EFFICIENT FACIAL LANDMARK DETECTION VIA PRIOR KNOWLEDGE-GUIDED AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a highly efficient, agent-based framework for facial landmark detection that prioritizes model compactness and computational efficiency over maximum accuracy. Unlike conventional approaches that rely on large, fully supervised models, our method assigns each agent to a specific landmark, enabling it to infer its position solely from local observations and prior knowledge without explicit location awareness or inter-agent communication. Prior knowledge is modeled in two embedding spaces—feature and coordinate—using class-conditional Gaussian distributions. Agents navigate by minimizing deviations from these priors via a lightweight policy network. To enhance representation learning, we introduce a proximity-weighted contrastive learning strategy that incorporates spatial proximity into the training objective. A multi-stage detection strategy further reduces redundant computation by detecting sub-landmarks relative to core landmarks. While our method produces slightly higher normalized mean error than state-of-the-art (SoTA) methods, it achieves over $16\times$ and $41\times$ improvements in space and time complexities, respectively, compared to the SoTA lightweight model, running at 4.19 and 1.29 frames per second on an i5 CPU (2.5 GHz) for the COFW and 300W datasets, respectively.

1 INTRODUCTION

Facial landmark detection is a fundamental component in many computer vision applications, including face recognition (Zhao et al., 2003), expression analysis (Yang et al., 2018), and 3D face reconstruction (Liu et al., 2018). Over the past decade, advances in deep learning have greatly improved detection accuracy. However, the increasing demand for real-time performance on resource-constrained platforms, such as mobile devices, AR/VR headsets, and embedded AI modules, has shifted attention toward efficiency-oriented solutions. Such platforms impose strict limits on power consumption, computation, and memory, motivating the need for new algorithms that balance accuracy with efficiency.

Facial landmark detection aims to localize key facial points in 2D images. State-of-the-art (SoTA) methods often achieve high accuracy using large-scale models, especially those based on supervised heatmap and coordinate regression (Feng et al., 2018; Lin et al., 2021; Wang et al., 2020; Huang et al., 2021; Dang et al., 2025). However, these models incur high computational and memory costs, making them unsuitable for embedded or low-power environments. In contrast, our method adopts a lightweight, agent-based approach that leverages prior knowledge and local observations to detect landmarks efficiently. Although our normalized mean error (NME) is higher than that of SoTA models, our approach offers orders-of-magnitude gains in efficiency with only 577k parameters and < 30 MFLOPs, compared to 9.7–67M parameters and 1.2–26.8 GFLOPs for conventional methods. This trade-off makes our method highly practical for real-time, embedded applications.

Our contributions are as follows:

- We propose an agent-based framework in which each agent independently localizes a specific landmark using only local observations and learned priors without access to absolute coordinates.
- We model prior knowledge in both latent feature and coordinate spaces via class-conditional generative models, enabling effective search under weak supervision.

- We introduce a spatially aware contrastive learning method that weights positives according to spatial proximity, improving robustness in occluded or noisy conditions.
- Despite slightly higher NME than SoTA models, our method achieves $16.8\times$ lower space complexity and $41.1\times$ lower time complexity than the best lightweight baseline, enabling real-time CPU inference.

2 Related work

2.1 Regression-based methods

Facial landmark detection methods are commonly categorized into coordinate regression (CR) and heatmap regression (HR) approaches. CR methods (Feng et al., 2018; Qian et al., 2019; Lin et al., 2021) directly predict landmark coordinates using deep neural networks, learning both spatial mappings and local features. While conceptually straightforward, CR approaches are highly sensitive to noise and bias, and typically require extensive supervision to achieve accurate predictions. HR methods (Wang et al., 2020; Huang et al., 2021) generate heatmaps for each landmark, from which coordinates are derived via a decoding step. Despite their strong accuracy, HR models face two notable limitations:
Quantization error: Because heatmaps are typically of lower resolution than the input image, the decoding process introduces quantization errors (Bulat et al., 2021; Lan et al., 2021), degrading coordinate precision.
Lack of landmark correlation modeling: Standard HR methods generate heatmaps independently for each landmark, ignoring spatial relationships.
D-ViT (Dang et al., 2025) addresses this via spatial-split and channel-split vision transformers. Both CR and HR methods require full-image access and combine feature extraction with coordinate prediction in a single large model, leading to high parameter counts and computational cost.

2.2 Multiple landmark detection with agents

Detecting multiple landmarks with agents is challenging due to the need for coordination and reliance on partial visual observations. A key difficulty lies in effectively leveraging prior knowledge of both morphological and spatial correlations among landmarks. MARL (Vlontzos et al., 2019) uses a Deep Q-Network with inter-agent communication to implicitly capture morphological relationships through joint actions. The Multiscale Agent (Alansary et al., 2019) method addresses spatial relations by incorporating multiscale search. SGMaRL (Wan et al., 2023) integrates a statistical shape model (Cootes et al., 1995) to refine landmark positions based on spatial structure. While these approaches consider landmark correlations, they do so only partially—handling either morphological or spatial aspects in isolation. None fully integrate both dimensions of prior knowledge, limiting their ability to guide agents efficiently and accurately in complex visual conditions.

2.3 Contrastive learning

Contrastive learning projects representations into a latent space where similar instances are pulled together and dissimilar ones are pushed apart (Chen et al., 2020; Tian et al., 2020). Typically, two augmented views per sample yield $2N$ views for a batch of N samples. In self-supervised settings, only views from the same source are considered positive pairs. A limitation is that different-class instances may be treated as negatives, even if semantically related. Supervised Contrastive Loss (SupConLoss) (Khosla et al., 2020) addresses this by incorporating label information, allowing multiple positive pairs per class. This improves representation quality, leading to more accurate and robust classification.

