SapiensID: Foundation for Human Recognition

Supplementary Material

A. Method Details

A.1. Training Details

The training pipeline of SapiensID is largely similar to the setting of training a ViT model in face recognition [37]. This is possible because WebBody4M is a labeled dataset with a sufficient number of subjects, just as face recognition datasets. We use the AdaFace [34] loss and optimize the model with the AdamW [49] optimizer for 33 epochs. The learning rate is scheduled by the Cosine Annealing Learning Rate Scheduler [48] with an additional warm-up period of 3 epochs. The maximum learning rate is set to 0.0001. We use 7 A100 GPUs with a batch size of 128. We also change the classifier to PartialFC [2] with a sampling ratio of 0.1 to save GPU memory and gain computation efficiency. Overview of the model is shown in Fig. 8.

For data augmentation, we find that it is important to use a moderate amount of geometric augmentation (zoom in-out: $0.9 \sim 1.1$, translation: ± 0.05) and aspect ratio adjustments ($0.95 \sim 1.05$). We also find it effective for improving aligned face recognition performance to include face-zoomed-in images frequently (40%). We also oversample images that contain more visible keypoints because those images are relatively scarce (note Tab. 16).

A.2. Notation Clarification in the Main Paper

In Semantic Attention Pooling's SAH, the equation presented as Eq. 8:

$$\mathbf{O}^i_{\text{part}} = \text{Attention}\left(\mathbf{Q}^i_{kp}, \text{PE}, \text{backbone}(\mathbf{X}^i)\right), \tag{11}$$

$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ is specifically defined as:

$$\mathbf{O}^i_{\text{part}} = \text{softmax}\left(\frac{\mathbf{W}_q\mathbf{Q}\mathbf{W}_k\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{W}_v\mathbf{V}, \tag{12}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ represent the query, key, and value matrices, respectively, and $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$ are their associated projection weights. This is how the size of the attenion is modulated during learning.

Also notice that without the learnable projections $\mathbf{W}_{q,k,v}$ and a small d, the attention simply focuses on the position with the highest proximity to the keypoint. To make sure that we have this feature from the sharp peak at the keypoint location, we additionally use

$$\mathbf{O}^i_{\text{peak}} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}. \tag{13}$$

The final feature vector is computed by concatenating the two sets of semantic features $\mathbf{O}^i_{\text{part}}$ and $\mathbf{O}^i_{\text{peak}}$ and flattening them for MLP projection. Specifically, it is

$$f^i = \text{MLP}(\text{flatten}([\mathbf{O}^i_{\text{part}}, \mathbf{O}^i_{\text{peak}}])). \tag{14}$$

The addition of $\mathbf{O}^i_{\text{peak}}$ is simply to ensure that the model always has the feature from the keypoint location. We have not tested how much performance gap is created by removing this inductive bias in SAH. The final number of part features is 152 (19 keypoints $\times$ 4 offset repeats $\times$ 2 from concatenating $\mathbf{O}^i_{\text{part}}$ and $\mathbf{O}^i_{\text{peak}}$. We realize that the readers could be confused about the formulation of SAH attention, so we will make it clearer in the main paper.

A.3. Things We Tried That Did Not Make it into the Main Algorithm

- We tried to initialize the model with the Sapiens [33] pretrained backbone, thinking it would be a good starting point that leads to better generalization. However, it did not lead to better performance. We believe this is because: 1) our patch scheme is dramatically different from the original patch scheme, and 2) Sapiens is trained with the MAE [22] objective, which is suitable for dense prediction tasks. However, SapiensID is a classification (or metric learning) task. Dense prediction tasks prioritize spatial consistency and detailed reconstruction, whereas classification tasks focus on extracting discriminative features, which may require different feature representations.
- We tried using the differential layerwise learning rate [72], but it did not help and the learning was only slower.
- We tried not learning the size and offset for the Semantic Attention Head (SAH) by simply taking the feature from the keypoint locations. This led to worse performance in general.

A.4. Transforming Keypoints to ROIs

SapiensID relies on predicted keypoints to define Regions of Interest (ROIs). Assuming we have an input image roughly cropped around the visible body area (typically using a person detector's bounding box), we start with a set of predicted keypoints $\mathbf{K} = \{(x_k, y_k)\}_{k=1}^N$, where N is the number of keypoints. Our goal is to generate bounding boxes for each ROI. Specifically, we generate two bounding boxes—for the face and the upper torso—in the format (x_1, y_1, x_2, y_2), representing the top-left and bottom-right corners.

1. **Valid Keypoint Selection**:
   Let $\mathcal{K} = \{1, 2, \ldots, N\}$ be the set of keypoint indices. For each keypoint $k \in \mathcal{K}$, the coordinates are $(x_k, y_k) \in \mathbb{R}^2$. We define a visibility indicator v_k for each keypoint:

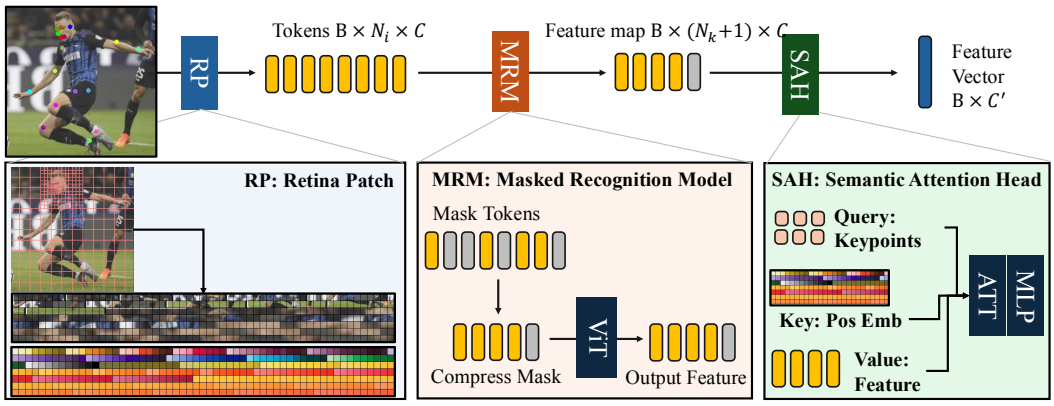


Figure 8. Illustration of the feature vector generation in SapiensID. First, Retina Patch (RP) generates image patches. Then, Masked Recognition Model (MRM) modifies the number of tokens. Finally, Semantic Attention Head (SAH) produces the feature vector from the set of tokens.

