Supplementary Material
AdaFace: Quality Adaptive Margin for Face Recognition

Minchul Kim, Anil K. Jain, Xiaoming Liu
Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI, 48824
{kimminc2, jain, liuxm}@cse.msu.edu

A. Gradient Scaling Term

In Sec. 3.1 of the main paper, the gradient scaling term (GST), g is introduced. Specifically, it is derived from the gradient equation for the margin-based softmax loss and defined as

$$g := \left(P_j^{(i)} - \mathbb{1}(y_i = j)\right) \frac{\partial f(\cos\theta_j)}{\partial \cos\theta_j}, \tag{1}$$

where

$$P_j^{(i)} = \frac{\exp(f(\cos\theta_{y_i}))}{\exp(f(\cos\theta_{y_i})) + \sum_{j\neq y_i}^{n} \exp(s\cos\theta_j)}. \tag{2}$$

This scalar term, g affects the magnitude of the gradient during backpropagation from the margin-based softmax loss. The form of g depends on the form of the margin function $f(\cos\theta_j)$. In Tab. 1, we summarize the margin function $f(\cos\theta_j)$ and the corresponding GST when $j = y_i$, the ground truth index.

Methods	$f(\cos\theta_j)$, $j \neq y_i$	$f(\cos\theta_j)$, $j = y_i$	g when $j = y_i$
Softmax	$s \cdot \cos\theta_{y_i}$	$s \cdot \cos\theta_{y_i}$	$\left(P_{y_i}^{(i)} - 1\right) s$
Additive Margin (CosFace [15])	$s \cdot \cos\theta_{y_i}$	$s(\cos\theta_{y_i} - m)$	$\left(P_{y_i}^{(i)} - 1\right) s$
Angular Margin (ArcFace [5])	$s \cdot \cos\theta_{y_i}$	$s \cdot \cos(\theta_{y_i} + m)$	$\left(P_{y_i}^{(i)} - 1\right) s \left(\cos(m) + \frac{\cos\theta_{y_i}\sin(m)}{\sqrt{1-\cos^2\theta_{y_i}}}\right)$
Adaptive Angular Margin	$s \cdot \cos\theta_{y_i}$	$s \cdot \cos(\theta_{y_i} + m(\|z_i\|))$	$\left(P_{y_i}^{(i)} - 1\right) s \left(\cos(m(\|z_i\|)) + \frac{\cos\theta_{y_i}\sin(m(\|z_i\|))}{\sqrt{1-\cos^2\theta_{y_i}}}\right)$
	$m(\|z_i\|) =$ a monotonically inc. function of $\|z_i\|$. In this table, g is derived with $\|z_i\|$ as a constant.		
CurricularFace [7]	$N(t, \cos\theta_j)$	$s \cdot \cos(\theta_{y_i} + m)$	$\left(P_{y_i}^{(i)} - 1\right) s \left(\cos(m) + \frac{\cos\theta_{y_i}\sin(m)}{\sqrt{1-\cos^2\theta_{y_i}}}\right)$
	$N(t, \cos\theta_j) = \cos(\theta_j)(t + \cos\theta_j)$ if $s\cos(\theta_{y_i} + m) < \cos\theta_j$ else $\cos(\theta_j)$		
AdaFace (ours)	$s \cdot \cos\theta_{y_i}$	$s \cdot \cos(\theta_{y_i} + g_{\text{angle}}) - g_{\text{add}}$	$\left(P_{y_i}^{(i)} - 1\right) s \left(\cos(g_{\text{angle}}) + \frac{\cos\theta_{y_i}\sin(g_{\text{angle}})}{\sqrt{1-\cos^2\theta_{y_i}}}\right)$
	$g_{\text{angle}} = -m \cdot \widehat{\|\boldsymbol{z}_i\|}$, $g_{\text{add}} = m \cdot \widehat{\|\boldsymbol{z}_i\|} + m$		$\widehat{\|\boldsymbol{z}_i\|} = \left\lfloor \frac{\|\boldsymbol{z}_i\| - \mu_z}{\sigma_z/h} \right\rceil_{-1}^{1}$

Table 1. Table of margin functions and their gradient scale terms. The concept of Adaptive Angular Margin is explored in MagFace [12]. However, unlike other works, MagFace is treating $m(\|z_i\|)$ as a term to optimize (*i.e.* $\|z_i\|$ is a function of $\cos\theta_j$), as oppose to treating it as a constant. In this table, we treat $\|z_i\|$ as a constant to highlight the effect of the margin. The exact form of g for MagFace will be different. In Fig. 3 of the main paper, Adaptive Angular Margin is visualized using the equation from this table.