3 Method

3.1 Algorithm overview

Our method employs N_c agents, each assigned to search a specific landmark ($c \in \boldsymbol{Cl}; |\boldsymbol{Cl}| = N_c$) on a given image of size $C \times H \times W$. In total, the agents simultaneously search all N_c landmarks, with

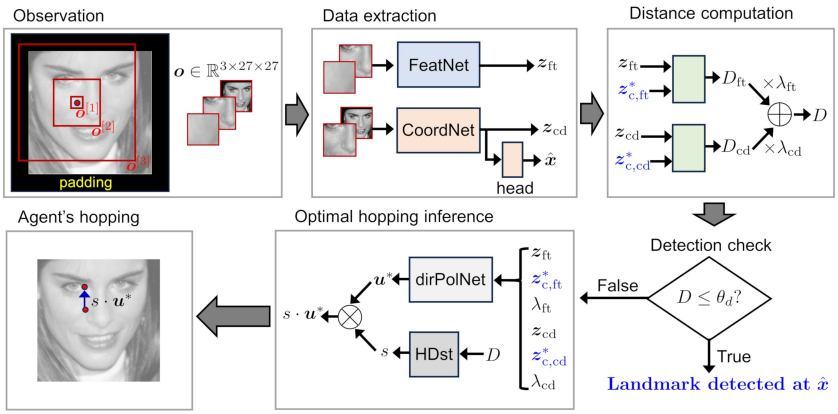

Figure 1: Overview of our algorithm for facial landmark detection.

each agent exclusively responsible for one landmark. Their coordinates $\boldsymbol{x} = (x_h, x_w)$ correspond to multiple fixation points, which are normalized to the image size ($x_h, x_w \in [-1, 1]$). We also use normalized coordinate for landmarks. Agents have access to:

1. Prior knowledge of landmarks in both the coordinate and latent feature spaces.
2. Multiscale local observations $\boldsymbol{o} \in \mathbb{R}^{3C\times a\times a}$, with $a \ll H, W$.

Agents **do not** know their absolute positions; positions are inferred from observations $\boldsymbol{o}$. No inter-agent communication occurs. Fig. 1 illustrates the framework at timestep t.

Observation by agents. To let each agent understand how the local visual structure relates to its spatial context, we consider three patches ($o^{[1]}$: $C\times a\times a$, $o^{[2]}$: $C\times s_1 a\times s_1 a$, and $o^{[3]}$: $C\times s_2 a\times s_2 a$; $1 < s_1 < s_2$) centered at its current location $\boldsymbol{x}$. The $o^{[2]}$ and $o^{[3]}$ patches are resized to $C \times a \times a$ and concatenated with the $o^{[1]}$ patch to construct the observation $\boldsymbol{o} \in \mathbb{R}^{3C\times a\times a}$. Throughout this study, we set $a = 27$, $s_1 = 4$, and $s_2 = 10$, respectively. When the patches exceeds the original image boundaries, Constant padding is applied when patches exceed image boundaries.

Data extraction from local observation. Observation $\boldsymbol{o}$ by an agent located at $\boldsymbol{x}$ is processed to extract the latent feature and coordinate of the local view in its spatial context using the following models.

- FeatNet : $\mathbb{R}^{2C\times a\times a} \to \mathbb{R}^{d_{\text{ft}}}$. FeatNet projects current local observation $\boldsymbol{o}$ by an agent at $\boldsymbol{x}$ to an embedding $\boldsymbol{z}_{\text{ft}} \in \mathbb{R}^{d_{\text{ft}}}$. This network is trained using proximity-weighted contrastive learning, which projects observation $\boldsymbol{o}$ to an embedding similar to spatially proximal landmarks. Tensor $\boldsymbol{o}_{0:2C,:,:}$ is provided as input. $\boldsymbol{z}_{\text{ft}} = \text{FeatNet}(\boldsymbol{o}_{0:2C,:,:})$. We omit $\boldsymbol{o}_{2C:3C,:,:}$ due to its minimal contribution. Unless otherwise stated, we fix $d_{\text{ft}} = 128$.
- CoordNet : $\mathbb{R}^{2C\times a\times a} \to \mathbb{R}^{2+d_{\text{cd}}}$, which infers the agent's current absolute coordinate. CoordNet function infers the agent's current coordinate $\hat{\boldsymbol{x}}$ from the local observation, which is also trained using proximity-weighted contrastive learning. Simultaneously, CoordNet projects the observation to a d_{cd}-dimensional vector $\boldsymbol{z}_{\text{cd}} (\equiv \hat{\boldsymbol{x}})$. We omit $\boldsymbol{o}_{0:C,:,:}$ due to its minimal contribution. $[\hat{\boldsymbol{x}}, \boldsymbol{z}_{\text{cd}}] = \text{CoordNet}(\boldsymbol{o}_{C:3C,:,:})$. We fix $d_{\text{cd}} = 128$.
- RelCoordNet : $\mathbb{R}^{2(C+1)\times a\times a} \to \mathbb{R}^{2+d_{\text{cd}}}$. This function infers the agent's current relative coordinate ($\Delta\hat{\boldsymbol{x}}$; $\Delta\hat{x}_h, \Delta\hat{x}_w \in [-2, 2]$) with reference to a given coordinate $\boldsymbol{x}^0$. We used RelCoordNet instead of CoordNet for landmarks with high positional variability.

Computation of deviation from knowledge. For current observation $\boldsymbol{o}$, the latent feature and coordinate embeddings ($\boldsymbol{z}_{\text{ft}}$ and $\boldsymbol{z}_{\text{cd}}$) are each compared to their respective preferred embeddings (prior knowledge $\boldsymbol{z}^*_{\text{c,ft}} \in \mathbb{R}^{d_{\text{ft}}}$ and $\boldsymbol{z}^*_{\text{c,cd}} \in \mathbb{R}^{d_{\text{cd}}}$ for landmarks in class c) to compute the distance D.

$$D = \lambda_{\text{ft}} D_{\text{ft}} + \lambda_{\text{cd}} D_{\text{cd}}, \quad D_{(\cdot)} = ||z_{(\cdot)} - z^*_{c,(\cdot)}||_2^2, \quad \text{where} \quad (\cdot) \in \{\text{ft}, \text{cd}\}\,. \tag{1}$$

The distance D is a weighted combination of distances from each embedding space, using balance parameters (λ_{ft} and λ_{cd}) such that $\lambda_{\text{ft}} + \lambda_{\text{cd}} = 1$. If the distance D is lower than a preset threshold θ_d,

Algorithm 1 Generative model for prior knowledge.

