature scaling, optimized on the validation set to improve the alignment between the predicted confidence scores and the actual frequency of observed outcomes.

As shown in Fig. 6, the proposed SOARFace framework achieves better performance than the baseline method in terms of both ECE and Brier Score. The reduction in these two metrics indicates that the predicted probabilities are better aligned with the actual outcomes, demonstrating that the model not only improves recognition accuracy but also provides more reliable confidence estimates. These results suggest that the joint application of TTA and temperature scaling effectively enhances the calibration quality of the model and improves its robustness against distribution shifts in the test data.

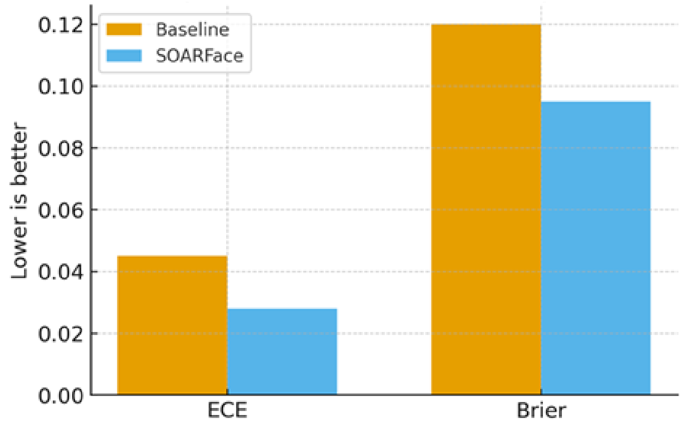

Figure 6. Calibration performance comparison between the proposed SOARFace framework and the baseline method using ECE and Brier Score metrics.

4.6. Cost–Latency Trade-off Analysis

To evaluate the computational and temporal efficiency of the proposed framework, the average inference latency, 95th-percentile latency (P95), and energy consumption per image were measured. In addition, the EER–latency trade-off curve was analyzed to investigate the balance between recognition performance and processing time. Unless otherwise specified, all experiments were conducted under a fixed operational load and using the same hardware configuration.

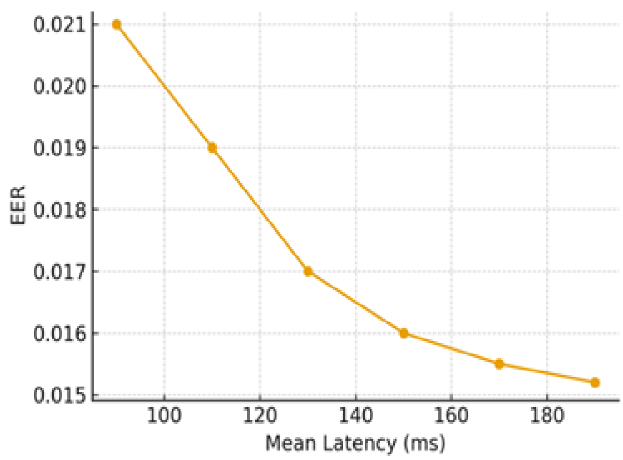

Figure 7. EER–Latency trade-off analysis of the proposed SOARFace framework under different computational budgets.

As shown in Fig. 7, increasing the available time budget leads to a reduction in EER, indicating improved recognition performance. However, SOARFace achieves the target EER level even within a low-latency regime below 180 ms. This result demonstrates that the proposed SOAR + RL-based meta-control mechanism activates computationally expensive processing paths, such as few-shot adaptation, TTA, and PAD, only for challenging or ambiguous samples, while processing the majority of inputs through the fast pathway. Consequently, the framework maintains low average latency without causing a noticeable degradation in recognition accuracy. These findings indicate that SOARFace establishes an effective trade-off between recognition performance and computational cost, making it suitable for real-time applications and time-constrained recognition systems.

4.7. Ablation Analysis of SOARFace Components

To analyze the individual contribution of each component in the proposed SOARFace framework, we performed an incremental ablation study. The evaluation started from the standard ArcFace baseline, and the proposed components were gradually incorporated into the model in a step-by-step manner, including PEFT, domain adaptation (DA), test-time adaptation (TTA), few-shot adaptation, SOAR + RL-based meta-control, and the final selective PAD module.