Note that P_{y_i} is also affected by the choice of the margin function $f(\cos\theta_{y_i})$ as in Eqn. 2. So, g is a function of m, except for Softmax, and g is affected by m through $f(\cos\theta_{y_i})$ in P_{y_i}. For Angular Margin, m appears in the equation for g directly. We derive g for Angular Margin below. The term g for the Adaptive Angular Margin and CurricularFace [7] can be obtained using the g from the Angular Margin. The GST term for AdaFace can be obtained by using g for the Angular Margin and the Additive Margin, and replacing m with adaptive terms g_{angle} and g_{add}. This is possible because $\|z_i\|$ is treated as a constant.

A.1. Derivation of Angular Margin

We can rewrite $f(\cos\theta_{y_i})$ as

$$\begin{aligned} f(\cos\theta_{y_i}) &= s \cdot (\cos(\theta_{y_i} + m)) \\ &= s \cdot (\cos\theta_{y_i}\cos m - \sin\theta_{y_i}\sin m) \\ &= s \cdot \left(\cos\theta_{y_i}\cos m - \sqrt{1-\cos^2\theta_{y_i}}\sin m\right), \end{aligned} \tag{3}$$

by the laws of trignometry. Therefore,

$$\frac{\partial f(\cos\theta_{y_i})}{\partial \cos\theta_{y_i}} = s\left(\cos(m) + \frac{\cos\theta_{y_i}\sin(m)}{\sqrt{1-\cos^2\theta_{y_i}}}\right). \tag{4}$$

A.2. Interpretation of g

For Softmax and Additive Margin, we see that $g = (P_{y_i}^{(i)} - 1)s$. Since the softmax operation in $P_{y_i}^{(i)}$ has a tendency to scale the result to be close to either 0 or 1, the first term in g, $(P_j^{(i)} - 1)$ tends to be close to 1 or 0 far away from the decision boundary. In the equation for P_{y_i}, there is also s which is a scaling hyper-parameter, and is often set to $s = 64$ [5, 7, 10, 15]. This high s makes the softmax operation even steeper near the decision boundary. This results in almost equal GST for samples away from the decision boundary, regardless of how far they are from the decision boundary. This is evident in Fig. 1, where the blue curve is flat except near the decision boundary when s is high.

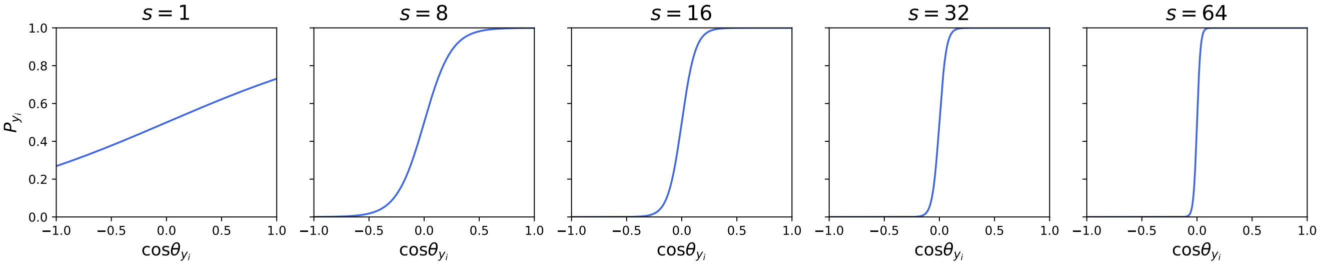


Figure 1. Plot of P_{y_i} for different values of s. In this figure, P_{y_i} is calculated with $f(\cos\theta_j)$ from Softamx (*i.e.* $m = 0$).

For Softmax and Additive Margin, $\frac{\partial f(\cos\theta_{y_i})}{\partial \cos\theta_{y_i}} = s$. This term is different for Angular Margin due to $\frac{\partial f(\cos\theta_{y_i})}{\partial \cos\theta_{y_i}}$ being a function of $\cos\theta_{y_i}$. The exact form of $\frac{\partial f(\cos\theta_{y_i})}{\partial \cos\theta_{y_i}}$ for Angular Margin is found in Eqn. 4. As shown in Fig. 2, Eqn. 4 is monotonically increasing with respect to $\cos\theta_{y_i}$ when $m > 0$ and vice versa. Note that $\cos\theta_{y_i}$ is how close the sample is to the ground truth weight vector, and it is closely related to the difficulty of the sample during training. Therefore, this partial derivative term from the angular margin, $\frac{\partial f(\cos\theta_{y_i})}{\partial \cos\theta_{y_i}}$, can be viewed as scaling the importance of sample based on the difficulty.

