Towards Self-Supervised Gaze Estimation: Supplemental Material

Arya Farkhondeh[1,3]
farkhondeh.1860768@studenti.uniroma1.it
Cristina Palmero[2,3]
crpalmec7@alumnes.ub.edu
Simone Scardapane[1]
simone.scardapane@uniroma1.it
Sergio Escalera[2,3]
sergio@maia.ub.es

[1] Sapienza University of Rome
Rome, Italy
[2] University of Barcelona
Barcelona, Spain
[3] Computer Vision Center (CVC)
Barcelona, Spain

The supplemental material is organized as follows: implementation details can be found in Sec. A. The details of transformations and identifying top-performing transformations appear in Sec. B. We then perform robustness analysis in Sec. C and show the results of ablation studies in Sec. D. Lastly, we visualize estimated gaze estimation directions in Sec. E.

A Implementation Details

In this section, we provide the implementation details of the evaluation settings. Throughout all the experiments we use Adam as the optimizer, batch size of 512, an input size of 224×224 pixels, and a learning rate decay factor of 0.1 unless otherwise stated.

A.1 Further details of pretraining

As mentioned in Sec. 3.1 of the main submission, SwAV [1] performs score adjustment using the Sinkhorn-Knopp [3] algorithm to avoid trivial solutions. We refer the reader to the SwAV paper [1] for the details of the Sinkhorn-Knopp algorithm. This algorithm has two hyperparameters, namely, the number of iterations and Sinkhorn regularization parameter (ε). We perform 3 Sinkhorn iterations as in SwAV and set $\varepsilon = 0.03$. Note that a high value of ε leads to trivial solutions, i.e., same cluster assignment for every image within a batch, whereas a too low value results in numerical instability.

A.2 Implementation details of linear evaluation

For linear evaluation (Sec. B.1 of this supplementary material and Sec. 4.2 of the main submission), we freeze the backbone and train a linear regressor on top for 100 epochs. We set the initial learning rate to 0.01, which is decayed using cosine decay with a final value of 0.0001. We also used a weight decay of 0.0001.

A.3 Implementation details of semi-supervised learning

In semi-supervised learning (Sec. 4.3), we finetune the whole network using two subsets (10% and 30%) from the ETH-XGaze dataset, at the subject level. We finetune SwAT for 100 epochs with an initial learning rate of 0.001 for the backbone and 0.01 for the linear regression head. Then, we decay the learning rates after 40 and 80 epochs. We also used a weight decay of 0.0001. The supervised baseline is trained in the same manner except we initialize the learning rate of the backbone with 0.01.

A.4 Implementation details of supervised finetuning

This section provides implementation details of Sec. 4.4 in the main submission. For supervised finetuning, we use different hyperparameters for each dataset. In the case of ETH-XGaze, we finetune SwAT for 25 epochs following [6]. The learning rates of both the backbone and linear regressor are set to 0.001, which are then decayed at epoch 15. In addition, we use a weight decay of 0.0001. However, we train the supervised counterpart for 100 epochs with an initial learning rate of 0.01, decayed after 40 and 80 epochs. On Gaze360, we finetune SwAT for 80 epochs following [4]. The learning rate of backbone and the linear head are set to 0.001 and 0.01, respectively, which are decayed using cosine decay with a final value of 0.0001. In the case of MPIIFaceGaze, we perform finetuning for 40 epochs with an initial learning rate of 0.0005, decayed at 20 and 30 epochs. For MPIIFaceGaze*, we finetune for 25 epochs, decaying the learning rate at 10 and 20 epochs. For all datasets, we use horizontal flip and scaling $s \in [0.7, 1.4]$ as data augmentation.

A.5 Details of evaluation protocol on each dataset

ETH-XGaze contains 756K images and 80 subjects for training. The test set is composed of 15 subjects with a total of 159K samples. Since the dataset does not have an official validation set, we manually split the training set into two subject-independent sets, i.e., 90% (72 subjects) training set and 10% (8 subjects) validation set. We selected the subjects via visual inspection ensuring diversity across gender, ethnicity, and eyewear accessories. The validation set was only used for ablation study (Sec. D), evaluation of transformations (Sec. B.1), and evaluating the unsupervised features (Sec. 4.2). The rest of the experiments are trained using 100% of the training data and evaluated with the test set. Note that the test set of ETH-XGaze is kept private and online evaluation is performed via the dedicated submission webpage. For semi-supervised learning, we selected two subsets from the training data at subject level, i.e., 10% (8 subjects) and 30% (24 subjects). The ID of the subjects are as follows:

- **10% subset (8 subjects)** = {3, 32, 48, 52, 80, 88, 101, 109}
- **30% subset (24 subjects)** = {0, 3, 8, 9, 13, 24, 28, 32, 33, 36, 38, 40, 45, 48, 52, 62, 79, 80, 88, 92, 101, 103, 109, 111}

Gaze360 contains 129K training, 17K validation, and 26K test samples collected from 238 subjects. We use a subset of the dataset whose faces come with bounding boxes, resulting in around 85K, 11K, and 16K samples for training, validation, and test, respectively.