$$v_k = \begin{cases} 1, & \text{if } x_k \neq -1 \text{ and } y_k \neq -1, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Define the sets of keypoint indices relevant to each ROI:

K_1: Left Eye	K_6: Left Mouth Corner
K_2: Right Eye	K_7: Right Mouth Corner
K_3: Left Ear	K_8: Left Shoulder
K_4: Right Ear	K_9: Right Shoulder
K_5: Nose	

Then Face Keypoints are

$$\mathcal{M}_f = \{K_1, K_2, K_3, K_4, K_5, K_6, K_7\}.$$

And Upper Torso Keypoints are

$$\mathcal{M}_u = \mathcal{M}_f \cup \{K_8, K_9, K_{10}, K_{11}\}.$$

The valid keypoints for each ROI are those that are both visible and relevant:

$$\mathcal{V}^{\text{face}} = \{k \in \mathcal{M}_f \mid v_k = 1\}, \tag{16}$$
$$\mathcal{V}^{\text{torso}} = \{k \in \mathcal{M}_u \mid v_k = 1\}. \tag{17}$$

2. **Bounding Box Center and Size Calculation**:
For each ROI (face or upper torso), we compute the center using the set $\mathcal{V}$, which is either $\mathcal{V}^{\text{face}}$ or $\mathcal{V}^{\text{torso}}$:
First compute the minimum and maximum coordinates among valid keypoints:

$$x_{\min} = \min_{k \in \mathcal{V}} x_k, \quad y_{\min} = \min_{k \in \mathcal{V}} y_k, \tag{18}$$
$$x_{\max} = \max_{k \in \mathcal{V}} x_k, \quad y_{\max} = \max_{k \in \mathcal{V}} y_k. \tag{19}$$

Then calculate the center of the bounding box:

$$c_x = \frac{x_{\min} + x_{\max}}{2}, \quad c_y = \frac{y_{\min} + y_{\max}}{2}. \tag{20}$$

Then determine the maximum distance d from the center to the valid keypoints:

$$d = \max_{k \in \mathcal{V}} \sqrt{(x_k - c_x)^2 + (y_k - c_y)^2}. \tag{21}$$

3. **Bounding Box with Padding**:
First define the bounding box size s with a padding factor p (e.g., $p = 0.3$):

$$s = d \times (1 + p). \tag{22}$$

Then calculate the coordinates of the bounding box:

$$x_1 = c_x - s, \quad y_1 = c_y - s, \tag{23}$$
$$x_2 = c_x + s, \quad y_2 = c_y + s. \tag{24}$$

4. **Making Bounding Box Divisible**: To ensure that the patches cover the image without any overlap, the boundaries of the bounding box must *snap* onto the patch grid. In other words, the bounding box coordinate should be divisible by the patch size (p_w, p_h) of the enclosing ROI. Let n_r and n_c be the desired number of rows and columns for patches within the ROI. We modify the bounding box size s to ensure divisibility.

$$x_1' = \lfloor \frac{x_1}{p_w} \rfloor \times p_w, \quad y_1' = \lfloor \frac{y_1}{p_h} \rfloor \times p_h \tag{25}$$
$$x_2' = \lceil \frac{x_2}{p_w} \rceil \times p_w, \quad y_2' = \lceil \frac{y_2}{p_h} \rceil \times p_h \tag{26}$$

The final, grid-aligned bounding box is then:

$$\mathbf{b} = (x_1', y_1', x_2', y_2') \in \mathbb{R}^4. \tag{27}$$

This snapping process ensures that the bounding box boundaries coincide with patch boundaries, resulting in clean, non-overlapping patch extraction. We compute two bounding boxes, $\mathbf{b}^{\text{face}}$ and $\mathbf{b}^{\text{torso}}$, using this process. All these steps can be conducted in GPU for efficient computation.

A.5. Proof of Scaled Attention Equivalence

Let the scaled dot-product attention mechanism for self attention is defined as:

$$A = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V},$$

We aim to prove that when a scaling factor $\boldsymbol{\delta} \in \mathbb{R}^{1\times M}$ is added to the logits:

$$A = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \boldsymbol{\delta}\right)\mathbf{V},$$

this is equivalent to repeating each key $\mathbf{K}_j$ and value $\mathbf{V}_j$ exactly m_j times, where $\delta_j = \log m_j$.
Proof: Consider the following term:

$$\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \boldsymbol{\delta}.$$

For a query i and key j, the element of this matrix is:

$$\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \boldsymbol{\delta}\right)_{ij} = \frac{\mathbf{Q}_i \cdot \mathbf{K}_j^\top}{\sqrt{d}} + \log m_j,$$

where $\mathbf{Q}_i$ is the i-th query and $\mathbf{K}_j$ is the j-th key. Applying the softmax function, we get:

$$A_{ij} = \frac{\exp\left(\frac{\mathbf{Q}_i\cdot\mathbf{K}_j^\top}{\sqrt{d}} + \log m_j\right)}{\sum_k \exp\left(\frac{\mathbf{Q}_i\cdot\mathbf{K}_k^\top}{\sqrt{d}} + \log m_k\right)}.$$

Using the property $\exp(a+b) = \exp(a)\exp(b)$, this simplifies to:

$$A_{ij} = \frac{\exp\left(\frac{\mathbf{Q}_i\cdot\mathbf{K}_j^\top}{\sqrt{d}}\right) m_j}{\sum_k \exp\left(\frac{\mathbf{Q}_i\cdot\mathbf{K}_k^\top}{\sqrt{d}}\right) m_k}.$$

This is equivalent to each key $\mathbf{K}_j$ and corresponding value $\mathbf{V}_j$ are duplicated m_j times. We discard the values corresponding to the mask, so the result of the attenion mechanism is the same. Thus, the attention mechanism with $\boldsymbol{\delta}$ scaling is mathematically equivalent to duplicating the keys and values proportionally to the number of times the mask appears.

