# Deep Differentiable Random Forests for Age Estimation

Wei Shen, Yilu Guo, Yan Wang, Kai Zhao, Bo Wang, and Alan Yuille, *Fellow, IEEE*

**Abstract**—Age estimation from facial images is typically cast as a label distribution learning or regression problem, since aging is a gradual progress. Its main challenge is the facial feature space w.r.t. ages is inhomogeneous, due to the large variation in facial appearance across different persons of the same age and the non-stationary property of aging. In this paper, we propose two Deep Differentiable Random Forests methods, Deep Label Distribution Learning Forest (DLDLF) and Deep Regression Forest (DRF), for age estimation. Both of them connect split nodes to the top layer of convolutional neural networks (CNNs) and deal with inhomogeneous data by jointly learning input-dependent data partitions at the split nodes and age distributions at the leaf nodes. This joint learning follows an alternating strategy: (1) Fixing the leaf nodes and optimizing the split nodes and the CNN parameters by Back-propagation; (2) Fixing the split nodes and optimizing the leaf nodes by Variational Bounding. Two Deterministic Annealing processes are introduced into the learning of the split and leaf nodes, respectively, to avoid poor local optima and obtain better estimates of tree parameters free of initial values. Experimental results show that DLDLF and DRF achieve state-of-the-art performance on three age estimation datasets.

**Index Terms**—Age estimation, random forest, regression, label distribution learning, deterministic annealing

## 1 Introduction

THERE has been a growing interest in age estimation from facial images, driven by the increasing demands for a variety of potential applications in forensic research [1], security control [2], human-computer interaction (HCI) [2] and social media [3]. In this paper, we focus on estimating the precise chronological age (i.e., not age group estimation [4]). Although considerable progress has been made recently [5], [6], [7], estimating ages accurately and reliably from facial images is still a challenging problem.

To address age estimation, the characteristics of this task should be considered. First, aging is a slow and gradual progress, thus there is a strong correlation between close ages of the same individual, e.g., a person's facial images taken at close ages are similar. Due to this fact, age estimation is usually formulated as a label distribution learning (LDL) [8], [9], [10] or regression [11], [12], [13] problem rather than a classification problem, because in a classification problem, class labels are uncorrelated. Unlike classification, LDL assigns a distribution over the set of labels to an instance, which can be obtained by fitting a Gaussian or Triangle distribution whose peak is the label of this instance and represents the relative importance of each label involved in the description of an instance; by contrast, regression considers labels as continuous numerical values. Therefore, LDL and regression can explicitly and implicitly model cross age correlations of the same individual, respectively.

Second, learning the mapping between facial image features and ages is challenging. The main difficulty is the facial feature space w.r.t. ages is inhomogeneous, due to two factors: 1) there is a large variation in facial appearance across different persons of the same age (Fig. 1a); 2) the human face matures in different ways at different ages, e.g., bone growth in childhood and skin wrinkles in adulthood [14] (Fig. 1b). This inhomogeneity suggests applying divide-and-conquer models, such as Random Forests [15], [16], [17], to partition the data space and learn multiple local age estimators [18]. However, traditional Random Forests make hard data partitions based on heuristics, such as using a greedy algorithm where locally-optimal hard decisions are made at each split node [15], thus have limitations in representation learning, e.g., they can not learn deep facial features to perform data partition in an end-to-end manner.

To address this issue, we propose two Deep Differentiable Random Forests for age estimation, where one is an LDL model, named by Deep Label Distribution Learning Forest (DLDLF), the other is a regression model, named by Deep Regression Forest (DRF). Our Deep Differentiable Random Forests are inspired by [19], which introduced differentiable decision classification trees and integrated them with CNNs by connecting the split nodes in trees to a fully connected layer of a CNN. We extend the differentiable trees to deal

- *W. Shen, Y. Wang, and A. Yuille are with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218-2608. E-mail: {shenwei1231, wyanny.9, alan.l.yuille}@gmail.com.*
- *Y. Guo is with the School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China. E-mail: gyl.luan0@gmail.com.*
- *K. Zhao is with the College of Computer and Control Engineering, Nankai University, Tianjin 300071, China. E-mail: zhaok1206@gmail.com.*
- *B. Wang is with Vector Institute and Peter Munk Cardiac Center of University Health Network, Toronto, ON, Canada. E-mail: bowang@vectorinstitute.ai.*

*Manuscript received 24 Aug. 2018; revised 22 July 2019; accepted 20 Aug. 2019. Date of publication 0 . 0000; date of current version 0 . 0000.*
*(Corresponding author: Yan Wang.)*
*Recommended for acceptance by G. Wang.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TPAMI.2019.2937294*

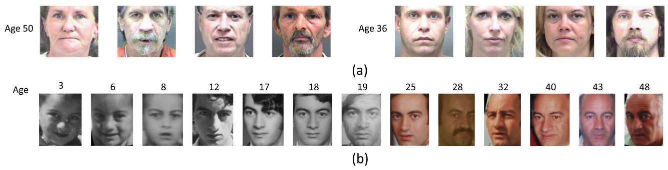



Fig. 1. (a) The large variation in facial appearance across different persons of the same age. (b) Facial images of a person from childhood to adulthood. Note that, Facial aging effects appear as changes in the shape of the face during childhood and changes in skin texture during adulthood, respectively.

with LDL and regression problems, which is non-trivial (see the discussion in Section 2). Differentiable trees perform soft data partition at split nodes, so that an input-dependent partition function can be learned to handle inhomogeneous data. In addition, the deep facial features at split nodes (input feature space) and the age distributions at leaf nodes (local estimators) can be learned jointly, which ensures that the local input-output correlation is homogeneous at the leaf node.

To jointly learn the deep facial features at split nodes and the age distributions at leaf nodes in our Deep Differentiable Random Forests, we apply an alternating optimization strategy: first we fix the leaf nodes parameters and optimize split node parameters as well as the CNN parameters (feature learning) by Back-propagation. Then, we fix split node parameters and optimize the age distributions at leaf nodes by Variational Bounding [20], [21]. These two learning steps are alternatively performed to jointly optimize feature learning and estimator modeling for age estimation. Additionally, these two learning steps are non-convex optimization problems (except for optimizing the age distributions at leaf nodes in DLDLFs), thus both Gradient Descent and Variational Bounding require “good” parameter initializations to avoid converging to poor local minimal. To address this problem, we introduce two Deterministic Annealing (DA) processes [22], [23], [24] into these two learning steps, respectively, which can avoid many poor local optima during optimization and obtain better estimates of tree parameters free of initializations. Finally, to learn the ensemble of multiple trees (forest), we explicitly define the forest loss as the average of the losses of all the individual trees and allow the split nodes from different trees to be connected to the same output unit of the feature learning function. In this way, the split node parameters of all the individual trees can be learned jointly. Fig. 2 illustrates a sketch chart of our DLDLF and DRF, where each forest consists of two trees is shown.

We evaluate our algorithms on three standard datasets for real age estimation methods: MORPH [25], FG-NET [26] and the Cross-Age Celebrity Dataset (CACD) [27]. Experimental results demonstrate that our algorithms outperform several state-of-the-art methods on these three datasets.

The contributions of this paper are five folds:

1) We propose Deep Label Distribution Learning Forest and Deep Regression Forest, two end-to-end models, to deal with inhomogeneous data by jointly learning input-dependent data partition at split nodes and age distribution at leaf nodes.
2) Based on Variational Bounding, the convergences of our update rules for leaf nodes in DLDLFs and DRFs are mathematically guaranteed.
3) We introduce Deterministic Annealing processes into the learning of DLDLFs and DRFs, which can avoid many poor local optima during optimization and obtain better estimates of tree parameters free of initial parameter values.
4) We propose a strategy to learn the ensemble of multiple trees, which is different from [19], but we show it is effective.
5) We apply DLDLFs and DRFs to three standard age estimation benchmarks, and achieve state-of-the-art results.

This paper summarizes two of our preliminary works [28], [29] into a unified optimization framework, i.e., alternatively learning split nodes by Back-propagation and learning leaf nodes by Variational Bounding and has following extensions: First, we introduce two methodological improvements, i.e., the two Deterministic Annealing processes introduced into the learning of split and leaf nodes, respectively, to avoid poor local optima and obtain better estimates of tree parameters free of initial parameter values. Second, we provide more experimental results and discussions, such as ablation experiments to study the influence of different designs and variants of our methods and updated state-of-the-art results on the three age estimation datasets.

## 2 Related Work

*Age Estimation.* One way to tackle precise facial age estimation is to search for a kernel-based global non-linear mapping, like kernel support vector regression [30] or kernel partial least squares regression [11]. The basic idea is to learn a low-dimensional embedding of the aging manifold [31]. However, global non-linear mapping algorithms may be biased [13], due to the inhomogeneous properties of the input data. Another way is to adopt divide-and-conquer approaches, which partition the data space and learn multiple local regressors. But hierarchical regression [18] or tree based regression [32] approaches made hard partitions according to ages, which is problematic because the subsets of facial images may not be homogeneous for learning local regressors. Huang et al. [13] proposed Soft-margin Mixture of Regressions (SMMR) to address this issue, which found homogeneous partitions in the joint input-output space, and learned a local regressor for each partition. But their regression model cannot be integrated with any deep networks as an end-to-end model.

Several researchers formulated age estimation as an ordinal regression problem [5], [12], [33], because the relative order among the age labels is also important information. They trained a series of binary classifiers to partition the samples according to ages, and estimated ages by summing over the classifier outputs. Thus, ordinal regression is limited by its lack of scalability [13]. Some other researchers formulated age estimation as a label distribution learning problem [34], which paid attention to modeling the cross-age correlations, based on the observation that faces at close ages look similar. LDL based age estimation methods [8], [9], [35] achieved promising results, but these LDL methods assume that a label distribution should be represented by a maximum entropy model [36], where the exponential part of this model restricts the generality of the distribution

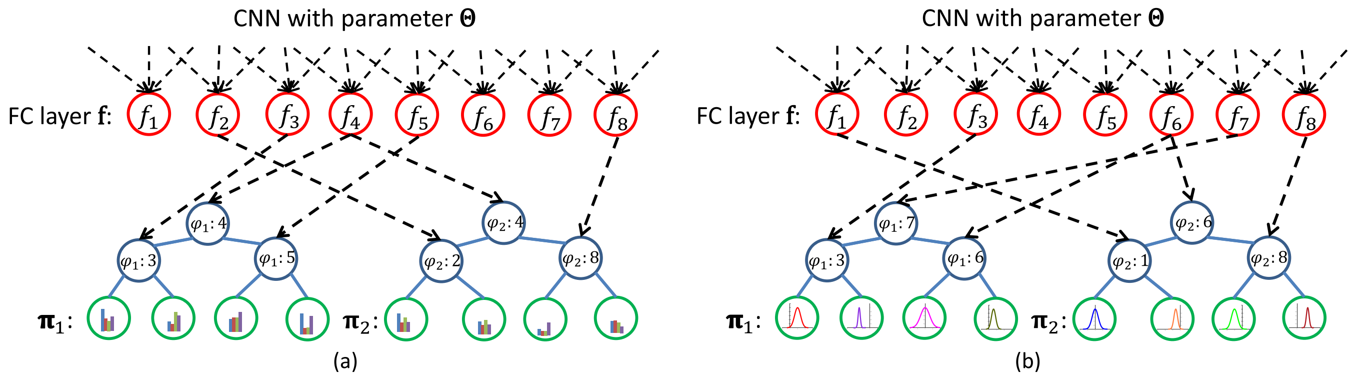



Fig. 2. Illustration of (a) a deep label distribution learning forest (DLDLF) and (b) a deep regression forest (DRF). Each forest consists of two trees. The top red circles denote the output units of the function $\mathbf{f}$ parameterized by $\mathbf{\Theta}$. Here, they are the units of a fully-connected (FC) layer in a CNN. The blue and green circles are split nodes and leaf nodes, respectively. in each forest, two index functions $\varphi_1$ and $\varphi_2$ are randomly assigned to the two trees respectively before training and then fixed. The black dash arrows indicate the correspondence between the split nodes of the two trees and the output units of the FC layer. Note that, one output unit may correspond to the split nodes belonging to different trees. Each tree has independent leaf node distribution $\pi$ (denoted by distribution histograms and curves in the leaf nodes of the DLDLF and the DRF, respectively). The output of the forest is a mixture of the tree predictions. $\mathbf{f}(\cdot;\mathbf{\Theta})$ and $\pi$ are learned jointly in an end-to-end manner.

form. On the contrary, our method, DLDLF, expresses a label distribution by a linear combination of the label distributions of training data, and thus have no restrictions on the distributions (e.g., no requirement of the maximum entropy model).

With the rapid development of deep networks, more and more end-to-end CNN based age estimation methods [3], [6], [7], [12], [37] have been proposed to address this nonlinear regression problem. But how to deal with inhomogeneous data is still an open issue.

*Random Forests.* Random Forests or randomized decision trees [15], [16], [17], [38], are a popular ensemble predictive model suitable for many machine learning tasks, such as supervised learning [16], semi-supervised learning [39] and multiple instance learning [40]. Each decision tree consists of several split nodes and leaf nodes. Tree growing is usually based on greedy algorithms which make locally-optimal hard data partition decisions at each split node. Thus, this makes it intractable to integrate decision trees with deep networks in an end-to-end learning manner. Some efforts have been made to combine these two worlds [19], [41], [42]. The newly proposed Deep Neural Decision Forests (DNDFs) [19] overcame this problem by introducing a soft differentiable decision function at the split nodes and a global loss function defined on a tree, which ensured that the split node parameters can be learned by back-propagation and leaf node predictions can be updated by a discrete iterative function.