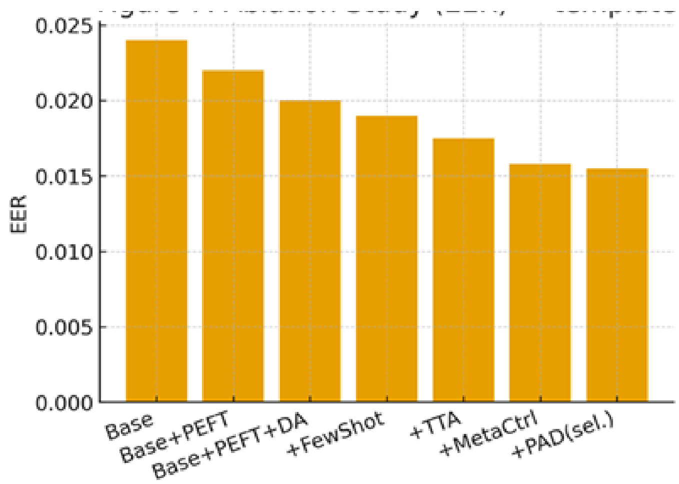

Figure 8. Ablation study results evaluating the contribution of individual components of SOARFace based on the EER metric.

As illustrated in Fig. 8, progressively adding the proposed components results in a gradual reduction in EER. This consistent improvement demonstrates the complementary contributions of the proposed components to overall system performance. The integration of PEFT into the baseline model provides an initial performance gain, demonstrating its effectiveness in adapting the model efficiently while significantly reducing training costs. Subsequently, domain adaptation (DA) improves robustness by reducing the distribution mismatch between the training and test domains.

Few-shot adaptation further improves recognition performance when only a limited number of samples is available, while TTA provides additional EER reduction by adapting the model to the characteristics of test-time data. Furthermore, the SOAR + RL-based meta-control mechanism achieves the most notable improvement in the final configuration, as it selectively activates computationally expensive processing paths only for challenging and uncertain samples. This strategy reduces recognition errors without introducing a significant latency increase. Finally, the selective PAD module improves the system's capability to handle presentation attacks while maintaining a lower computational cost compared with continuously enabled PAD.

Overall, these findings demonstrate that the performance improvement achieved by SOARFace is the result of the complementary and cumulative contributions of its individual components. Each module plays a distinct role in reducing EER and improving the overall efficiency and reliability of the recognition system.

5. Conclusion

In this study, we developed a neuro-symbolic framework for face recognition by integrating a SOAR-based cognitive meta-controller with reinforcement learning. The proposed framework dynamically selects an appropriate processing path according to input characteristics, prediction uncertainty, and computational constraints. Unlike conventional face recognition approaches that apply a fixed processing pipeline to all samples, SOARFace adaptively activates the processing modules required for each input. This strategy enables a balance between recognition performance and computational efficiency by allocating additional processing to challenging or uncertain samples while allowing straightforward samples to follow a more efficient processing path.

A key contribution of this study is the integration of embedding-based face recognition with parameter-efficient fine-tuning (PEFT), domain adaptation, test-time adaptation (TTA), uncertainty estimation, open-set recognition, and a SOAR-based neuro-symbolic meta-controller within a unified framework. The symbolic component provides a structured representation of the decision process and supports the selection of appropriate processing actions. By combining adaptive learning with symbolic decision-making, the proposed framework provides an interpretable mechanism for dynamically managing computational resources during face recognition.

The experimental results demonstrate that SOARFace improves face recognition performance across the evaluated verification, identification, and open-set recognition scenarios. In particular, the framework achieves a lower EER in face verification and improved TPIR@FPIR performance in identification. The open-set evaluation also indicates improved discrimination between known and unknown samples. In addition, the integration of PEFT and domain adaptation facilitates adaptation to different data distributions with a reduced number of trainable parameters, while TTA further improves robustness to test-time distribution changes. The adaptive activation of processing modules also supports computationally efficient inference.

Despite these encouraging results, the proposed framework has several limitations. The development of symbolic rules within the SOAR architecture still depends, to some extent, on predefined domain knowledge. Moreover, the effectiveness of the adaptive decision-making process depends on the quality of uncertainty estimation and the identification of distribution shifts. The current evaluation is also limited to the datasets and experimental conditions considered in this study. Therefore, additional validation on larger and more diverse datasets and under more challenging real-world conditions is required to further assess the robustness and generalization capability of the framework.