A.6. Token Length in MRM during Inference

To clarify the MRM's mechanism during training and inference, we include a more detailed explaination. One single masked token replaces all selected image tokens to mask during training. Eq.4 computes exactly same attention between □ □ □ ■ ■ and □ □ □ ■ where the black box is the mask token the number inside represents the attention offset (δ in Eq.4). So in inference, we append ■ with 1 (essentially no repeat) to make the token length same. *Eg*:

Sample 1: □ □ □ □ Sample 2: □ □ ■ ■

Dataset	Avg	LFW	CPLFW	CFPFP	CALFW	AGEDB
WF4M	97.44	99.80	94.97	98.94	96.03	97.48
WB4M-Facecrop	**97.63**	**99.82**	**95.12**	**99.19**	**96.07**	**97.97**

Table 7. Performance Comparison between WebFace4M and WebBody4M in the Face Recognition Task.

	AVG	LTCC CC		PRCC CC	
		Top1	mAP	Top1	mAP
Body	42.04	38.01	**18.84**	55.69	55.63
Face	36.56	17.60	4.91	72.62	51.10
Fused-Max	42.93	39.80	13.25	61.22	57.45
Fused Min-Max	49.92	39.80	12.95	79.00	67.93
Fused-Mean	49.99	39.80	12.82	**79.48**	67.85
SapiensID	**52.87**	**42.35**	17.79	78.75	**72.60**

Table 8. Performance table of score fusion (Body and Face).

B. Performance

B.1. WebBody4M vs WebFace4M Comparison

To assess the quality of the face image data within WebBody4M, we create WebBody-Facecrop by cropping face from the WebBody datset. And we compare its face recognition performance against WebFace4M [86], a dedicated large-scale face recognition dataset. We train the same ViT-based model with AdaFace loss on both datasets. Tab. 7 presents the results on standard face recognition benchmarks (LFW, CPLFW, CFPFP, CALFW, and AGEDB). The model trained on WebBody4M achieves a slightly higher average accuracy (97.63%) compared to that of WebFace4M (97.44%). This indicates WebBody4M label is of comparable quality, even slightly exceeding WebFace4M label.

B.2. Fusion Performance

While SapiensID inherently handles both face and body information within a single model, a common alternative approach involves training separate face and body recognition models and fusing their outputs. We compare SapiensID's performance with such multi-modal fusion methods. We consider a baseline where a body model (CAL [20]) is trained on either PRCC or LTCC, and a face model (ViT-Base [34]) is trained on WebFace4M. We then fuse the similarity scores of these two dedicated face and body models using three common fusion strategies: Max Fusion, Min-Max Normalization Fusion, and Mean Fusion. Tab. 8 presents the performance.

As shown in the table, even the best fusion strategy (Mean Fusion) achieves an average mAP of 49.99%, lower than SapiensID's 52.87%. Fusion is more helpful in PRCC but not much in LTCC with an increase in Top1 and a decrease in mAP. This result highlights the advantage of SapiensID's unified architecture, which learns to integrate face and body information more effectively than post-hoc fusion methods. Fusion methods treat each modality independently, potentially missing valuable contextual information that arises from their combined analysis.

	Method	KPR [57] + SOLDIER	SapiensID
	Training Data	LUPerson4M + OccludedReID	WebBody4M
OccludedReID	top1	84.80	**87.30**
	mAP	**82.60**	75.57
LTCC General	top1	68.15	**74.24**
	mAP	32.42	**36.88**
LTCC CC	top1	21.17	**42.60**
	mAP	10.19	**17.39**

Table 9. Generalization performance comparison under occlusion. SapiensID demonstrates superior generalization to unseen datasets (LTCC) compared to KPR+SOLDIER.

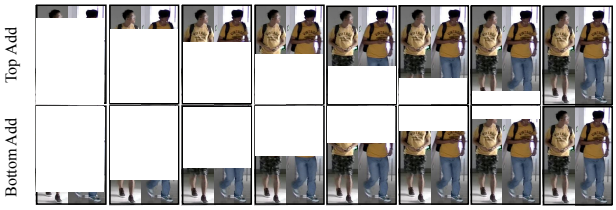


Figure 9. Illustration of how Images are erased from top to bottom or bottom to top.

B.3. Occluded ReID

Occlusions pose a significant challenge for robust human recognition. While specialized methods can be effective within their training domain, generalization to unseen scenarios is crucial for real-world deployment. We compare SapiensID's performance with KPR [57] combined with SOLDIER, a state-of-the-art occlusion handling method, to evaluate their respective generalization capabilities. KPR+SOLDIER is trained on a combination of LUPerson4M and the OccludedReID [87] dataset, while SapiensID is trained on our WebBody4M dataset without any OccludedReID data.

Tab. 9 presents the results on OccludedReID and the LTCC dataset (both General and Clothing Change protocols). KPR+SOLDIER and SapiensID similar performance on OccludedReID, SapiensID demonstrates significantly better generalization performance. On LTCC, SapiensID substantially outperforms KPR+SOLDIER across both protocols, highlighting the limitations of specialized training. This underscores the importance of training on diverse datasets like WebBody4M to achieve robust generalization in real-world human recognition. SapiensID, by learning from a wide range of poses, viewpoints, and clothing styles, is more adaptable and effective in unseen scenarios.

B.4. Impact of Body Part Features

We investigate the relative importance of different body parts in human recognition by conducting an ablation study on the Semantic Attention Head (SAH). Starting from part features ($\mathbf{O}^i_{part}$ in Eq. 8) multiplied by zero, we progressively undo masking, either from nose-to-ankles (top-down) or ankles-to-nose (bottom-up). We evaluate performance on LTCC (Clothing Change protocol) and PRCC (Clothing Change protocol). Results are presented side-by-side in Tab. 10. The top-down approach generally yields faster performance gains than bottom-up, suggesting that upper-body features contribute more significantly to recognition.

