Geometric Order Learning for Rank Estimation

Anonymous Author(s)
Affiliation
Address
email

Abstract

A novel approach to rank estimation, called geometric order learning (GOL), is proposed in this paper. First, we construct an embedding space, in which the direction and distance between objects represent order and metric relations between their ranks, by enforcing two geometric constraints: the order constraint compels objects to be sorted according to their ranks, while the metric constraint makes the distance between objects reflect their rank difference. Then, we perform the simple k nearest neighbor (k-NN) search in the embedding space to estimate the rank of a test object. Moreover, to assess the quality of embedding spaces for rank estimation, we propose a metric called discriminative ratio for ranking (DRR). Extensive experiments on facial age estimation, historical color image (HCI) classification, and aesthetic score regression demonstrate that GOL constructs effective embedding spaces and thus yields excellent rank estimation performances.

1 Introduction

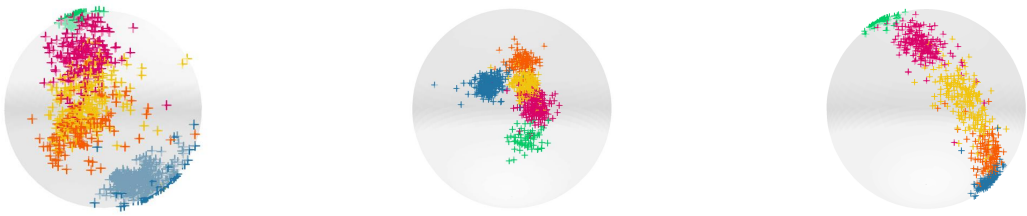
In rank estimation, we estimate the rank (or ordered class) of an object. It is different from ordinary classification in that its classes are arranged in a natural order. For example, in movie rating, classes can be ordered from 'outstanding' to 'very good,' 'satisfactory,' 'unsatisfactory,' and 'poor.' Rank estimation is a fundamental problem and, *e.g.*, used for various computer vision tasks including facial age estimation (Shin et al., 2022), aesthetic quality assessment (Schifanella et al., 2015), and HCI classification (Palermo et al., 2012).

For rank estimation, many techniques (Li et al., 2021; Liu et al., 2018) adopt the ordinal regression framework, which employs a classifier or a regressor to predict the rank of an object directly. However, they may fail to yield reliable estimates, for there is no clear distinction between ranks in many cases. For instance, in facial age estimation, the aging process — causing variations in facial shapes, sizes, and texture — has large individual differences due to factors such as genes, diet, and lifestyle.

To address this issue, comparison-based algorithms (Lim et al., 2020; Lee & Kim, 2021; Li et al., 2014; Nguyen et al., 2018) have been proposed. Instead of predicting the rank directly, they learn a binary relation between objects, such as order or metric (Hrbacek & Jech, 1984). These relations provide useful information for rank estimation: an order indicates the relative priority between objects x and y, while a metric informs of the distance between them.

In order learning (Lim et al., 2020; Lee & Kim, 2021), a comparator is learned to classify the relationship between x and y into one of three cases: x is 'greater than,' 'similar to,' or 'smaller than' y. Then, the rank of a test object is estimated by comparing it with multiple reference objects with known ranks. This approach is based on the idea that it is easier to predict ordering relationships between objects than to estimate the absolute ranks; telling the older one between two people is easier than estimating their exact ages. However, order learning disregards how much x is different from y. In other words, it ignores metric information.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) Order learning (b) Metric learning (c) Proposed

Figure 1: Comparison of embedding spaces of (a) order learning (Lim et al., 2020), (b) metric learning (Schroff et al., 2015), and (c) the proposed algorithm. For each method, we train ResNet18 (He et al., 2016) to classify the RetinaMNIST data (Yang et al., 2021) into the five ordinal classes, depicted by blue, orange, yellow, pink, and green markers. For visualization on the unit sphere, the output dimension of a network is reduced to 3 via a fully connected layer, and each feature is normalized.

On the other hand, metric learning algorithms (Li et al., 2014; Nguyen et al., 2018) employ the triplet constraint on three objects (x, y, z). It enforces the distance between x and y to be less than that between x and z in the embedding space if the ranks of (x, y, z) are in the increasing or decreasing order. By its design, the triplet constraint does not fully exploit the order among objects.

Figures 1 (a) and (b) compare embedding spaces, obtained by order learning and metric learning, respectively. Order learning sorts instances according to ranks in general, but instances in each class are scattered in the embedding space. In contrast, metric learning reduces within-class scattering but does not sort instances properly. For example, both blue and orange ranks are adjacent to the yellow one, although they should be arranged in the order of blue, orange, and yellow. Unlike ordinary classification, different errors have different severities in rank estimation: misclassifying an object in the blue rank as a yellow one is severer than mistaking it for an orange one. Thus, ordering in the embedding space is important to avoid such severe errors.

We propose a novel algorithm, GOL, to estimate the rank of an object reliably by exploiting both order and metric relations. To this end, we construct an embedding space, in which the direction and distance between objects represent the order and metric relations between their ranks. For the construction, we formulate two geometric constraints in the embedding space: 1) the order constraint enforces the feature vectors of instances to be arranged according to their ranks, and 2) the metric constraint makes the distance between instances reflect their rank difference. To satisfy these two constraints simultaneously, we introduce reference points that guide the region of each rank in the embedding space. Then, we use the simple k-NN rule in the embedding space to estimate the rank of a test instance. Extensive experiments show that GOL constructs high-quality embedding spaces and thus provides excellent rank estimation performances.

The contributions of this paper can be summarized as follows.

- To the best of our knowledge, GOL is the first attempt to design an embedding space in which the direction and distance between objects represent their order and metric relations.
- We introduce a novel metric, called DRR, to assess the quality of embedding spaces for rank estimation. Then, it is shown that GOL effectively sorts and separates instances according to their ranks in an embedding space, as illustrated in Figure 1 (c).
- The proposed GOL algorithm provides state-of-the-art performances on various benchmark datasets for facial age estimation, HCI classification, and aesthetic score regression. Specifically, GOL performs the best in 20 out of 25 benchmark tests.

2 Related Work

Ordinal regression: Many ordinal regression methods have been developed to estimate the rank of an object directly using classifiers or regressors. For example, Rothe et al. (2015) employed tens of classifiers to yield the average of their predictions as output. Yi et al. (2014) developed a regressor to estimate the rank of an image using multi-scale patches. Also, Frank & Hall (2001) employed multiple binary classifiers, each of which tells whether the rank of an object is higher than a series of thresholds or not. However, such direct estimation of ranks is challenging even for human beings in

general; *e.g.*, humans usually predict only a rough range of another one's age with limited confidence. Thus, Diaz & Marathe (2019) trained a regressor using soft ordinal labels to alleviate penalties on close predictions. Furthermore, Li et al. (2021) and Li et al. (2022) modeled the uncertainty of each prediction as a Gaussian distribution. In contrast to these methods, the proposed algorithm provides more accurate rank estimates, although it uses only a single encoder network with the simple k-NN rule and makes no complicated probabilistic assumptions.

Order learning: Rank-ordered (or ordinal) descriptors have been used in computer vision. For example, Toews & Wells (2009) proposed SIFT-rank descriptors for correspondence matching, which were also used to model anatomical development in MRI data (Toews et al., 2012). Recently, with the great success of deep learning, Lim et al. (2020) first proposed the notion of order learning, which learns ordering relationships between objects and determines the rank of an unseen object by comparing it with references with known ranks. It yields promising results because relative assessment is easier than absolute assessment in general. Lee & Kim (2021) improved the performance of order learning by finding more reliable references. They decomposed object information into an order-related feature and an identity feature and showed that objects with similar identity features can be compared more reliably. Also, Shin et al. (2022) extended the classification approach in (Lim et al., 2020; Lee & Kim, 2021) to regression-based one. These order learning methods, however, require a significant computational cost to find reliable references from an entire training set. Moreover, for rank estimation, they should do comparisons with many references with different ranks because they consider only relative priorities between objects. On the contrary, the proposed algorithm simply carries out the k-NN search to yield outstanding rank estimation results.

Metric learning: Metric learning aims to construct an embedding space in which the distance between objects reflects their semantic difference. Most metric learning algorithms (Schroff et al., 2015; Deng et al., 2019) are for ordinary classification, clustering, or image retrieval tasks. Therefore, they enforce an object to be located near other objects in the same class in the embedding space but far from objects in different classes. However, in rank estimation, this approach may be suboptimal because it does not consider ordinal relationships among classes. For example, in movie rating, it does not discriminate the class difference between 'outstanding' and 'very good' from that between 'outstanding' and 'poor.' To alleviate this problem, Xiao et al. (2009) designed a metric, called labeled distance, to measure semantic similarities between objects and attempted to preserve local semantic structures in the feature space. Moreover, to preserve the ordinal relationships, Li et al. (2012) developed a metric learning algorithm to make the distances between objects proportional to their rank differences. Also, Tian et al. (2016) employed a series of margins to explicitly impose different embedding distances according to rank differences. Suárez et al. (2021) attempted to make the embedding distances between pairs of objects sorted according to their rank differences.

3 Proposed Algorithm

3.1 Preliminary – Order and Metric

Mathematically, both order and metric are binary relations (Hrbacek & Jech, 1984). An order, denoted by $\leq$, on a set should satisfy the properties of reflexivity, antisymmetry, and transitivity (Schröder, 2003). On the other hand, a metric is a distance function d satisfying nonnegativity, commutativity, and triangle inequality (Rudin, 1991).

In rank estimation, an order describes the priorities of ranks or classes in the set $\Theta = \{\theta_0, \ldots, \theta_{M-1}\}$, where each rank represents one or more object instances. For example, in age estimation, θ_i may represent i-year-olds, and $\theta_{17} < \theta_{32}$ indicates that 17-year-olds are younger than 32-year-olds. Let $\theta(\cdot)$ be the rank function, and let x and y be instances. Then, $\theta(x) = \theta_{17}$ means that person x is 17-year-old. Also, a metric describes the difference between ranks in Θ. For example, $d(\theta(x), \theta(y)) = 15$ means that two people x and y are 15 years apart.