Our methods are inspired by Deep Neural Decision Forests [19], but differ in their objectives (label distribution learning/regression *vs* classification). Extending differentiable decision trees to deal with label distribution learning/regression is non-trivial, since there are some technical difficulties in learning leaf node predictions. Although a step-size free update function was given in DNDFs to update leaf node predictions, it was only proved to converge for a classification loss. Consequently, it is unclear how to obtain such an update function for other objectives, especially for regression, since the distribution of the output space for regression is continuous, but the distribution of the output space for classification is discrete. We show that the update functions for both the LDL loss and the regression loss can be derived from Variational Bounding and the one given in [19] is also a special case of Variational Bounding. In addition, we introduce two DA processes into the optimization of our Deep Differentiable Random Forests, which lead to better estimates of tree parameters free of initial parameter values. Last but not least, the strategies used in our deep Random Forests and DNDFs to learn the ensemble of multiple trees (forests) are different: We explicitly define a loss function for a forest, which allows the split nodes from different trees to be connected to the same output unit of the feature learning function (See Fig. 2) and enables that all trees in a DLDLF or a DRF can be learned jointly; while only the loss function for a single tree is defined in DNDFs, which only allows trees in a DNDF to be learned alternatively. As shown in our experiments (Section 6.4.3), our ensemble strategy can get better results by using more trees, but by using the ensemble strategy proposed in DNDFs, the results of forests are even worse than those for a single tree.

One recent work proposed Neural Regression Forest (NRF) [43] for depth estimation, which is similar to our DRF, but there are two main differences between an NRF and a DRF. The first difference is all the split nodes in a DRF are connected to the top layer of a single CNN, but every split node in an NRF is connected to a distinct CNN. Therefore, an NRF can be only connected to very shallow CNNs (as they did in their experiments), otherwise, the computational cost is extremely high. But, the representation learning ability of these shallow CNNs is limited. The second difference is the convergence of our update rule for leaf nodes is mathematically guaranteed by Variational Bounding, but the convergence of the update rule for leaf nodes used in the NRF was not guaranteed.

## 3 Differential Decision Trees

Since both DLDLFs and DRFs are based on differential decision trees [19], we introduce this tree model first in this section.

Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output spaces, respectively. A differential decision tree $\mathcal{T}$ consists of a set of split

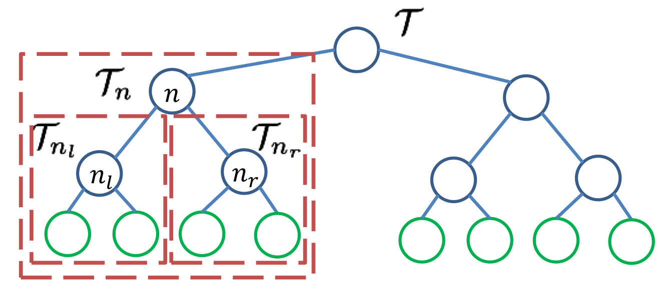



Fig. 3. The subtree rooted at node $n$: $\mathcal{T}_n$ and its left and right subtrees: $\mathcal{T}_{n_l}$ and $\mathcal{T}_{n_r}$.

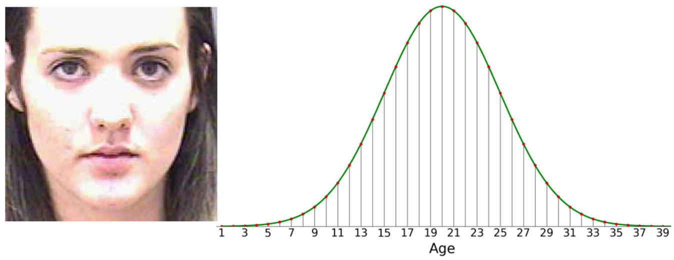



Fig. 4. Generated label distribution for a facial image at the chronological age of 20 ($\alpha = 20$).

nodes $\mathcal{N}$ and a set of leaf nodes $\mathcal{L}$. Each split node $n \in \mathcal{N}$ defines a split function $s_n(\cdot;\boldsymbol{\Theta}) : \mathcal{X} \rightarrow [0,1]$ parameterized by $\boldsymbol{\Theta}$ to determine whether a sample is sent to the left or right subtree. Each leaf node $\ell \in \mathcal{L}$ holds a distribution $\boldsymbol{\pi}_\ell$ over $\mathcal{Y}$. Following [19], we use a soft split function $s_n(\mathbf{x};\boldsymbol{\Theta}) = \sigma(f_{\varphi(n)}(\mathbf{x};\boldsymbol{\Theta}))$, where $\sigma(\cdot)$ is a sigmoid function, $\mathbf{x} \in \mathcal{X}$ and $\mathbf{f} : \mathbf{x} \rightarrow \mathbb{R}^M$ is a real-valued feature learning function depending on the sample $\mathbf{x}$ and the parameter $\boldsymbol{\Theta}$. $\mathbf{f}$ can take any forms. In our DLDLFs and DRFs, it is a CNN and $\boldsymbol{\Theta}$ is the network parameter. $\varphi(\cdot)$ is an index function to specify the correspondence between the split nodes and output units of $\mathbf{f}$, which is randomly assigned before tree learning and then fixed. An example to demonstrate $\varphi(\cdot)$ is shown in Fig. 2 (There are two trees with index functions in each forest, $\varphi_1$ and $\varphi_2$ for each). Then, the probability of the sample $\mathbf{x}$ falling into leaf node $\ell$ is given by

$$P(\ell|\mathbf{x};\boldsymbol{\Theta}) = \prod_{n\in\mathcal{N}} s_n(\mathbf{x};\boldsymbol{\Theta})^{\mathbf{1}(\ell\in\mathcal{L}_{n_l})}(1 - s_n(\mathbf{x};\boldsymbol{\Theta}))^{\mathbf{1}(\ell\in\mathcal{L}_{n_r})}, \tag{1}$$

where $\mathbf{1}(\cdot)$ is an indicator function and $\mathcal{L}_{n_l}$ and $\mathcal{L}_{n_r}$ denote the sets of leaf nodes held by the subtrees $\mathcal{T}_{n_l}$, $\mathcal{T}_{n_r}$ rooted at the left and right children $n_l, n_r$ of node $n$ (shown in Fig. 3), respectively.

Note that, there are two parameters introduced in this tree model: 1) the split node parameter $\boldsymbol{\Theta}$ and 2) the distributions $\boldsymbol{\pi}$ held by the leaf nodes. Tree learning requires the estimation of these two parameters. Let $\mathcal{S}$ be a training set and $R(\boldsymbol{\pi},\boldsymbol{\Theta};\mathcal{S})$ be an objective function for an arbitrary learning task, e.g., classification, label distribution learning or regression, then the best parameters $(\boldsymbol{\Theta}^*,\boldsymbol{\pi}^*)$ are determined by solving

$$(\boldsymbol{\Theta}^*,\boldsymbol{\pi}^*) = \arg\min_{\boldsymbol{\Theta},\boldsymbol{\pi}} R(\boldsymbol{\pi},\boldsymbol{\Theta};\mathcal{S}). \tag{2}$$

To solve Eq. (2), we consider an alternating optimization strategy: First, we fix $\boldsymbol{\pi}$ and optimize $\boldsymbol{\Theta}$ by Back-propagation; Then, we fix $\boldsymbol{\Theta}$ and optimize $\boldsymbol{\pi}$ by Variational Bounding [20], [21]. These two learning steps are performed alternatively, until convergence or a maximum number of iterations is reached (as described in the experiments). We show that this optimization framework is unified for learning both DLDLFs and DRFs in Sections 4 and 5, respectively, as well as for learning the tree models for classification [19] in the Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2019.2937294.

## 4 Age Estimation by Deep Label Distribution Forests

In this section, we describe Deep Label Distribution Learning Forests (DLDLFs) for age estimation. Since a forest is an ensemble of decision trees, We first introduce how to learn a single decision tree by label distribution learning, then describe the learning of a forest.

### 4.1 Problem Formulation

We formulate age estimation as an LDL problem: Let $\mathcal{X} = \mathbb{R}^m$ denote the input facial image space and $\mathcal{Y} = \{y_1, y_2, \ldots, y_C\}$ denote the complete and order set of age labels, where $C$ is the number of possible age values. For a facial image $\mathbf{x} \in \mathcal{X}$, its chronological age value is $y \in \mathcal{Y}$. To generate a proper label distribution $\mathbf{d} = (d_{\mathbf{x}}^{y_1}, d_{\mathbf{x}}^{y_2}, \ldots, d_{\mathbf{x}}^{y_C})^\top \in \mathbb{R}^C$, where $d_{\mathbf{x}}^{y_c} \in [0,1]$ and $\sum_{c=1}^{C} d_{\mathbf{x}}^{y_c} = 1$, for this facial image $\mathbf{x}$, following [9], [37], we use a Gaussian distribution whose mean is the chronological age $y$:

$$d_{\mathbf{x}}^{y_c} = \frac{p_{\mathcal{N}}(y_c; y, \alpha)}{\sum_{k=1}^{C} p_{\mathcal{N}}(y_k; y, \alpha)}, \tag{3}$$

where $p_{\mathcal{N}}(y_c; y, \alpha) = \frac{1}{\sqrt{2\pi}\alpha}\exp(-\frac{(y_c-y)^2}{2\alpha^2})$ and $\alpha$ is a pre-defined standard deviation. Fig. 4 illustrates an example of such a label distribution generated for a facial image at the chronological age of 20 ($\alpha = 10$). Observed that, this label distribution explicitly model the cross age correlations, since both the chronological age 20 and the neighboring ages 19 and 21 can be used to describe the appearance of this 20-year-old face, due to the appearance similarity of the neighboring ages.

We have formulated age estimation as an LDL problem, then our goal is to learn a mapping function $\mathbf{g} : \mathbf{x} \rightarrow \mathbf{d}$ between an facial image $\mathbf{x}$ and its corresponding label distribution $\mathbf{d}$ by a decision tree based model $\mathcal{T}$ described in Section 3. Since our target, label distribution, is a discrete distribution, accordingly each leaf node $\ell \in \mathcal{L}$ in the LDL tree $\mathcal{T}$ holds a probability mass distribution $\boldsymbol{\pi}_\ell = (\pi_{\ell_1}, \pi_{\ell_2}, \ldots, \pi_{\ell_C})^\top$ over $\mathcal{Y}$, i.e., $\pi_{\ell_c} \in [0,1]$ and $\sum_{c=1}^{C} \pi_{\ell_c} = 1$. Then the output of the tree $\mathcal{T}$ w.r.t. $\mathbf{x}$, i.e., the mapping function $g$, is given by

$$\mathbf{g}(\mathbf{x};\boldsymbol{\Theta},\mathcal{T}) = \sum_{\ell\in\mathcal{L}} P(\ell|\mathbf{x};\boldsymbol{\Theta})\boldsymbol{\pi}_\ell. \tag{4}$$

### 4.2 Tree Optimization

Given a training set $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{d}_i)\}_{i=1}^{N}$, our goal is to learn a LDL tree $\mathcal{T}$ described in in Section 4.1 which can output a distribution $\mathbf{g}(\mathbf{x}_i;\boldsymbol{\Theta},\mathcal{T})$ similar to $\mathbf{d}_i$ for each sample $\mathbf{x}_i$. To this end, a straightforward way is to minimize the Kullback-Leibler (K-L) divergence between each $\mathbf{g}(\mathbf{x}_i;\mathbf{\Theta},\mathcal{T})$ and $\mathbf{d}_i$, or equivalently to minimize the following cross-entropy loss:

$$\begin{aligned}R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S}) &= -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} d_{\mathbf{x}_i}^{y_c}\log\left(g_c(\mathbf{x}_i;\mathbf{\Theta},\mathcal{T})\right)\\ &= -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} d_{\mathbf{x}_i}^{y_c}\log\Big(\sum_{\ell\in\mathcal{L}} P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_{\ell_c}\Big),\end{aligned} \tag{5}$$

where $\boldsymbol{\pi}$ denotes the distributions held by all the leaf nodes $\mathcal{L}$ and $g_c(\mathbf{x}_i;\mathbf{\Theta},\mathcal{T})$ is the $c$th output unit of $\mathbf{g}(\mathbf{x}_i;\mathbf{\Theta},\mathcal{T})$.

Learning the tree $\mathcal{T}$ requires the estimation of two parameters: 1) the split node parameter $\mathbf{\Theta}$ and 2) the distributions $\boldsymbol{\pi}$ held by the leaf nodes. Next we introduce the optimization process in detail following the framework described in Section 3.