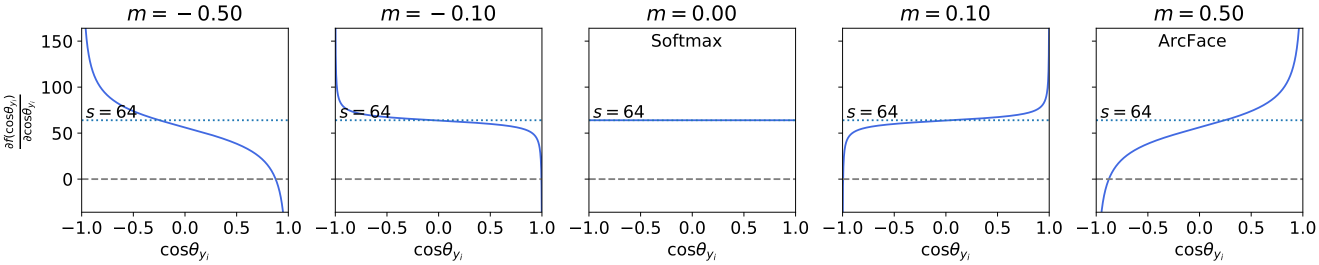
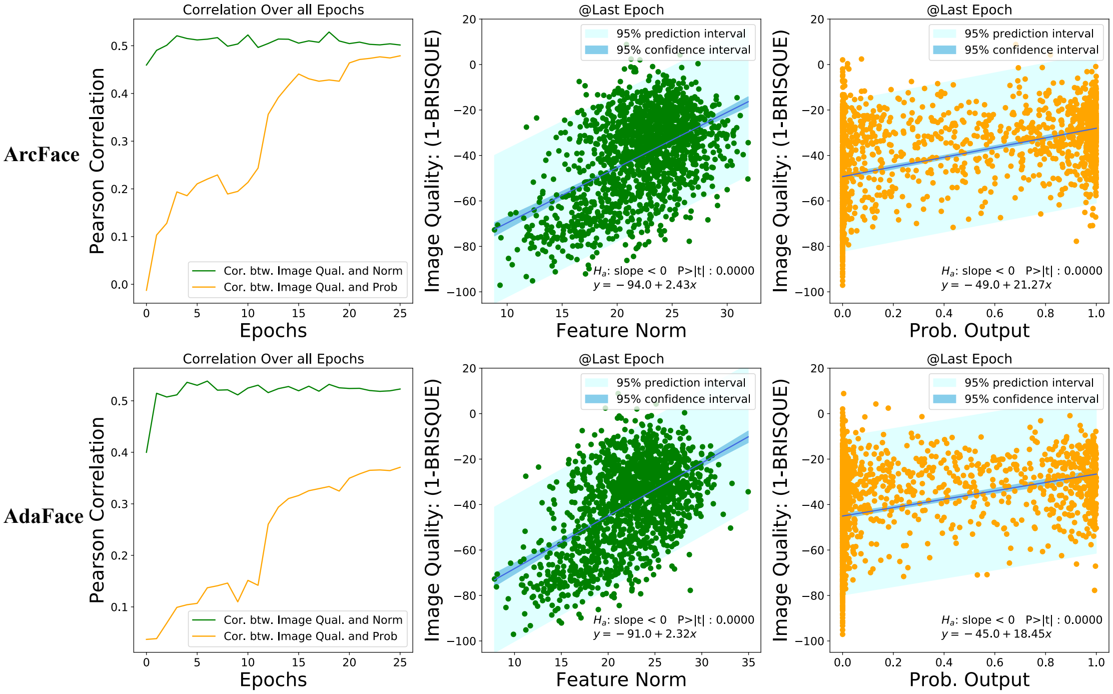


Figure 2. Plot of $\frac{\partial f(\cos\theta_{y_i})}{\partial \cos\theta_{y_i}}$ for different value of m when the margin function is Angluar Margin.