3.2 Embedding Space Construction

Given an object instance x, the objective is to estimate its rank $\theta(x)$. In such rank estimation, an order and a metric convey complementary information: the order provides directional information between ranks, while the metric does length (or magnitude of difference) information. In age estimation, let us consider three age ranks θ_2, θ_{17}, and θ_{32}. Since $\theta_2 < \theta_{17} < \theta_{32}$, the order informs that θ_2 and θ_{32}

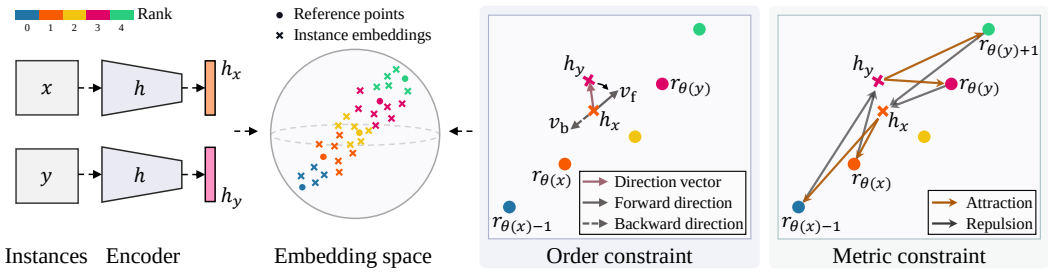


Figure 2: An overview of the proposed GOL algorithm. It is recommended to watch the accompanying video of the proposed algorithm.

are at the opposite sides with respect to θ_{17}. On the other hand, since $d(\theta_2, \theta_{17}) = d(\theta_{32}, \theta_{17}) = 15$, the metric indicates how far θ_2 and θ_{32} are from θ_{17}. In this case, both lengths are identically 15.

For rank estimation, the pairwise comparison methods (Lim et al., 2020; Lee & Kim, 2021; Nguyen et al., 2018) attempt to learn these relations. However, Lim et al. (2020) and Lee & Kim (2021) exploit the order relation only, whereas Nguyen et al. (2018) use the metric relation only to train their neural networks. Thus, the conventional methods may yield sub-optimal results.

To learn both order and metric relations, we propose a geometric approach called GOL. It contains two types of geometric constraints that enforce directional (order) and distance (metric) relationships between object instances according to their ranks in an embedding space. Specifically, the order constraint sorts instances directionally according to the ranks, while the metric constraint separates two instances farther if their rank difference is larger. Figure 2 is an overview of GOL.

Order constraint: Suppose that there are M ranks in a training set $\mathcal{X}$. Without loss of generality, the ranks are assumed to be consecutive integers in $\Theta = \{0, 1, \ldots, M-1\}$. In Figure 2, an encoder h maps each instance $x \in \mathcal{X}$ into a feature vector $h_x = h(x)$ in an embedding space. As h, we adopt VGG16 (Simonyan & Zisserman, 2015) without fully connected layers. The output of the last pooling layer is normalized so that $h_x^t h_x = 1$. Thus, the embedding space is a unit hypersphere.

As in Lim et al. (2020) and Lee & Kim (2021), we classify the ordering between two instances x and y in $\mathcal{X}$ into three categories:

$$x \succ y \text{ if } \theta(x) - \theta(y) > \tau, \quad x \approx y \text{ if } |\theta(x) - \theta(y)| \leq \tau, \quad x \prec y \text{ if } \theta(x) - \theta(y) < -\tau, \tag{1}$$

where τ is a threshold. For instance ordering, notations '$\prec, \approx, \succ$' are used instead of '$<, =, >$.'

The order constraint encourages instances to be sorted according to their ordering relationships. In other words, for two instances x and y with ordering $x \prec y$, the vector from h_x to h_y should be aligned with the direction of the rank increment in the embedding space. To model such rank directions, we introduce M reference points, $r_0, r_1, \ldots, r_{M-1}$, which are learnable parameters guiding the positions of the M ranks in the embedding space. These reference points are randomly initialized by the Glorot normal method (Glorot & Bengio, 2010) and jointly optimized with encoder parameters during training.

Let us define the direction vector $v(r, s)$ from point r to point s on the unit hypersphere as

$$v(r, s) = (s - r)/\|s - r\|. \tag{2}$$

Then, $v(r_i, r_j)$ is called the rank direction from rank i to rank j. Also, the rank direction $v(r_i, r_j)$ is forward if $i < j$, and backward if $i > j$. Note that forward directions may differ from one another, for they may represent different physical changes. For example, in age estimation, visual variations from 0-years-old to 5-years-old are mainly due to craniofacial development, whereas those from 45-years-old to 50-years-old are due to skin aging (Geng et al., 2007).

If $x \prec y$, we determine the forward and backward rank directions, respectively, by

$$v_{\mathrm{f}} = v(r_{\theta(x)}, r_{\theta(y)}), \tag{3}$$

$$v_{\mathrm{b}} = v(r_{\theta(x)}, r_{\theta(x)-1}). \tag{4}$$

Then, the encoder is trained so that the embedded features h_x and h_y satisfy the order constraint:

$$x \prec y \quad \Leftrightarrow \quad v_{\mathrm{f}}^t v(h_x, h_y) > v_{\mathrm{b}}^t v(h_x, h_y). \tag{5}$$

In other words, the direction vector $v(h_x, h_y)$ should be aligned more with the forward direction v_f than with the backward direction v_b.

To enforce the order constraint in (5), we compute the softmax probability $p^{xy} = [p_\mathrm{f}^{xy}, p_\mathrm{b}^{xy}]^t$, where $p_\mathrm{f}^{xy} = e^{v_\mathrm{f}^t v(h_x,h_y)}/(e^{v_\mathrm{f}^t v(h_x,h_y)} + e^{v_\mathrm{b}^t v(h_x,h_y)})$ and $p_\mathrm{b}^{xy} = 1 - p_\mathrm{f}^{xy}$. We then define the order loss L_order as the cross entropy between p^{xy} and $q^{xy} = [q_\mathrm{f}^{xy}, q_\mathrm{b}^{xy}]^t = [1, 0]^t$, given by

$$L_\mathrm{order} = q_\mathrm{f}^{xy} \log p_\mathrm{f}^{xy} + q_\mathrm{b}^{xy} \log p_\mathrm{b}^{xy}. \tag{6}$$

The order loss for case $x \succ y$ is formulated similarly in a symmetric manner.

Metric constraint: Next, we formulate a metric constraint to make the distance between instances in the embedding space reflect their rank difference. Specifically, it is desirable that

$$|\theta(x) - \theta(y)| > \tau \quad \Leftrightarrow \quad d_\mathrm{e}(h_x, h_y) > \gamma \tag{7}$$

where d_e is the Euclidean distance in the embedding space, and γ is a margin. Note that if $|\theta(x) - \theta(y)| > \tau$, either $x \prec y$ or $x \succ y$ in (1).

Hence, the metric constraint in (7) is equivalent to

$$x \approx y \quad \Leftrightarrow \quad d_\mathrm{e}(h_x, h_y) \leq \gamma \tag{8}$$

From the triangle inequality, we have $|d_\mathrm{e}(r_i, h_x) - d_\mathrm{e}(r_i, h_y)| \leq d_\mathrm{e}(h_x, h_y)$ for every reference point r_i. So, if $x \approx y$, we also have $|d_\mathrm{e}(r_i, h_x) - d_\mathrm{e}(r_i, h_y)| \leq \gamma$. To encourage this inequality, we define a loss $L_{x\approx y}$ as

$$L_{x\approx y} = \sum_{i\in\Theta} \max(|d_\mathrm{e}(r_i, h_x) - d_\mathrm{e}(r_i, h_y)| - \gamma, 0). \tag{9}$$

On the contrary, if $x \prec y$, it should be that $d_\mathrm{e}(h_x, h_y) > \gamma$. Thus, we define another loss

$$L_{x\prec y} = \sum_{i:i\leq\theta(x)} \max(d_\mathrm{e}(r_i, h_x) - d_\mathrm{e}(r_i, h_y) + \gamma, 0) + \sum_{j:j\geq\theta(y)} \max(d_\mathrm{e}(r_j, h_y) - d_\mathrm{e}(r_j, h_x) + \gamma, 0). \tag{10}$$

To minimize the first sum, $d_\mathrm{e}(r_i, h_x)$ should be reduced, while $d_\mathrm{e}(r_i, h_y)$ should be increased. Thus, reference points $r_i, 0 \leq i \leq \theta(x)$, are trained to attract h_x and repel h_y, as illustrated in Figure 2. The second sum in (10) is similar. We do not use reference points $r_l, \theta(x) < l < \theta(y)$, which tend to be between h_x and h_y and unhelpful for guiding them. The derivation of $L_{x\prec y}$ in (10) from the metric constraint in (7) is provided in Appendix B. Also, $L_{x\succ y}$ is formulated symmetrically.

Then, we define the metric loss L_metric as

$$L_\mathrm{metric} = [x \succ y] \cdot L_{x\succ y} + [x \approx y] \cdot L_{x\approx y} + [x \prec y] \cdot L_{x\prec y} \tag{11}$$

where $[\cdot]$ is the indicator function. Note that, in (9) or (10), we make h_x and h_y attract or repel each other indirectly through reference points r_i. This is because the direct attraction or repulsion using $d_\mathrm{e}(h_x, h_y)$ may cause h_x and h_y to move in arbitrary directions. Because the reference points are trained also with the order loss in (6), they provide proper directional guidance of attraction or repulsion in (9) or (10). Experimental analysis on $L_{x\prec y}$ in (10) is in Appendix D.6.