Interestingly, ankle features alone appear more discriminative than nose features alone. However, this counter-intuitive finding does not imply that ankles are inherently more informative than noses for person identification. We hypothesize that this observation arises because each part feature within SAH is not solely derived from the corresponding body part. Due to the preceding ViT backbone's attention mechanism, each part feature incorporates information from other body regions. Therefore, the presented results reflect the discriminative power of a part plus peripheral information from other parts, rather than the isolated contribution of each part.

A more accurate assessment of a part's individual discriminative ability would involve manipulating the input image directly, such as by occluding specific body parts. This approach, which isolates the impact of each part, is explored in the following section.

B.5. Impact of Actual Image Erased

To isolate the contribution of each body region, we conduct a second ablation study where we progressively erase sections of the input image, either top-down or bottom-up, as illustrated in Fig. 9. We erase equal-sized horizontal strips, starting with a single strip and progressively adding more until the whole image is erased (represented as "None" in the tables). The "Full" row represents the baseline performance with the complete image. Results are presented in Tab. 11.

The direct manipulation of the image confirms the importance of upper body regions. On both datasets, removing the top portion of the image drastically reduces performance. It comes as a surprise that PRCC can achieve a very good performance with only 1 top strip of image. But for LTCC, the lower parts are necessary to obtain a good performance. This indicates that different datasets exhibit different characteristics that can be exploited for conducting ReID.

B.6. Sensitivity to Pose Estimation

To understand the sensitivity of SapiensID to the pose estimation, we compare OpenPose [7], and YoloV8 [31] and add Gaussian noise ($\sigma = 0.01$). Tab (a) shows minimal impact from detector choice, but systematic keypoint errors reduce performance. Contrarily, in (b) we show how 5% zoom degrades CLIP3DReID, while SapiensID remains robust, making it the first ReID model robust to input extrinsics.

B.7. (Ablation on Model Size

We investigate the relationship between performance and the model and dataset size. In Tab. 12, we include ViT size variation (small vs base). The trend shows that the larger model

		LTCC CC		PRCC CC	
		Top1	mAP	Top1	mAP
1	None	0.00	3.56	1.47	4.28
2	1+Nose	25.77	5.78	27.21	21.04
3	2+Eye	30.61	8.87	63.87	55.17
4	3+Mouth	38.01	11.81	73.36	65.05
5	4+Ear	39.80	14.05	77.65	70.45
6	5+Shoulder	41.84	15.82	79.73	73.14
7	6+Elbow	41.07	16.64	**80.55**	**73.54**
8	7+Wrist	41.07	17.16	79.34	73.16
9	8+Hip	40.56	17.50	79.99	73.38
10	9+Knee	42.35	17.73	79.00	72.88
11	10+Ankle (full)	**42.35**	**17.79**	78.75	72.60

(a) top-down

		LTCC CC		PRCC CC	
		Top1	mAP	Top1	mAP
1	None	0.00	3.56	1.47	4.28
2	1+Ankle	27.04	7.37	45.05	35.32
3	2+Knee	32.14	9.55	55.12	44.97
4	3+Hip	35.71	12.34	66.07	55.04
5	4+Wrist	37.24	13.83	67.63	58.43
6	5+Elbow	40.05	15.72	69.57	62.61
7	6+Shoulder	41.33	16.87	73.84	67.80
8	7+Ear	41.58	17.61	76.21	70.62
9	8+Mouth	41.58	17.95	78.18	72.63
10	9+Eye	41.58	**17.80**	**79.23**	**72.92**
11	10+Nose (Full)	**42.35**	17.79	78.75	72.60

(b) bottom-up

Table 10. Comparison of feature erasing performance. (a) shows the performance as we progressively introduce features from Nose to Ankle (top-down approach). (b) demonstrates the performance when adding features from Ankle to Nose (bottom-up approach). Results are evaluated on LTCC and PRCC Cloth Changing (CC) protocol.

		LTCC CC		PRCC CC	
		Top1	mAP	Top1	mAP
1	None	2.30	1.89	12.67	4.78
2	1+Top1	5.10	2.61	78.04	67.29
3	2+Top2	27.04	11.88	79.25	70.53
4	3+Top3	29.34	13.20	78.35	69.85
5	4+Top4	33.67	13.88	77.82	69.55
6	5+Top5	37.24	14.65	76.97	69.28
7	6+Top6	36.48	15.49	78.55	70.39
8	7+Top7	41.07	16.63	**80.07**	**71.52**
9	Full	**42.35**	**17.79**	78.75	72.60

(a) top-add

		LTCC CC		PRCC CC	
		Top1	mAP	Top1	mAP
1	None	2.30	1.87	12.50	4.78
2	1+Bottom1	2.81	2.26	24.56	10.89
3	2+Bottom2	6.12	3.08	31.22	16.94
4	3+Bottom3	5.87	3.62	33.78	20.65
5	4+Bottom4	10.20	4.26	33.08	24.59
6	5+Bottom5	12.50	5.33	22.10	21.31
7	6+Bottom6	16.07	6.48	24.47	24.80
8	7+Bottom7	35.46	13.20	29.07	28.63
9	Full	**42.35**	**17.79**	**78.75**	**72.60**

(b) bottom-add

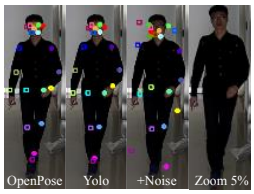
Table 11. Impact of progressively adding visible parts from the (a) top and from the (b) bottom. In contrast to Tab. 10 which measures the performance with the intermediate features zeroed out, here the actual input image is masked out.

Keypoint Predictor	Whole Body ReID	
	Short	Long
Open Pose	66.30	73.05
Yolo V8	65.62	72.76
Open Pose + ϵ	56.08	65.72

(a) SapiensID with keypoint changes

Extrinsic Change	LTCC (CC)	
	Original	Zoom 5%
CLIP3DReID[44]	41.84	31.88
SapiensID	42.35	41.58

(b) Different camera extrinsics



(c) Example visualization

has higher performance. We also created WebBody12M, in addition to 4M and the dataset increase further improves the performance.