#### *4.2.1 Learning Split Nodes w/ Deterministic Annealing by Gradient Descent*

In this section, we describe how to learn the parameter $\mathbf{\Theta}$ for split nodes, when the distributions held by the leaf nodes $\boldsymbol{\pi}$ are fixed. We found that, empirically (also shown in [19]) minimization of $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$ w.r.t. $\mathbf{\Theta}$ would gradually produce almost hard data partitions, i.e., $P(\ell|\mathbf{x}_i;\mathbf{\Theta})$ approaches 0 or 1. But it is better to enforce $P(\ell|\mathbf{x}_i;\mathbf{\Theta})$ to be uniform for all $\ell\in\mathcal{L}$, i.e., maintain more uncertainty, at the beginning of minimization. Inspired by [22], we introduce a Deterministic Annealing process into this optimization, which minimizes $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$ subject to a specified level of uncertainty. The level of uncertainty is measured by the Shannon entropy

$$H(\mathbf{\Theta};\mathcal{S}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{\ell\in\mathcal{L}} P(\ell|\mathbf{x}_i;\mathbf{\Theta})\log P(\ell|\mathbf{x}_i;\mathbf{\Theta}). \tag{6}$$

Then we reformulate the original loss function Eq. (5) to be

$$E(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S},T) = R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S}) - TH(\mathbf{\Theta};\mathcal{S}), \tag{7}$$

where $T$ is the temperature parameter. During the DA process, $E(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S},T)$ is then gradually deformed to its original form, i.e., $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$, by decreasing the temperature $T\rightarrow 0$. From the DA viewpoint, minimization the original loss function corresponds to a "zero temperature" system, where each input sample must make a hard decision about which leaf node it would fall into. This is hard at the beginning of the minimization. On the other hand, starting at high $T$ smoothes the loss function $E(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S},T)$ making it easier to get a good minimum, which can be traced by slowly decreasing $T$ ("cooling" the system). By introducing this DA process, we start with each input sample equally influencing all leaf nodes and gradually localize the influence. This gives us some intuition as to how the system searches for a better optimum [22]. We use a simple cooling schedule to decrease $T$ during optimization: $T\leftarrow\eta T$, where $\eta$ is a constant less than 1.

We compute the gradient of the loss $E(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S},T)$ w.r.t. $\mathbf{\Theta}$ by the chain rule

$$\frac{\partial E(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S},T)}{\partial\mathbf{\Theta}} = \sum_{i=1}^{N}\sum_{n\in\mathcal{N}}\frac{\partial E(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S},T)}{\partial f_{\varphi(n)}(\mathbf{x}_i;\mathbf{\Theta})}\frac{\partial f_{\varphi(n)}(\mathbf{x}_i;\mathbf{\Theta})}{\partial\mathbf{\Theta}}, \tag{8}$$

where only the first term depends on the tree. The second term depends on the specific type of the function $f_{\varphi(n)}$. The first term is given by

$$\begin{aligned}\frac{\partial E(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S},T)}{\partial f_{\varphi(n)}(\mathbf{x}_i;\mathbf{\Theta})} &= \frac{\partial R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})}{\partial f_{\varphi(n)}(\mathbf{x}_i;\mathbf{\Theta})} - T\frac{\partial H(\mathbf{\Theta};\mathcal{S})}{\partial f_{\varphi(n)}(\mathbf{x}_i;\mathbf{\Theta})}\\ &= \frac{1}{N}\Big(s_n(\mathbf{x}_i;\mathbf{\Theta})\big(D_i^{n_r} - TS_i^{n_r}\big)\\ &\quad - \big(1-s_n(\mathbf{x}_i;\mathbf{\Theta})\big)\big(D_i^{n_l} - TS_i^{n_l}\big)\Big),\end{aligned} \tag{9}$$

where for a generic node $n\in\mathcal{N}$

$$D_i^n = \sum_{c=1}^{C} d_{\mathbf{x}_i}^{y_c}\frac{g_c(\mathbf{x}_i;\mathbf{\Theta},\mathcal{T}_n)}{g_c(\mathbf{x}_i;\mathbf{\Theta},\mathcal{T})} = \sum_{c=1}^{C} d_{\mathbf{x}_i}^{y_c}\frac{\sum_{\ell\in\mathcal{L}_n} P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_{\ell_c}}{\sum_{\ell\in\mathcal{L}} P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_{\ell_c}}, \tag{10}$$

and

$$S_i^n = \sum_{\ell\in\mathcal{L}_n}\Big(P(\ell|\mathbf{x}_i;\mathbf{\Theta}) + P(\ell|\mathbf{x}_i;\mathbf{\Theta})\log P(\ell|\mathbf{x}_i;\mathbf{\Theta})\Big). \tag{11}$$

Both $D_i^n$ and $S_i^n$ can be efficiently computed for all nodes $n$ in the tree $\mathcal{T}$ by a single pass over the tree. Observing that $D_i^n = D_i^{n_l} + D_i^{n_r}$ and $S_i^n = S_i^{n_l} + S_i^{n_r}$, the computation for $D_i^n$ and $S_i^n$ can be started at the leaf nodes and conducted in a bottom-up manner. Following Eq. (9), the split node parameters $\mathbf{\Theta}$ can be learned by standard Back-propagation.

#### *4.2.2 Learning Leaf Nodes by Variational Bounding*

Note that, since the entropy term introduced in Eq. (7) is constant w.r.t. to $\boldsymbol{\pi}$, thus it does not influence the learning of leaf nodes. By fixing the parameter $\mathbf{\Theta}$, we show how to learn the distributions at the leaf nodes $\boldsymbol{\pi}$, which is a constrained convex optimization problem

$$\min_{\boldsymbol{\pi}} R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S}), \mathbf{s.t.}, \forall\ell, \sum_{c=1}^{C}\pi_{\ell_c} = 1. \tag{12}$$

We address this constrained convex optimization problem by Variational Bounding [20], [21]. In Variational Bounding, the original objective function to be minimized gets replaced by tight upper bounds in an iterative manner. A tight upper bound for the loss function $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$ can be obtained by Jensen's inequality

$$\begin{aligned}R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S}) &= -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} d_{\mathbf{x}_i}^{y_c}\log\Big(\sum_{\ell\in\mathcal{L}} P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_{\ell_c}\Big)\\ &\le -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} d_{\mathbf{x}_i}^{y_c}\sum_{\ell\in\mathcal{L}}\rho(\ell|\bar{\pi}_{\ell_c},\mathbf{x}_i)\log\Big(\frac{P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_{\ell_c}}{\rho(\ell|\bar{\pi}_{\ell_c},\mathbf{x}_i)}\Big),\end{aligned} \tag{13}$$

where $\rho(\ell|\pi_{\ell_c},\mathbf{x}_i) = \frac{P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_{\ell_c}}{g_c(\mathbf{x}_i;\mathbf{\Theta},\mathcal{T})}$ and it has the property that $\rho(\ell|\pi_{\ell_c},\mathbf{x}_i)\in[0,1]$ and $\sum_{\ell\in\mathcal{L}}\rho(\ell|\pi_{\ell_c},\mathbf{x}_i) = 1$. Note that when $\boldsymbol{\pi} = \bar{\boldsymbol{\pi}}$, the equality holds, which indicates this upper bound is tight. We define

$$\phi(\boldsymbol{\pi}, \bar{\boldsymbol{\pi}}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} d_{\mathbf{x}_i}^{y_c} \sum_{\ell\in\mathcal{L}} \rho(\ell|\bar{\pi}_{\ell_c}, \mathbf{x}_i) \log\left(\frac{P(\ell|\mathbf{x}_i;\boldsymbol{\Theta})\pi_{\ell_c}}{\rho(\ell|\bar{\pi}_{\ell_c}, \mathbf{x}_i)}\right). \tag{14}$$

Then $\phi(\boldsymbol{\pi}, \bar{\boldsymbol{\pi}})$ is a tight upper bound for $R(\boldsymbol{\pi}, \boldsymbol{\Theta}; \mathcal{S})$, which has the properties that for any $\boldsymbol{\pi}$ and $\bar{\boldsymbol{\pi}}$, $\phi(\boldsymbol{\pi}, \bar{\boldsymbol{\pi}}) \geq \phi(\boldsymbol{\pi}, \boldsymbol{\pi}) = R(\boldsymbol{\pi}, \boldsymbol{\Theta}; \mathcal{S})$ and $\phi(\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{\pi}}) = R(\bar{\boldsymbol{\pi}}, \boldsymbol{\Theta}; \mathcal{S})$. These two properties hold the conditions for Variational Bounding.

Assume that we are at a point $\boldsymbol{\pi}^{(t)}$ corresponding to the $t$th iteration, then $\phi(\boldsymbol{\pi}, \boldsymbol{\pi}^{(t)})$ is a tight upper bound for $R(\boldsymbol{\pi}, \boldsymbol{\Theta}; \mathcal{S})$. In the next iteration, $\boldsymbol{\pi}^{(t+1)}$ is chosen such that $\phi(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\pi}^{(t)}) \leq R(\boldsymbol{\pi}^{(t)}, \boldsymbol{\Theta}; \mathcal{S})$, which implies $R(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\Theta}; \mathcal{S}) \leq R(\boldsymbol{\pi}^{(t)}, \boldsymbol{\Theta}; \mathcal{S})$. Consequently, we can minimize $\phi(\boldsymbol{\pi}, \bar{\boldsymbol{\pi}})$ instead of $R(\boldsymbol{\pi}, \boldsymbol{\Theta}; \mathcal{S})$ after ensuring that $R(\boldsymbol{\pi}^{(t)}, \boldsymbol{\Theta}; \mathcal{S}) = \phi(\boldsymbol{\pi}^{(t)}, \bar{\boldsymbol{\pi}})$, i.e., $\bar{\boldsymbol{\pi}} = \boldsymbol{\pi}^{(t)}$. So we have

$$\boldsymbol{\pi}^{(t+1)} = \arg\min_{\boldsymbol{\pi}} \phi(\boldsymbol{\pi}, \boldsymbol{\pi}^{(t)}), \mathbf{s.t.}, \forall\ell, \sum_{c=1}^{C}\pi_{\ell_c} = 1, \tag{15}$$

which leads to minimizing the Lagrangian defined by

$$\varphi(\boldsymbol{\pi}, \boldsymbol{\pi}^{(t)}) = \phi(\boldsymbol{\pi}, \boldsymbol{\pi}^{(t)}) + \sum_{\ell\in\mathcal{L}} \lambda_\ell \left(\sum_{c=1}^{C}\pi_{\ell_c} - 1\right), \tag{16}$$

where $\lambda_\ell$ is the Lagrange multiplier. By setting $\frac{\partial\varphi(\boldsymbol{\pi}, \boldsymbol{\pi}^{(t)})}{\partial\pi_{\ell_c}} = 0$, we have

$$\begin{aligned} \lambda_\ell &= \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} d_{\mathbf{x}_i}^{y_c} \rho(\ell|\pi_{\ell_c}^{(t)}, \mathbf{x}_i), \\ \pi_{\ell_c}^{(t+1)} &= \frac{\sum_{i=1}^{N} d_{\mathbf{x}_i}^{y_c} \rho(\ell|\pi_{\ell_c}^{(t)}, \mathbf{x}_i)}{\sum_{c=1}^{C}\sum_{i=1}^{N} d_{\mathbf{x}_i}^{y_c} \rho(\ell|\pi_{\ell_c}^{(t)}, \mathbf{x}_i)}. \end{aligned} \tag{17}$$

Note that, $\pi_{\ell_c}^{(t+1)}$ satisfies that $\pi_{\ell_c}^{(t+1)} \in [0, 1]$ and $\sum_{c=1}^{C}\pi_{\ell_c}^{(t+1)} = 1$. Eq. (17) is the update scheme for distributions held by the leaf nodes. The starting point $\boldsymbol{\pi}_\ell^{(0)}$ can be simply initialized by the uniform distribution: $\pi_{\ell_c}^{(0)} = \frac{1}{C}$.

### 4.3 Learning an LDL Forest

An LDL forest is an ensemble of LDL decision trees $\mathcal{F} = \{\mathcal{T}^1, \ldots, \mathcal{T}^K\}$. In the training stage, all trees in the forest $\mathcal{F}$ use the same parameters $\boldsymbol{\Theta}$ for feature learning function $\mathbf{f}(\cdot; \boldsymbol{\Theta})$ (but correspond to different output units of $\mathbf{f}$ assigned by $\varphi$, see Fig. 2), but each tree has independent leaf node predictions $\boldsymbol{\pi}$. The loss function for a forest is given by averaging the loss functions for all individual trees

$$E_{\mathcal{F}} = \frac{1}{K}\sum_{k=1}^{K} E_{\mathcal{T}^k}, \tag{18}$$

where $E_{\mathcal{T}^k}$ is the loss function for tree $\mathcal{T}^k$ defined by Eq. (7). To learn $\boldsymbol{\Theta}$ by fixing the leaf node predictions $\boldsymbol{\pi}$ of all the trees in the forest $\mathcal{F}$, based on the derivation in Section 4.2 and referring to Fig. 2, we have

$$\frac{\partial E_{\mathcal{F}}}{\partial\boldsymbol{\Theta}} = \frac{1}{K}\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{n\in\mathcal{N}_k} \frac{\partial E_{\mathcal{T}^k}}{\partial f_{\varphi_k(n)}(\mathbf{x}_i; \boldsymbol{\Theta})} \frac{\partial f_{\varphi_k(n)}(\mathbf{x}_i; \boldsymbol{\Theta})}{\partial\boldsymbol{\Theta}}, \tag{19}$$

where $\mathcal{N}_k$ and $\varphi_k(\cdot)$ are the split node set and the index function of $\mathcal{T}^k$, respectively, and the first term of the right side $\frac{\partial E_{\mathcal{T}^k}}{\partial f_{\varphi_k(n)}(\mathbf{x}_i; \boldsymbol{\Theta})}$ is computed by Eq. (9). Note that, the index function $\varphi_k(\cdot)$ for each tree is randomly assigned before tree learning, and thus split nodes correspond to a subset of output units of $\mathbf{f}$. This strategy is similar to the random subspace method [44], which increases the randomness in training to reduce the risk of over-fitting. For the temperature parameter $T$ introduced in each $E_{\mathcal{T}^k}$, we initialize it as a large value $T_0$ ($T_0 > 0$) and decrease it by the simple cooling schedule described in Section 4.2.1: $T \leftarrow \eta T$, where $\eta$ is a constant cooling factor less than 1.