Loss function: In addition to the losses for the order and metric constraints, we employ the center loss (Nguyen et al., 2018), which aims at locating each reference point r_i at the center of all instances with rank i.

$$L_\mathrm{center} = d_\mathrm{e}(r_{\theta(x)}, h_x) + d_\mathrm{e}(r_{\theta(y)}, h_y). \tag{12}$$

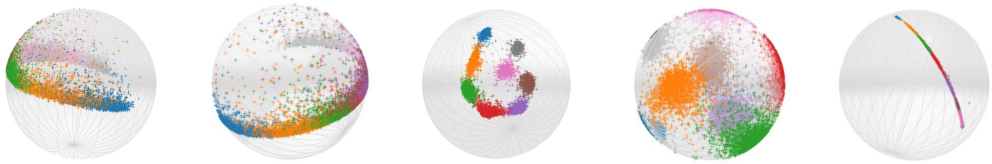
Finally, we use the overall loss function to optimize the encoder parameters and the reference points r_i, which is given by

$$L_\mathrm{total} = L_\mathrm{order} + L_\mathrm{metric} + L_\mathrm{center}. \tag{13}$$

It is worth pointing out that reference points play essential roles in all three loss terms in (13). They are used to define forward and backward rank directions in L_order, so they help to sort instances directionally according to the ranks. The reference points themselves are also sorted since instances are clustered around the reference points because of L_center. Also, as mentioned before, the reference points provide proper directional guidance of attraction or repulsion in L_metric to satisfy the metric constraint. In other words, GOL uses the reference points to satisfy both order and metric constraints simultaneously and thus constructs a well-arranged, well-clustered embedding space.

Table 1: Comparison of B2W and DRR scores on the MORPH II, CACD, and Adience datasets.

	MORPH II (setting A)			CACD (validation split)			Adience		
Algorithm	B2W	$\text{DRR}_{1.0}$	$\text{DRR}_{0.5}$	B2W	$\text{DRR}_{1.0}$	$\text{DRR}_{0.5}$	B2W	$\text{DRR}_{1.0}$	$\text{DRR}_{0.5}$
ML (Schroff et al., 2015)	41.20	10.06	8.94	3.90	2.89	2.55	5.12	3.21	2.98
MV (Pan et al., 2018)	72.47	10.37	9.80	24.21	6.23	5.84	7.08	3.29	3.24
OL (Lim et al., 2020)	43.94	12.79	11.44	8.77	5.60	4.86	13.29	7.27	6.53
MWR-G (Shin et al., 2022)	30.42	10.66	9.28	9.99	6.00	5.24	7.84	5.20	4.71
Proposed GOL	292.72	29.40	25.77	53.06	12.59	10.82	70.31	32.09	28.37



(a) OL (b) MWR-G (c) ML (d) MV (e) GOL

Figure 3: Comparison of embedding spaces on Adience: (a) OL (Lim et al., 2020), (b) MWR-G (Shin et al., 2022), (c) ML (Schroff et al., 2015), (d) MV (Pan et al., 2018), and (e) GOL.

3.3 k-NN Rank Estimation

For rank estimation, we use the simple k-NN rule. Given a test instance x, in the embedding space, we find a set $\mathcal{N}$ of its k NNs among all training instances in $\mathcal{X}$. Then, the rank of x is estimated by

$$\hat{\theta}(x) = \frac{1}{k} \sum_{y \in \mathcal{N}} \theta(y). \tag{14}$$

4 Experimental Results

We conduct experiments on embedding space construction and rank estimation. Implementation details and more results are available in Appendices C and D, respectively.

4.1 Implementation

We initialize an encoder h with VGG16 pre-trained on ILSVRC2012 (Deng et al., 2009) and reference points with the Glorot normal method (Glorot & Bengio, 2010). The Adam optimizer (Kingma & Ba, 2015) is used with a batch size of 32 and a weight decay of 5×10^{-4}, and the initial learning rates for the encoder and the reference points are set to 10^{-4} and 10^{-3}, respectively. We perform the scheduled learning according to cosine annealing cycles (Huang et al., 2017). For data augmentation, we do random horizontal flips only.

4.2 Embedding Spaces

The proposed GOL algorithm attempts to design an embedding space in which both order and metric constraints are satisfied: instances should be well sorted according to their ranks and well separated if they have big rank differences.

To measure the quality of an embedding space for rank estimation, we may adopt the between-class variance to within-class variance (B2W) criterion in Fisher's linear discriminant (Duda et al., 2006),

$$\text{B2W} = \Big(\sum_{i \in \Theta} |\mathcal{X}_i| d_{\text{e}}^2(c_i, c)\Big) / \Big(\sum_{i \in \Theta} \sum_{x \in \mathcal{X}_i} d_{\text{e}}^2(h_x, c_i)\Big) \tag{15}$$

where $\mathcal{X}_i = \{x \in \mathcal{X} \mid \theta(x) = i\}$, $c_i = \sum_{x \in \mathcal{X}_i} h_x / |\mathcal{X}_i|$ is the centroid for $\mathcal{X}_i$, and $c = \sum_{x \in \mathcal{X}} h_x / |\mathcal{X}|$ is the centroid for all instances in $\mathcal{X}$. A high B2W score indicates that the rank sets $\mathcal{X}_i$, $0 \leq i \leq M-1$, are well separated from one another, while instances in each rank set are compactly distributed.

However, B2W does not consider the ordinal relationships of ranks in Θ since it was formulated for ordinary classification. Therefore, to assess embedding spaces for rank estimation, we propose the

Table 2: Comparison of facial age estimation results in the four evaluation settings (A, B, C, and D) of MORPH II. Here, * means that IMDB-WIKI pre-training is performed.

	Setting A		Setting B		Setting C		Setting D	
Algorithm	MAE	CS (%)	MAE	CS (%)	MAE	CS (%)	MAE	CS (%)
DRFs (Shen et al., 2018)	2.91	82.9	2.98	-	-	-	2.17	91.3
MV (Pan et al., 2018)*	-	-	-	-	2.79	-	2.16	-
C3AE (Chao et al., 2019)*	-	-	-	-	-	-	2.75	-
BridgeNet (Li et al., 2019)*	2.38	91.0	2.63	86.0	-	-	-	-
AVDL (Wen et al., 2020)*	2.37	-	2.53	-	-	-	**1.94**	-
OL (Lim et al., 2020)*	2.41	91.7	2.75	88.2	2.68	88.8	2.22	93.3
DRC-ORID (Lee & Kim, 2021)*	2.26	**93.8**	**2.51**	89.7	2.58	89.5	2.16	93.5
MWR-G (Shin et al., 2022)*	2.24	93.5	2.55	**90.1**	2.61	89.5	2.16	93.0
Proposed GOL	**2.17**	**93.8**	2.60	89.3	**2.51**	**90.0**	2.09	**94.2**

discriminative ratio for ranking (DRR), given by

$$\mathrm{DRR}_\beta = \frac{\sum_{i,j\in\Theta:i<j}(j-i)^\beta d_{\mathrm{e}}(c_i,c_j)/\sum_{i,j\in\Theta:i<j}(j-i)^\beta}{\sum_{i\in\Theta}\sum_{x,y\in\mathcal{X}_i:x\neq y} d_{\mathrm{e}}(h_x,h_y)/\sum_{i\in\Theta}\sum_{x,y\in\mathcal{X}_i:x\neq y} 1} \tag{16}$$

which is the ratio of the average pairwise centroid distance to the average pairwise instance distance in each rank set. Note that, in the numerator, ordinal weights $(j-i)^\beta$ are used to emphasize the difference between a pair of rank sets with a large rank difference. Here, β is a nonnegative parameter for controlling the level of emphasis. A high DRR score is obtained when instances are well sorted and well separated according to their ranks in the embedding space.

Table 1 compares the B2W and DRR scores on the MORPH II (Ricanek & Tesafaye, 2006), CACD (Chen et al., 2015), and Adience (Levi & Hassner, 2015) train data. To compare the embedding spaces as fairly as possible, we use the same encoder backbone of VGG16 for all algorithms. GOL significantly outperforms all conventional algorithms in all tests. In GOL, instances within each rank set $\mathcal{X}_i$ are located compactly around reference point r_i by L_{center} in (12), while those in different rank sets are well separated by L_{metric} in (11). Hence, GOL yields outstanding B2W scores. Moreover, GOL excels with a larger score gap in $\mathrm{DRR}_{1.0}$ than in $\mathrm{DRR}_{0.5}$, which means that it arranges the rank sets effectively in the embedding space to reflect their ordinal relationships using L_{order} in (6).

Figure 3 visualizes the embedding spaces on Adience. As in Figure 1, we add a fully connected layer with 3 output neurons to each encoder for the visualization. In (a) and (b), instances are sorted but scattered over the spaces, as the order learning algorithms (Lim et al., 2020; Shin et al., 2022) ignore the metric relation of ranks. In (c), instances in each rank set are well clustered, but the rank sets are not sorted because ML (Schroff et al., 2015) neglects the order relation. In (d), MV (Pan et al., 2018), which is an ordinal regressor, exhibits large within-class scattering, as well as large between-class scattering. In contrast, in (e), GOL constructs a well-sorted, well-clustered embedding space.

4.3 Rank Estimation

Since GOL constructs high-quality embedding spaces, it provides excellent rank estimation performances even with the simple k-NN rule.

Facial age estimation: We use four datasets of MORPH II (Ricanek & Tesafaye, 2006), CACD (Chen et al., 2015), UTK (Zhang et al., 2017), and Adience (Levi & Hassner, 2015), as detailed in Appendix D.3.

In Table 2, we compare the performances in the four evaluation settings of MORPH II, which is one of the most popular datasets in age estimation. We use the mean absolute error (MAE) and cumulative score (CS) metrics. MAE is the average absolute error between estimated and ground-truth ages (*i.e.* ranks), and CS computes the percentage of images whose absolute errors are less than or equal to a tolerance level $l = 5$. Note that all algorithms in Table 2 use VGG16 as the encoder backbones, except for C3AE employing a shallow CNN. As for Shin et al. (2022), the scores of the global ρ-regressor are compared since their local scheme employs as many as six independent VGG16 encoders recursively. GOL performs the best in 5 out of 8 tests, including setting C, which is the most challenging task. Furthermore, unlike most conventional algorithms, GOL does not exploit the

Table 3: Comparison in the train and validation settings of CACD and also on UTK.