MPIIFaceGaze comes with 45K samples and 15 subjects each having 3K samples. We perform a leave-one-person-out cross-validation for each subject to evaluate and compare the methods.

Method	Color	Blur	Noise	Flip	Rotate	Scale	Cutout	Sobel	Composition
SwA**V**	27.1	28.9	28.4	33.8	30.8	29.7	30.7	27.7	26.4
SwA**T**	27.1	28.9	28.4	**28.6**	**29.2**	29.7	30.7	27.7	**26.0**

Table i: **Evaluation of Transformations.** Performance of SwAV and SwAT for each individual transformation on the validation set of ETH-XGaze, in terms of average angular gaze error (degrees). Note that SwAV and SwAT only differ in terms of Flip and Rotation while behaving identically in the case of other transformations. The last column shows the results of composition of transformations using the soft assignment policy.

MPIIFaceGaze* is the unnormalized version of the MPIIFaceGaze dataset. We performed 3-fold cross-validation where the folds were chosen uniformly at random.

B Transformations

Fig. 1 (right) of the main submission shows the explored transformations which fall into two groups, namely appearance, and geometric transformations. For appearance transformation, we consider color drop, color jitter, Gaussian blur, Gaussian noise, cutout, and Sobel filtering. As geometric transformations, we examine horizontal flip, rotation, and scale.

B.1 Evaluation of Transformations

To identify the most effective transformations, we perform individual transformation evaluation. So, we pretrain an encoder on the ETH-XGaze dataset (without labels) using each transformation. Then, we freeze the backbone and train a linear gaze regressor on top. For this experiment, we use ResNet-50 as the backbone and we set the input size to 112×112.

Individual Transformation Evaluation. Tab. i shows the results of individual transformations for SwAV and SwAT methods. Note that both methods behave identically under appearance and scale transformations, whereas they differ in terms of horizontal flip and rotation. As can be seen, SwAT outperforms SwAV in the case of horizontal flip and rotation, achieving around 15% and 5% relative improvements, respectively. This demonstrates the benefit of enforcing equivariance under affine transformations via SwAT, producing feature representations that are more aligned with the gaze estimation task.

Composition of Transformations. A stronger image distortion can be realized via composing multiple transformations in a sequential manner. To achieve that, we compose the transformations using a soft assignment policy. Let us denote p as the probability of applying a transformation. We compute p by mapping the individual performances (Tab. i) to [0,1] via scaling, such that:

$$p_t = 1 - \frac{\mathrm{e}_t - \mathrm{e}_{min}}{\mathrm{e}_{max} - \mathrm{e}_{min}}, \tag{1}$$

where t is a given transformation chosen from the transformation catalog, e_t corresponds to the gaze error of the transformation during individual transformation evaluation, and e_{max} and e_{min} are the minimum and maximum gaze error across all the individual transformations of the method i.e., SwAV and SwAT (rows of Tab. i). The computed probabilities for each

Method	Color	Blur	Noise	Flip	Rotate	Scale	Cutout	Sobel
SwA**V**	1.0	0.7	0.8	0.0	0.4	0.6	0.5	0.9
SwA**T**	1.0	0.5	0.6	0.6	0.4	0.3	0.0	0.8

Table ii: Computed probabilities (p) using the soft assignment policy.

transformation depending on the pretraining approach (SwAV or SwAT) can be found in Tab. ii. This way, all the transformations contribute to data augmentation with respect to their individual performances. The last column of Tab. i (*Composition*) shows that the soft assignment policy improves the performance compared to individual transformations. We find such a soft assignment policy more promising than a hard assignment counterpart such as selecting top-k transformations and then performing an exhaustive search as in [5]. In particular, we get 0.9° improvement using the soft assignment policy compared to the best composition of the hard assignment method.

B.2 Details of Transformations

In this section, we provide the details of transformations for self-supervised pretraining. When composed together, each transformation is applied with probability p_t, which is determined by soft assignment policy (Tab. ii). In the following, we provide the details of each transformation in the same order they are applied during implementation.

Sobel. Since the two transformed views are assumed to be different, we apply the Sobel filter to only one view.