Future work will focus on developing more adaptive symbolic rules and improving uncertainty estimation for more reliable decision-making. Further research will also investigate more efficient adaptation strategies for new domains and the extension of the proposed framework to video-based and multimodal face recognition. In addition, privacy-preserving learning, lifelong adaptation, and systematic evaluation under diverse deployment conditions represent promising directions for developing more reliable and practical face recognition systems.

References

[1]	Bendale, A. and Boult, T.E. (2016), Towards Open Set Deep Networks. in *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1563–1572.
[2]	Coleman, E.N. Quarantiello, L. Liu Z. Yang, Q. Mukherjee, S. Hurtado, J. and Lomonaco, V. (2025), Parameter-Efficient Continual Fine-Tuning: A Survey, *Machine Learning*, 1-57.
[3]	Deng, J. Guo, J. Xue, N. and Zafeiriou, S. (2019), ArcFace: Additive Angular Margin Loss for Deep Face Recognition. in *Proceeding of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 4690–4699.
[4]	Falade, A.A. Agarwal, G. Sanghi, A. and Gupta, A.K. (2025), An Intelligent Cognitive Framework Integrating SOAR Architecture with Large Language Models and AI Agents for Cyber Defence, *2025 IEEE 1st International Conference on Recent Trends in Computing and Smart Mobility (RCSM)*, 1-6.

[5] Galvan, A. Higuero, M. Sanz, A. Atutxa, A. Jacob, E. and Saavedra, A. (2025), Comparing CNN and ViT for Open-Set Face Recognition, *Electronics*, vol. 14, no.19, 1-19.

[6] Ganin, Y. Ustinova, E. Ajakan, H. Germain, P. Larochelle, H. Laviolette, F. Marchand, M. and Lempitsky, V. (2016), Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research (JMLR)*, vol. 17, no. 59, 1–35.

[7] Houlsby, N. Giurgiu, A. Jastrzebski, S,. Morrone, B. de Laroussilhe, Q. Gesmundo, A. Attariyan, M. and Gelly, S. (2019), Parameter-Efficient Transfer Learning for NLP. in *Proceedings of the International Conference on Machine Learning (ICML)*, 2790–2799.

[8] Hu, E.J. Shen, Y. Wallis, P. Allen-Zhu, Z. Li, Y. Wang, S. Wang, L. and Chen, W. (2022), LoRA: Low-Rank Adaptation of Large Language Models. in *Proceedings of the International Conference Learning Representations (ICLR)*.

[9] Huang, Y. Wang, Y. Wang, Y. Liu, W. and Huang, D. (2020), CurricularFace: Adaptive Curriculum Learning Loss for Deep Face Recognition. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5901–5910.

[10] Ibrahim, S.M. Ibrahim, M.S. Khan, S. Ko, Y.W. and Lee, J.G. (2025), Improving Face Presentation Attack Detection through Deformable Convolution and Transfer Learning, *IEEE ACCESS*, 31228-31238.

[11] ISO/IEC 30107-3:2017, (2017), *Information Technology — Biometric Presentation Attack Detection — Part 3: Testing and Reporting*, International Organization for Standardization.

[12] Khosla, P. Teterwak, P. Wang, C. Sarna, A. Tian, Y. Isola, P. Maschinot, A. Liu, C. and Krishnan, D. (2020), Supervised Contrastive Learning. in *Proceedings of the 34th Conference on Neural Information Processing System (NeurIPS)*.

[13] Kim, Y. Kim, J. Choi, J. and Ro, Y. (2022), AdaFace: Quality Adaptive Margin for Face Recognition. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 18750–18759.

[14] Kumar, I. Tiwari, A. Jasti, S.K. and Lade, A.H. (2026), RG-TTA: Regime-Guided Meta-Control for Test-Time Adaptation in Streaming Time Series, *Machine Learning*, 1-18.

[15] Laird, J.E. (2012), *The Soar Cognitive Architecture*. Cambridge, MA, USA: MIT Press.

[16] Lee, K. Lee, K. Lee, H. and Shin, J. (2018), A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS).*

[17] Lieder, F. and Griffiths, T.L. (2020), Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, vol. 43, p. e1.