Input: Training dataset $\mathcal{T}$ of $N_{\mathcal{T}}$ samples.
Output: Generative model parameters $\boldsymbol{\mu}_{\boldsymbol{z}_{\text{ft}}|c}$, $\boldsymbol{\sigma}^2_{\boldsymbol{z}_{\text{ft}}|c}$, $\boldsymbol{\mu}_{\boldsymbol{z}_{\text{cd}}|c}$, $\boldsymbol{\sigma}^2_{\boldsymbol{z}_{\text{cd}}|c}$

$\boldsymbol{K}_{\text{ft}} \leftarrow$ zero tensor of shape $(N_{\mathcal{T}}, N_c, d_{\text{ft}})$
$\boldsymbol{K}_{\text{cd}} \leftarrow$ zero tensor of shape $(N_{\mathcal{T}}, N_c, d_{\text{cd}})$
for each sample y_i in $\mathcal{T}$ **do**
  for each landmark c_j in $\boldsymbol{C}$ **do**
    Compute $\boldsymbol{o}$ for landmark c_j in sample y_i
    $K_{\text{ft}}[i, j] \leftarrow \text{FeatNet}(\boldsymbol{o}_{0:2C,:,:})$
    $K_{\text{cd}}[i, j] \leftarrow \text{CoordNet}(\boldsymbol{o}_{C:3C,:,:})$
  end for
end for
$\boldsymbol{\mu}_{\boldsymbol{z}_{(\cdot)}|c_j} \leftarrow K_{(\cdot)}.\text{mean}(\text{dim} = 0)$ for $(\cdot) \in \{\text{ft}, \text{cd}\}$
$\boldsymbol{\sigma}^2_{\boldsymbol{z}_{(\cdot)}|c_j} \leftarrow K_{(\cdot)}.\text{var}(\text{dim} = 0)$ for $(\cdot) \in \{\text{ft}, \text{cd}\}$

the agent has successfully arrived at the landmark, outputting its current coordinate ($\hat{\boldsymbol{x}}$ if CoordNet was used, and $\boldsymbol{x}^0 + \Delta\hat{\boldsymbol{x}}$ if RelCoordNet was used). Otherwise, the agent continues searching.

Hopping policy. The agent infers optimal hopping direction and distance using PolNet that is factorized into two sub-functions (dirPolNet and HDst): PolNet = HDst · dirPolNet.

- dirPolNet : $\mathbb{R}^{2d_{\text{ft}}+2d_{\text{cd}}+2} \rightarrow \mathbb{R}^8$. dirPolNet is a categorical classifier that infers the optimal hopping direction $\boldsymbol{u}^* \in \boldsymbol{U}$. We defined the direction space $\boldsymbol{U}$ as follows: $\boldsymbol{U} = \{(u_1, u_2) \,|\, (u_1 \in \boldsymbol{U}_0 \vee u_2 \in \boldsymbol{U}_0) \wedge (u_1, u_2 \neq 0)\}$, where $\boldsymbol{U}_0 = \{-1, 0, 1\}$. Therefore, $|\boldsymbol{U}| = 8$. This model takes $\boldsymbol{z}_{\text{ft}}/\boldsymbol{z}^*_{\text{c,ft}}/\lambda_{\text{ft}}$ and $\boldsymbol{z}_{\text{cd}}/\boldsymbol{z}^*_{\text{c,cd}}/\lambda_{\text{cd}}$ for the current observation $\boldsymbol{o}$ as input.
- HDst : $\mathbb{R} \rightarrow \mathbb{Z}_{>0}$. This function determines the hopping distance $s \in [1, s_{\text{max}}]$ as a function of the distance D.

3.2 Prior knowledge modeling

We built generative models for prior knowledge of landmarks projected to two independent spaces (latent feature space $\mathbb{R}^{d_{\text{ft}}}$ and coordinate space $\mathbb{R}^{d_{\text{cd}}}$).

$$p\left(\boldsymbol{z}_{\text{ft}}, c\right) = p\left(\boldsymbol{z}_{\text{ft}}|c\right) p\left(c\right) \quad \text{for latent features,} \quad p\left(\boldsymbol{z}_{\text{cd}}, c\right) = p\left(\boldsymbol{z}_{\text{cd}}|c\right) p\left(c\right) \quad \text{for coordinate,}$$

where c denotes a landmark class. We modeled the class-conditional probability distribution function $p\left(\boldsymbol{z}|c\right)$ using a Gaussian function:

$$p\left(\boldsymbol{z}|c\right) = \mathcal{N}(\boldsymbol{\mu}_{z|c}, \boldsymbol{\Sigma}_{z|c}) \approx \mathcal{N}(\boldsymbol{\mu}_{z|c}, \boldsymbol{\sigma}^2_{z|c}\boldsymbol{I}), \tag{2}$$

where we apply a diagonal approximation to the covariance matrix $\Sigma_{\boldsymbol{z}|c}$ for computational simplicity. The parameters $\boldsymbol{\mu}_{z|c}$ and $\boldsymbol{\sigma}^2_{z|c}$ were separately computed for each of the two embeddings (to $\boldsymbol{z}_{\text{ft}}$ and $\boldsymbol{z}_{\text{cd}}$) over all landmarks of the same class in a given training dataset. This procedure is detailed in **Algorithm** 1. Prior knowledge $\boldsymbol{z}^*_{\text{c,ft}}$ and $\boldsymbol{z}^*_{\text{c,cd}}$ is sampled from the conditional generative model in Eq. 2.