B. Feature Norm Analysis

B.1. Correlation between Norm and BRISQUE during Training

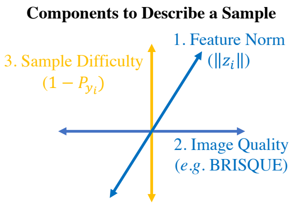
In the Sec. 3.2 of the main paper, we introduce the idea of using the feature norm as a proxy of the image quality. We observe that in models trained with a margin-based softmax loss, the feature norm exhibits a trend that is correlated with the image quality. Here, we show for ArcFace and AdaFace, both loss functions exhibit this trend, in Fig. 3. Regardless of the form of the margin function, the correlation between the feature norm and the image quality is quite similar (green plot in 1st and 2nd columns). We leverage this behavior to design the proxy for the image quality.



a) Correlation for all Epochs **b) Feature Norm vs Img. Qual.** **c) Prob. Output vs Img. Qual.**

Figure 3. Comparison between ArcFace and AdaFace on the correlation between the feature norm and the image quality. We randomly sampled $1,534$ images from the training dataset (MS1MV2 [5]) to show this plot.

We use three concepts (image quality, feature norm and sample difficulty) to describe a sample, as illustrated in Fig. 4. We leverage the correlation between the feature norm and the image quality to apply different emphasis to different difficulty of samples. In contrast, MagFace learns a representation that aligns the feature norm with recognizability. The term, *image quality* in MagFace paper [12] refers to the face recognizability, which is closer in meaning to the sample difficulty than the term, image quality, we use in our paper. Please refer to the Fig. 1 (a) and the first contribution claim of the MagFace paper [12]. Also note the difference in gradient flow through the feature norm, $\|z_i\|$. MagFace relies on learning the feature that has $\|z_i\|$ aligned with the recognizability of the sample, requiring the gradient to flow through $\|z_i\|$ during backpropagation. The loss function has the incentive to reduce the margin by reducing $\|z_i\|$. However, our objective is to adaptively change the loss function, itself, so we treat $\|z_i\|$ as a constant. Finally, from Tab. 3 of our main paper, AdaFace substantially outperforms MagFace, *e.g.* reducing the errors of MagFace on IJB-B and IJB-C relatively by 21% and 23% respectively.



Method	Relationship	Gradient Flow to $\|z_i\|$
MagFace [12]	Sample Difficulty vs. $\|z_i\|$	Yes
AdaFace	Image Qual. vs. $\|z_i\|$	No

Figure 4. An illustration of different components to describe a sample and their usage in previous works.

B.2. Training Sample Visualization

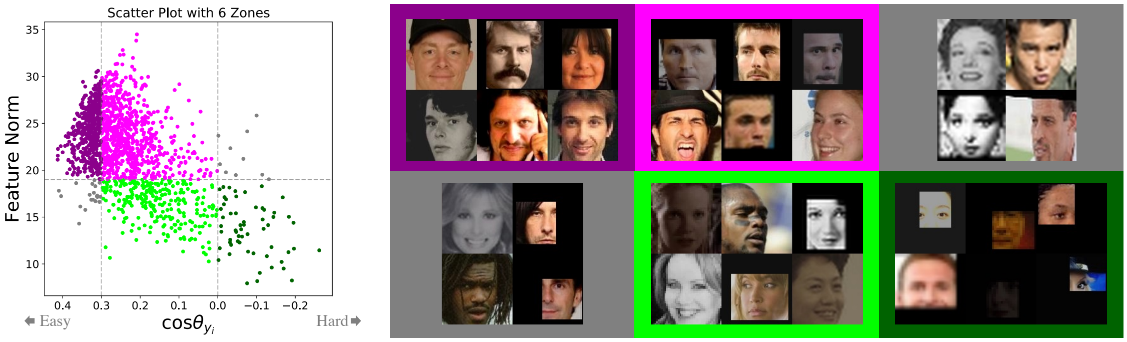
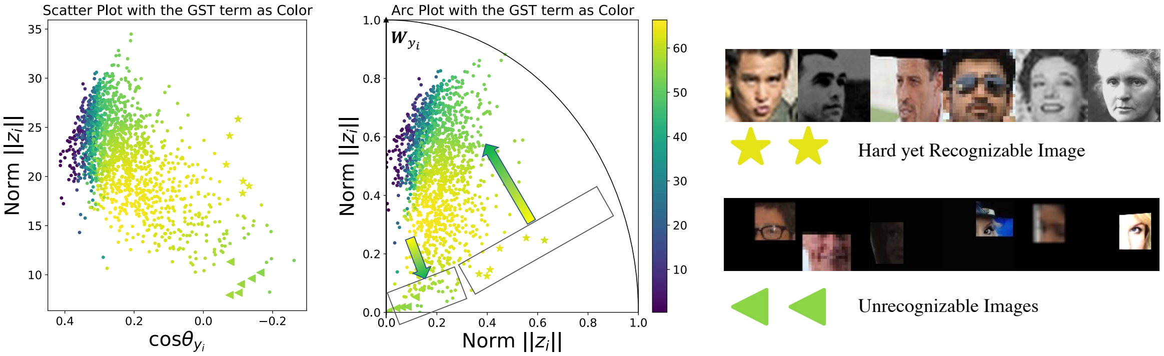