As for $\boldsymbol{\pi}$, since each tree in the forest $\mathcal{F}$ has its own leaf node predictions $\boldsymbol{\pi}$, we can update them independently by Eq. (17). For implementational convenience, we do not conduct this update scheme on the whole dataset $\mathcal{S}$ but on a set of mini-batches $\mathcal{B}$. The training procedure of an LDLF is shown in Algorithm. 1.

**Algorithm 1.** The Training Procedure of a DLDLF

**Require:** $\mathcal{S}$: training set
**Require:** $n_B$: the number of mini-batches to update $\boldsymbol{\pi}$,
**Require:** $T_0$: a starting temperature parameter
**Require:** $\eta$: a constant cooling factor
  Initialize $\boldsymbol{\Theta}$ randomly and $\boldsymbol{\pi}$ uniformly
  Set $\mathcal{B} = \{\emptyset\}$ and $T = T_0$
  **while** Not converge **do**
    **while** $|\mathcal{B}| < n_B$ **do**
      Randomly select a mini-batch $B$ from $\mathcal{S}$
      Update $\boldsymbol{\Theta}$ by Gradient Descent (Eqs. (19), (9)) on $B$
      $\mathcal{B} = \mathcal{B} \bigcup B$
    **end while**
    Update $\boldsymbol{\pi}$ by iterating Eq. (17) on $\mathcal{B}$
    $\mathcal{B} = \{\emptyset\}$
    **if** $T \geq 0$ **then**
      $T \leftarrow \eta T$
    **end if**
  **end while**

In the testing stage, the output of the forest $\mathcal{F}$ is given by averaging the predictions from all the individual trees

$$\mathbf{g}(\mathbf{x}; \boldsymbol{\Theta}, \mathcal{F}) = \frac{1}{K}\sum_{k=1}^{K}\mathbf{g}(\mathbf{x}; \boldsymbol{\Theta}, \mathcal{T}^k). \tag{20}$$

Then the predicted age value is given by $\hat{y} = y_{c^*}$, where $c^* = \arg\min_c g_c(\mathbf{x}; \boldsymbol{\Theta}, \mathcal{F})$.

## 5 Age Estimation by Deep Regression Forests

In this section, we describe Deep Regression Forests (DRFs) for age estimation. Similar to the previous section, we first introduce how to learn a single differentiable regression tree, then describe how to learn tree ensembles to form a forest.

### 5.1 Problem Formulation

We formulate age estimation as a regression problem, where we regard age as a continues numerical value: Let $\mathcal{X} = \mathbb{R}^m$ denote the input facial image space and $\mathcal{Y} = \mathbb{R}$ denote the output age space. For a facial image $\mathbf{x} \in \mathcal{X}$, its chronological age value is $y \in \mathcal{Y}$. The objective of regression is to find a mapping function $g : \mathbf{x} \rightarrow y$ between an input sample $\mathbf{x}$ and its output target $y$. A standard way to address this problem is to model the conditional probability function $p(y|\mathbf{x})$, so that the mapping is given by

$$\hat{y} = \mathbf{g}(\mathbf{x}) = \int yp(y|\mathbf{x})dy. \tag{21}$$

We propose to model this conditional probability by a decision tree based structure $\mathcal{T}$ described in Section 3. Each leaf node $\ell \in \mathcal{L}$ in the regression tree $\mathcal{T}$ holds a probability density distribution $\pi_\ell(y)$ over $\mathcal{Y}$, i.e, $\int \pi_\ell(y)dy = 1$. The conditional probability function $p(y|\mathbf{x};\mathcal{T})$ given by the tree $\mathcal{T}$ is

$$p(y|\mathbf{x};\mathcal{T}) = \sum_{\ell\in\mathcal{L}} P(\ell|\mathbf{x};\mathbf{\Theta})\pi_\ell(y). \tag{22}$$

Then the mapping between $\mathbf{x}$ and $y$ modeled by tree $\mathcal{T}$ is given by $\hat{y} = \mathbf{g}(\mathbf{x};\mathcal{T}) = \int yp(y|\mathbf{x};\mathcal{T})dy$.

## 5.2 Tree Optimization

Given a training set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, learning a regression tree $\mathcal{T}$ leads to minimizing the following negative log likelihood loss:

$$\begin{aligned} R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S}) &= -\frac{1}{N}\sum_{i=1}^{N}\log\left(p(y_i|\mathbf{x}_i,\mathcal{T})\right) \\ &= -\frac{1}{N}\sum_{i=1}^{N}\log\Big(\sum_{\ell\in\mathcal{L}} P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_\ell(y_i)\Big), \end{aligned} \tag{23}$$

where $\boldsymbol{\pi}$ denotes the density distributions contained by all the leaf nodes $\mathcal{L}$. To optimize $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$ w.r.t. the split node parameter $\mathbf{\Theta}$ and the density distributions $\boldsymbol{\pi}$ held by leaf nodes, we also follow the optimization framework described in Section 3, i.e., alternating the following two steps: (1) fixing $\boldsymbol{\pi}$ and optimizing $\mathbf{\Theta}$; (2) fixing $\mathbf{\Theta}$ and optimizing $\boldsymbol{\pi}$, until convergence or a maximum number of iterations is reached.

### 5.2.1 Learning Split Nodes w/ Deterministic Annealing by Gradient Descent

Now, we discuss how to learn the parameter $\mathbf{\Theta}$ for split nodes, when the density distributions held by the leaf nodes $\boldsymbol{\pi}$ are fixed. We introduce the same DA process described in Section 4.2.1 into the optimization for split node parameter $\mathbf{\Theta}$, which reformulates the original regression loss Eq. (23) as the same form as Eq. (7), i.e., $E(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S},T) = R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S}) - TH(\mathbf{\Theta};\mathcal{S})$. We use the same cooling schedule in Section 4.2.1 to decrease $T$ during optimization: $T \leftarrow \eta T$. Similarly, we compute the gradient $\frac{\partial E(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S},T)}{\partial\mathbf{\Theta}}$ by the chain rule, and we have

$$\begin{aligned} \frac{\partial E(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S},T)}{\partial f_{\varphi(n)}(\mathbf{x}_i;\mathbf{\Theta})} &= \frac{\partial R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})}{\partial f_{\varphi(n)}(\mathbf{x}_i;\mathbf{\Theta})} - T\frac{\partial H(\mathbf{\Theta};\mathcal{S})}{\partial f_{\varphi(n)}(\mathbf{x}_i;\mathbf{\Theta})} \\ &= \frac{1}{N}\Big(s_n(\mathbf{x}_i;\mathbf{\Theta})\big(\Gamma_i^{n_r} - TS_i^{n_r}\big) \\ &\quad - \big(1 - s_n(\mathbf{x}_i;\mathbf{\Theta})\big)\big(\Gamma_i^{n_l} - TS_i^{n_l}\big)\Big), \end{aligned} \tag{24}$$

where for a generic node $n \in \mathcal{N}$

$$\Gamma_i^n = \frac{p(y_i|\mathbf{x}_i;\mathcal{T}_n)}{p(y_i|\mathbf{x}_i;\mathcal{T})} = \frac{\sum_{\ell\in\mathcal{L}_n} P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_\ell(y_i)}{p(y_i|\mathbf{x}_i;\mathcal{T})}, \tag{25}$$

and $S_i^n$ is computed by Eq. (11). $\Gamma_i^n$ can be also efficiently computed for all nodes $n$ in the tree $\mathcal{T}$ by a single pass over the tree. Observing that $\Gamma_n^i = \Gamma_{n_l}^i + \Gamma_{n_r}^i$, the computation for $\Gamma_n^i$ can be started at the leaf nodes and conducted in a bottom-up manner. Based on Eq. (24), the split node parameters $\mathbf{\Theta}$ can be learned by standard Back-propagation.

### 5.2.2 Learning Leaf Nodes by Variational Bounding

By fixing the split node parameters $\mathbf{\Theta}$, learning the leaf nodes parameters $\boldsymbol{\pi}$ becomes a constrained optimization problem

$$\min_{\boldsymbol{\pi}} R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S}), \mathbf{s.t.}, \forall\ell, \int \pi_\ell(y)dy = 1. \tag{26}$$

For efficient computation, we represent each density distribution $\pi_\ell(y)$ by a parametric model. Since ideally each leaf node corresponds to a compact homogeneous subset, we assume that the density distribution $\pi_\ell(y)$ in each leaf node is a Gaussian distribution, i.e.,

$$\pi_\ell(y) = \frac{1}{\sqrt{2\pi}\sigma_\ell}\exp\bigg(-\frac{(y-\mu_\ell)^2}{2\sigma_\ell^2}\bigg), \tag{27}$$

where $\mu_\ell$ and $\sigma_\ell$ are the mean and the covariance matrix of the Gaussian distribution. Based on this assumption, Eq. (26) is equivalent to minimizing $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$ w.r.t. $\mu_\ell$ and $\sigma_\ell$. We also propose to address this optimization problem by Variational Bounding [20], [21]. To obtain a tight upper bound of $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$, we apply Jensen's inequality to it

$$\begin{aligned} R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S}) &= -\frac{1}{N}\sum_{i=1}^{N}\log\bigg(\sum_{\ell\in\mathcal{L}} P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_\ell(y_i)\bigg) \\ &= -\frac{1}{N}\sum_{i=1}^{N}\log\bigg(\sum_{\ell\in\mathcal{L}}\rho(\ell|\bar{\boldsymbol{\pi}},y_i,\mathbf{x}_i)\frac{P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_\ell(y_i)}{\rho(\ell|\bar{\boldsymbol{\pi}},y_i,\mathbf{x}_i)}\bigg) \\ &\le -\frac{1}{N}\sum_{i=1}^{N}\sum_{\ell\in\mathcal{L}}\rho(\ell|\bar{\boldsymbol{\pi}},y_i,\mathbf{x}_i)\log\bigg(\frac{P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_\ell(y_i)}{\rho(\ell|\bar{\boldsymbol{\pi}},y_i,\mathbf{x}_i)}\bigg) \\ &= R(\bar{\boldsymbol{\pi}},\mathbf{\Theta};\mathcal{S}) - \frac{1}{N}\sum_{i=1}^{N}\sum_{\ell\in\mathcal{L}}\rho(\ell|\bar{\boldsymbol{\pi}},y_i,\mathbf{x}_i)\log\bigg(\frac{\pi_\ell(y_i)}{\bar{\pi}_\ell(y_i)}\bigg), \end{aligned} \tag{28}$$

where $\rho(\ell|\boldsymbol{\pi},y_i,\mathbf{x}_i) = \frac{P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_\ell(y_i)}{p(y_i|\mathbf{x}_i;\mathcal{T})}$ and it has the property that $\rho(\ell|\boldsymbol{\pi},y_i,\mathbf{x}_i) \in [0,1]$ and $\sum_{\ell\in\mathcal{L}}\rho(\ell|\boldsymbol{\pi},y_i,\mathbf{x}_i) = 1$. Note that, when $\boldsymbol{\pi} = \bar{\boldsymbol{\pi}}$, the equality holds, which indicates this upper bound is tight. Let us define

$$\phi(\boldsymbol{\pi},\bar{\boldsymbol{\pi}}) = R(\bar{\boldsymbol{\pi}},\mathbf{\Theta};\mathcal{S}) - \frac{1}{N}\sum_{i=1}^{N}\sum_{\ell\in\mathcal{L}}\rho(\ell|\bar{\boldsymbol{\pi}},y_i,\mathbf{x}_i)\log\Big(\frac{\pi_\ell(y_i)}{\bar{\pi}_\ell(y_i)}\Big). \tag{29}$$

Then $\phi(\boldsymbol{\pi},\bar{\boldsymbol{\pi}})$ is a tight upper bound for $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$, which has the properties that for any $\boldsymbol{\pi}$ and $\bar{\boldsymbol{\pi}}$, $\phi(\boldsymbol{\pi},\bar{\boldsymbol{\pi}}) \ge \phi(\boldsymbol{\pi},\boldsymbol{\pi}) = R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$ and $\phi(\bar{\boldsymbol{\pi}},\bar{\boldsymbol{\pi}}) = R(\bar{\boldsymbol{\pi}},\mathbf{\Theta};\mathcal{S})$. These two properties give the conditions for Variational Bounding.