3.3 Network models

FeatNet. This model (2C9(6C9)-9C16-16C32-32C64-FC256-FC128-L2Norm for gray-scale (RGB) images) projects the local observation $\boldsymbol{o}_{0:2C,:,:}$ by an agent at $\boldsymbol{x}$ to the feature space $\mathbb{R}^{d_{\text{ft}}}$. Inspired by supervised contrastive learning (Khosla et al., 2020), we trained FeatNet using a novel proximity-weighted contrastive learning algorithm that locates the embedding $\boldsymbol{z}_{\text{ft}}$ for a given observation close to landmarks of spatial proximity. To this end, all landmarks within a $o^{[2]}$ patch of $C \times s_1 a \times s_2 a$ size centered at given $\boldsymbol{x}$ are considered as positive landmarks while the others as negative ones. Furthermore, we defined the degree of positiveness for the positive landmarks based on their distances from the coordinate $\boldsymbol{x}$.

For proximity-weighted contrastive learning, the ith sample in a given batch $\mathcal{B}$ includes a single anchor at $\boldsymbol{x}_A^{(i)}$, which is placed on a landmark $c_{A(i)}$ (at $\boldsymbol{x}_{c_{A(i)}}^{(i)}$) that is randomly sampled from total N_c landmarks ($\boldsymbol{x}_A^{(i)} = \boldsymbol{x}_{c_{A(i)}}^{(i)}$). We define an augmented batch $\mathcal{B}'$ of the same samples (and sequence) as $\mathcal{B}$ but with a random anchor for each sample, and thus unnecessarily $\boldsymbol{x}_A^{(i)} = \boldsymbol{x}_{c_{A(i)}}^{(i)}$. The key is the use of a proximity-weighted contrastive loss (PWConLoss) for $\mathcal{B}$ and $\mathcal{B}'$.

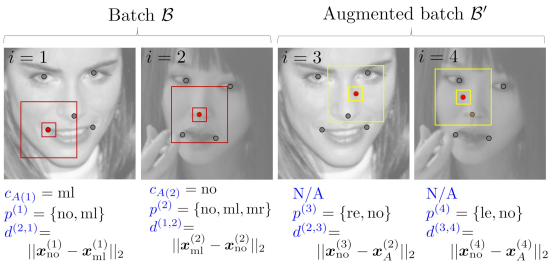

Figure 2: Example of sample augmentation for proximity-weighted contrastive learning. Right eye, left eye, nose, mouth left, and mouth right are denoted by `re`, `le`, `no`, `ml`, and `mr`, respectively.

$$\mathcal{L}^{\text{PWS}} = \frac{-1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\Bigg[\underbrace{\frac{1}{N^{(i)}}\sum_{j\in\mathcal{B}\setminus\{i\}} w^{(i,j)}\mathbb{1}_{\left\{c_{A(i)}\in\boldsymbol{p}^{(j)}\right\}}l^{(i,j)}}_{\text{between } c_{A(i)} \text{ and } c_{A(j)}\ (j\neq i)} + \underbrace{\frac{1}{N'^{(i)}}\sum_{j\in\mathcal{B}'} w^{(i,j)}\mathbb{1}_{\left\{c_{A(i)}\in\boldsymbol{p}^{(j)}\right\}}l^{(i,j)}}_{\text{between } c_{A(i)} \text{ and random observations } \boldsymbol{o}}\Bigg],$$

$$N^{(i)} = \sum_{j\in\mathcal{B}\setminus\{i\}}\mathbb{1}_{\left\{c_{A(i)}\in\boldsymbol{p}^{(j)}\right\}},\ N'^{(i)} = \sum_{j\in\mathcal{B}'}\mathbb{1}_{\left\{c_{A(i)}\in\boldsymbol{p}^{(j)}\right\}},\ l^{(i,j)} = \log\frac{\exp(\boldsymbol{z}_A^{(i)}\cdot\boldsymbol{z}_A^{(j)}/\tau)}{\sum_{k\in\mathcal{B}_{\text{s}}\setminus\{i\}}\exp(\boldsymbol{z}_A^{(i)}\cdot\boldsymbol{z}_A^{(k)}/\tau)}, \tag{3}$$

where $\boldsymbol{p}^{(j)} = \left\{c\in\boldsymbol{Cl}|\boldsymbol{x}_c^{(j)}\in o^{[2]} \text{ for } \boldsymbol{x}_A^{(j)}\right\}$. The weight $w^{(i,j)}$ is given by

$$w^{(i,j)} = 1 + \exp\left(-0.025d^{(i,j)}\right), \tag{4}$$

where $d^{(i,j)}$ denotes the distance between $\boldsymbol{x}_{c_{A(i)}}^{(j)}$ and $\boldsymbol{x}_A^{(j)}$. Note that $\boldsymbol{x}_{c_{A(i)}}^{(j)}$ means the coordinate of the anchor landmark type of the ith sample $c_{A(i)}$ on the jth sample. For $j\in\mathcal{B}\setminus\{i\}$, the equality $\boldsymbol{x}_A^{(j)} = \boldsymbol{x}_{c_{A(j)}}^{(j)}$ holds. In Eq. 3, $\mathcal{B}_{\text{s}} = \text{concat}(\mathcal{B},\mathcal{B}')$, and $\boldsymbol{z}_{\text{c,ft}}^{(i)}$ and τ denote the embedding of the anchor in the ith sample and temperature, respectively. An example of sample augmentation for proximity-weighted contrastive learning is shown in Fig. 2.

CoordNet/RelCoordNet. CoordNet projects the local observation $\boldsymbol{o}_{C:3C,:,:}$ by an agent at $\boldsymbol{x}$ to the coordinate space $\mathbb{R}^{d_{\text{cd}}}$. It consists of four convolutional layers and one linear layer: 2C9(6C9)-9C16-16C32-32C64-FC128-L2Norm for gray-scale (RGB) images. We deploy an additional head for coordinate regression, Linear$(128\rightarrow 2)$ + Tanh, which infers the normalized coordinate $\hat{\boldsymbol{x}}$ ($\hat{x}_h,\hat{x}_h\in[-1,1]$) for the agent at $\boldsymbol{x}$. Similar to FeatNet, the main network (except the head) is trained using proximity-weighted contrastive learning using the PWConLoss in Eq. 3 with a weight function $w^{(i,j)} = 2 + 9.26\cdot 10^{-3}d^{(i,j)}$ instead of Eq. 4. However, the anchor for each samples in batch $\mathcal{B}$ is placed on a random coordinate on the sample unlike FeatNet (for which the anchor is on a landmark only), and thus, the embedding $\boldsymbol{z}_{\text{cd}}$ for a given observation becomes similar to other observations of spatial proximity. The additional head is trained using a mean squared error loss function (MSELoss).