[18] Liu, W. Wang, X. Owens, J.D. and Li, Y. (2020), Energy-based Out-of-distribution Detection. in *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS).*

[19] Liu, W. Wen, Y. Yu, Z. and Yang, M. (2017), SphereFace: Deep Hypersphere Embedding for Face Recognition. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 212–220.

[20] Long, M. Cao, Y. Wang, J. and Jordan, M.I. (2015), Learning Transferable Features with Deep Adaptation Networks. in *Proceedings of the International Conference on Machine Learning (ICML)*, 97–105.

[21] Majer, M. Cheung, V. and Lieder, F. (2023), A reinforcement-learning meta-control architecture based on the dual-process theory of moral decision-making, *37th Conference on Neural Information Processing Systems*, 1-10.

[22] Manhaeve, R. Dumancic, S. Klaesema, A.D. De Raedt, M. and De Raedt, L. (2018), DeepProbLog: Neural Probabilistic Logic Programming. in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS).*

[23] Mao, J. Gan, C. Kohli, P. Tenenbaum, J.B. and Wu, J. (2019), The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences from Natural Supervision. in *Proceedings of the International Conference on Learning Representations (ICLR).*

[24] Maze, B. Adams, J. Duncan, J.A. Kalka, N. Miller, T. Otto, C. Jain, A.K. Niggel, W.T. Anderson, J. Cheney, J. and Grother, P. (2018), IARPA Janus Benchmark–C: Face Recognition at Scale. in *Proc. IEEE Int. Conf. Biometrics Theory, Applications and Systems (BTAS)*.

[25] Rajendran, S.S.P. and Gebremedhin, A. (2026), Meta-reinforcement learning for multi-energy building

	microgrids: curriculum design, rapid adaptation, and scalable deployment, *Energy Conversion and Management*, vol. 363, 1-22.
[26]	Russell, S.J. and Wefald, E.H. (1991), Do the Right Thing: Studies in Limited Rationality. *Cambridge*, MA, USA: MIT Press, 1991.
[27]	Schneider, S. Cherepanova, A. Raghunathan, A. Kurakin, A. and Song, D. (2020), Improving Robustness against Common Corruptions with Batch Normalization. *arXiv:2006.10963*.
[28]	Schroff, F. Kalenichenko, D. and Philbin, J. (2015), FaceNet: A Unified Embedding for Face Recognition and Clustering. in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–823.
[29]	Snell, J. Swersky, K. and Zemel, R. (2017), Prototypical Networks for Few-shot Learning. in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 4077–4087.
[30]	Sun, B. and Saenko, K. (2016), Deep CORAL: Correlation Alignment for Deep Domain Adaptation. in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2016: 443–450.
[31]	Teerapittayanon, S. McDanel, B. and Kung, H.T. (2016), BranchyNet: Fast Inference via Early Exiting from Deep Neural Networks. in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2464–2469.
[32]	Tian, Z. Liu, Y. and Sun, Q. (2025), Meta-Learning Hyperparameters for Parameter Efficient Fine-Tuning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 23037-23047.
[33]	Wang, D. Shelhamer, E. Liu, S. Olshausen, B. and Darrell, T. (2021), TENT: Fully Test-Time Adaptation by Entropy Minimization. in *Proceedings of the International Conference on Learning Representations (ICLR)*.
[34]	Wang, H. Wang, Y. Zhou, Z. Ji, X. Gong, D. Zhou, J. Li, J. and Liu, W. (2018), CosFace: Large Margin Cosine Loss for Deep Face Recognition. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 5265–5274.
[35]	Wang, X. Yu, F. Dou, Z.Y. Darrell, T. and Gonzalez, J.E. (2018), SkipNet: Learning Dynamic Routing in Convolutional Networks. in *Proceedings of the European Conference on Computer Vision (ECCV),* 409–424.
[36]	Wu, Z. Nagarajan, T. Kumar, A. Rennie, S. Davis, L.S. Grauman, K. and Feris, R. (2018), BlockDrop: Dynamic Inference Paths in Residual Networks. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 8817–8826.
[37]	Yuan, F. Zeng, J. Hu, Y. Zhu, Z, Yin, Q. and Xie, Y. (2025), NL2Gen Sym: Natural Language to Generative Symbolic Rules for SOAR Cognitive Architecture via Large Language Models, *Journal of IEEE transactions on artificial, intelligence*, 1-21 (arXiv:2510.09355)