Algorithm	Train MAE	Validation MAE	UTK MAE
dLDLF (Shen et al., 2017)	4.73	6.77	-
AGEn (Tan et al., 2017)	4.68	-	-
DRFs (Shen et al., 2018)	4.64	5.77	-
CORAL (Cao et al., 2020)	-	-	5.47
Gustafsson et al. (2020)	-	-	4.65
Berg *et al.* (Berg et al., 2021)	-	-	4.55
MWR-G (Shin et al., 2022)	4.76	5.75	4.49
Proposed GOL	**4.52**	**5.58**	**4.35**

Table 4: Accuracy (%) and MAE comparison on the Adience and HCI datasets.

Algorithm	Adience Acc.	Adience MAE	HCI Acc.	HCI MAE
OR-CNN (Niu et al., 2016)	56.7	0.54	38.7	0.95
CNNPOR (Liu et al., 2018)	57.4	0.55	50.1	0.82
GP-DNNOR (Liu et al., 2019)	57.4	0.54	46.6	0.76
SORD (Diaz & Marathe, 2019)	59.6	0.49	-	-
DRC-ORID (Lee & Kim, 2021)	-	-	44.7	0.80
POE (Li et al., 2021)	60.5	0.47	54.7	0.66
MWR-G (Shin et al., 2022)	62.2	0.46	52.2	0.60
Proposed GOL	**62.5**	**0.43**	**56.2**	**0.55**

Table 5: Accuracy (%) and MAE comparison on the aesthetics dataset.

Algorithm	Nature Acc.	Nature MAE	Animal Acc.	Animal MAE	Urban Acc.	Urban MAE	People Acc.	People MAE	Overall Acc.	Overall MAE
CNNm (Liu et al., 2018)	71.0	0.31	68.0	0.34	68.2	0.36	71.6	0.32	69.5	0.33
CNNPOR (Liu et al., 2018)	71.9	0.29	69.3	0.32	69.1	0.33	69.9	0.32	70.1	0.32
SORD (Diaz & Marathe, 2019)	73.6	**0.27**	70.3	0.31	73.3	0.28	70.6	0.31	72.0	0.29
POE (Li et al., 2021)	73.6	**0.27**	71.1	0.30	72.8	0.28	**72.2**	**0.29**	72.4	0.29
Proposed GOL	**73.8**	**0.27**	**72.4**	**0.28**	**74.2**	**0.26**	69.6	0.31	**72.7**	**0.28**

IMDB-WIKI pretraining to boost performances. Even without the pretraining, *i.e.* by employing only MORPH II training data, GOL yields outstanding results in Table 2.

Table 3 compares the results on CACD, which is a bigger dataset containing over 100,000 natural face shots in diverse environments. GOL outperforms the second-best methods with meaningful gaps of 0.12 and 0.17 in the train and validation settings, respectively, which indicates that it can cope with large and diverse data effectively as well.

Table 3 also compares the results on UTK. Note that Gustafsson et al. (2020) and Berg et al. (2021) employ the deep ResNet50 network (He et al., 2016) as their encoders, and MWR-G (Shin et al., 2022) predicts ranks using a complicated, recursive regressor. In contrast, GOL uses the shallower VGG16 encoder and performs rank estimation based on the simple k-NN search. Nevertheless, it yields an excellent result, for its geometric constraints help to construct an effective embedding space.

Table 4 shows the results on Adience, where each image is labeled as one of 8 age groups. Compared with the second-best MWR-G, GOL improves the accuracy by 0.3% and reduces MAE by 0.03.

HCI classification: The HCI dataset (Palermo et al., 2012) is used for estimating the shooting decade of a photograph. It contains 1,325 images from five decades 1930s $\sim$ 1970s. Table 4 lists the results on HCI. GOL yields the best performances as well, by improving the accuracy by 1.5% and reducing MAE by 0.05 as compared with the second-best methods. This means that GOL yields reliable results even for a small dataset, which may be unfavorable for the k-NN estimation.

Aesthetic score regression: The aesthetics dataset (Schifanella et al., 2015) contains 15,687 images in four categories. Each image is annotated with a 5-scale aesthetic score. In Table 5, we follow the experimental setting in (Liu et al., 2018). It is challenging to estimate aesthetic scores reliably due to the subjectivity and ambiguity of aesthetic criteria, but GOL performs the best in 8 out of 10 tests. Compared with the state-of-the-art POE (Li et al., 2021), GOL improves the accuracy by 0.3% and reduces MAE by 0.1 overall.

4.4 Analysis

Ablation study: Table 6 compares ablated methods for the loss function in (13). Method I employs L_{center} only. In II and III, L_{metric} and L_{order} are excluded, respectively. Compared with IV (GOL), method I degrades the performances severely, since the center loss alone cannot construct a meaningful embedding space, as discussed in Appendix D.6. Also, from II and IV, we see that the metric constraint significantly improves the performances by separating instances with different ranks from each other. From III and IV, we see that the order constraint also improves the results by arranging instances directionally according to their ranks. To summarize, both order and metric constraints improve the results and are complementary to each other.

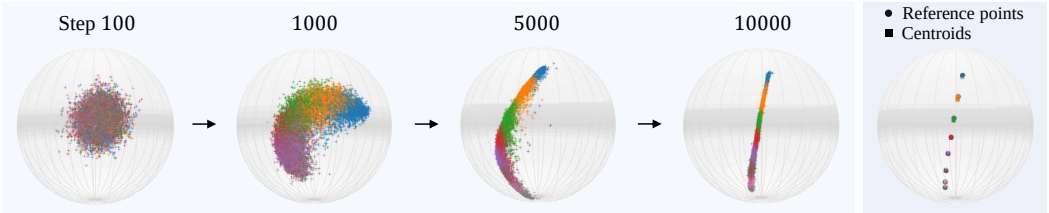


Figure 4: The transition of an embedding space for the Adience dataset during the GOL training.

Table 6: Ablation studies for the loss function in (13) on the MORPH II and CACD datasets.

				MORPH II (setting A)				CACD (validation split)			
	L_{order}	L_{metric}	L_{center}	MAE	B2W	$\text{DRR}_{1.0}$	$\text{DRR}_{0.5}$	MAE	B2W	$\text{DRR}_{1.0}$	$\text{DRR}_{0.5}$
I			✓	8.71	$9 \cdot 10^{-4}$	0.07	0.06	10.0	0.05	0.42	0.38
II	✓		✓	2.71	21.48	8.89	7.69	8.59	5.16	2.66	2.65
III		✓	✓	2.43	44.94	10.93	9.63	6.14	12.72	5.79	5.07
IV	✓	✓	✓	2.17	292.7	29.40	25.77	5.58	53.06	12.59	10.82

Table 7: Comparison with order learning algorithms in terms of testing time and network complexity.

Algorithm	Time (ms)	#parameters (M)
OL (Lim et al., 2020)	11.18	15.51
DRC-ORID (Lee & Kim, 2021)	4.93	45.86
MWR-G (Shin et al., 2022)	1.61	15.77
Proposed GOL	0.07	14.75

Comparison with order learning: The conventional order learning algorithms (Lim et al., 2020; Lee & Kim, 2021; Shin et al., 2022) provide competitive ranking performances, but they require relatively heavy computations. Table 7 lists the testing times. To estimate the rank of a test instance, OL uses five references for each of the M ranks; *e.g.*, 300 comparisons should be made if $M = 60$ as in typical age estimation. Moreover, OL and DRC-ORID need additional processes, such as MAP estimation, to estimate a rank based on ordering relationships. Also, MWR-G estimates the rank by comparing a test instance with multiple reference pairs recursively. In contrast, GOL simply performs the k-NN estimation, which can be done efficiently in a parallel manner, and thus is about 160 and 23 times faster than OL and MWR-G, respectively.

Furthermore, the order learning algorithms should select references among a training set $\mathcal{X}$ through a complicated optimization process; *e.g.*, OL compares all possible pairs of training instances with $O(|\mathcal{X}|^2)$ complexity to select the most reliable references, and DRC-ORID (Lee & Kim, 2021) performs joint network training and clustering for reference selection. On the contrary, GOL requires no such process. Also, in Table 7, the proposed algorithm demands the fewest parameters because it directly estimates the rank of an instance in the embedding space through the k-NN search, whereas the order learning algorithms should adopt comparators or regressors tailored for each task.

Embedding space transition: Figure 4 visualizes the transition of an embedding space for Adience. As the GOL training step goes on, we see that instances are gradually sorted and separated according to their ranks to satisfy the geometric constraints. After the convergence, for each rank, the reference point r_i and the centroid c_i are close to each other, indicating that instances are clustered around their reference points. Moreover, those reference points are well sorted as well.

5 Conclusions

The GOL algorithm for rank estimation was proposed in this work. First, we construct an embedding space based on the two geometric constraints, which enforce the direction and distance between instances to represent the order and metric relations between their ranks. Then, we perform the simple k-NN search in the embedding space for rank estimation. Extensive experiments on various rank estimation tasks demonstrated that GOL constructs high-quality embedding spaces and yields excellent rank estimation results.