RelCoordNet has the same architecture as CoordNet except the first convolutional layer and head for coordinate regression: 4C9(8C9)-9C16-16C32-32C64-FC128–L2Norm-FC2-2Tanh for gray-scale (RGB) images. RelCoordNet takes the reference coordinate $\boldsymbol{x}^0$ (represented using $x_w^0\mathbf{1}_{1\times a\times a}$ and $x_h^0\mathbf{1}_{1\times a\times a}$; $x_w^0, x_h^0\in[-1,1]$) as its input alongside the local observation $\boldsymbol{o}_{C:3C,:,:}$, so that the input consists of $2C+2$ channels. Instead of Tanh, 2Tanh is applied because of the range of relative coordinate $\Delta\hat{\boldsymbol{x}}$ ($\Delta\hat{x}_h,\Delta\hat{x}_w\in[-2,2]$). This model is also trained using proximity-weighted contrastive learning using PWConLoss in Eq. 3 with random anchors for the samples in batch $\mathcal{B}$ as for CoordNet. The head is trained using MSELoss.

Algorithm 2 Delayed decision algorithm.

Input: $\hat{\mathbf{\Lambda}}$, D_{ft}, D_{cd}, θ_{d}, λ_{ft}, $\hat{\boldsymbol{x}}$
Output: Updated SHT and $\hat{\boldsymbol{x}}_c$

$D_{\text{min}} \leftarrow$ MAX; $\hat{\boldsymbol{x}}_c \leftarrow$ NULL
for $i = 0$ to $N_\lambda - 1$ **do**
  $D_{\text{tmp}} \leftarrow \hat{\mathbf{\Lambda}}[i] D_{\text{ft}} + (1 - \hat{\mathbf{\Lambda}}[i]) D_{\text{cd}}$
  if SHT$[i, 0] > D_{\text{tmp}}$ **then**
    SHT$[i, 0] \leftarrow D_{\text{tmp}}$; SHT$[i, 1] \leftarrow \hat{\boldsymbol{x}}$
  end if
end for
$i \leftarrow 0$
while $\hat{\mathbf{\Lambda}}[i] \geq \lambda_{\text{ft}}$ **do**
  if SHT$[i, 0] \leq \theta_{\text{d}}$ and SHT$[i, 0] \leq D_{\text{min}}$ **then**
    $\hat{\boldsymbol{x}}_c \leftarrow \hat{\boldsymbol{x}}$; $D_{\text{min}} \leftarrow$ SHT$[i, 0]$
  end if
  $i \leftarrow i + 1$
end while

PolNet. PolNet infers the optimal hopping direction $\boldsymbol{u}^*$ and distance $s \in [1, s_{\text{max}}]$ for the current local observation $\boldsymbol{o}$ using its sub-functions, dirPolNet and HDst, respectively.

$$\begin{aligned} &\text{PolNet} = \text{HDst}(D) \cdot \text{dirPolNet}(I), \\ &I = \text{concat}(\boldsymbol{z}_{\text{ft}}, \boldsymbol{z}^*_{\text{c,ft}}, \lambda_{\text{ft}}, \boldsymbol{z}_{\text{cd}}, \boldsymbol{z}^*_{\text{c,cd}}, \lambda_{\text{cd}}), \\ &\text{HDst}(D) = \min\left(\left(\lceil D/\Delta_D \rceil\right), s_{\text{max}}\right), \end{aligned} \tag{5}$$

where Δ_D denotes a unit step for uniform quantization of distance D in Eq. 1. dirPolNet (FC512-FC256-FC8-Softmax) infers the optimal hopping direction $\boldsymbol{u}^* \in \boldsymbol{U}$, which is trained using supervised learning on a dataset $\mathcal{T}$. We define $\boldsymbol{U} = \{(u_1, u_2) \,|\, (u_1 \in \boldsymbol{U}_0 \vee u_2 \in \boldsymbol{U}_0) \wedge (u_1, u_2 \neq 0)\}$, where $\boldsymbol{U}_0 = \{-1, 0, 1\}$. That is, $|\boldsymbol{U}| = 8$. For a given image, a pair of coordinate $\boldsymbol{x}$ and landmark $c \in \boldsymbol{Cl}$ are randomly sampled. Similar to the habitual network (Cushman & Morris, 2015), each sample $\boldsymbol{y}_i = (I_i, \hat{u}_i)$ in $\mathcal{T}$ consists of (1) input I_i (in Eq. 5) for the local observation $\boldsymbol{o}$ at the random coordinate $\boldsymbol{x}$ and (2) $\hat{u}_i = \arg\min A_{u \in \boldsymbol{U}} D$ for the coordinate $\boldsymbol{x}$ and landmark c.

3.4 Hyperparameter setting

Detection threshold. Detection threshold θ_d is a primary hyperparameter that determines the detection accuracy and speed, which are measured in NME and duration required for detection (T_{d}), respectively. A lower θ_{d} generally yields a lower NME but a larger T_{d}. To balance this trade-off, threshold θ_{d} is initially set to its minimum ($\theta_{\text{d,min}}$) and is monotonously increased by $\Delta\theta_{\text{d}}$ at each timestep if detection fails.

Balance parameters. Balance parameters λ_{ft} and $\lambda_{\text{cd}} (= 1 - \lambda_{\text{ft}})$ in Eq. 1 govern the complementary contributions of the distances from distinct representation spaces (D_{ft} and D_{cd}) to the total distance D. A higher λ_{ft} generally yields a lower NME but a larger T_{d}. We initially set λ_{ft} is initially set to its maximum($\lambda_{\text{ft,max}}$ and monotonically decrease by $\Delta\lambda_{\text{ft}}$ once every two timesteps if detection fails.