SapiensID	Dataset	LTCC	CCDA	Celeb	LFW	AGEDB
Small	WB4M	71.40	57.04	91.29	99.67	96.58
Base	WB4M	74.24	61.84	92.77	99.77	97.18
Base	WB12M	**75.66**	**66.80**	**94.01**	**99.85**	**98.02**

Table 12. SapiensID backbone and dataset size Variation.

B.8. FLOP Analysis

In this subsection, we provide the FLOP analysis of SapiensID. The backbone model shares face model backbone (ViT-base). The major difference with ViT-base is the number of tokens. In inference, RetinaPatch produces 281 tokens on average (vs. 196 in ViT), increasing FLOPs from 24.69G to 35.39G. RetinaPatch (0.45G FLOPs) and Head (1.09G FLOPs, 336.65M params) add minimal overhead. Similarity measure is cosine dist, same as ArcFace.

B.9. Role of Masked Recognition Model (MRM)

In this subsection, we provide more ablation of MRM to showcase the importance of variable masking rate. Starting from simple ViT, we progressively add elements that comprises MRM. First we introduce token masking to handle varying token counts from RetinaPatch and improve training speed. Yet, simple masking significantly reduces performance due to discrepancies between training and testing samples. Thus, we propose variable rate masking (MRM), which restores performance to full-token training levels (see the table below, row 1 vs 3). All performance is measure without Retina Patch or Semantic Attention Pooling.

Metric same as Tab.5 (main paper)	Face	Whole Body ReID	
		Short	Long
(1) ViT (Full Token)	90.63	56.17	31.81
(1) + Token Masking (always remove 33%)	57.73	49.23	25.83
(1) + MRM (variable remove rate)	89.54	55.56	30.76
(1) + MRM + Retina Patch	92.93	59.16	46.95

Table 13. Performance of ViT as measured in Tab.5 of main paper. MRM is needed to allow Retina

B.10. Additional Face Recognition (FR) Perf.

We include more face recognition performances to investigate the performance of SapiensID in more challenging face recognition scenarios. We include the performance measured in IJB-B [69], IJB-C [69], TinyFace [11]. TinyFace

measures the face recognition performance in low quality imageries. WebBody4M is actually rich in small faces due to whole body images. It results in better TinyFace performances (row 2,3) than WebFace4M. SapiensID, inherently a body ReID model works well on aligned faces is because RetinaPatch always focuses on the face region.

Aligned FR	Training Data	IJBB TAR@FAR0.01%	IJBC	TinyFace R1	R5
ViT-AdaFace	WebFace4M	95.60	97.14	74.81	77.58
ViT-AdaFace	WB4M (Face crop)	**95.92**	**97.22**	75.32	78.76
SapiensID	WB4M	95.07	96.43	**75.97**	**79.69**

Table 14. Face Recognition Performance with ViT-Base. IJB,C measured in TAR@FAR=0.01%. All input images are aligned.

B.11. IJB-S Evaluation

A unified model is useful when matching cross modality imagery. In IJB-S [32] evaluation Surv2Single protocol, probe surveillance videos are matched to close-up gallery face images. UAV2Book presents an even greater challenge, with drone-captured probe videos featuring smaller faces and high-pitch angles. In such case, facial regions are too small. With a shared representation for both the whole body and face, the unified model (SapiensID) *opportunistically* captures more contextual cues, leading to improved matching, as shown below. Separate face or body models don't share the same representation space to conduct cross-modality matching. All models are finetuned on LQ BRAIR dataset.

IJB-S Evaluation Model	Input Type Probe	Gallery	Surv2Single R1	R5	UAV2Book R1	R5
Body Models	Body	Face	NA because raw gallery is face.			
ViT-AdaFace	Face	Face	75.6	79.7	29.1	38.0
SapiensID	Face	Face	**75.8**	**80.0**	31.6	44.3
	Body	Face	72.6	77.9	**39.2**	**49.4**

Table 15. Performance in IJB-S Evaluation Dataset.

B.12. Unaligend Face Recognition

We also show unaligned IJB-B/C results to see the face recognition performance without alignment. A dedicated FR model is better in aligned, but SapiensID has less performance drop in unaligned settings.

Metric TAR@FAR=0.01%	Unaligned IJB-B	IJB-C	Aligned IJB-B	IJB-C
ViT-AdaFace	93.26	94.97	**95.60**	**97.14**
SapiensID	**94.30**	**96.05**	95.07	96.43

C. Visualization

C.1. Token Length Sampling Distribution

In Masked Recognition Model (MRM), we propose an adaptive token sampling strategy during training to enhance the robustness and generalization of our masked recognition model. Fig. 10 illustrates the sampling distribution and its effect on the input image. The number of tokens to keep, $\hat{n}_k$, is determined by Eqn. 6:

$$\hat{n}_k = n_k + (n_i - n_k) \cdot e^{-\lambda \cdot U(0,1)},$$

where n_i is the maximum possible number of tokens (432 in our case, with 3 ROIs of 12x12 patches each), n_k is the minimum number of tokens to keep, $U(0,1)$ is a uniform random variable, and λ controls the decay rate (set to 4).