References

Axel Berg, Magnus Oskarsson, and Mark O'Connor. Deep ordinal regression with label diversity. In *ICPR*, 2021. 8, 18

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. Rank-consistent ordinal regression for neural networks. *Pattern Recog. Lett.*, 140:325–331, 2020. 8

Jaime S. Cardoso and Joaquim F. Pinto da Costa. Learning to classify ordinal data: The data replication method. *Journal of Machine Learning Research*, 8:1393–1429, 2007. 19

Davide Castelvecchi. Is facial recognition too biased to be let loose? *Nature*, 587(7834):347–349, 2020. 15

Kuang-Yu Chang, Chu-Song Chen, and Yi-Ping Hung. A ranking approach for human age estimation based on face images. In *ICPR*, 2010. 19

Kuang-Yu Chang, Chu-Song Chen, and Yi-Ping Hung. Ordinal hyperplanes ranker with cost sensitivities for age estimation. In *CVPR*, 2011. 19

Zhang Chao, Shuaicheng Liu, Xun Xu, and Ce Zhu. C3AE: Exploring the limits of compact model for age estimation. In *CVPR*, 2019. 7, 19

Bor-Chun Chen, Chu-Song Chen, and Winston H. Hsu. Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. *IEEE Trans. Multimedia*, 17:804–815, 2015. 7, 18

Shixing Chen, Caojin Zhang, Ming Dong, Jialiang Le, and Mike Rao. Using ranking-CNN for age estimation. In *CVPR*, 2017. 19

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 3

Raul Diaz and Amit Marathe. Soft labels for ordinal regression. In *CVPR*, 2019. 3, 8, 18, 20

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley & Sons, 2006. 6

Eibe Frank and Mark Hall. A simple approach to ordinal classification. In *ECML-PKDD*, 2001. 2, 19

Xin Geng, Zhi-Hua Zhou, and Kate Smith-Miles. Automatic age estimation based on facial aging patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:2234–2240, 2007. 4

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 4, 6

Guodong Guo and Guowang Mu. Simultaneous dimensionality reduction and human age estimation via kernel partial least squares regression. In *CVPR*, 2011. 19

Fredrik K. Gustafsson, Martin Danelljan, Goutam Bhat, and Thomas B. Schon. Energy-based models for deep probabilistic regression. In *ECCV*, 2020. 8, 18

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 8

Karel Hrbacek and Thomas Jech. *Introduction to Set Theory*. Marcel Dekker, Inc., 2nd edition, 1984. 1, 3

Han Hu, Anil K. Jain, Fang Wang, Shiguang Shan, and Xilin Chen. Heterogeneous face attribute estimation: A deep multi-task learning approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40: 2597–2609, 2017. 19

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get M for free. In *ICLR*, 2017. 6

Ivan Huerta, Carles Fernández, Carlos Segura, Javier Hernando, and Andrea Prati. A deep analysis on age estimation. *Pattern Recog. Lett.*, 68:239–249, 2015. 19

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 16

Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

Seon-Ho Lee and Chang-Su Kim. Deep repulsive clustering of ordered data based on order-identitiy decomposition. In *ICLR*, 2021. 1, 3, 4, 7, 8, 9, 18, 19

Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *CVPR Workshops*, 2015. 7, 18

Changsheng Li, Qingshan Liu, Jing Liu, and Hanqing Lu. Learning distance metric regression for facial age estimation. In *ICPR*, 2012. 3

Changsheng Li, Qingshan Liu, Jing Liu, and Hanqing Lu. Ordinal distance metric learning for image ranking. *TNNLS*, 26:1551–1559, 2014. 1, 2

Qiang Li, Jingjing Wang, Zhaoliang Yao, Yachun Li, Pengju Yang, Jingwei Yan, Chunmao Wang, and Shiliang Pu. Unimodal-concentrated loss: Fully adaptive label distribution learning for ordinal regression. In *CVPR*, 2022. 3

Wanhua Li, Jiwen Lu, Jianjiang Feng, Chunjing Xu, Jie Zhou, and Qi Tian. BridgeNet: A continuity-aware probabilistic network for age estimation. In *CVPR*, 2019. 7, 19

Wanhua Li, Xiaoke Huang, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Learning probabilistic ordinal embeddings for uncertainty-aware regression. In *CVPR*, 2021. 1, 3, 8, 18, 19, 20

Kyungsun Lim, Nyeong-Ho Shin, Young-Yoon Lee, and Chang-Su Kim. Order learning and its application to age estimation. In *ICLR*, 2020. 1, 2, 3, 4, 6, 7, 9, 18, 19

Hsuan-Tien Lin and Ling Li. Reduction from cost-sensitive ordinal ranking to weighted binary classification. *Neural Computation*, 24(5):1329–1367, 2012. 19, 20

Yanzhu Liu, Adams W. K. Kong, and Chi Keong Goh. A constrained deep neural network for ordinal regression. In *CVPR*, 2018. 1, 8, 18, 19, 20

Yanzhu Liu, Fan Wang, and Adams W. K. Kong. Probabilistic deep ordinal regression based on Gaussian processes. In *CVPR*, 2019. 8, 18, 19

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 20, 21

Paul Martin, Antoine Doucet, and Frédéric Jurie. Dating color images with ordinal classification. In *Proc. ACM ICMR*, 2014. 19

Bac Nguyen, Carlos Morell, and Bernard D. Baets. Distance metric learning for ordinal classification based on triplet constraints. *Knowledge-Based Systems*, 142:17–28, 2018. 1, 2, 4, 5

Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output CNN for age estimation. In *CVPR*, 2016. 8, 15, 19, 20

Richard V. Noorden. The ethical questions that haunt facial-recognition research. *Nature*, 587(7834): 354–358, 2020. 15

Roy Or-El, Soumyadip Sengupta, Ohad Fried, Eli Shechtman, and Ira Kemelmacher-Shlizerman. Lifespan age transformation synthesis. In *ECCV*, 2020. 15

Frank Palermo, James Hays, and Alexei A. Efros. Dating historical color images. In *ECCV*, 2012. 1, 8, 19

Hongyu Pan, Hu Han, Shiguang Shan, and Xilin Chen. Mean-variance loss for deep age estimation from a face. In *CVPR*, 2018. 6, 7, 18, 19

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaisonet, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NIPS*, 2019. 16

Karl Ricanek and Tamirat Tesafaye. MORPH: A longitudinal image database of normal adult age-progression. In *FGR*, 2006. 7, 15, 18

Rasmus Rothe, Radu Timofte, and Luc V. Gul. DEX: Deep expectation of apparent age from a single image. In *ICCV Workshops*, 2015. 2

Rasmus Rothe, Radu Timofte, and Luc V. Gul. Deep expectation of real and apparent age from a single image without facial landmarks. *Int. J. Comput. Vis.*, 126(2-4):144–157, 2018. 18, 19

Antoaneta Roussi. Resisting the rise of facial recognition. *Nature*, 587(7834):350–353, 2020. 15

Walter Rudin. *Functional Analysis*. McGraw-Hill, 2nd edition, 1991. 3

Rossano Schifanella, Miriam Redi, and Luca M. Aiello. An image is worth more than a thousand favorites: Surfacing the hidden beauty of Flickr pictures. In *ICWSM*, 2015. 1, 8, 19

Bernd S. W. Schröder. *Ordered Sets: An Introduction*. Springer, 2003. 3

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 2, 3, 6, 7, 18

Wei Shen, Kai Zhao, Yilu Guo, and Alan Yuille. Label distribution learning forests. In *NIPS*, 2017. 8

Wei Shen, Yilu Guo, Yan Wang, Kai Zhao, Bo Wang, and Alan Yuille. Deep regression forests for age estimation. In *CVPR*, 2018. 7, 8, 18, 19

Nyeong-Ho Shin, Seon-Ho Lee, and Chang-Su Kim. Moving window regression: A novel approach to ordinal regression. In *CVPR*, 2022. 1, 3, 6, 7, 8, 9, 18, 19

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICML*, 2015. 4

Juan L. Suárez, Salvador García, and Francisco Herrera. Ordinal regression with explainable distance metric learning based on ordered sequences. *Machine Learning*, 110:2729–2762, 2021. 3

Zichang Tan, Jun Wan, Zhen Lei, Ruicong Zhi, Guodong Guo, and Stan Z. Li. Efficient group-n encoding and decoding for facial age estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(11): 2610–2623, 2017. 8, 19

Qing Tian, Songcan Chen, and Lishan Qiao. Ordinal margin metric learning and its extension for cross-distribution image data. *Information Sciences*, 349:50–64, 2016. 3

Matthew Toews and William Wells. SIFT-Rank: Ordinal description for invariant feature correspondence. In *CVPR*, 2009. 3

Matthew Toews, William M Wells, and Lilla Zöllei. A feature-based developmental model of the infant brain in structural MRI. In *MICCAI*, 2012. 3

Xin Wen, Biying Li, Haiyun Guo, Zhiwei Liu, Guosheng Hu, Ming Tang, and Jinqiao Wang. Adaptive variance based label distribution learning for facial age estimation. In *ECCV*, 2020. 7, 15, 19

Bo Xiao, Xiaokang Yang, Yi Xu, and Hongyuan Zha. Learning distance metric for regression by semidefinite programming with application to human age estimation. In *Proc. ACM Multimedia*, 2009. 3

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. MedMNIST v2: A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *arXiv preprint arXiv:2110.14795*, 2021. 2

Dong Yi, Zhen Lei, and Stan Z. Li. Age estimation by multi-scale convolutional network. In *ACCV*, 2014. 2, 19

Byungin Yoo, Youngjun Kwak, Youngsung Kim, Changkyu Choi, and Junmo Kim. Deep facial age estimation using conditional multitask learning with weak label expansion. *IEEE Signal Process. Lett.*, 25:808–812, 2018. 19

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *CVPR*, 2017. 7, 15, 18

Tan Zichang, Shuai Zhou, Jun Wan, Zhen Lei, and Stan Z. Li. Age estimation based on a single network with soft softmax of aging modeling. In *ACCV*, 2016. 19

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See failure cases in Appendix D.6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix A.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 4.1 and Appendix C.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We will release the source codes.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix C.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See Appendix D.3.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Appendix D.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Broader Impacts

Recently, ethical concerns about the fairness of deep-learning-based systems have been raised (Castelvecchi, 2020; Roussi, 2020; Noorden, 2020). Especially, due to the intrinsic imbalance of facial datasets (Ricanek & Tesafaye, 2006; Zhang et al., 2017; Niu et al., 2016), most deep learning methods on facial analysis (Wen et al., 2020; Or-El et al., 2020) have unwanted gender or racial bias. The proposed algorithm is not free from this bias either when trained on such datasets. Hence, the bias should be resolved before any practical usage. Also, even though the proposed algorithm discovers some subclasses by ranking instances, these results should never be misinterpreted in such a way as to encourage any kind of discrimination. We recommend using the proposed algorithm for research only.