3.5 Efficiency Enhancements

Delayed decision algorithm. Critical detection inefficiency with varying hyperparameters (θ_{d} and λ_{ft}) arises when previously visited locations with previous hyperparameter values yield successful detection with the current θ_{d}. Revisiting such locations and repeatedly recalculating the distance D leads to redundant computation. To mitigate this issue, we introduce a delayed decision algorithm for the following case.

$$\theta_{\text{d}}[t'] < D[t'] = \sum_{i \in \{\text{ft,cd}\}} \lambda_i[t'] D_i[t'] \leq \theta_{\text{d}}[t] \text{ for } t' < t.$$

We define a balance parameter set $\Lambda = \{\lambda[t] | \forall t \in [0, t_{\text{max}}]\}$ ($|\Lambda| = N_\lambda$) and a corresponding tuple $\hat{\mathbf{\Lambda}}$ of $\lambda (\in \mathbf{\Lambda})$ sorted in descending order. This delayed decision algorithm is based on a $N_\lambda \times 2$

search history table (SHT). **Algorithm** 2 explains SHT organization and delayed decision based on the SHT.

Two-stage detection. To reduce the detection duration T_{d}, we introduce a two-stage detection strategy for all landmarks. The first stage detects coarse coordinates of landmarks using the higher detection threshold $\theta_{\mathrm{d}}^{(1)}$ and lower balance parameter $\boldsymbol{\lambda}_{\mathrm{ft}}^{(1)}$, and the second stage refines the coordinate using the lower threshold $\theta_{\mathrm{d}}^{(2)}(<\theta_{\mathrm{d}}^{(1)})$ and higher balance parameter $\boldsymbol{\lambda}_{\mathrm{ft}}^{(2)}(>\boldsymbol{\lambda}_{\mathrm{ft}}^{(1)})$. In each stage, the hyperparameters change following the rule explained in the previous section.

Cascaded detection. Facial landmarks can be grouped based on their spatial proximity in the latent feature space $\mathbb{R}^{d_{\mathrm{ft}}}$. We choose a single core-landmark for each group and considered the others in the same group as sub-landmarks. Agents responsible for detecting landmarks in the same group often exhibit overlapping trajectories in the initial detection phase, leading to redundant computation. To address this, we propose a cascaded detection strategy. In this strategy, a single agent first detects the core-landmark using FeatNet and CoordNet for local observation $\boldsymbol{o}$ at each timestep t. This core-landmark serves as a reference. Subsequently, multiple agents simultaneously search the sub-landmarks with reference to the core-landmark. This approach mitigates redundant agent movements during the early detection steps, thereby reducing redundant computation and processing time. Fig. 3 shows an example of cascaded detection for a Right-eye group. Note that we use RelCoordNet in place of CoordNet for the sub-landmarks in a particular group.

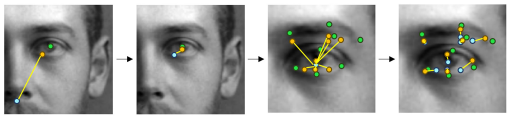

Figure 3: Example of cascaded detection.

4 Experimental Results

We used the COFW (Burgos-Artizzu et al., 2013) and 300W (Sagonas et al., 2016) datasets as a proof of concept. COFW comprises 1,345 training and 507 test images (gray-scale), each annotated with 29 landmarks. COFW with frequent occlusions is well-suited for evaluating our method's ability to leverage prior knowledge of landmarks. 300W comprises 3,148 training and 689 test images (RGB), each annotated with 68 landmarks. This dataset exhibits a wide range of variations in pose and lighting conditions.

4.1 Implementation Details

Each image is cropped to include the full head, resized to 256×256, then randomly rescaled ($\pm 5\%$) and horizontally flipped (50%). For cascaded detection, the landmarks in each dataset are grouped based in their spatial proximity as follows.
COFW: `left_eye` (left pupil), `right_eye` (right pupil), and `others` (nose tip)
300W: `left_eye` (left inner canthus), `right_eye` (right inner canthus), `mouth` (Cupid's bow), `nose` (nose tip), `jaw_line` (none).
The landmarks in parentheses indicate core landmarks. We used different FeatNets with different sets of parameters (but the same CoordNet and RelCoordNet) for each group. For the Others group in COFW, RelCoordNet was used for detecting the sub-landmarks. Note that we used the means $\boldsymbol{\mu}_{\boldsymbol{z}_{\mathrm{ft|c}}}$ and $\boldsymbol{\mu}_{\boldsymbol{z}_{\mathrm{cd|c}}}$ as prior knowledge $\boldsymbol{z}_{\mathrm{c,ft}}^{*}$ and $\boldsymbol{z}_{\mathrm{c,ft}}^{*}$, respectively, unless otherwise stated. The models were trained using the Pytorch framework (Paszke et al., 2019) on a GPU workstation (RTX A6000; Xeon Gold CPU 2.9GHz; 256 GB DRAM). Landmark detection experiments were conducted on both the GPU workstation and a desktop equipped with an i5 CPU (2.5GHz) and 32 GB DRAM. The hyperparameters were optimized using Optuna (Akiba et al., 2019). All hyperparameters are summarized in Technical Appendix.

4.2 Performance of Network Models

FeatNet. We analyzed a fully trained FeatNet whose learning curve is shown in Technical Appendix. Fig. 4**a** shows L2 distance between landmarks in the same classes in COFW, identifying successful landmark clustering. Fig. 4**b** shows L2 distance between a left pupil (Class 17) and the others on the

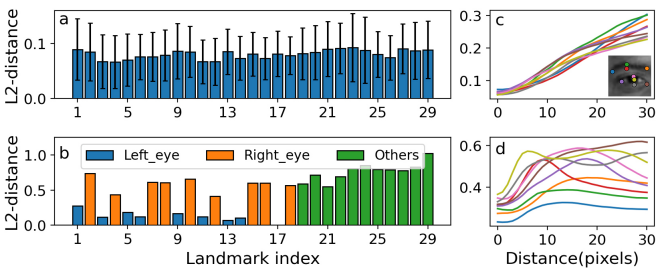

Figure 4: Performance of FeatNet embeddings. (**a**) L2 distances between embeddings of landmarks within the same class. (**b**) L2 distance between a left-pupil embedding and embeddings of other landmarks. (**c**) L2 distances as a function of spatial distance from a given landmark. (**d**) Comparison with a supervised contrastive learning baseline.