This sampling strategy allows us to retain between 26% and 80% of the tokens (112 to 345 tokens), with an average of 166 tokens per batch. As depicted in Fig. 10, heavy masking can significantly distort the input image. Fixing the masking rate to such high levels could introduce a distribution shift between training and testing (where all tokens are used), causing a performance drop. Our adaptive sampling mitigates this issue by exposing the model to a variety of masking ratios, encouraging it to learn robust representations

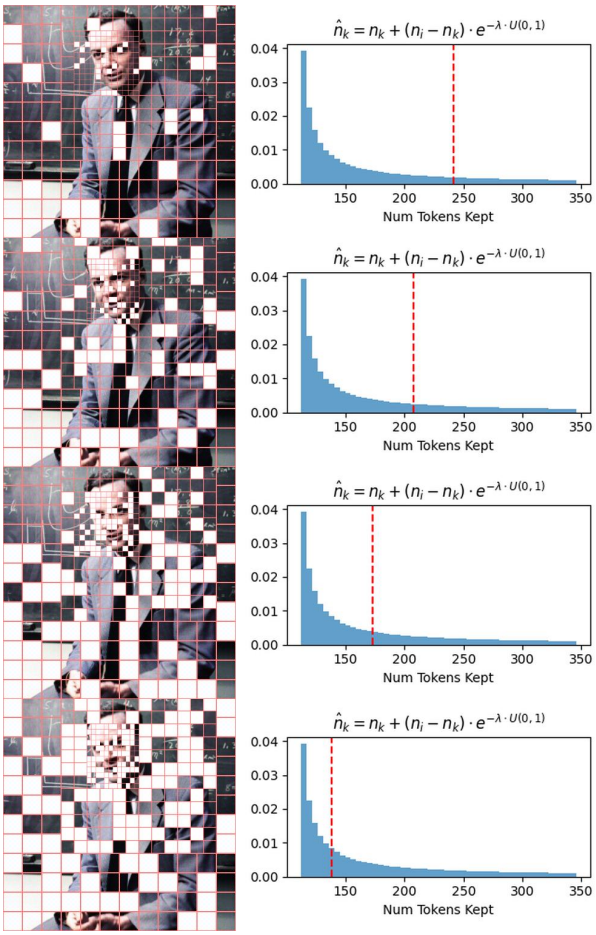


Figure 10. Illustration of the masked image and the sampling distribution of the number of tokens to keep $\hat{n}_k$. The red vertical line shows where the sampling took place for the right image. From top to bottom, less samples are kept (more masking).

Visibility	Left (%)	Right (%)
Eye	93.49	93.59
Ear	76.87	74.48
Shoulder	88.15	90.04
Elbow	53.76	53.80
Wrist	49.98	50.35
Hip	45.68	45.70
Knee	23.92	23.95
Ankle	16.98	17.00

Table 16. Keypoint Visibility in WebBody Dataset.

that generalize well to full token input during inference.

One thing to note is that the sampling of $\hat{n}_k$ happens per batch. And when a larger $\hat{n}_k$ is sampled per batch, we reduce the batch size accordingly for the given GPU memory (See Sec. 3.2 for more details).

C.2. WebBody4M Dataset Body Parts Visibility

WebBody4M dataset encompasses a wide range of human poses and viewpoints, resulting in varying visibility of body keypoints. Tab. 16 presents the percentage of images in which each keypoint (left and right sides) is visible. As expected, keypoints in the upper body, such as eyes and shoulders, exhibit high visibility rates (over 74% and 88% respectively). Visibility decreases progressively down the body, with elbows and wrists around 50%, hips around 45%, and knees and ankles below 24% and 17% respectively. This distribution reflects the natural tendency for upper body parts to be more frequently visible in unconstrained images, as lower body parts are often occluded by clothing, objects, or the image frame itself. This distribution also helps explain why upper body parts provide greater discriminative power for person ReID in our earlier analysis (Supp B.4).

C.3. Visualization of Part Weights

To facilitate effective learning from a mixture of short-term and long-term ReID datasets, we hypothesize that it would be helpful to add learnable weights that modulate the importance of individual part features within the Semantic Attention Head (SAH). Our conjecture is the discriminative characteristics of body parts can vary significantly depending on whether clothing remains constant or varying in the training dataset.

Fig. 11 visualizes the learned weights (Eqn. 14) for WebBody4M and several additional whole-body ReID datasets. WebBody4M, primarily composed of web-collected images, exhibits a higher emphasis on facial features compared to lower body parts. This is expected, as the WebBody4M was collected largely based on facial similarity.

In contrast to WebBody4M, auxiliary datasets like Market1501, LTCC, and PRCC, which feature many images with consistent clothing (e.g., 1-3 outfits across 20-30 images per person), show increased emphasis on body features for recognition. This highlights the importance of body shape, pose, and clothing appearance as discriminative cues when attire remains relatively constant. However, Celeb-ReID, similar to WebBody4M, primarily contains images with clothing changes across captures. Consequently, Celeb-ReID exhibits a similar weighting pattern, with less emphasis on body features and a relatively higher focus on other cues, likely emphasizing facial features.

	All	Face	Whole Body ReID	
			Short	Long
SapiensID	**78.67**	**96.66**	73.05	**66.30**
SapiensID-Weight	78.59	**96.66**	**75.72**	63.39

Table 17. Performance comparison of SapiensID and SapiensID without weight masking during training across different metrics.

To validate the hypothesis, we conducted an ablation study to evaluate the impact of training with learnable weights. Tab. 17 presents a comparison between SapiensID and SapiensID without the learnable weights. In the latter, all aspects remain the same except that the learnable weights are removed during training.

From the results, it is evident that the inclusion of learnable weights does not yield a significant overall improvement. Instead, it shows a specific enhancement in long-term ReID performance, possibly because WebBody4M's learning was not hindered by the influence of short-term datasets with same clothings. However, for short-term datasets, the addition of weights does not result in performance gains. This suggests that while the weighting mechanism provides insights into dataset-specific learning behaviors, it is not a definitive factor for achieving better ReID performance.

In conclusion, while the introduction of learnable weights is interesting for analytical purposes, we want to let the readers clearly know that it is not a deciding factor for learning universal representation that works for both short-term and long-term ReID. Future research could explore alternative methods that better balance the learning from diverse dataset characteristics without negatively impacting specific subsets.