Figure 5. Actual training data examples corresponding to 6 zones. A pretrained AdaFace model is used as a feature extractor.

We show some visualization of the actual training images. From the randomly sampled $1,534$ images from the training dataset (MS1MV2 [5]), we divide the samples into 6 different zones. We plot the samples by $\cos\theta_{y_i}$ (decreasing) as the x-axis and the feature norm $\|z_i\|$ as y-axis in Fig. 5. We divide the plot into 6 zones and sample a few images from each group. Clearly, there are not many samples in the zones highlighted by the gray area (top right and bottom left). This indicates that the sample difficulty distribution is different for each level of feature norm. Furthermore, the samples in the dark green area are mostly unrecognizable images. AdaFace de-emphasizes these samples. Also, the samples in the bright pink area are more difficult samples than the dark pink area. AdaFace puts more emphasis on the harder samples when the feature norm is high. We would like to reminde the readers thatthis figure may serve as *an empirical validation* of the two-dimensional face image categorization we made in Fig. 1 of the main paper.

B.3. Training Samples' Gradient Scaling Term for AdaFace



(a) Scatter Plot between $\cos\theta_{y_i}$ and $\|z_i\|$

(b) Scatter Plot in Angular Space

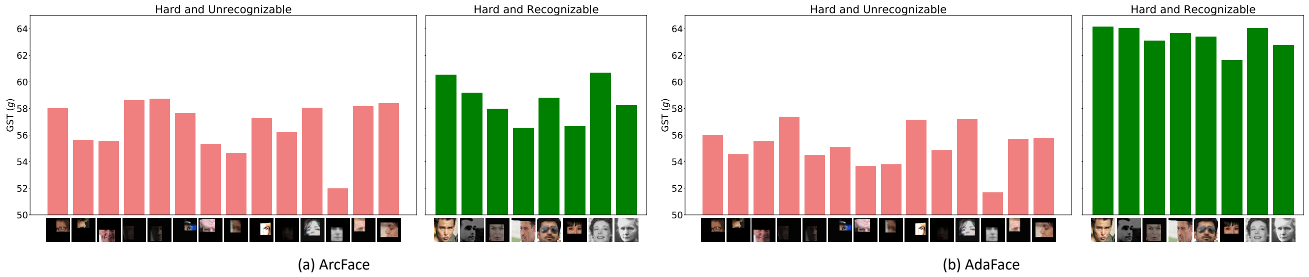
(c) Selected Samples Visualization form the scatter plot

Figure 6. (a) Scatter plot of samples from Fig. 5 with the color as the GST term. (b): Scatter plot of the same $1,534$ points in angular space. For each feature, the angle from W_{y_i} is calculated from $\cos\theta_{y_i}$ and the distance from the origin is calculated from $\|z_i\|$. Both terms are normalized for visualization. (c): Sample image visualization from the low norm and high norm regions of similar $\cos\theta_{y_i}$.

In Fig. 6 (a), we plot the actual GST term for AdaFace. We use the same $1,534$ images from the training dataset (MS1MV2 [5]) as in Fig. 5. The color of points indicates the magnitude of the GST term. The purple points on the left side of the scatter plot are samples past the decision boundary. Therefore the magnitude of GST term is low. The effective difference in GST term for samples outside the decision boundary can be seen by the color change from green to yellow. Note that AdaFace de-emphasizes samples of low feature norm and high difficulty. This is shown in the lower right region of the plot. In Fig. 6 (b), we warp the plot into the angular space to make a correspondence with the Fig. 3 of the main paper, where we illustrate the GST term for AdaFace. We illustrate how actual training samples are distributed in this angular space. In Fig. 6 (b) and (c), we visualize two groups of images where one is from the low feature norm area (triangle) and the other is from the high feature norm area (star). AdaFace exploits images that are hard yet recognizable, as indicated by the yellow star regions, and lowers the learning signal from the unrecognizable images, as indicated by the green triangle regions.