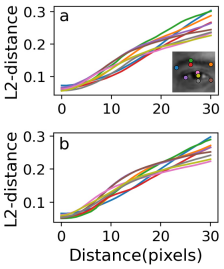

Figure 5: L2 distances between coordinate embeddings and landmarks at varying spatial distances from the landmarks in the inset for (**a**) CoordNet and **(b)** RelCoordNet.

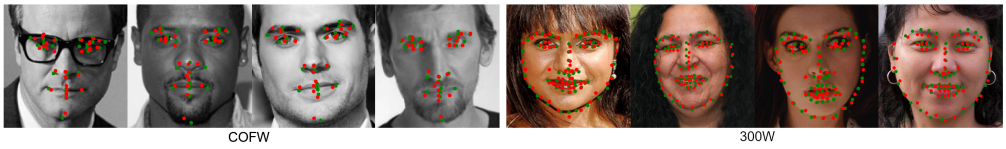

Figure 6: Detected landmarks (red circles) and ground-truth annotations (green circles) on sample images from COFW and 300W.

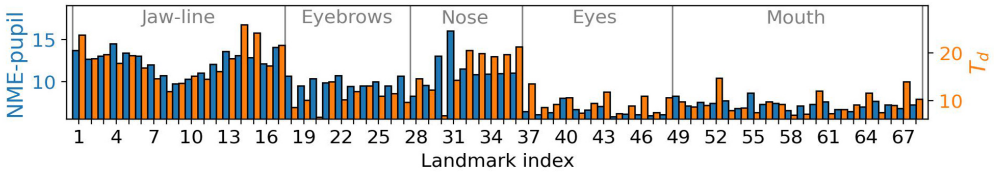

Figure 7: Average detection accuracy (NME) and detection duration for each landmark on 300W.

same image. This identifies the separation of landmark clusters based on their spatial proximity. We analyzed the capability of FeatNet to encode observations $\boldsymbol{o}$ at random coordinates $\boldsymbol{x}$ as $\boldsymbol{z}_{\text{ft}}$ based on their distances from landmarks. Fig. 4**c** highlights (1) a gradual increase in L2 distance with distance between the observation $\boldsymbol{o}$ and landmark and (2) marginal variability in L2 distance at aero distance upon different landmarks. As a counterpart, we used the same network trained with supervised contrastive learning, where spatial proximity was addressed in binary form, employing the same loss as in Eq. 3 but without the proximity weight $w^{(i,j)}$. The performance of this counterpart is shown in Fig. 4**d**. The comparison with this counterpart highlights proximity-weighted contrastive learning as a means to encode random observations based on spatial separation from the landmarks.

CoordNet/RelCoordNet. Fully trained CoordNet and RelCoordNet (whose learning curves are shown in Technical Appendix) successfully infer the coordinate of the current observation as $\boldsymbol{z}_{\text{cd}}$-dimensional embeddings. Fig. 5 shows the L2 distance between the coordinate embedding $\boldsymbol{z}_{\text{cd}}$ and several landmarks (in the inset) with spatial distance. Compared with Fig. 4, CoordNet/RelCoordnet can infer the coordinate of distal observations with higher precision (lower variability).

4.3 Detection Precision and Efficiency

The detected landmarks on several samples in COFW and 300W are shown in Fig. 6, demonstrating successful landmark detection using our algorithm. Nevertheless, our method has higher NME than regression-based SoTA techniques as listed in Table 1. This is largely due to the fact that our approach rely on not supervised learning but prior knowledge of landmarks' features. For instance, although our method well detects the jaw-line landmarks in 300W samples (Fig. 6), NME for these landmarks is large due to their deviation from the semi-automatically annotated jaw-line landmarks

Table 1: Comparison of our method with SoTA approaches on COFW and 300W datasets. The NME value in parentheses for 300W excludes the `jaw_line` landmarks.

Method	COFW		300W	# Params (M)	FLOPs	Duration T_d
	NME-ocular	NME-pupil	NME-pupil			
LAB (Wu et al., 2018)	3.92	5.58	3.49	25.1	18.9G	-
AWing (Wang et al., 2019)	-	4.94	3.07	24.2	26.8G	-
AVS (Qian et al., 2019)	-	4.43	3.86	28.3	2.40G	-
HRNet (Wang et al., 2020)	3.45	-	3.32	9.66	4.75G	-
PIP (Jin et al., 2021)	-	-	3.36	12.0	2.40G	-
ADNet (Huang et al., 2021)	-	4.69	2.93	13.4	17.0G	-
SDFL (Lin et al., 2021)	3.63	-	-	-	5.17G	-
HIH (Lan et al., 2021)	3.21	4.63	3.09	22.7	17.2G	-
SLPT (Xia et al., 2022)	3.32	4.63	3.17	13.2	6.12G	-
STARLoss (Zhou et al., 2023)	-	4.62	2.87	13.4	-	-
D-ViT (Dang et al., 2025)	-	4.13	2.85	67.3	21.8G	-
PoPos (Xiang et al., 2025)	-	3.80	3.28	9.70	1.20G	-
Ours on COFW/300W	**8.28**	**11.96**	**9.36 (8.33)**	**0.577**	**21.1M/29.1M**	**10.42/12.77**

Table 2: FPS on different processors.

	CPU i5 2.5GHz	CPU Xeon 2.9GHz	GPU A6000
COFW	4.19±0.11	3.00 ± 0.14	19.73 ± 1.15
300W	1.29±0.19	1.25 ± 0.15	5.21 ± 0.68

on 300W samples. Fig. 7 plots the average NME and detection duration T_d for each landmark on 300W sample, identifying high NME for the `jaw_line` landmarks.