Blur. Gaussian blur is applied using a Gaussian kernel where we randomly sample the radius $\sigma \in [0.1, 2.0]$. We do not apply the blur transformation to views with Sobel transformation applied.

Color. We apply color transformation following SimCLR [2]. More concretely, this transformation is composed of two sub-transformations, i.e., color jittering (brightness, contrast, saturation, and hue) and color dropping (grayscale). We randomly apply color jittering with probability of 0.8, and color dropping with 0.2. We do not apply the color transformation to views with Sobel transformation applied.

Noise. We add Gaussian noise $N \sim (0, 30)$ to an image. Once Sobel is applied, we do not apply Gaussian noise.

Cutout. We randomly cutout a patch of size $h \times h$ pixels, where $h = 64 \times (H_x/224)$ and H_x is the height (or width) of the input image (x).

Flip. Applying horizontal flip to both views results in the same image. Thus, we only applied it to one view.

Rotate. We apply rotation via randomly sampling the rotation angle (θ) from $\theta \in [-45, 45]$ degrees.

Scale. Scaling (s) is applied via randomly sampling the scale factor $s \in [0.7, 1.4]$.

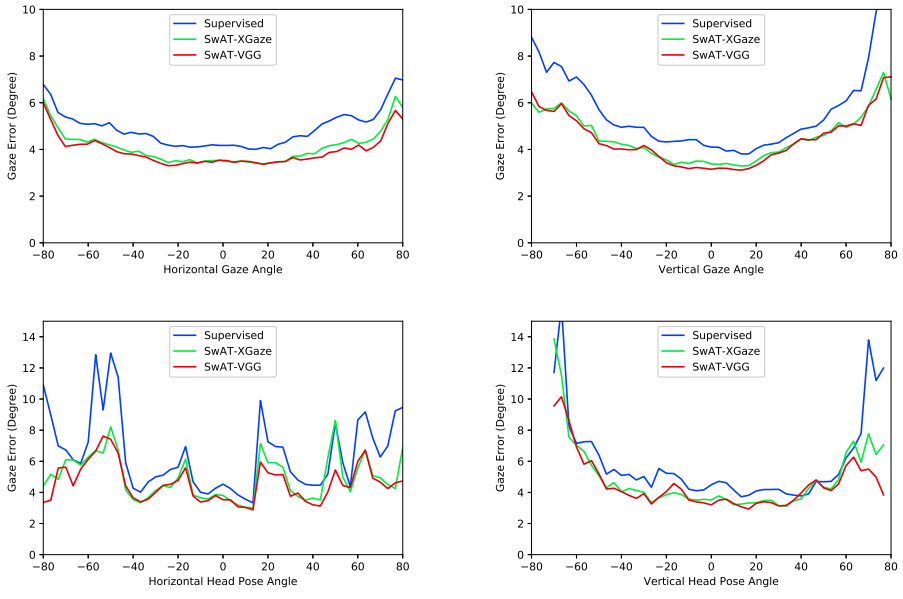

Figure i: **Robustness Analysis for Supervised Finetuning Setting.** Gaze estimation error across horizontal (left) and vertical (right) for gaze and head pose directions in degrees.

C Robustness Analysis

Mean gaze error is not quite an informative indicator of how a method performs within a specific gaze direction and head pose range. Thus, we conduct a robustness analysis to shed light on the performance of our method across gaze and head pose angles. Fig. i depicts the gaze error in degrees across horizontal and vertical gaze and head pose angles on the test set of ETH-XGaze. We observe that the performance of the supervised baseline substantially decreases as a function of the number of samples in ETH-XGaze (gaze angles follow a Gaussian-like distribution centered at 0, whereas the head pose distribution is multimodal [5]). In contrast, SwAT demonstrates superior robustness across all directions compared to the supervised baseline. However, SwAT pretrained with VGGFace is consistently more stable than SwAT pretrained on ETH-XGaze (without labels), especially in case of extreme gaze and head pose angles. We repeat the same analysis for the semi-supervised setting (Sec. 4.3). As shown in Fig. ii, overall, across both horizontal and vertical directions, the performance of SwAT is superior to the supervised (Random Init.) baselines. Nevertheless, the error curves slightly fluctuate for extreme head poses.