B Derivation of $L_{\mathrm{x}\prec\mathrm{y}}$ in (10)

From the metric constraint in (7), if $x \prec y$, it should be that $d_{\mathrm{e}}(h_x, h_y) > \gamma$, or equivalently

$$-d_{\mathrm{e}}(h_x, h_y) < -\gamma \tag{17}$$

Note that reference points and instances are roughly sorted according to their corresponding ranks in the embedding space by the order constraint in (5). Therefore, for $x \prec y$, h_x tends to be closer to each reference point r_i, $0 \leq i \leq \theta(x)$, than h_y is. In other words, $x \prec y$, it is likely that

$$d_{\mathrm{e}}(r_i, h_x) - d_{\mathrm{e}}(r_i, h_y) < 0. \tag{18}$$

However, to satisfy the constraint in (17), we impose a stricter upper bound on the difference $d_{\mathrm{e}}(r_i, h_x) - d_{\mathrm{e}}(r_i, h_y)$ as follows. From the triangle inequality, we have

$$d_{\mathrm{e}}(r_i, h_y) \leq d_{\mathrm{e}}(r_i, h_x) + d_{\mathrm{e}}(h_x, h_y), \tag{19}$$

which is equivalent to

$$-d_{\mathrm{e}}(h_x, h_y) \leq d_{\mathrm{e}}(r_i, h_x) - d_{\mathrm{e}}(r_i, h_y). \tag{20}$$

Therefore, if

$$d_{\mathrm{e}}(r_i, h_x) - d_{\mathrm{e}}(r_i, h_y) < -\gamma, \tag{21}$$

the constraint in (17) is satisfied. This desirable condition in (21) is formulated into the first sum in $L_{\mathrm{x}\prec\mathrm{y}}$ in (10).

Similarly, for each reference point r_j, $\theta(y) \leq j \leq M-1$, it is likely that

$$d_{\mathrm{e}}(r_j, h_y) - d_{\mathrm{e}}(r_j, h_x) \leq 0. \tag{22}$$

However, to satisfy the metric constraint, it is desirable that

$$d_{\mathrm{e}}(r_j, h_y) - d_{\mathrm{e}}(r_j, h_x) < -\gamma, \tag{23}$$

which is formulated into the second sum in $L_{\mathrm{x}\prec\mathrm{y}}$ in (10).

C Implementation Details

C.1 Network Architecture

The structure of the encoder h is detailed in Table 8, where '$k_h \times k_w$-s-c Conv' denotes the 2D convolution with kernel size $k_h \times k_w$, stride s, and c output channels. Similarly, '$k_h \times k_w$-s MaxPool' and '$k_h \times k_w$-s AvgPool' represent the 2D max pooling and 2D average pooling with a $k_h \times k_w$ kernel at stride s, respectively. Also, BN means batch normalization (Ioffe & Szegedy, 2015). The encoder is based on the VGG16 network and takes a $224 \times 224 \times 3$ image as input.

Table 8: The structure of the encoder h.

Layers	Output
3×3-1-64 Conv BN ReLU	224×224×64
3×3-1-64 Conv BN ReLU	224×224×64
3×3-2 MaxPool	112×112×64
3×3-1-128 Conv BN ReLU	112×112×128
3×3-1-128 Conv BN ReLU	112×112×128
3×3-2 MaxPool	56×56×128
3×3-1-256 Conv BN ReLU	56×56×256
3×3-1-256 Conv BN ReLU	56×56×256
3×3-1-256 Conv BN ReLU	56×56×256
3×3-2 MaxPool	28×28×256
3×3-1-512 Conv BN ReLU	28×28×512
3×3-1-512 Conv BN ReLU	28×28×512
3×3-1-512 Conv BN ReLU	28×28×512
3×3-2 MaxPool	14×14×512
3×3-1-512 Conv BN ReLU	14×14×512
3×3-1-512 Conv BN ReLU	14×14×512
3×3-1-512 Conv BN ReLU	14×14×512
14×14-1 AvgPool	1×1×512

C.2 Embedding Space Visualization

In Figure 1, ResNet18 is employed as an encoder, but the output dimension of its last fully connected layer is reduced to 3 for the visualization. The encoder takes a $28 \times 28 \times 3$ image as input. We use the same training setting in Section 4.1.

Similarly, for the other 3D embedding space visualizations, including Figures 3 and 4, we modify the encoder in Table 8 to append a fully connected layer with output dimension 3. We train the modified encoder using the same setting in Section 4.1.

C.3 Computing Environment

We do all experiments using PyTorch (Paszke et al., 2019) and an NVIDIA GeForce RTX 3090 GPU.

C.4 Training Time

We train the encoder until the loss converges. Table 9 lists the training epochs and time for each dataset.

Table 9: Training epochs and time.

	MORPH (setting A)	MORPH (setting C)	CACD (train split)	CACD (validation split)	UTK	Adience	HCI	Aesthetics
# epochs	250	150	10	30	50	80	150	50
Time (hrs)	5	30	7	1	3	4	1	3

D More Experimental Results

D.1 Hyper-Parameters

The proposed GOL algorithm has three hyper-parameters; τ in (1), γ in (7), and k in (14). Unless specified otherwise, we set $\tau = 0$ and $\gamma = 0.05$. Also, we fix $k = 50$ and $k = 16$ for facial age estimation tasks and the others, respectively. Let us describe how the hyper-parameters affect performances.

Hyper-parameter γ: Table 10 compares the MAE scores at different γ's on the MORPH II, CACD, and UTK datasets. In this test, $\tau = 0$ and $k = 50$. Note that γ is a margin to impose the minimum distance between instances with a rank difference larger than τ in the embedding space. We see that $\gamma = 0.05$ performs well in general, so we set $\gamma = 0.05$ as the default option. Only for UTK, we set $\gamma = 0.25$, even though GOL outperforms the state-of-the-art MWR-G (4.49) at the other γ levels as well.

Table 10: MAE performances according to γ on MORPH II, CACD, and UTK.

γ	MORPH (setting D)	CACD (validation split)	UTK
0.05	2.08	5.58	4.48
0.15	2.13	5.78	4.44
0.25	2.09	5.60	4.35

Hyper-parameter τ: Table 11 compares the MAE scores at different τ's on MORPH II, CACD, and UTK. In this test, $\gamma = 0.05$ and $k = 50$. We see that $\tau = 0$ provides decent results in general. We hence set the default $\tau = 0$.

Table 11: MAE performances according to τ on MORPH II, CACD, and UTK.

τ	MORPH II (setting D)	CACD (validation split)	UTK
0	2.08	5.58	4.48
1	2.12	5.71	4.45
2	2.18	5.99	4.46

Hyper-parameter k: Table 12 compares the MAE scores according to k. In each age estimation task, there are more than 40 ranks, but there are no clear distinctive characteristics for each rank. Therefore, we set a relatively large $k = 50$ as the default option. However, note that the results are not very sensitive as long as $k \geq 18$.

Table 12: MAE performances according to k on MORPH II, CACD, and UTK.

k	2	10	18	26	34	42	50	58
MORPH (setting D)	2.14	2.11	2.11	2.10	2.09	2.09	2.08	2.09
CACD (validation split)	5.73	5.62	5.59	5.59	5.58	5.58	5.58	5.57
UTK	4.63	4.52	4.51	4.50	4.49	4.48	4.48	4.49

Next, Table 13 compares the MAE scores on different folds of the HCI dataset according to k. The HCI and aesthetics datasets contain only five ranks, respectively. Thus, we use a relatively small $k = 16$ for these tasks.

D.2 Embedding Spaces

Table 14 compares the B2W and DRR scores on the test data of MORPH II, CACD, and Adience. GOL performs the best in all tests, as well as for the training data in Table 1. These results indicate that GOL can arrange the rank sets effectively in the embedding space to reflect their ordinal relationships even for unseen test instances.

Table 13: MAE performances according to k on the HCI dataset. The performances for 5 folds out of 10 are listed.

k	4	8	12	16	20	24	28	32
Fold 1	0.528	0.532	0.512	0.508	0.500	0.504	0.504	0.504
Fold 2	0.584	0.596	0.588	0.584	0.584	0.584	0.588	0.588
Fold 3	0.548	0.544	0.540	0.532	0.532	0.528	0.528	0.528
Fold 4	0.516	0.512	0.512	0.508	0.508	0.508	0.512	0.512
Fold 5	0.532	0.532	0.528	0.524	0.532	0.536	0.540	0.540
Average	0.542	0.543	0.536	0.531	0.531	0.532	0.534	0.534

Table 14: Comparison of B2W and DRR scores on the MORPH II, CACD and Adience test data.

	MORPH II (setting A)			CACD (validation split)			Adience		
Algorithm	B2W	$\text{DRR}_{1.0}$	$\text{DRR}_{0.5}$	B2W	$\text{DRR}_{1.0}$	$\text{DRR}_{0.5}$	B2W	$\text{DRR}_{1.0}$	$\text{DRR}_{0.5}$
ML (Schroff et al., 2015)	6.84	4.14	3.64	0.83	1.39	1.20	3.15	2.45	2.24
MV (Pan et al., 2018)	3.89	2.90	2.63	0.75	1.30	1.15	1.45	1.69	1.60
OL (Lim et al., 2020)	5.16	3.80	3.38	1.31	1.99	1.72	5.58	4.16	3.71
MWR-G (Shin et al., 2022)	7.23	4.85	4.20	1.09	1.80	1.55	4.71	3.63	3.25
Proposed GOL	9.47	5.21	4.54	1.71	2.29	1.97	11.49	7.24	6.39

D.3 Rank Estimation – Facial Age Estimation

Datasets: We provide facial age estimation results of GOL on four datasets: MORPH II (Ricanek & Tesafaye, 2006), CACD (Chen et al., 2015), UTK (Zhang et al., 2017), and Adience (Levi & Hassner, 2015). MORPH II has the Institutional Review Board approval. The other datasets were made for academic research purposes only. Any images getting deletion requests from the original owners will be discarded from the datasets. There are no name labels, except for CACD containing celebrity names. We exploit the datasets only for the performance assessment of GOL.

MORPH II (Ricanek & Tesafaye, 2006) is a popular dataset for facial age estimation, containing about 55,000 facial images of 13,617 subjects in the age range [16, 77]. In each image, the gender and race labels are annotated. Based on these labels, various evaluation protocols have been proposed. We employ the four evaluation settings A, B, C, and D (Lim et al., 2020; Lee & Kim, 2021; Shin et al., 2022).