B.4. Train Samples' Gradient Scaling Term Comparison with ArcFace

In Fig. 7, we compare the GST term placed on training samples. We have two groups of images. One group is comprised of unrecognizable images, shown under the red bar. Another group is comprised of hard yet recognizable images, shown under the green bar. Each bar corresponds to one training sample, and the height of the bar indicates the magnitude of the gradient scaling term (GST). For ArcFace shown on the left, the same level of GST is placed on all samples. However, in AdaFace, unrecognizable samples are less emphasized relative to the recognizable samples.



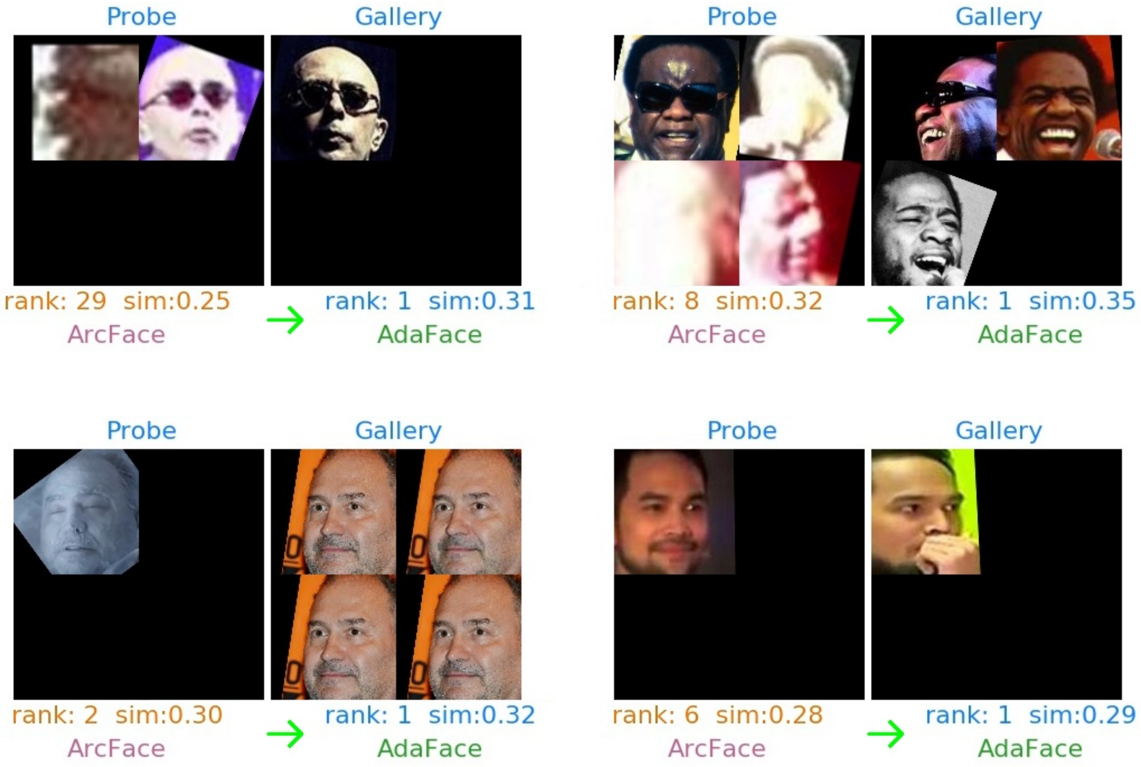
(a) ArcFace

(b) AdaFace

Figure 7. Comparison of the magnitude of GST term between ArcFace and AdaFace.

C. Visualization of Success and Failed Test Images

We show samples from IJB-C [11] dataset to show which samples are correctly classified in AdaFace, compared to ArcFace [5]. In each pair of probe and gallery images, we write the rank and the similarity score for both ArcFace and AdaFace. Rank= 1 is the correct match and a high similarity score is desired. Note that the majority of the cases where AdaFace successfully matches the hard samples for ArcFace are comprised of low quality samples. This shows that indeed AdaFace works well on low quality images.