D Ablation Studies

In this section, we vary some of the key hyperparameters of SwAT such as the number of prototypes (Sec. D.1), the number of epochs (Sec. D.2), and the dimensionality of the projection head (Sec. D.3). We also present the comparison with PeCLR [5] in Sec. D.4. Throughout these experiments, we use ResNet-50 as the backbone encoder and ETH-XGaze (without labels) dataset for pretraining. Then, we freeze the backbone and train a linear regressor on top using the training set of ETH-XGaze, and measure the performance on the

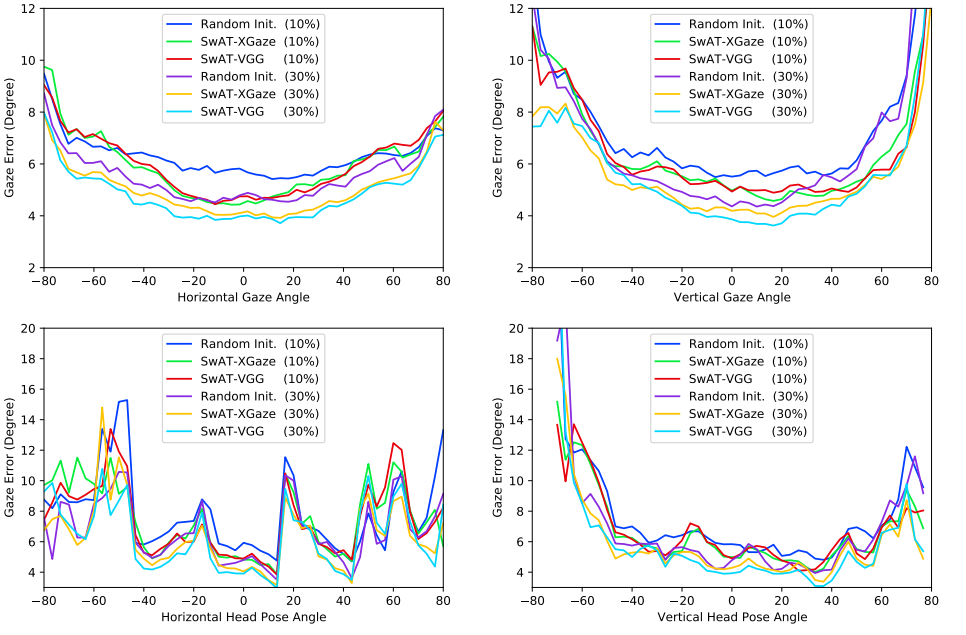

Figure ii: **Robustness Analysis for Semi-Supervised Setting.** Gaze estimation error across horizontal (left) and vertical (right) for gaze and head pose directions in degrees. The percentages show the amount of labeled data used for finetuning.

	Number of prototypes				d		epochs		
	500	1500	3000	6000	128-D	256-D	100	200	400
Gaze Error	**25.8**	26.0	25.9	26.0	26.0	**25.7**	**26.0**	26.3	26.4

Table iii: Results of ablation study on the number of prototypes, the dimensionality of the projection head (d), and the number of epochs. Numbers denote gaze error in degrees. Best results are bolded.

validation set. We set the input image size to 112×112. The default values for the number of prototypes, number of epochs, and dimensionality of projection-head are 500, 100, and 128, respectively, unless otherwise specified.

D.1 Number of prototypes

In this experiment, we investigate the effect of the number of prototypes (M) on the performance of SwAT. To achieve that, we consider four candidates, i.e., 500, 1500, 3000, and 6000. As shown in Tab. iii, we observe a slight difference in the performance of SwAT with different numbers of prototypes. This shows that the number of prototypes has a negligible impact on the performance of SwAT.

D.2 Number of epochs

We aim at increasing the number of epochs for pretraining from 100 to 200 and 400 epochs to assess whether SwAT takes advantage of longer pretraining. Results in Tab. iii suggest that 100 epochs is sufficient and further pretraining leads to worse results.

D.3 Dimensionality of projection-head

In this experiment, we increase the dimensionality of the projection head d from 128-D to 256-D. As shown in Tab. iii, SwAT achieves a slightly better result with 256-D.

D.4 Comparison with PeCLR [5]

In this subsection, we shed light on the differences between our equivariance formulation (SwAT) and PeCLR [5], a self-supervised approach for the task of 3D hand pose estimation. To avoid trivial solutions, PeCLR uses a contrastive loss that attracts the positive pairs while repelling the negative pairs. Furthermore, PeCLR achieves equivariance via inverting the image-space affine transformations in feature space which results in having the same affine information for both positive and negative pairs. Thus, the contrastive loss has to push apart representations with the same affine information in feature space. Additionally, the negative pairs may also contain faces with similar gaze or head poses. In contrast, SwAT equalizes the feature vectors in terms of affine information and does not require negative samples. SwAT learns more geometry-aware representations as throughout training iterations SwAT sees the same image under various transformation information in feature space. Thus, the same image can have different cluster assignments depending on the randomly sampled transformation. Whereas, throughout training, PeCLR observes the same image with the same transformation information in feature space.