C.4. SAH Visualization

The Semantic Attention Head (SAH) plays a crucial role in SapiensID by generating pose-invariant features. To understand how SAH behaves after training, we visualize its attention maps in Fig. 13. To be specific, we visualize the following. Let $\mathbf{Q}^i_{kp} = \text{GridSample}(\text{PE}, \text{kp}^i) + \mathbf{B}$ be the semantic query embedding for i-th image created by sampling from the fixed 2D position embeddings (PE) at the 19 keypoint locations. The dimension is $\mathbf{Q}^i_{kp} \in \mathbb{R}^{nk \times C}$, where $k = 19$ and $n = 4$ because it is repeated 4 times to learn 4 different offsets. In SAH, we perform attention with $\mathbf{Q}^i_{kp}$ and PE by

$$\mathbf{O}^i_{\text{part}} = \text{softmax}\left(\frac{\mathbf{W}_q\mathbf{Q}\mathbf{W}_k\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{W}_v\mathbf{V}. \quad (28)$$

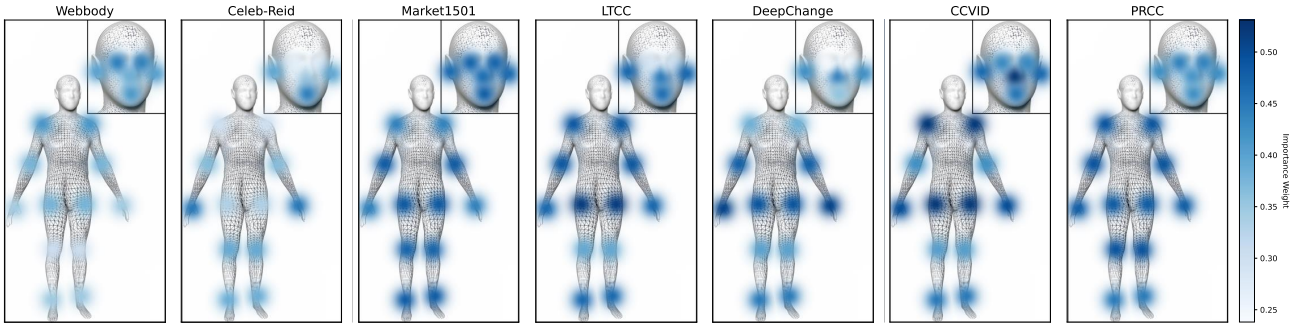


Figure 11. Comparison of learned part weights across seven datasets. Left and right sides are averaged together before visualization.

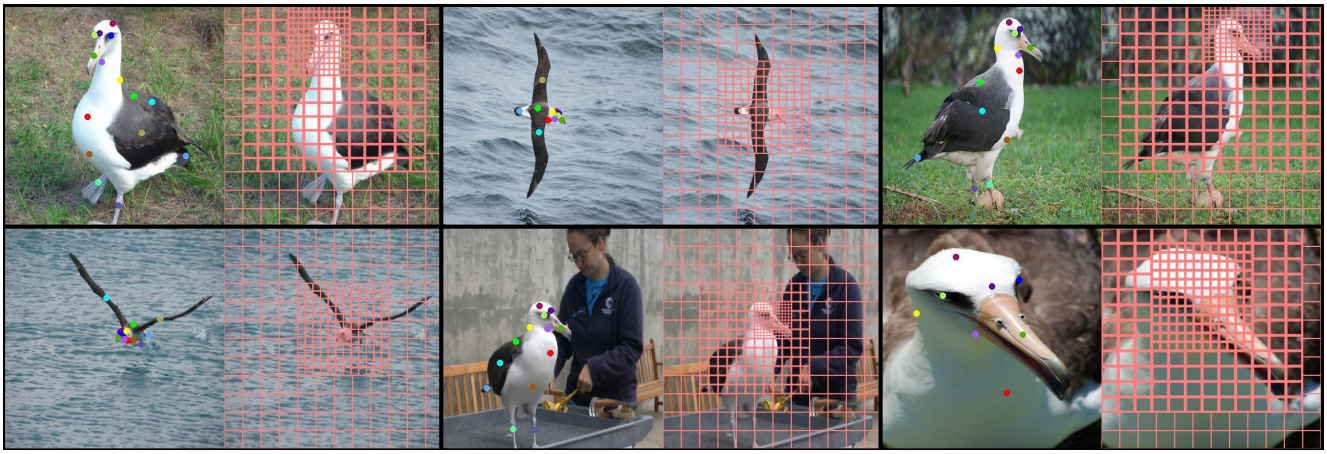


Figure 12. Keypoint visualization (left) and corresponding Retina Patch results (right) for images from the CUB dataset.

In our visualization, we are showing

$$\text{softmax}\left(\frac{\mathbf{W}_q\mathbf{Q}\mathbf{W}_k\mathbf{K}^\top}{\sqrt{d}}\right),$$

for each keypoint and each offset. We have nk attention maps as shown by the visualization.

For each input image, we show each row corresponds to a different offset. There are 4 rows because we learn $n = 4$ offsets for each of 19 keypoints. Offest refers to $\mathbf{B} \in \mathbb{R}^{nk\times C}$ in Eqn. 7. Offset bias allows the keypoints to move slightly from its original position. Each column correspond to different keypoints used by SAH (e.g., nose, left right shoulder, *etc*). As the visualization shows, the learned attenion maps are not limited to the keypoint location but also move around the keypoints and vary in size.

D. Potential Application of Retina Patch

While SapiensID focuses on human recognition, the Retina Patch (RP) mechanism has broader applicability to other domains. Figure 12 demonstrates its potential for fine-grained visual recognition, using the CUB birds dataset as an example. This dataset provides semantic keypoints, enabling the definition of meaningful regions of interest (ROIs) for RP. We define two ROIs: "head" (beak, forehead, crown, left eye, right eye, throat) and "body" (back, belly, breast, nape, left wing, right wing) excluding tail, left leg and right leg.

The figure showcases multiple bird images processed with RP, illustrating its ability to handle variations in bird size and head size. By dynamically allocating more patches to these regions, RP ensures consistent representation of crucial features, regardless of their scale within the image. Though we do not know whether the performance of CUB bird classification will be improved with RP, we want to suggest that RP could be beneficial for general recognition tasks where image naturally contains large pose and scale variation. Future work could explore the integration of RP into models for more broad set of datasets to quantitatively evaluate its benefits.