- Setting A: 5,492 Caucasian images are randomly sampled and divided into training and testing sets with a ratio of 8:2.
- Setting B: About 21K images of Caucasians and Africans are randomly chosen so that the ratio between Caucasians and Africans is 1:1 and that between females and males is 1:3. Then, it is divided into three subsets (S1, S2, S3). The training and testing are repeated twice — 1) training on S1, testing on S2+S3, and 2) training on S2, testing on S1+S3.
- Setting C: The entire dataset is randomly split into five folds, subject to the constraint that the same person should belong to only one fold, and the 5-fold cross-validation is performed.
- Setting D: The whole dataset is randomly divided into five folds without any constraint, and the 5-fold cross-validation is performed.

CACD (Chen et al., 2015) provides 160k images from 2,000 celebrities. It is split into three subsets by celebrities: 1,800 for training, 80 for validation, and 120 for testing. We provide two results by training on the train set and on the validation set, respectively, as done in (Rothe et al., 2018; Shen et al., 2018; Shin et al., 2022). The age range is [14, 62].

UTK (Zhang et al., 2017) consists of 20,000 facial images in a wide age range [0, 116]. We adopt the evaluation protocol in (Gustafsson et al., 2020; Berg et al., 2021).

Adience (Levi & Hassner, 2015) is used for age group estimation. There are 26,580 facial images from 2,284 subjects, which are grouped into 8 ordinal classes: 0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, and over 60-year-olds. We employ the 5-fold subject-exclusive (SE) cross-validation evaluation setting in (Liu et al., 2018, 2019; Diaz & Marathe, 2019; Li et al., 2021).

More comparison: Table 15 provides comparison results with more conventional algorithms on MORPH II. It is an extended version of Table 2.

Table 15: Comparison of facial age estimation results in the four evaluation settings (A, B, C, and D) of MORPH II. Here, * means that IMDB-WIKI pre-training is performed.

	Setting A		Setting B		Setting C		Setting D	
	MAE	CS(%)	MAE	CS(%)	MAE	CS(%)	MAE	CS(%)
RED-SVM (Chang et al., 2010)	-	-	-	-	-	-	6.49	49.0
OHRank (Chang et al., 2011)	-	-	-	-	-	-	6.07	56.3
KPLS (Guo & Mu, 2011)	-	-	4.18	-	-	-	-	-
CPLF (Yi et al., 2014)	-	-	3.63	-	-	-	-	-
Huerta *et al.* (Huerta et al., 2015)	-	-	-	-	3.88	-	-	-
OR-CNN (Niu et al., 2016)	-	-	-	-	-	-	3.27	73.0
Tan *et al.* (Zichang et al., 2016)	-	-	3.03	-	-	-	-	-
Ranking-CNN (Chen et al., 2017)	-	-	-	-	-	-	2.96	85.0
DEX (Rothe et al., 2018)*	2.68	-	-	-	-	-	-	-
DMTL (Hu et al., 2017)	-	-	-	-	3.00	85.3	-	-
CMT (Yoo et al., 2018)	-	-	-	-	2.91	-	-	-
DRFs (Shen et al., 2018)	2.91	82.9	2.98	-	-	-	2.17	91.3
AGEn (Tan et al., 2017)*	2.52	85.0	2.70	83.0	-	-	-	-
MV (Pan et al., 2018)*	-	-	-	-	2.79	-	2.16	-
C3AE (Chao et al., 2019)*	-	-	-	-	-	-	2.75	-
BridgeNet (Li et al., 2019)*	2.38	91.0	2.63	86.0	-	-	-	-
AVDL (Wen et al., 2020)*	2.37	-	2.53	-	-	-	**1.94**	-
OL (Lim et al., 2020)*	2.41	91.7	2.75	88.2	2.68	88.8	2.22	93.3
DRC-ORID (Lee & Kim, 2021)*	2.26	**93.8**	**2.51**	89.7	2.58	89.5	2.16	93.5
MWR-G (Shin et al., 2022)*	2.24	93.5	2.55	**90.1**	2.61	89.5	2.16	93.0
Proposed GOL	**2.17**	**93.8**	2.60	89.3	**2.51**	**90.0**	2.09	**94.2**

D.4 Rank Estimation – HCI Classification

Table 16 is an extended version of the HCI results in Table 4.

Table 16: Accuracy (%) and MAE comparison on the HCI dataset.

	HCI	
Algorithm	Accuracy (%)	MAE
Frank & Hall (Frank & Hall, 2001)	41.4	0.99
Cardoso *et al.* (Cardoso & da Costa, 2007)	41.3	0.95
Palermo *et al.* (Palermo et al., 2012)	44.9	0.93
RED-SVM (Lin & Li, 2012)	35.9	0.96
Martin *et al.* (Martin et al., 2014)	42.8	0.87
OR-CNN (Niu et al., 2016)	38.7	0.95
CNNPOR (Liu et al., 2018)	50.1	0.82
GP-DNNOR (Liu et al., 2019)	46.6	0.76
DRC-ORID (Lee & Kim, 2021)	44.7	0.80
POE (Li et al., 2021)	54.7	0.66
MWR-global (Shin et al., 2022)	52.2	0.60
Proposed GOL	**56.2**	**0.55**

D.5 Rank Estimation – Aesthetic Score Regression

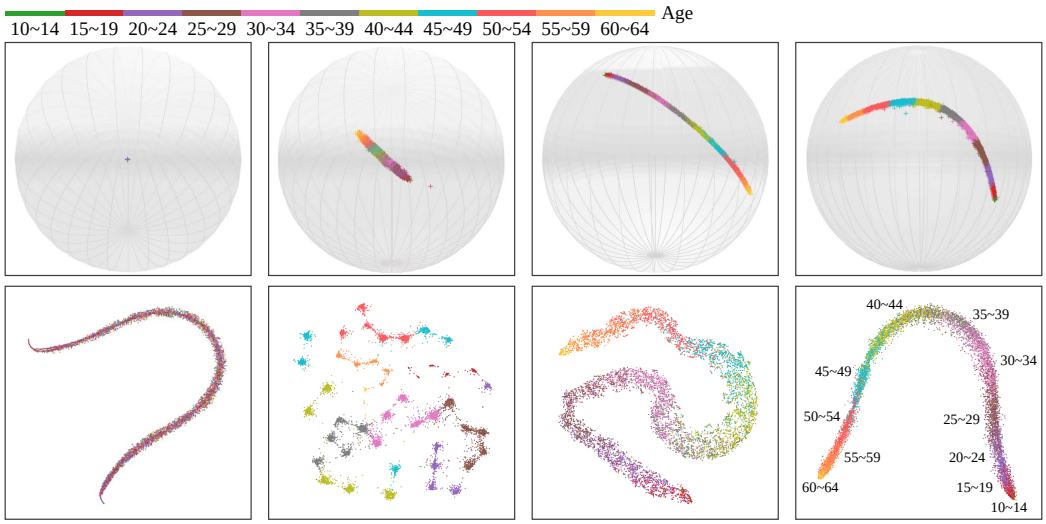
The aesthetics dataset (Schifanella et al., 2015) provides 15,687 image URLs on Flickr, where 13,929 images are available but the others are lost. Each image is annotated with a 5-scale aesthetic score. We adopt the 5-fold cross-validation. For training, we set the learning rate of all parameters, including reference points, to 10^{-6}. The other settings are the same as in Section 4.1. Table 17 is an extended version of Table 5.

Table 17: Accuracy (%) and MAE comparison on the aesthetics dataset.

Algorithm	Nature		Animal		Urban		People		Overall	
	Accuracy (%)	MAE	Accuracy (%)	MAE	Accuracy (%)	MAE	Accuracy (%)	MAE	Accuracy (%)	MAE
RED-SVM (Lin & Li, 2012)	70.7	0.31	61.1	0.41	65.4	0.37	61.2	0.41	64.6	0.38
CNNm (Liu et al., 2018)	71.0	0.31	68.0	0.34	68.2	0.36	71.6	0.32	69.5	0.33
OR-CNN (Niu et al., 2016)	69.8	0.31	69.1	0.33	66.5	0.35	70.4	0.31	69.0	0.33
CNNPOR (Liu et al., 2018)	71.9	0.29	69.3	0.32	69.1	0.33	69.9	0.32	70.1	0.32
SORD (Diaz & Marathe, 2019)	73.6	**0.27**	70.3	0.31	73.3	0.28	70.6	0.31	72.0	0.29
Li *et al.* (Li et al., 2021)	73.6	**0.27**	71.1	0.30	72.8	0.28	**72.2**	**0.29**	72.4	0.29
Proposed GOL	**73.8**	**0.27**	**72.4**	**0.28**	**74.2**	**0.26**	69.6	0.31	**72.7**	**0.28**

D.6 Analysis

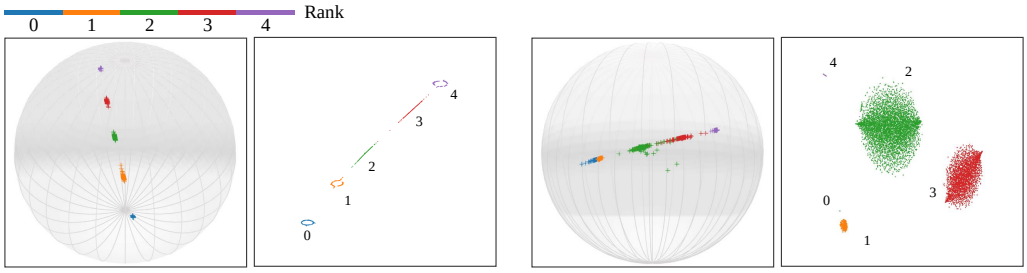
Ablation study: Figure 5 shows the embedding spaces for the ablated methods in Table 6. The top row visualizes the reduced 3D embedding spaces, as described in Appendix C.2, while the bottom row does the original 512D embedding spaces via t-SNE (Maaten & Hinton, 2008). In (a), only L_{center} is used, so the 3D embedding space almost collapses to a single point. It is because a trivial solution to minimize L_{center} can be obtained by merging all instances and all reference points into a single point. In (b), instances are directionally aligned, but adjacent rank sets overlap one another. In (c) and (d), the 3D embedding spaces seem similar. However, GOL yields more clearly ordered instances in the t-SNE visualization of the original embedding space than '$L_{\text{center}} + L_{\text{metric}}$' does.