We compare both methods in the same setting and we use rotation as the only affine transformation. We evaluate the quality of the features by linear evaluation where we freeze the backbone and train a linear regressor on top. The results show that SwAT achieves better performance (29.2°) compared to PeCLR (29.6°).

E Qualitative Results

Fig. iii shows the estimated gaze direction on the test set of Gaze360. As shown, the supervised model demonstrates a large discrepancy compared to the ground-truth vectors while SwAT estimations better match the ground truth. It can be seen that SwAT is able to better estimate the gaze direction in extreme head-pose conditions. In the last column, we show some failure cases where SwAT and supervised model are not on par with the ground truth. In addition to Gaze360, we also show the estimated gaze direction on MPIIFaceGaze in Fig. iv. Note that in the case of MPIIFaceGaze, we performed leave-one-person-out evaluation for two subjects. The visual results in Fig. iv suggest that SwAT achieves higher performance than the supervised baseline in the challenging case of extreme illumination condition. The last column in Fig. iv shows some failure cases where both SwAT and supervised model fail to follow the ground-truth. Nevertheless, as can be seen from the failure case of closed eyes in both figures (Fig. iii bottom row, Fig. iv top row), despite the fact that the ground truth indicates the theoretical gaze direction, SwAT estimates a downward direction, which is more aligned with the closed eye direction.

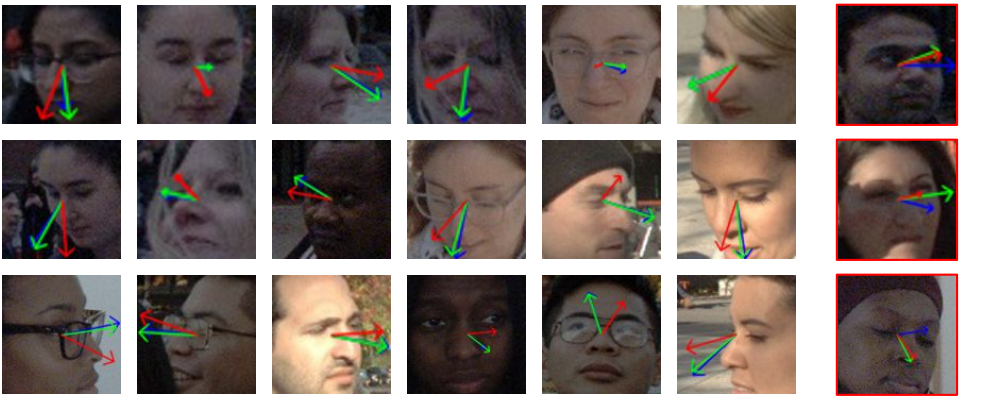

Figure iii: Visual results of estimated gaze direction on the test set of Gaze360. The green, red, and blue colors are, SwAT (⟶), Supervised (⟶), Ground-truth (⟶) , respectively. The last column shows examples of failure cases.

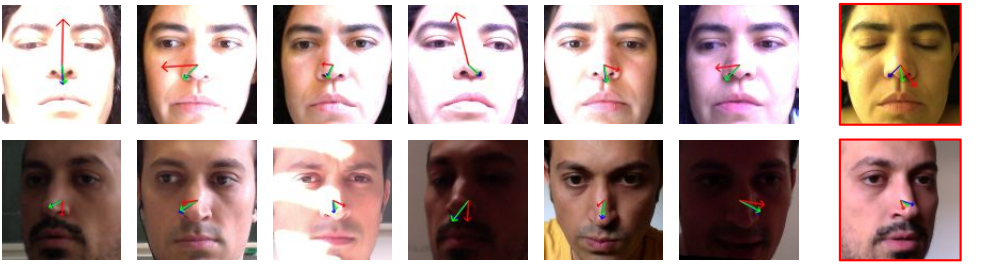

Figure iv: Visual results of estimated gaze direction on the test set of MPIIFaceGaze. The green, red, and blue colors are, SwAT (⟶), Supervised (⟶), Ground-truth (⟶) , respectively. The last column shows examples of failure cases.

References

[1] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[3] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013.

[4] Petr Kellnhofer, Adrià Recasens, Simon Stent, W. Matusik, and A. Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *ICCV*, 2019.

[5] Adrian Spurr, Aneesh Dahiya, Xi Wang, Xucong Zhang, and Otmar Hilliges. Self-supervised 3d hand pose estimation from monocular rgb via contrastive learning. In *ICCV*, 2021.

[6] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *ECCV*, 2020.