E. Limitations

While SapiensID demonstrates promising results for human recognition, its reliance on predefined Regions of Interest (ROIs) introduces certain limitations. The effectiveness of the Retina Patch mechanism hinges on the ability to define

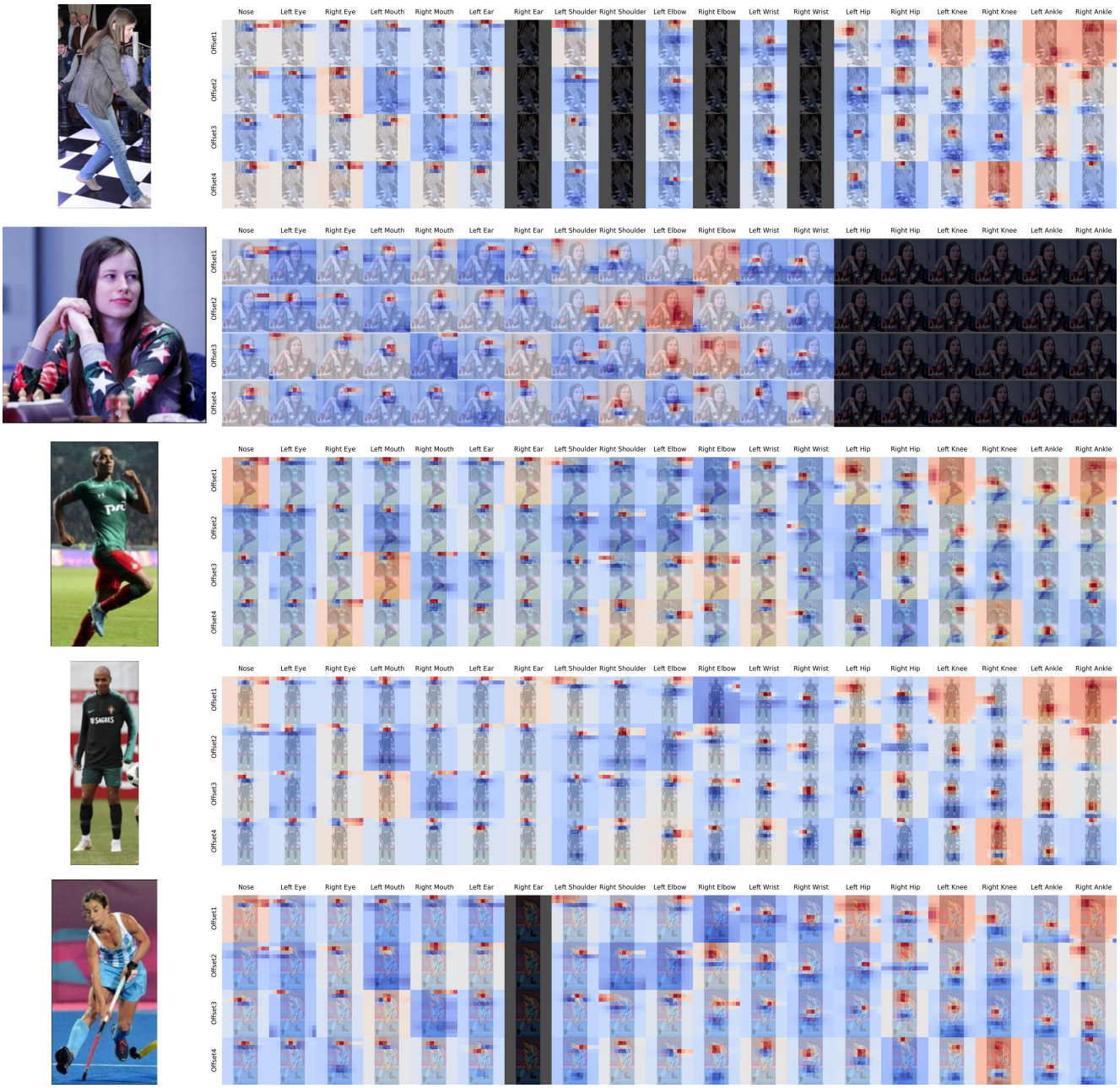


Figure 13. Visualization of attention maps in the Semantic Attention Head (SAH). Regions with higher attention values are highlighted in red, while regions with lower attention values are shown in blue. Blacked-out areas represent parts of the images without visible keypoints. The visualizations provides how SAH allows learning both varied size and offsets based on a set of keypoints.

meaningful ROIs that capture discriminative features. This approach works well for humans, who share a consistent body topology and where keypoints like the face, torso, and limbs provide valuable cues for recognition.

However, this reliance on ROIs poses challenges when dealing with objects or entities that lack a consistent or well-defined structure. For instance, applying SapiensID to amorphous objects, scenes with highly variable elements, or categories with significant intra-class topological differences would require alternative strategies. In such cases, predefined ROIs might not adequately capture the relevant information, or might even be detrimental by focusing on irrelevant or inconsistent features. Future research could explore more flexible or adaptive mechanisms for defining regions of interest, enabling the application of similar principles to a wider range of object recognition tasks.

While SapiensID achieves state-of-the-art performance in long-term ReID, its short-term ReID accuracy lags behind methods like Soldier [9] and HAP [79]. This discrepancy stems from a fundamental conflict between short-term cues—such as clothing—and long-term biometric traits like facial features and body shape. Soldier and HAP leverage masked reconstruction objectives that emphasize visible appearance cues, including clothing, making them more effective for short-term scenarios. In contrast, SapiensID is trained on the WebBody4M dataset, which features frequent clothing changes and thus prioritizes identity over appearance. Addressing this trade-off remains an open challenge, and future work could explore unified models that balance both short-term appearance cues and long-term identity features.

F. Ethical Concerns

Our goal is to facilitate research in human recognition while operating strictly within the bounds of copyright law, privacy regulations, and ethical considerations. For large-scale image datasets, it is a common practice to release datasets in URL format [3, 54] because researchers do not hold the rights to redistribute the data directly. By providing permanent link URLs, labels and a one step code to download and prepare dataset, researchers can have access and utilize the data responsibly, while respecting the rights of copyright holders and individuals. We believe this approach balances the need for large-scale datasets to advance research with the imperative to protect intellectual property and privacy.