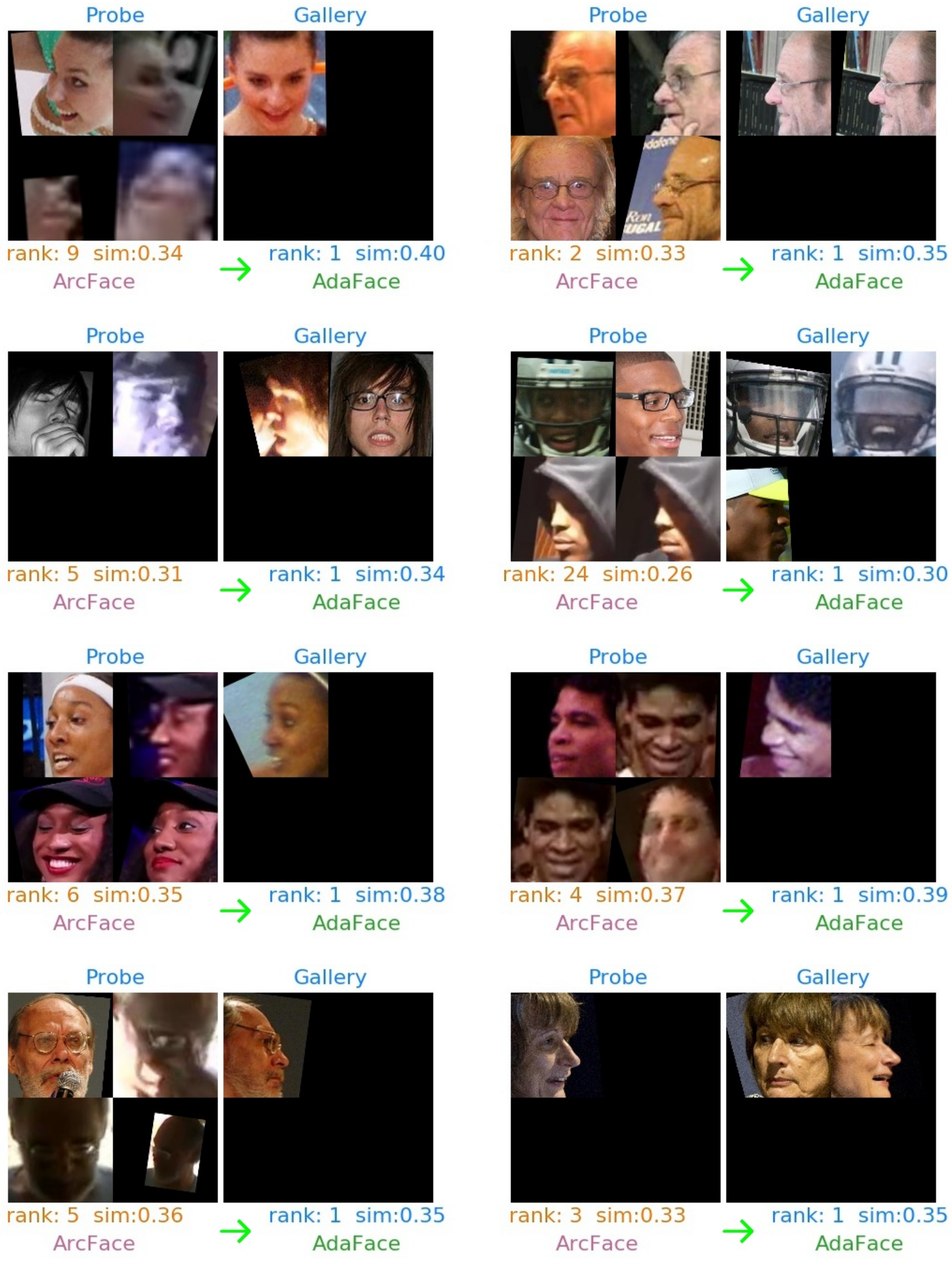


Figure 8. Examples from IJB-C [11] dataset, where ArcFace fails to identify the subject whereas AdaFace successfully finds the correct match between the probe and the gallery. On the left is the set of probe images and on the right is the set of gallery images.