Recall that we parameterize $\pi_\ell(\mathbf{y})$ by two parameters: the mean $\mu_\ell$ and the covariance matrix $\sigma_\ell$. Let $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ denote these two parameters held by all the leaf nodes $\mathcal{L}$. We define $\psi(\boldsymbol{\mu},\bar{\boldsymbol{\mu}}) = \phi(\boldsymbol{\pi},\bar{\boldsymbol{\pi}})$, then $\psi(\boldsymbol{\mu},\bar{\boldsymbol{\mu}}) \ge \phi(\boldsymbol{\pi},\boldsymbol{\pi}) = \psi(\boldsymbol{\mu},\boldsymbol{\mu}) = R(\boldsymbol{\pi},$

$\mathbf{\Theta};\mathcal{S})$, which indicates that $\psi(\boldsymbol{\mu},\bar{\boldsymbol{\mu}})$ is also a tight upper bound for $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$. Assume that we are at a point $\boldsymbol{\mu}^{(t)}$ corresponding to the $t$th iteration, then $\psi(\boldsymbol{\mu},\boldsymbol{\mu}^{(t)})$ is a tight upper bound for $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$. In the next iteration, $\boldsymbol{\mu}^{(t+1)}$ is chosen such that $\psi(\boldsymbol{\mu}^{(t+1)},\boldsymbol{\mu}) \leq R(\boldsymbol{\pi}^{(t)},\mathbf{\Theta};\mathcal{S})$, which implies $R(\boldsymbol{\pi}^{(t+1)},\mathbf{\Theta};\mathcal{S}) \leq R(\boldsymbol{\pi}^{(t)},\mathbf{\Theta};\mathcal{S})$. Therefore, we can minimize $\psi(\boldsymbol{\mu},\bar{\boldsymbol{\mu}})$ instead of $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$ after ensuring that $R(\boldsymbol{\pi}^{(t)},\mathbf{\Theta};\mathcal{S}) = \psi(\boldsymbol{\mu}^{(t)},\bar{\boldsymbol{\mu}})$, i.e., $\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}^{(t)}$. Thus, we have

$$\boldsymbol{\mu}^{(t+1)} = \arg\min_{\boldsymbol{\mu}} \psi(\boldsymbol{\mu},\boldsymbol{\mu}^{(t)}). \tag{30}$$

The partial derivative of $\psi(\boldsymbol{\mu},\boldsymbol{\mu}^{(t)})$ w.r.t. $\mu_\ell$ is computed by

$$\begin{aligned}\frac{\partial\psi(\boldsymbol{\mu},\boldsymbol{\mu}^{(t)})}{\partial\mu_\ell} &= \frac{\partial\phi(\boldsymbol{\pi},\boldsymbol{\pi}^{(t)})}{\partial\mu_\ell} \\ &= -\frac{1}{N}\sum_{i=1}^{N}\rho(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i)\sigma_\ell^{-1}(y_i-\mu_\ell).\end{aligned} \tag{31}$$

By setting $\frac{\partial\psi(\boldsymbol{\mu},\boldsymbol{\mu}^{(t)})}{\partial\mu_\ell}=0$, we have

$$\mu_\ell^{(t+1)} = \frac{\sum_{i=1}^{N}\rho(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i)y_i}{\sum_{i=1}^{N}\rho(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i)}. \tag{32}$$

Similarly, we define $\nu(\boldsymbol{\sigma},\bar{\boldsymbol{\sigma}}) = \phi(\boldsymbol{\pi},\bar{\boldsymbol{\pi}})$, then

$$\boldsymbol{\sigma}^{(t+1)} = \arg\min_{\boldsymbol{\sigma}} \nu(\boldsymbol{\sigma},\boldsymbol{\sigma}^{(t)}). \tag{33}$$

The partial derivative of $\nu(\boldsymbol{\sigma},\boldsymbol{\sigma}^{(t)})$ w.r.t. $\sigma_\ell$ is obtained by

$$\begin{aligned}\frac{\partial\nu(\boldsymbol{\sigma},\boldsymbol{\sigma}^{(t)})}{\partial\sigma_\ell} &= \frac{\partial\phi(\boldsymbol{\pi},\boldsymbol{\pi}^{(t)})}{\partial\sigma_\ell} \\ &= -\frac{1}{N}\sum_{i=1}^{N}\rho(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i)\Big[-\frac{1}{2}\sigma_\ell^{-1}+\frac{1}{2}\sigma_\ell^{-2}(y_i-\mu_\ell^{(t+1)})^2\Big].\end{aligned} \tag{34}$$

By Setting $\frac{\partial\nu(\boldsymbol{\sigma},\boldsymbol{\sigma}^{(t)})}{\partial\sigma_\ell}=0$, we have

$$\sigma_\ell^{(t+1)} = \frac{\sum_{i=1}^{N}\rho(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i)(y_i-\mu_\ell^{(t+1)})^2}{\sum_{i=1}^{N}\rho(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i)}. \tag{35}$$

Eqs. (32) and (35) are the update functions for the density distribution $\boldsymbol{\pi}$ held by all leaf nodes, which are step-size free and fast-converged.

### 5.2.3 Learning Leaf Nodes w/ Deterministic Annealing w/o Initialization

One issue remained is how to initialize the starting point $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{\sigma}^{(0)}$. Note that $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$ is convex w.r.t. $\boldsymbol{\pi}$, but is non-convex w.r.t. $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. Consequently, based on the update functions Eqs. (32) and (35), $R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$ may converge to a poor local minimum, if $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{\sigma}^{(0)}$ are not well initialized. In our previous work [29], we did k-means clustering on $\{y_i\}_{i=1}^N$ to obtain $|\mathcal{L}|$ subsets, then initialized $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{\sigma}^{(0)}$ according to cluster assignment. Here, inspired by [23], [24] we propose a deterministic annealing algorithm for the above optimization problem, which leads to an initialization free solution to avoid poor local minimum. In [23], [24], a deterministic annealing Expectation-Maximization (EM) algorithm was presented for Maximum Likelihood Estimation (MLE) problems to obtain better estimates free of the initial parameter values, in which a new posterior parameterized by "temperature" is derived by using the principle of maximum entropy and is used for controlling the annealing process. To apply this strategy to our optimization problem, we first rewrite Eq. (28) in the form of Neal and Hinton's free energy [45]

$$\begin{aligned}&J(\rho,\boldsymbol{\pi}) = \\ &-\frac{1}{N}\sum_{i=1}^{N}\Big(\mathbb{E}_\rho\big[\log\big(P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_\ell(y_i)\big)\big]-\mathbb{E}_\rho[\log\rho]\Big),\end{aligned} \tag{36}$$

where $\mathbb{E}_\rho[\cdot]$ denotes the expectation w.r.t. conditional probability $\rho(\ell|\boldsymbol{\pi},y_i,\mathbf{x}_i)$. For a fixed $\boldsymbol{\pi}$, when $\rho(\ell|\boldsymbol{\pi},y_i,\mathbf{x}_i) = \frac{P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_\ell(y_i)}{p(y_i|\mathbf{x}_i;\mathcal{T})}$, $J(\rho,\boldsymbol{\pi})$ achieves its minimum, i.e., $J(\rho,\boldsymbol{\pi}) \equiv R(\boldsymbol{\pi},\mathbf{\Theta};\mathcal{S})$. $\rho(\ell|\boldsymbol{\pi},y_i,\mathbf{x}_i)$ is the posterior in this MLE problem, which plays an important role in the optimization (see Eqs. (32) and (35)). However, since initial values $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{\sigma}^{(0)}$ are not guaranteed to be near the true ones, $\rho(\ell|\boldsymbol{\pi},y_i,\mathbf{x}_i)$ may be unreliable at early stages of optimization. Ideally, the influence of this conditional probability should be weakened at the beginning, and as the optimization proceeds, the effect should be strengthened. To this ends, we want to seek another conditional probability $\varrho(\ell|\boldsymbol{\pi},y_i,\mathbf{x}_i)$ to replace $\rho(\ell|\boldsymbol{\pi},y_i,\mathbf{x}_i)$ by extending Eq. (36) to a deterministic annealing variant

$$\begin{aligned}&J'(\varrho,\boldsymbol{\pi},\tau) = \\ &-\frac{1}{N}\sum_{i=1}^{N}\Big(\mathbb{E}_\varrho\big[\log\big(P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_\ell(y_i)\big)\big]-\frac{1}{\tau}\mathbb{E}_\varrho[\log\varrho]\Big),\end{aligned} \tag{37}$$

where $\frac{1}{\tau}$ is the temperature parameter. Note that, when $\tau=1$, $J'\equiv J$. According to Neal and Hinton's theory [45], the minimization of $J'(\varrho,\boldsymbol{\pi},\tau)$ can be performed by the following Coordinate Descent iterations:

- Set $\varrho^{(t+1)}$ to $\varrho$ that minimizes $J'(\varrho,\boldsymbol{\pi}^{(t)},\tau)$
- Set $\boldsymbol{\pi}^{(t+1)}$ to $\boldsymbol{\pi}$ that minimizes $J'(\varrho^{(t+1)},\boldsymbol{\pi},\tau)$

Given $\boldsymbol{\pi}^{(t)}$, $\varrho^{(t+1)}$ is obtained by minimizing $J'(\varrho,\boldsymbol{\pi}^{(t)},\tau)$ w.r.t. $\varrho$ under the constraint $\sum_{\ell\in\mathcal{L}}\varrho=1$

$$\begin{aligned}&\frac{\partial J'(\varrho,\boldsymbol{\pi}^{(t)},\tau)}{\partial\varrho} = \\ &-\log\big(P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_\ell^{(t)}(y_i)\big)+\frac{1}{\tau}(\log\varrho+1)+\lambda=0,\end{aligned} \tag{38}$$

where $\lambda$ is a Lagrange multiplier. Thus, we have

$$\varrho^{(t+1)}(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i) = \frac{\big(P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_\ell^{(t)}(y_i)\big)^\tau}{\sum_{\ell\in\mathcal{L}}\big(P(\ell|\mathbf{x}_i;\mathbf{\Theta})\pi_\ell^{(t)}(y_i)\big)^\tau}. \tag{39}$$

Then, by fixing $\varrho(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i) = \varrho^{(t+1)}(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i)$, minimizing $J'(\varrho^{(t+1)},\boldsymbol{\pi},\tau)$ w.r.t. $\boldsymbol{\pi}$ leads to

$$\mu_\ell^{(t+1)} = \frac{\sum_{i=1}^{N}\varrho(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i)y_i}{\sum_{i=1}^{N}\varrho(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i)}, \tag{40}$$

and

$$\sigma_\ell^{(t+1)} = \frac{\sum_{i=1}^{N}\varrho(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i)(y_i-\mu_\ell^{(t+1)})^2}{\sum_{i=1}^{N}\varrho(\ell|\boldsymbol{\pi}^{(t)},y_i,\mathbf{x}_i)}. \tag{41}$$

Comparing the two groups of update functions for leaf nodes, i.e., Eq. (32) *versus* Eq. (40) and Eq. (35) *versus* Eq. (41), the only difference between them is that the posterior is changed from $\rho(\ell|\boldsymbol{\pi}, y_i, \mathbf{x}_i)$ to $\varrho(\ell|\boldsymbol{\pi}, y_i, \mathbf{x}_i)$. When $\tau \to 0$, the posterior $\varrho(\ell|\boldsymbol{\pi}, y_i, \mathbf{x}_i)$ becomes uniform distribution, i.e., $\varrho(\ell|\boldsymbol{\pi}, y_i, \mathbf{x}_i) \leftarrow \frac{1}{|\mathcal{L}|}$. Thus, for a small enough $\tau$, no matter how $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{\sigma}^{(0)}$ are initialized, $\varrho(\ell|\boldsymbol{\pi}, y_i, \mathbf{x}_i)$ does not influence the optimization, since $J'(\varrho, \boldsymbol{\pi}, \tau)$ always has the minimum: $\mu_\ell = \frac{1}{N}\sum_{i=1}^N y_i,\ \sigma_\ell = \frac{1}{N}\sum_{i=1}^N (y_i - \mu_\ell)^2$. Then, by gradually increasing $\tau$ (decreasing the temperature), $\varrho(\ell|\boldsymbol{\pi}, y_i, \mathbf{x}_i)$ gradually becomes nonuniform, and the influence of $\varrho(\ell|\boldsymbol{\pi}, y_i, \mathbf{x}_i)$ in the optimization is increasedly strengthened. In this annealing process, we also start with each input sample equally influencing all leaf nodes and gradually localize the influence, which is a well-known strategy to obtain better optima [22].

### 5.3 Learning a Regression Forest

We use the same strategies in Section 4.3 to learn a regression forest. A regression forest is an ensemble of regression trees $\mathcal{F} = \{\mathcal{T}^1, \ldots, \mathcal{T}^K\}$, where all trees can possibly share the same parameters in $\boldsymbol{\Theta}$, but each tree can have a different set of split functions (assigned by $\varphi$, as shown in Fig. 2), and independent leaf node distribution $\boldsymbol{\pi}$. The loss function for a regression forest is also defined as the averaged loss functions of all individual trees (Eq. (18)). Learning the forest $\mathcal{F}$ also follows the alternating optimization strategy described in Section 5.2. To learn $\boldsymbol{\Theta}$, we have the gradient of the forest loss given in Eq. (18), but the first term of the right side is computed by Eq. (24). For the temperature parameter $T$ introduced when optimizing $\boldsymbol{\Theta}$, we use the same initialization and cooling schedule described in Section 4.3. The leaf node distribution $\boldsymbol{\pi}$ of each tree in the forest $\mathcal{F}$ is updated independently according to Eqs. (40) and (41). For the temperature parameter $\tau$ introduced in optimizing $\boldsymbol{\pi}$, we initialize it as a small value $\tau_0$ ($\tau_0 < 1$) and gradually increase it by $\tau \leftarrow \tau/\eta$ until $\tau = 1$, where $\eta$ is a constant cooling factor less than 1. The training procedure of a DRF is shown in Algorithm 2.

**Algorithm 2.** The Training Procedure of a DRF

```
Require: S: training set
Require: n_B: the number of mini-batches to update π
Require: T_0, τ_0: two starting temperature parameters
Require: η: a constant cooling factor
  Initialize Θ and π randomly.
  Set B = {∅}, T = T_0 and τ = τ_0
  while Not converge do
    while |B| < n_B do
      Randomly select a mini-batch B from S
      Update Θ by Gradient Descent (Eqs. (19), (24)) on B
      B = B ∪ B
    end while
    Update π by iterating Eqs. (40) and (41) on B
    B = {∅}
    if T ≥ 0 then
      T ← ηT
    end if
    if τ < 1 then
      τ ← τ/η
    end if
end while
```

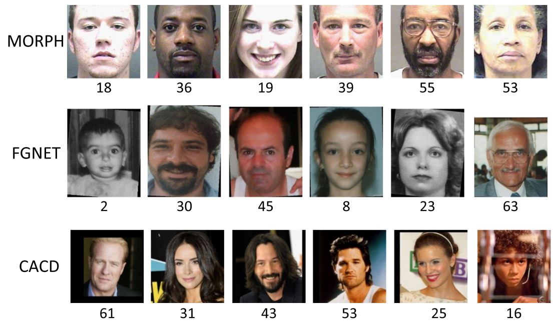



Fig. 5. Some examples of MORPH [25], FG-NET [26] and CACD [27]. The number below each image is the chronological age of each subject.