However, our method demonstrates significantly low computational complexity.
Space complexity: Total 577k parameters (FeatNet/CoordNet/PolNet with 123k/58k/396k parameters).
Time complexity: Total 21.1 MFLOPs for a 10.42 T_d on COFW and 29.1 MFLOPs for a 12.77 T_d on 300W. Relative to the lightweight PoPos (Xiang et al., 2025) model with 9.70M parameters, our method reduces space complexity by $16.8\times$ and time complexity by $41.1\times$. Our method with extremely low complexity runs at 4.19 (COFW) and 1.29 (300W) frames per second (FPS) on a desktop with an i5-13400 CPU (Table 2).

4.4 Ablation Study

The balance parameter λ_{ft} is an important hyperparameter that governs both detection accuracy and duration, and its value is scheduled over time. We analyzed the impact of $\boldsymbol{\lambda}_{\text{ft}}$ on detection accuracy and duration by varying its value over the range $[0, 1]$ while keeping it fixed during each detection run. As shown in Fig. 8, $\boldsymbol{\lambda}_{\text{ft}}$ exhibits a clear trade-off between accuracy and duration, highlighting the need for parameter scheduling to achieve optimal performance.

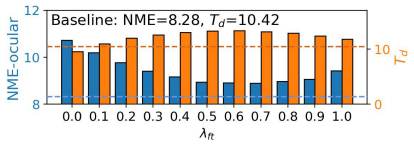

Figure 8: Relationship between detection accuracy and detection duration for different fixed values of λ_{ft} on COFW.

5 Conclusion

We proposed a lightweight, agent-based framework for facial landmark detection that leverages prior knowledge and local observations without relying on strong supervision. Each agent infers its location independently using embeddings from dual spaces—feature and coordinate—guided by class-conditional generative models. To train robust embeddings, we introduced proximity-weighted contrastive learning, and we further improved efficiency with a multi-stage detection strategy and delayed decision mechanism to reduce redundant computation. While our method shows slightly higher NME than SoTA approaches, it achieves exceptional efficiency by reducing space complexity by $16.8\times$ and time complexity by $41.1\times$ compared to the SoTA lightweight model, making it ideal for real-time or embedded applications. This work demonstrates that prior knowledge-guided agent-based detection is a practical and scalable alternative for efficient landmark localization.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Amir Alansary, Ozan Oktay, Yuanwei Li, Loic Le Folgoc, Benjamin Hou, Ghislain Vaillant, Konstantinos Kamnitsas, Athanasios Vlontzos, Ben Glocker, Bernhard Kainz, et al. Evaluating reinforcement learning agents for anatomical landmark detection. *Medical image analysis*, 53: 156–164, 2019.

Adrian Bulat, Enrique Sanchez, and Georgios Tzimiropoulos. Subpixel heatmap regression for facial landmark localization. *arXiv preprint arXiv:2111.02360*, 2021.

Xavier P Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *Proceedings of the IEEE international conference on computer vision*, pp. 1513–1520, 2013.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61:38–59, 1995.

Fiery Cushman and Adam Morris. Habitual control of goal selection in humans. *Proceedings of the National Academy of Sciences*, 112:13817–13822, 2015.

Ziqiang Dang, Jianfang Li, and Lin Liu. Cascaded dual vision transformer for accurate facial landmark detection. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5884–5894. IEEE, 2025.

Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2235–2245, 2018.

Yangyu Huang, Hao Yang, Chong Li, Jongyoo Kim, and Fangyun Wei. Adnet: Leveraging error-bias towards normal direction in face alignment. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3080–3090, 2021.

Haibo Jin, Shengcai Liao, and Ling Shao. Pixel-in-pixel net: Towards efficient facial landmark detection in the wild. *International Journal of Computer Vision*, 129:3174–3194, 2021.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Xing Lan, Qinghao Hu, Qiang Chen, Jian Xue, and Jian Cheng. Hih: Towards more accurate face alignment via heatmap in heatmap. *arXiv preprint arXiv:2104.03100*, 2021.

Chunze Lin, Beier Zhu, Quan Wang, Renjie Liao, Chen Qian, Jiwen Lu, and Jie Zhou. Structure-coherent deep feature learning for robust face alignment. *IEEE Transactions on Image Processing*, 30:5313–5326, 2021.

Feng Liu, Qijun Zhao, Xiaoming Liu, and Dan Zeng. Joint face alignment and 3d face reconstruction with application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 42:664–678, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Shengju Qian, Keqiang Sun, Wayne Wu, Chen Qian, and Jiaya Jia. Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10153–10163, 2019.

Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, 47:3–18, 2016.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 776–794. Springer, 2020.

Athanasios Vlontzos, Amir Alansary, Konstantinos Kamnitsas, Daniel Rueckert, and Bernhard Kainz. Multiple landmark detection using multi-agent reinforcement learning. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part IV 22*, pp. 262–270. Springer, 2019.

Kaiwen Wan, Lei Li, Dengqiang Jia, Shangqi Gao, Wei Qian, Yingzhi Wu, Huandong Lin, Xiongzheng Mu, Xin Gao, Sijia Wang, et al. Multi-target landmark detection with incomplete images via reinforcement learning and shape prior embedding. *Medical Image Analysis*, 89: 102875, 2023.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43: 3349–3364, 2020.

Xinyao Wang, Liefeng Bo, and Li Fuxin. Adaptive wing loss for robust face alignment via heatmap regression. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6971–6981, 2019.

Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2129–2138, 2018.

Jiahao Xia, Weiwei Qu, Wenjian Huang, Jianguo Zhang, Xi Wang, and Min Xu. Sparse local patch transformer for robust face alignment and landmarks inherent relation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4052–4061, 2022.

Chong-Yang Xiang, Jun-Yan He, Zhi-Qi Cheng, Xiao Wu, and Xian-Sheng Hua. Popos: Improving efficient and robust facial landmark detection with parallel optimal position search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8602–8610, 2025.

Huiyuan Yang, Umur Ciftci, and Lijun Yin. Facial expression recognition by de-expression residue learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2168–2177, 2018.

Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35:399–458, 2003.

Zhenglin Zhou, Huaxia Li, Hong Liu, Nanyang Wang, Gang Yu, and Rongrong Ji. Star loss: Reducing semantic ambiguity in facial landmark detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15475–15484, 2023.