D. Comparison with General Image-Quality Aware Learning Method

We compare our method with QualNet [9] (CVPR21) as a comparison with general image-quality aware learning method. The scope of general image-quality aware learning methods is not limited to face recognition, but the idea is applicable. In Tab. 2, we show the comparison with QualNet with models trained on CASIA-WebFace. AdaFace outperforms QualNet on the TinyFace test set. QualNet aligns the low quality (LQ) image feature distribution to the high quality (HQ) features' distribution via a fixed pretrained decoder. In contrast, AdaFace prevents LQ images from degrading the overall recognition performance by de-emphasizing heavily degraded LQ images. Since LQ facial images can often be devoid of identity, it helps to avoid overfitting on unidentifiable LQ images and learn to exploit the identifiable LQ images. This improves generalization across HQ and LQ.

Method	Training Set	Test set	Rank1	Rank5
QualNet [9]	CASIA-Webface	TinyFace	35.54	44.45
AdaFace			**44.39**	**47.23**

Table 2. Closed set identification performance (ranked match rate) on TinyFace. For a fair comparison, we adopt the train/test setting of QualNet. QualNet results are directly taken from the CVPR21 paper.

E. Effect of Batch Size

Our image quality proxy $\widehat{\|\boldsymbol{z}_i\|}$ does not depend on the batch size due to the exponential moving average in Eq.17 of the main paper (rewritten below).

$$\widehat{\|\boldsymbol{z}_i\|} = \left\lfloor \frac{\|\boldsymbol{z}_i\| - \mu_z}{\sigma_z / h} \right\rceil_{-1}^{1}, \tag{5}$$

$$\mu_z = \alpha \mu_z^{(k)} + (1-\alpha)\mu_z^{(k-1)}. \tag{6}$$

To empirically show this, we train R50 model on MS1MV2 with the batch size of 128, 256 and 512 and report their performance on IJB-B TAR@FAR=0.01%. As shown in Tab. 3, the difference due to the batch size is minimal.

Method	Batch size 128	Batch size 256	Batch size 512
AdaFace	94.32	94.42	94.35

Table 3. Performance comparison by varying the batch size. This shows that AdaFace performance not subject to different batch sizes.

F. Implementation Details and Code

The code is released at `https://github.com/mk-minchul/AdaFace`. For preprocessing the training data MS1MV2 [5], we reference InsightFace [1] and InsightFacePytorch [2], for the backbone model definition, TFace [3] and for evaluation of LFW [6], CFP-FP [14], CPLFW [18], AgeDB [13], CALFW [19], IJB-B [16], and IJB-C [11], we use InsightFace [1]. For preprocessing IJB-S [8] and TinyFace [4], we use MTCNN [17] to align faces.

References

[1] InsightFace. https://github.com/deepinsight/insightface.git. Accessed: 2021-9-1. 7

[2] InsightFacePytorch. https://github.com/TreB1eN/InsightFace_Pytorch.git. Accessed: 2021-9-1. 7

[3] TFace. https://github.com/Tencent/TFace.git. Accessed: 2021-10-3. 7

[4] Zhiyi Cheng, Xiatian Zhu, and Shaogang Gong. Low-resolution face recognition. In *Asian Conference on Computer Vision*, pages 605–621, 2018. 7

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 1, 2, 3, 4, 5, 7

[6] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled Faces in the Wild: A database forstudying face recognition in unconstrained environments. In *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008. 7

[7] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. CurricularFace: adaptive curriculum learning loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5901–5910, 2020. 1, 2

[8] Nathan D Kalka, Brianna Maze, James A Duncan, Kevin O'Connor, Stephen Elliott, Kaleb Hebert, Julia Bryan, and Anil K Jain. IJB–S: IARPA Janus Surveillance Video Benchmark. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–9, 2018. 7

[9] Insoo Kim, Seungju Han, Ji-won Baek, Seong-Jin Park, Jae-Joon Han, and Jinwoo Shin. Quality-agnostic image recognition via invertible decoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12257–12266, 2021. 7

[10] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. SphereFace: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 212–220, 2017. 2

[11] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, and Patrick Grother. IARPA Janus Benchmark-C: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, pages 158–165, 2018. 5, 6, 7

[12] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. MagFace: A universal representation for face recognition and quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14225–14234, 2021. 1, 3

[13] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. AGEDB: the first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 51–59, 2017. 7

[14] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, 2016. 7

[15] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. CosFace: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. 1, 2

[16] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. IARPA Janus Benchmark-B face dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 90–98, 2017. 7

[17] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 7

[18] Tianyue Zheng and Weihong Deng. Cross-Pose LFW: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 5:7, 2018. 7

[19] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-Age LFW: A database for studying cross-age face recognition in unconstrained environments. *CoRR*, abs/1708.08197, 2017. 7