In the testing stage, the output of the forest $\mathcal{F}$ is given by averaging the predictions from all the individual trees

$$\begin{aligned} \hat{y} = \mathbf{g}(\mathbf{x}; \mathcal{F}) &= \frac{1}{K}\sum_{k=1}^{K} \mathbf{g}(\mathbf{x}; \mathcal{T}^k) \\ &= \frac{1}{K}\sum_{k=1}^{K} \int y p(y|\mathbf{x}; \mathcal{T}^k) dy \\ &= \frac{1}{K}\sum_{k=1}^{K} \int y \sum_{\ell \in \mathcal{L}^k} P(\ell|\mathbf{x}; \boldsymbol{\Theta}) \pi_\ell(y) dy \\ &= \frac{1}{K}\sum_{k=1}^{K} \sum_{\ell \in \mathcal{L}^k} P(\ell|\mathbf{x}; \boldsymbol{\Theta}) \mu_\ell, \end{aligned} \tag{42}$$

where $\mathcal{L}^k$ is the leaf node set of the $k$th tree. Here, we take the fact that the expectation of the Gaussian distribution $\pi_\ell(y)$ is $\mu_\ell$.

## 6 Experiments

In this section, we first introduce our experimental setup including the datasets, evaluation metrics, and implementation details. Then we compare our results with others to show the effectiveness of our algorithms. After that, we conduct elaborate ablation experiments to study the influence of different designs and variants of our methods. Finally, we discuss the comparison between our DLDLF and DRF, visualizations of the learned leaf nodes, the hyper-parameters and performance variance brought by random assignment $\varphi(\cdot)$.

### 6.1 Experimental Setup

#### *6.1.1 Datasets*

We conduct our experiments on three standard benchmarks: MORPH [25], FG-NET [26] and the Cross-Age Celebrity Dataset [27]. Some examples of these three datasets are illustrated in Fig. 5.

*MORPH.* MORPH is the most popular dataset for age estimation, which contains more than 55,000 images from about 13,000 people of different races. Each of the facial image is annotated with a chronological age. The ethnicity of MORPH is very unbalanced, as more than 96 percent of the facial images are from African or European people.

Existing methods adopted different experimental setups on MORPH. The first setup (Setup I) is introduced in [3], [6], [31], [33], [46], [47], [48], which selects 5,492 images of

Caucasian people from the original MORPH dataset, to reduce the cross-ethnicity effects. In Setup I, these 5,492 images are randomly partitioned into two subsets: 80 percent of the images are selected for training and others for testing. The random partition is repeated 5 times, and the final performance is averaged over these 5 different partitions.

The second setup (Setup II) is used in [8], [34], [37], under which all of the images in MORPH are randomly split into training/testing (80/20 percent) sets. The random splitting is performed 5 times repeatedly and the final performance is obtained by averaging the performances of these 5 different splits.

There are also several methods [11], [49], [50] using the third setup (Setup III), which randomly selected a subset (about 21,000 images) from MORPH and restricted the ratio between Black and White and the one between Female and Male are 1:1 and 1:3, respectively.

*FG-NET*. FG-NET [26] is also a widely used dataset for age estimation. It contains 1,002 facial images of 82 individuals, in which most of them are white people. Each individual in FG-NET has more than 10 photos taken at different ages. The images in FG-NET have a large variation in lighting conditions, poses and expressions.

Following the experimental setup used in [3], [31], [46], [51], [52], we perform "leave one out" cross validation on this dataset, i.e., we leave images of one person for testing and take the remaining images for training.

*CACD*. CACD [27] is a large dataset which has around 160,000 facial images of 2,000 celebrities collected from the Internet. These celebrities are divided into three subsets: the training set, the testing set and the validation set which consist of 1,800 celebrities, 120 celebrities and 80 celebrities, respectively. The validation and testing sets are clean but the training set is noisy.

For evaluation we adopt the setup used in [3]. They report results on the testing set obtained by using the models trained on the training set and the validation set, respectively.

#### 6.1.2 Evaluation Metric

The performance of age estimation is evaluated in terms of mean absolute error (MAE) as well as Cumulative Score (CS). MAE is the average absolute error over the testing set, and the Cumulative Score is calculated by $\mathrm{CS}(l) = \frac{K_l}{K} \cdot 100\%$, where $K$ is the total number of testing images and $K_l$ is the number of testing facial images whose absolute error between the estimated age and the ground truth age is not greater than $l$ years. Here, we set the same error level 5 as in [33], [46], [53], i.e., $l = 5$. Note that, since not all the methods reported the Cumulative Score, we are only able to give CS values for some competitors.

#### 6.1.3 Implementation Details

Our realizations of DLDLFs and DRFs are based on the public available "caffe" [54] framework. Following recent deep learning based age estimation methods [3], [6], [37], [48], we use the VGG-16 Net [55] as the CNN part of the proposed DLDLFs and DRFs.

*Parameters Setting.* The forest model related hyper-parameters (and the default values we used) are: number of trees (5), tree depth (6), number of output units produced by the feature learning function (128), iterations to update leaf-node predictions (20), number of mini-batches used to update leaf node predictions (50).

TABLE 1
Performance Comparison on MORPH [25] (Setup I)(*: The Value Is Read from the CS Curve Shown in the Reference)

| Method | MAE | CS |
|---|---|---|
| Human workers [18] | 6.30 | 51.0 %* |
| AGES [58] | 8.83 | 46.8 %* |
| MTWGP [59] | 6.28 | 52.1%* |
| CA-SVR [46] | 5.88 | 57.9% |
| SVR [31] | 5.77 | 57.1% |
| DLA [47] | 4.77 | 63.4 %* |
| Rank [60] | 6.49 | 49.5% |
| Rothe [48] | 3.45 | N/A |
| DEX [3] | 3.25 | N/A |
| ARN [6] | 3.00 | N/A |
| DLDLF (ours) | 2.94 | 84.7% |
| DRF (ours) | **2.80** | **85.6%** |

The age distribution generation related hyper-parameter (and the default value we used) is: the pre-defined standard deviation $\alpha$ in Eq. (3) (2.0).

The network training related hyper-parameters (and the values we used) are: initial learning rate (0.05), mini-batch size (16), maximal iterations (30k). We decrease the learning rate ($\times 0.5$) every 10k iterations.

We fixed the values for the temperature parameters introduced in our DA processes: $T_0 = 1$, $\tau_0 = 0.5$ and $\eta = 0.9$.

*Preprocessing.* Face alignment is a common preprocessing operation for age estimation [3], [5], [6], [12], [37], [50].

Following these previous methods, we perform face alignment to guarantee all eyeballs stay at the same position in the image: Faces are first detected by using a standard face detector [56] and facial landmarks are localized by AAM [57].

### 6.2 Performance Comparison

In this section we compare our LDLF and DRF with other state-of-the-art age estimation methods on the three standard benchmarks: MORPH [25], FG-NET [26] and the Cross-Age Celebrity Dataset [27].

*MORPH.* We first compare the proposed LDLF and DRF with other state-of-the-art age estimation methods on MORPH. As we described before, there are three experimental setups used on this dataset. For a fair comparison, we test the proposed LDLF and DRF on MORPH under all these three setups. The quantitative results of the three settings are summarized in Tables 1, 2 and 3, respectively. As can be seen from these tables, our DRF and LDLF achieve the best and the second best performance on all of the setups, respectively, and outperform the current state-of-the-arts with a clear margin. This result shows the effectiveness of jointly learning input-dependent data partition and data distributions in local partitions for age estimation.

*FG-NET*. We then conduct experiments on FG-NET [26]. The quantitative comparisons on FG-NET dataset are shown in Table 4. As can be seen, DRF and DLDLF outperform other methods significantly. Note that, they are the

TABLE 2
Performance Comparison on MORPH [25] (Setup II)(*: The Value Is Read from the CS Curve Shown in the Reference)

| Method | MAE | CS |
|---|---|---|
| IIS-LDL [8] | 5.67 | 71.2%* |
| CPNN [9] | 4.87 | N/A |
| Huerta [53] | 4.25 | 71.2% |
| BFGS-LDL [34] | 3.94 | N/A |
| OHRank [33] | 6.07 | 56.3% |
| OR-SVM [60] | 4.21 | 68.1%* |
| CCA [61] | 4.73 | 60.5%* |
| LSVR [30] | 4.31 | 66.2%* |
| OR-CNN [12] | 3.27 | 81.5% |
| SMMR [13] | 3.24 | N/A |
| Ranking-CNN [5] | 2.96 | 85.2% |
| DLDL [37] | 2.42 | N/A |
| Mean-Variance Loss [7] | 2.41 | 91.2% |
| DLDLF (ours) | 2.19 | **93.0%** |
| DRF (ours) | **2.14** | 91.3% |

TABLE 4
Performance Comparison on FG-NET [26](*: The Value Is Read from the CS Curve Shown in the Reference)

| Method | MAE | CS |
|---|---|---|
| Human workers [18] | 4.70 | 69.5%* |
| Rank [60] | 5.79 | 66.5%* |
| DIF [18] | 4.80 | 74.3%* |
| AGES [58] | 6.77 | 64.1%* |
| IIS-LDL [8] | 5.77 | N/A |
| CPNN [9] | 4.76 | N/A |
| MTWGP [59] | 4.83 | 72.3%* |
| CA-SVR [46] | 4.67 | 74.5% |
| LARR [31] | 5.07 | 68.9%* |
| OHRank [33] | 4.48 | 74.4% |
| DLA [47] | 4.26 | N/A |
| CAM [62] | 4.12 | 73.5%* |
| Rothe [48] | 5.01 | N/A |
| DEX [3] | 4.63 | N/A |
| Mean-Variance Loss [7] | 4.10 | 78.5%* |
| DLDLF (Ours) | 3.71 | 84,8% |
| DRF (Ours) | **3.47** | **87.3%** |

only two methods that have a MAE below 4.0. The age distribution of FG-NET is strongly biased, moreover, the "leave one out" cross validation policy further aggravates the bias between the training set and the testing set. The ability of overcoming the bias between training and testing sets indicates that the proposed LDLF and DRF can handle inhomogeneous data well.

*CACD.* Finally, we conduct our experiments on CACD [27]. The detailed comparisons are shown in Table 5. The proposed DLDLF and DRF perform better than the competitor DEX [3], no matter which set they are trained on. It's worth noting that, the improvements of DLDLF and DRF to DEX are much more significant when they are trained on the validation set than the training set. This result can be explained as follow: As described earlier, the inhomogeneous data is the main challenge for training age estimation models. This challenge can be alleviated by enlarging the number of the training samples. Therefore, DEX, DLDLF and DRF achieve comparable results when they are trained on the training set. But when they are trained on the validation set, which is much smaller than the training set, DLDLF and DRF, especially DRF, outperform DEX significantly, because DLDLF and DRF directly address the inhomogeneity challenge. Therefore, DLDLF and DRF are capable of handling inhomogeneous data even when learned from a small set.

### 6.3 Ablation Study

We conduct some ablation experiments on the MORPH dataset (Setup I), to analyze the influence of different designs and components for our methods. We want to answer these questions from the ablation study: (1) Since we argue that age estimation is usually formulated as an LDL or regression problem rather than a classification problem, what the result would be if we addressed age estimation by a deep classification forest model [19]? (2) Since we argue that our forest structure is important for age estimation, what the result would be if we replaced our forest structure by an $\ell_2$ regression loss function? (3) Since we argue that the convergence of the update functions for leaf nodes used in NRF [43] is not guaranteed, what the result would be if we changed our update functions in DRFs to them? (4) What the result would be if the DA process for learning split nodes was not used ($T = 0$)? (5) What would the result be if the DA process for learning leaf nodes in DRFs was not used ($\tau = 1$), especially when leaf nodes are not initialized by kmeans clustering (w/o kmeans initialization)? To answer these questions, we consider these variants of our methods in the ablation experiments:

- DLDLF ($T = 0$): the DLDLF *without* the DA process for learning split nodes, i.e., the baseline DLDLFs proposed in our previous work [28].
- DLDLF (full model): the DLDLF *with* the DA process for learning split nodes, i.e., the method described in Section 4.
- DRF ($T = 0$, $\tau = 1$, w/ kmeans initialization): the DRF *without* the two DA processes for learning split and leaf nodes and *with* kmeans initialization for leaf nodes, i.e., the baseline DRF proposed in our previous work [29].