(a) L_{center} (b) $L_{\text{center}} + L_{\text{order}}$ (c) $L_{\text{center}} + L_{\text{metric}}$ (d) GOL

Figure 5: Visualizing the embedding spaces of the ablated methods in Table 6 on the CACD dataset: (top) reduced 3D spaces and (bottom) t-SNE visualizations of original 512D spaces. Note that the top and bottom rows use different color schemes because they use different software tools. The color scheme for the bottom row is depicted in the bottom right subfigure.

More visualizations: Figure 6 visualizes the embedding spaces of the proposed algorithm on the HCI and aesthetics datasets. For each dataset, the right subfigure is the t-SNE visualization (Maaten & Hinton, 2008).



(a) HCI (b) Aesthetics

Figure 6: Visualization of the embedding spaces for the HCI and aesthetics dataset.

Young, median, and old instances: Figure 7 shows sampled instances from each rank set $\mathcal{X}_i$ of the Adience dataset. Specifically, Figures 7 (a) and (c) list the instances in each $\mathcal{X}_i$ that are the farthest from the reference point r_i in the backward and forward directions, respectively. Also, Figures 7 (b) shows the closest instance to r_i among all instances in $\mathcal{X}_i$. In Adience, each rank set $\mathcal{X}_i$ consists of instances in an age range, marked at the top of the figure. Thus, from each rank set, young, median, and old instances tend to be selected in (a), (b), and (c), respectively, which means that the instances are well sorted in the embedding space although their exact ages are not annotated.

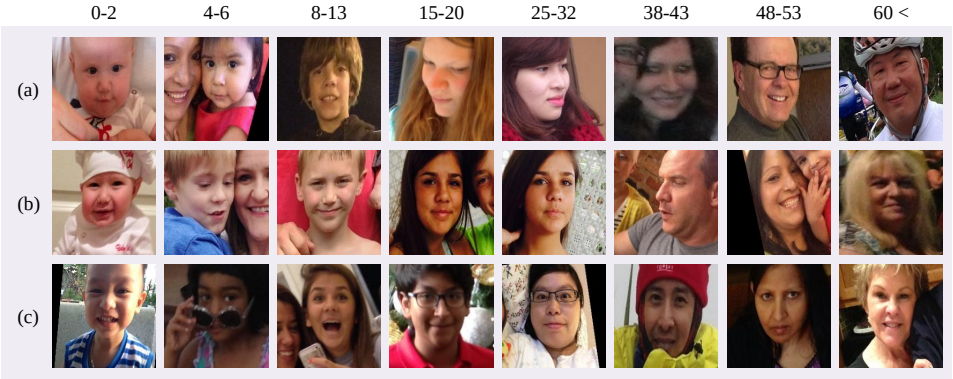


Figure 7: Sampling (a) young, (b) median, and (c) old instances from each rank set of Adience.

Alternatives to $L_{x \prec y}$: Table 18 compares the rank estimation results of alternative loss terms for $L_{x \prec y}$ in (10). Method I directly increases $d_{\mathrm{e}}(h_x, h_y)$ to make it larger than γ. Method II uses only two reference points $r_{\theta(x)}$ and $r_{\theta(y)}$. In other words, unlike method III (*i.e.* GOL), it does not employ reference points r_i, $0 \leq i < \theta(x)$, and $\theta(y) < r_j \leq M-1$. For each method, $L_{x \succ y}$ is also modified accordingly.

Method I degrades the performances significantly, as h_x and h_y move in arbitrary directions. Also, GOL performs better than II because attraction and repulsion with many reference points facilitate the positioning of instances in the embedding space, as well as the reference points themselves.

Table 18: MAE comparison of alternative choices for $L_{x \prec y}$ in (10) on MORPH and CACD.

Method	Alternative to $L_{x \prec y}$	MORPH (setting A)	CACD
I	$\max(\gamma - d_{\mathrm{e}}(h_x, h_y), 0)$	2.35	5.77
II	$\max(d_{\mathrm{e}}(r_{\theta(x)}, h_x) - d_{\mathrm{e}}(r_{\theta(x)}, h_y) + \gamma, 0) + \max(d_{\mathrm{e}}(r_{\theta(y)}, h_y) - d_{\mathrm{e}}(r_{\theta(y)}, h_x) + \gamma, 0)$	2.34	5.75
III	(10)	2.17	5.58

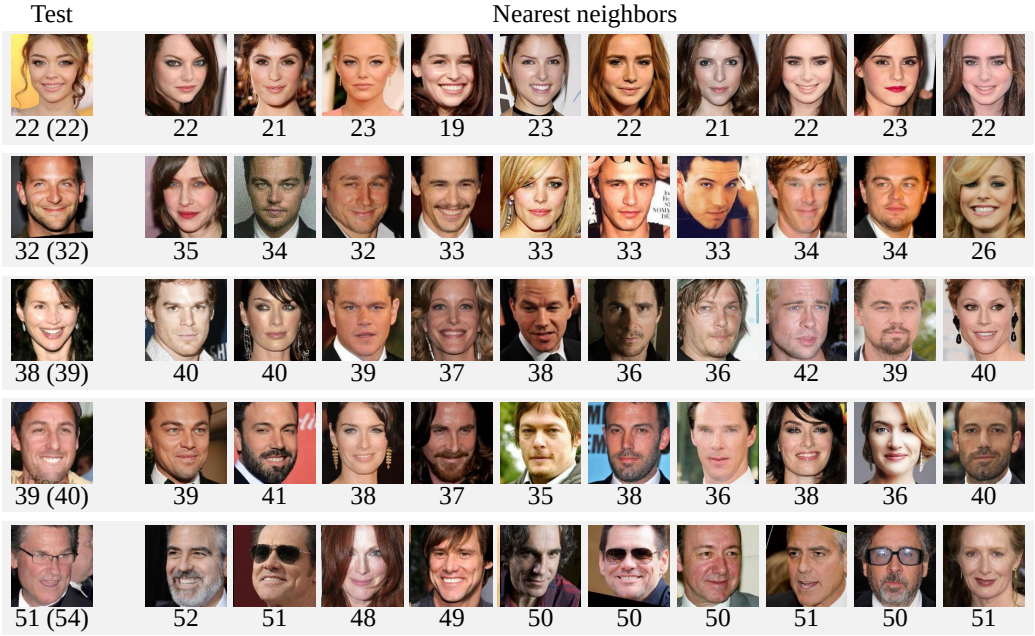
Testing time: Table 19 lists the testing time of GOL according to the number of samples for the k-NN search on the MORPH II and CACD datasets. Note that the distances to all samples can be computed efficiently in a parallel manner. Therefore, the proposed algorithm performs fast even with 44,000 samples.

Table 19: Testing time according to the number of samples for the k-NN search.

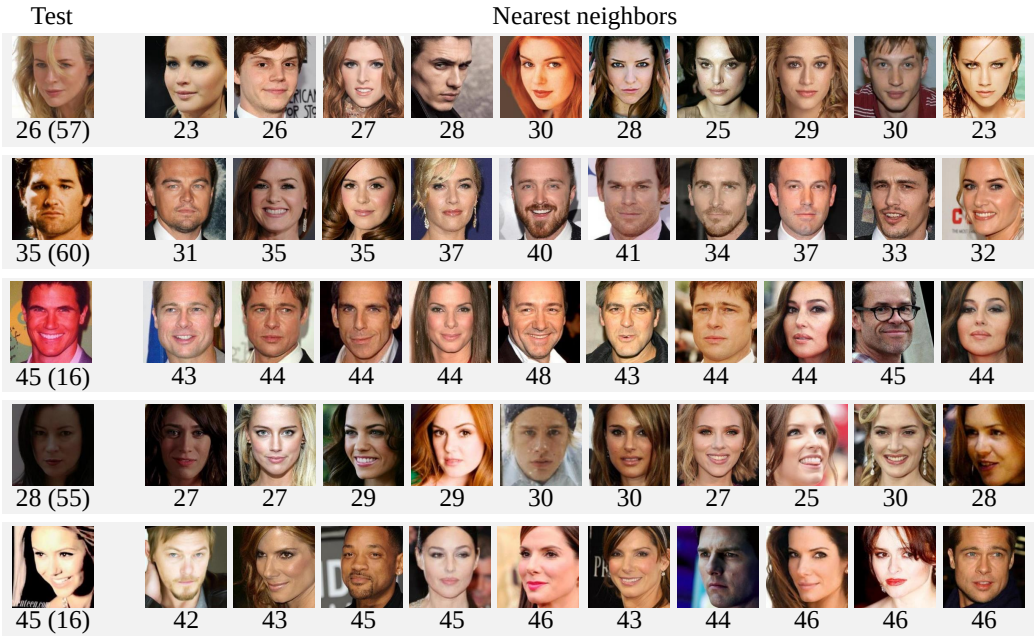
	MORPH (setting A)	MORPH (setting B)	MORPH (setting C)	CACD (validation split)
# samples	4,394	7,000	44,000	7,600
Time (ms)	0.05	0.06	0.08	0.06

Limitations: Figure 8 shows some success and failure cases of GOL on the CACD dataset. In Figure 8 (a), the ages of test instances are predicted accurately with absolute errors less than 4. However, in Figure 8 (b), GOL yields poor estimates on some hard cases, in which various factors, such as poor illumination, overexposure, and low-quality photographs, hinder accurate estimation.

Also, since GOL predicts a rank based on the k-NN search, it suffers when there are insufficient training instances. For example, GOL yields relatively poor results on MORPH II setting B, which consists of 7,000 training samples and 14,000 test samples. Similarly, GOL tends to yield less accurate estimates on minority classes, such as toddlers and elders, since most age estimation datasets contain fewer instances in such classes. These limitations of GOL might be alleviated by generating pseudo-references in the embedding space, which we leave as future work in this paper.



(a)



(b)

Figure 8: (a) Success and (b) failure cases of the proposed algorithm in facial age estimation. For each test image x, the estimate $\hat{\theta}(x)$ is reported with the ground-truth ($\theta(x)$) within the parentheses. Also, the ten nearest neighbors of x are shown with their ages.