TABLE 3
Performance Comparison on MORPH [25] (Setting III)

| Method | MAE | CS |
|---|---|---|
| KPLS [11] | 4.18 | N/A |
| Guo and Mu [49] | 3.92 | N/A |
| CPLF [50] | 3.63 | N/A |
| DLDLF (ours) | 2.99 | **85.6%** |
| DRF (ours) | **2.90** | 82.7% |

TABLE 5
Performance Comparison on CACD (Measured by MAE) [27]

| Trained on | Dex [3] | DLDLF (Ours) | DRF (Ours) |
|---|---|---|---|
| CACD (train) | 4.785 | 4.679 | **4.610** |
| CACD (val) | 6.521 | 6.162 | **5.630** |

TABLE 6
Ablation Study on MORPH [43] (Setup I)

| Method | MAE | CS |
|---|---|---|
| DNDF [19] | 3.32 | 80.7% |
| Deep Regression | 3.21 | 81.1% |
| DRF-NRF (w/ kmeans initialization) [43] | 3.51 | 75.4% |
| DLDLF ($T = 0$) | 3.02 | 81.3% |
| DLDLF (full model) | 2.94 | 84.7% |
| DRF ($T = 0$, $\tau = 1$, w/ kmeans initialization) | 2.91 | 82.9% |
| DRF ($T = 0$, $\tau = 1$) | 6.91 | 47.8% |
| DRF ($\tau = 1$, w/ kmeans initialization) | 2.85 | 83.9% |
| DRF ($T = 0$) | 2.85 | 84.1% |
| DRF (full model) | 2.80 | 85.6% |

- DRF ($T = 0$, $\tau = 1$): the DRF *without* the two DA processes for learning split and leaf nodes and also *without* kmeans initialization for leaf nodes.
- DRF ($\tau = 1$, w/ kmeans initialization): the DRF *with* the DA process for learning split nodes, *without* the two DA processes for leaf nodes, and *with* kmeans initialization for leaf nodes.
- DRF ($T = 0$): the DRF *without* the DA process for learning split nodes, *with* the two DA processes for leaf nodes, and *without* kmeans initialization for leaf nodes.
- DRF (full model): the DRF *with* the two DA processes for learning split and leaf nodes and *without* kmeans initialization for leaf nodes, i.e., the method described in Section 5.

The results of the ablation experiments are summarized in Table 6.

### 6.3.1 Age Estimation by Classification

To formulate age estimation as a classification problem, we treat each age value as a class. We apply a deep classification forest model, deep neural decision forests [19], to address this problem. As shown in Table 6, the age estimation result obtained by DNDFs is much worse than the results obtained by our baseline DLDLF and DRF, i.e., DLDLF ($T = 0$) and DRF ($T = 0, \tau = 1$, w/ kmeans initialization), which evidences that age estimation is not suitable to be formulated as a classification problem.

### 6.3.2 Age Estimation w/o Forest Structure

To verify whether our forest structure is important for age estimation, we replace our forest structure by an $\ell_2$ norm (euclidean) loss function and denote this method by Deep Regression. As shown in Table 6, the age estimation result obtained by Deep Regression is even worse than the result obtained by our baseline DRF, i.e., DRF ($T = 0, \tau = 1$, w/ kmeans initialization), which evidences the importance of our forest structure.

### 6.3.3 The Update Functions for Leaf Nodes

The method which replaces the update functions for leaf nodes in a NRF by those in a DRF is denoted by DRF-NRF. For fair comparison, we also initialize leaf node distributions in DRF-NRF by kmeans. As can be seen, using NRF's leaf node update functions leads to a much worse result than our baseline DRF, i.e., DRF ($T = 0, \tau = 1$, w/ kmeans initialization), which agrees with our concern about the convergence of NRF's leaf node update rule.

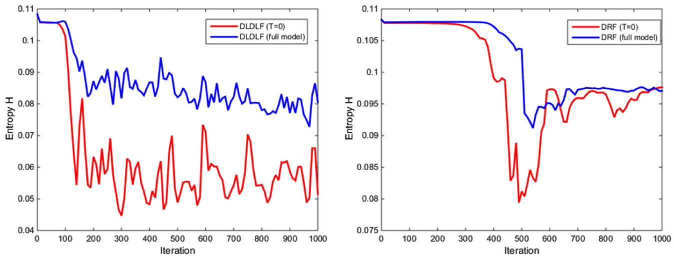



Fig. 6. The dynamics of the averaged entropy $H(\mathbf{\Theta}; \mathcal{S})$ over the training set $\mathcal{S}$ for DLDLFs (left) and DRFs (right) at the beginning of tree learning.

### 6.3.4 The DA Process for Learning Split Nodes

By comparing DLDLF (full model), DRF (full model) and DRF ($\tau = 1$, w/ kmeans initialization) with DLDLF ($T = 0$), DRF ($T = 0$) and DRF ($T = 0$, $\tau = 1$, w/ kmeans initialization), respectively, we see that using the DA process for learning split nodes always leads to better performances. We also show the dynamics of the averaged entropy $H(\mathbf{\Theta}; \mathcal{S})$ over the training set $\mathcal{S}$ at the beginning of tree learning in Fig. 6, where we see for both DLDLF (full model) and DRF (full model), the averaged entropies $H(\mathbf{\Theta}; \mathcal{S})$ of them are lager and decrease more slowly than those of DLDLF ($T = 0$) and DRF ($T = 0$). This result is consistent with our intuition about the DA process for learning split nodes described in Section 4.2.1.

### 6.3.5 The DA Process for Learning Leaf Nodes

We first compare DRF ($T = 0$, $\tau = 1$, w/ kmeans initialization) with DRF ($T = 0, \tau = 1$, w/o kmeans initialization). The result obtained by DRF ($T = 0, \tau = 1$, w/o kmeans initialization) is much worse than the one obtained by DRF ($T = 0$, $\tau = 1$, w/ kmeans initialization). This comparison shows that without the DA process for learning leaf nodes, the leaf node update may converge to a poor local minimum, if the leaf node parameters are not well initialized. We then compare DRF (full model) with DRF ($\tau = 1$, w/ kmeans initialization), where we see that using the DA process for learning leaf nodes, even without well initializing the leaf node parameters, can lead to a better performance. This comparison indicates that the proposed DA process for learning leaf nodes can avoid poor local optima and obtain better estimates of tree parameters free of initial parameter values.

## 6.4 Discussion

### 6.4.1 Comparison between DLDLFs and DRFs

Although DLDLFs and DRFs formulate age estimation as different problems, both of them take the strong correlation between close ages of the same individual into account. The difference is DLDLFs explicitly model cross age correlations of the same individual, while DRFs do this implicitly. The experimental results in Section 6.2 show that DRFs always achieve lower (better) MAE than DLDLFs. The reason might be that DRFs directly approach the precise chronological ages. However, an interesting result is, for both Setup II and

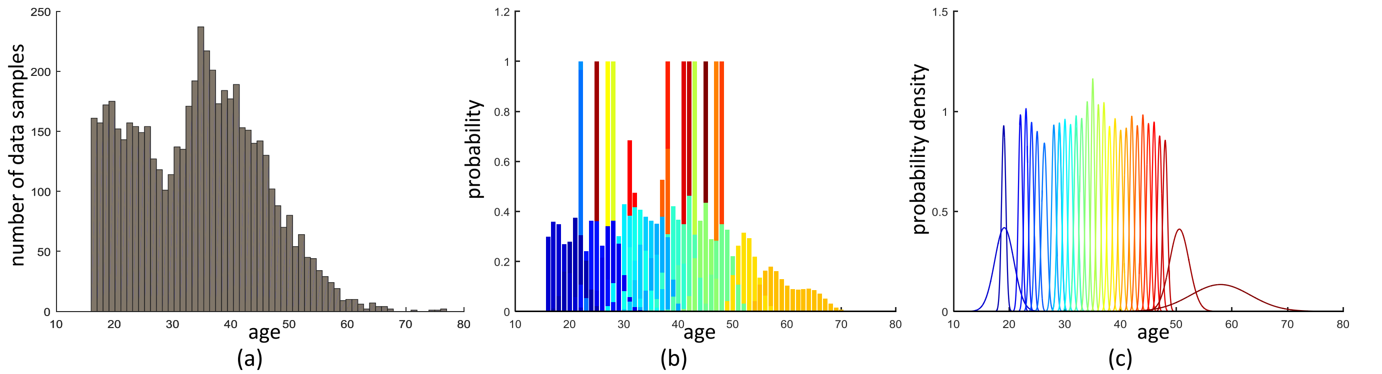



Fig. 7. (a) Histogram of data samples with respect to age on MORPH [25] (Setup I). (b) Visualization of the learned leaf node distributions in our DLDLF. (c) Visualization of the learned leaf node distributions in our DRF. The distributions held by different leaf nodes are in different (gradually varied) colors, which are best viewed in color.

Setup III of the MORPH dataset, that DLDLFs obtain a worse MAE than DRFs, but a better CS. This is because the CS metric tolerates small prediction errors, and DLDLFs learn an age distribution, which benefits fuzzy prediction. Another reason for the worse MAEs achieved by DLDLFs is that we generate age distributions by a fixed $\alpha$ for different individuals, which might be inflexible in adapting to complex face data domains with diverse cross-age correlations [10]. Our future work is to learn the age distributions adaptive to different individuals [7].

#### 6.4.2 Visualization of Learned Leaf Nodes

To better understand DLDLF and DRF, we visualize the distributions at the leaf nodes learned on MORPH [25] (Setup I) in Figs. 7b and 7c, respectively. For reference, we also display the histogram of data samples (the vertical axis) with respect to age (the horizontal axis). Each leaf node in our DLDLF contains a discrete probability distribution, which is represented by a colored histogram in Fig. 7b. Each leaf node in our DRF contains a Gaussian distribution, as visualized in in Fig. 7c. The horizontal axes in both Figs. 7b and 7c represent age and the vertical axes in them represent probability and probability density, respectively (we rescale some very sharp distributions in Fig. 7c for better visualization, since the peak densities of them are too large). According to Fig. 7a, the age data in MORPH was sampled mostly below age 60, and densely concentrated around 20's and 40's. As shown in Figs. 7b and 7c, both of the distributions learned by DLDLF and DRF fit the age data well: The discrete distributions around 60 are more uniform than those spread in the interval between 20 and 50; The Gaussian distribution centered around 60 has much larger variance than those centered in the interval between 20 and 50, but has smaller probability density. Note that, although these learned distributions represent homogeneous local partitions, the number of samples is not necessarily uniformly distributed among partitions. Another phenomenon is these distributions are heavily overlapped, which accords with the fact that different people with the same age but have quite different facial appearances.

#### 6.4.3 Sensitivity of Hyper-Parameters

Now we discuss three important hyper-parameters: the tree number, the tree depth and the standard deviation $\alpha$ in Eq. (3) used for age distribution generation. We vary each of them and fix the other one to the default value to see how the performance changes on MORPH (Setup I).

*Tree Number.* As we stated in Section 1, the ensemble strategy to learn a forest proposed in DNDFs [19] is different from ours. Therefore, it is necessary to see which ensemble strategy is better to learn a forest. Towards this end, we replace our ensemble strategy by the one used in DNDFs, and name the methods DLDLF-DNDF and DRF-DNDF, accordingly. As shown in Fig. 8a, our ensemble strategy can improve the performance by using more trees, as we expected, while the one used in DNDFs leads to an even worse performance than one for a single tree.

*Tree Depth.* Tree depth is another important parameter for decision trees. As shown in Fig. 8b, for both DLDLF and DRF, with the tree depth increase, the MAE first becomes lower and then stable. One concern about the tree depth is that a very deep tree may lead to underflow due to the continued product form of Eq. (1). However, this will not happen in

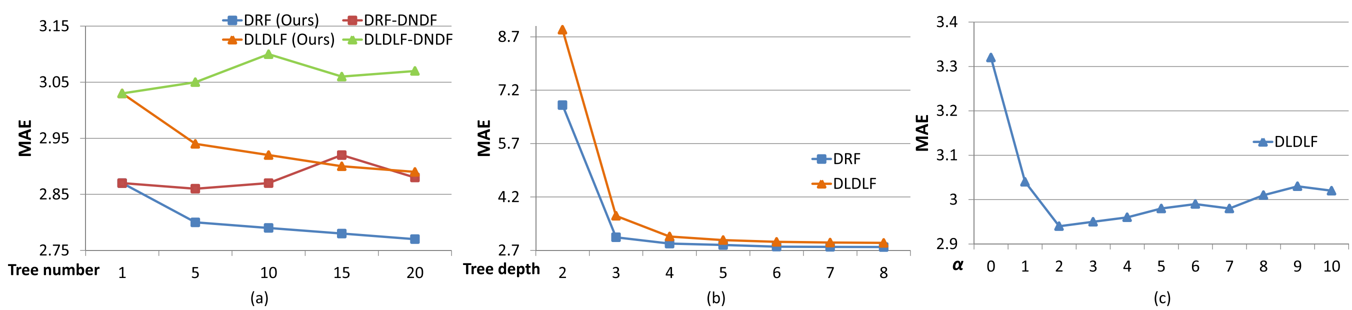



Fig. 8. Performance changes by varying (a) tree number, (b) tree depth and (c) standard deviation $\alpha$ on MORPH [25] (Setup I).

practice. First, according to Fig. 8b, the performance of our method becomes saturated when the tree depth is larger than 6, thus it is unnecessary to use very deep trees. Second, there is an implicit constraint between tree depth $h$ and unit number $m$ of the FC layer $\mathbf{f}$: $m \geq 2^{h-1} - 1$. The maximum unit number of the FC layers in a typical CNN architecture, e.g., AlexNet [63] and VGG-16 Net [55], is 4096, which implies that the tree depth should not be larger than 13.

*Standard Deviation* $\alpha$. Fig. 8c shows that how the MAE of our DLDLF changes by using Gaussian distributions with different standard deviation $\alpha$ to generate age distributions. Note that, when $\alpha = 0$, the generated age distributions are one-hot, i.e., the LDL problem becomes a standard classification problem, which leads to a significant performance reduction. When $\alpha$ is larger, the generated age distributions are dispersive, which also leads to performance decrease. This is consistent with our intuition that neighboring ages can help to describe the face appearance of a given age but should not change the priority of the original given age.

#### 6.4.4 Performance Variance Brought by Random Assignment $\varphi(\cdot)$

In our forests, CNN features are randomly selected and assigned to split nodes for forest training, as defined by the function $\varphi(\cdot)$. In order to to check whether this randomness will seriously affect performance, we train 50 DRFs on MORPH (Setup I) and find that the standard derivation of the results obtained by the 50 DRFs is only 0.01 MAE. Therefore, the random feature selection and assignment process does not seriously affect the performance. This result is plausible, since CNN features are first initialized by the values computed by random weights, or weights from a pre-trained model trained for classification on Imagenet, which is a very different task than age estimation, and then the selected CNN features used in the forest will be learned and optimized for age estimation during forest training.

## 7 Conclusion

We proposed two Deep Differentiable Random Forests, i.e., Deep Label Distribution Forest (DLDLF) and Deep Regression Forest for age estimation, which learn nonlinear mapping between inhomogeneous facial feature space and ages. In these two forests, by performing soft data partition at split nodes, the forests can be connected to a deep network and learned in an end-to-end manner, where data partition at split nodes is learned by Back-propagation and age distribution at leaf nodes is optimized by iterating a step-size free and fast-converged update rule derived from Variational Bounding. In addition, two Deterministic Annealing processes are introduced into the learning of split and leaf nodes, respectively, to avoid poor local optima and obtain better estimates of tree parameters free of initial parameter values. The end-to-end learning of split and leaf nodes ensures that partition function at each split node is input-dependent and the local input-output correlation at each leaf node is homogeneous. Experimental results showed that DLDLF and DRF achieved state-of-the-art results on three age estimation benchmarks. Our Deep Differentiable Random Forests are also applicable to other problems with inhomogeneous data, which will be investigated in our future work.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China No. 61672336 and in part by ONR N00014-15-1-2356. The authors would like to thank Chenxi Liu, Siyuan Qiao and Zhishuai Zhang for instructive discussions.

## References

[1] K. Alkass, B. A. Buchholz, S. Ohtani, T. Yamamoto, H. Druid, and K. L. Spalding, "Age estimation in forensic sciences: Application of combined aspartic acid racemization and radiocarbon analysis," *Mol. Cellular Proteomics*, vol. 9, pp. 1022–1030, 2010.

[2] H. Han, C. Otto, and A. K. Jain, "Age estimation from face images: Human vs. machine performance," in *Proc. Int. Conf. Biometrics*, 2013, pp. 1–8.

[3] R. Rothe, R. Timofte, and L. V. Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *Int. J. Comput. Vis.*, vol. 126, pp. 144–157, 2018.

[4] G. Levi and T. Hassner, "Age and gender classification using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 34–42.

[5] S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao, "Using ranking-CNN for age estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 742–751.

[6] E. Agustsson, R. Timofte, and L. V. Gool, "Anchored regression networks applied to age estimation and super resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1652–1661.

[7] H. Pan, H. Han, S. Shan, and X. Chen, "Mean-variance loss for deep age estimation from a face," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5285–5294.

[8] X. Geng, K. Smith-Miles, and Z. Zhou, "Facial age estimation by learning from label distributions," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 451–456.

[9] X. Geng, C. Yin, and Z. Zhou, "Facial age estimation by learning from label distributions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2401–2412, Oct. 2013.

[10] Z. He, X. Li, Z. Zhang, F. Wu, X. Geng, Y. Zhang, M.-H. Yang, and Y. Zhuang, "Data-dependent label distribution learning for age estimation," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3846–3858, Aug. 2017.

[11] G. Guo and G. Mu, "Simultaneous dimensionality reduction and human age estimation via kernel partial least squares regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 657–664.

[12] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, "Ordinal regression with multiple output CNN for age estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4920–4928.

[13] D. Huang, L. Han, and F. D. la Torre, "Soft-margin mixture of regressions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4058–4066.

[14] N. Ramanathan, R. Chellappa, and S. Biswas, "Age progression in human faces: A survey," *J. Visual Lang. Comput.*, vol. 15, pp. 3349–3361, 2009.

[15] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Comput.*, vol. 9, no. 7, pp. 1545–1588, 1997.

[16] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[17] A. Criminisi and J. Shotton, *Decision Forests for Computer Vision and Medical Image Analysis*. Berlin, Germany: Springer, 2013.

[18] H. Han, C. Otto, X. Liu, and A. K. Jain, "Demographic estimation from face images: Human vs. machine performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 6, pp. 1148–1161, Jun. 2015.

[19] P. Kontschieder, M. Fiterau, A. Criminisi, and S. R. Bulò, "Deep neural decision forests," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1467–1475.

[20] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, no. 2, pp. 183–233, 1999.

[21] A. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Comput.*, vol. 15, no. 4, pp. 915–936, 2003.

[22] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proc. IEEE*, vol. 86, no. 11, pp. 2210–2239, Nov. 1998.

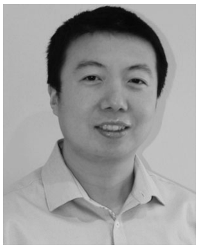
[23] N. Ueda and R. Nakano, "Deterministic annealing variant of the EM algorithm," in *Proc. 7th Int. Conf. Neural Inf. Process. Syst.*, 1994, pp. 545–552.
[24] N. Ueda and R. Nakano, "Deterministic annealing EM algorithm," *Neural Netw.*, vol. 11, no. 2, pp. 271–282, 1998.
[25] K. Ricanek and T. Tesafaye, "MORPH: A longitudinal image database of normal adult age-progression," in *Proc. 7th Int. Conf. Autom. Face Gesture Recognit.*, 2006, pp. 341–345.
[26] G. Panis, A. Lanitis, N. Tsapatsoulis, and T. F. Cootes, "Overview of research on facial ageing using the FG-NET ageing database," *IET Biometrics*, vol. 5, no. 2, pp. 37–46, 2016.
[27] B. Chen, C. Chen, and W. H. Hsu, "Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset," *IEEE Trans. Multimedia*, vol. 17, no. 6, pp. 804–815, Jun. 2015.
[28] W. Shen, K. Zhao, Y. Guo, and A. Yuille, "Label distribution learning forests," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 834–843.
[29] W. Shen, Y. Guo, Y. Wang, K. Zhao, B. Wang, and A. L. Yuille, "Deep regression forests for age estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2304–2313.
[30] G. Guo, G. Mu, Y. Fu, and T. S. Huang, "Human age estimation using bio-inspired features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 112–119.
[31] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang, "Image-based human age estimation by manifold learning and locally adjusted robust regression," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1178–1188, Jul. 2008.
[32] A. Montillo and H. Ling, "Age regression from faces using random forests," in *Proc. 16th IEEE Int. Conf. Image Process.*, 2009, pp. 2465–2468.
[33] K. Y. Chang, C. S. Chen, and Y. P. Hung, "Ordinal hyperplanes ranker with cost sensitivities for age estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 585–592.
[34] X. Geng, "Label distribution learning," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1734–1748, Jul. 2016.
[35] X. Yang, X. Geng, and D. Zhou, "Sparsity conditional energy label distribution learning for age estimation," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2259–2265.
[36] A. L. Berger, S. D. Pietra, and V. J. D. Pietra, "A maximum entropy approach to natural language processing," *Comput. Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.
[37] B. B. Gao, C. Xing, C. W. Xie, J. Wu, and X. Geng, "Deep label distribution learning with label ambiguity," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2825–2838, Jun. 2017.
[38] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, 1995, pp. 278–282.
[39] C. Leistner, A. Saffari, J. Santner, and H. Bischof, "Semi-supervised random forests," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 506–513.
[40] C. Leistner, A. Saffari, and H. Bischof, "Miforests: Multiple-instance learning with randomized trees," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 29–42.
[41] Y. Ioannou, D. P. Robertson, D. Zikic, P. Kontschieder, J. Shotton, M. Brown, and A. Criminisi, "Decision forests, convolutional networks and the models in-between," *arXiv:1603.01250*, 2016.
[42] C. Lee, P. W. Gallagher, and Z. Tu, "Generalizing pooling functions in CNNs: Mixed, gated, and tree," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 863–875, Apr. 2018.
[43] A. Roy and S. Todorovic, "Monocular depth estimation using neural regression forest," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5506–5514.
[44] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.
[45] R. M. Neal and G. E. Hinton, *Learning in Graphical Models.* M. I. Jordan, Ed. Cambridge, MA, USA: MIT Press, 1999, ch. A View of the EM Algorithm That Justifies Incremental, Sparse, and Other Variants, pp. 355–368.
[46] K. Chen, S. Gong, T. Xiang, and C. L. Chen, "Cumulative attribute space for age and crowd density estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2467–2474.
[47] X. Wang, R. Guo, and C. Kambhamettu, "Deeply-learned feature for age estimation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2015, pp. 534–541.
[48] R. Rothe, R. Timofte, and L. V. Gool, "Some like it hot - visual guidance for preference prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5553–5561.
[49] G. Guo and G. Mu, "A framework for joint estimation of age, gender and ethnicity on a large database," *Image Vis. Comput.*, vol. 32, no. 10, pp. 761–770, 2014.
[50] D. Yi, Z. Lei, and S. Z. Li, "Age estimation by multi-scale convolutional network," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 144–158.
[51] S. Yan, H. Wang, X. Tang, and T. S. Huang, "Learning auto-structured regressor from uncertain nonnegative labels," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
[52] K. Chang, C. Chen, and Y. Hung, "Ordinal hyperplanes ranker with cost sensitivities for age estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 585–592.
[53] I. Huerta, C. Fernández, and A. Prati, "Facial age estimation through the fusion of texture and local appearance descriptors," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2014, pp. 667–681.
[54] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.
[56] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2001, pp. 511–518.
[57] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Proc. Eur. Conf. Comput. Vis.*, 1998, pp. 484–498.
[58] X. Geng, Z.-H. Zhou, and K. Smith-Miles, "Automatic age estimation based on facial aging patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2234–2240, Dec. 2007.
[59] Y. Zhang and D.-Y. Yeung, "Multi-task warped gaussian process for personalized age estimation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2622–2629.
[60] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung, "A ranking approach for human ages estimation based on face images," in *Proc. 20th Int. Conf. Pattern Recognit.*, 2010, pp. 3396–3399.
[61] G. Guo and G. Mu, "Joint estimation of age, gender and ethnicity: CCA vs. PLS," in *Proc. 10th IEEE Int. Conf. Workshops Autom. Face Gesture Recognit.*, 2013, pp. 1–6.
[62] K. Luu, K. Seshadri, M. Savvides, T. D. Bui, and C. Y. Suen, "Contourlet appearance model for facial age estimation," in *Proc. Int. Joint Conf. Biometrics*, 2011, pp. 1–8.
[63] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

**Wei Shen** received the BS and PhD degrees both in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007 and in 2012. From April 2011 to November 2011, he worked in Microsoft Research Asia as an intern. In 2012, he joined the School of Communication and Information Engineering, Shanghai University and served as an assistant and associate professor until Oct 2018. He is currently an assistant research professor with the Department of Computer Science, Johns Hopkins University. His current research interests include computer vision, deep learning and biomedical image analysis.

**Yilu Guo** received the BS degree in electronic and information engineering from Shanghai University, Shanghai, China, in 2016. Currently, he is working toward the master's degree in the School of Communication and Information Engineering, Shanghai University. His research interests include computer vision and deep learning.

**Yan Wang** received the BE degree in electronics and information engineering from the Huazhong University of Science and Technology, in 2011, and the PhD degree in electrical and electronic engineering from Nanyang Technological University, in 2017. She is currently a post-doctoral researcher with the Center for Imaging Science, the Johns Hopkins University. She was a research scholar at electrical and computer engineering at the Ohio State University from 2016 to 2017. Her research interests include computer vision and medical image analysis.

**Kai Zhao** received the BS and MS degrees both in communication and information engineering from Shanghai University, Shanghai, China, in 2014 and 2017. He is currently working toward the PhD degree in the College of Computer and Control Engineering, Nankai University. His research interests include computer vision and deep learning.



**Bo Wang** received the master's degree in computer science from the University of Toronto, Canada, in 2012 and the PhD degree in computer science from Stanford University, in 2017. He is a scientist with the Peter Munk Cardiac Center, University Health Network, and an assistant professor with the Dept. of Medical Biophysics, University of Toronto. He started his own laboratory in Toronto, in 2018. He is also named as one of the CIFAR AI chairs at Vector Institute, Toronto, Canada, in 2019. His current research interests include computer vision, computational biology and machine learning.

**Alan Yuille** received the BA degree in mathematics from the University of Cambridge, in 1976, and the PhD degree in theoretical physics at Cambridge, in 1980. He then held a postdoctoral position with the Physics Department, University of Texas at Austin, and the Institute for Theoretical Physics, Santa Barbara. He then became a research scientist with the Artificial Intelligence Laboratory, MIT (1982-1986) and followed this with a faculty position in the Division of Applied Sciences at Harvard (1986-1995), rising to the position of associate professor. From 1995-2002 he worked as a senior scientist with the Smith-Kettlewell Eye Research Institute in San Francisco. From 2002-2016 he was a full professor with the Department of Statistics, UCLA with joint appointments in Psychology, Computer Science, and Psychiatry. In 2016 he became a Bloomberg distinguished professor in cognitive science and computer science at Johns Hopkins University. He has won a Marr prize, a Helmholtz prize, and is a fellow of the IEEE.