Skin tone Diagnosis in the Wild: Towards More Robust and Inclusive User Experience Using Oriented Aleatoric Uncertainty

Emmanuel Malherbe[1], Michel Remise[2], Shuai Zhang[1], and Matthieu Perrot[1]

[1] L'Oréal AI Research name.surname@rd.loreal.com
[2] ANEO mremise@aneo.fr

Abstract. The past decade has seen major advances in deep learning models that are trained to predict a supervised label. However, estimating the uncertainty for a predicted value might provide great information beyond the prediction itself. To address this goal, using a probabilistic loss was proven efficient for aleatoric uncertainty, which aims at capturing noise originating from the observations. For multidimensional predictions, this estimated noise is generally a multivariate normal variable, characterized by a mean value and covariance matrix. While most of literature have focused on isotropic uncertainty, with diagonal covariance matrix, estimating full covariance brings additional information, such as the noise orientation in the output space.
We propose in this paper a specific decomposition of the covariance matrix that can be efficiently estimated by the neural network. From our experimental comparison to the existing approaches, our model offers the best trade-off between uncertainty orientation likeliness, model accuracy and computation costs. Our industrial application is skin color estimation based on a selfie picture, which is at the core of an online make-up assistant but is a sensitive topic due to ethics and fairness considerations. Thanks to oriented uncertainty, we can reduce this risk by detecting uncertain cases and proposing a simplified color correction bar, thus making user experience more robust and inclusive.

1 Introduction

Even if they are still likely to make wrong predictions, Deep Learning models are now state of the art for many problems [28]. More and more industrial applications are now based on such models, such as face verification for security by [43], autonomous driving by [39], or cancer detection by [6], with a certain degree of risk in case of wrong prediction. The risk can be also in terms of financial costs, for instance when [36] estimate construction prices or when [38] predicts user retention. Cosmetics and Beauty Tech industries also suffers from imperfect deep learning models, which provide mass personalization via smartphone applications [1, 20]. Within this context, we currently develop a model that estimates facial skin color from selfie pictures taken in the wild. This model is the core diagnosis for an online make-up assistant service, which for instance

recommends foundation shades to the user. Such application highly suffers from model errors, since a poor estimation may lead to a degraded personalization and a disappointed customer. Moreover, the skin color is an ethically sensitive feature to predict, because it is linked to the ethnicity. Such a model is thus at the core of AI fairness and inclusivity issue, as explained by [17]. The risk is very high for the provider of such service, both legally and for brand reputation.

These models being trained to minimize *on average* the errors on samples (e.g, MSE, cross-entropy, ...), errors are always likely to occur, even for an ideal training with no overfitting. This is true for training samples but even more for unseen data in-the-wild. Among the possible causes to such errors, a good part find roots during the training of the model. For instance, the quality of the training data-set can explain noisy predictions, typically when the coverage is not good enough and there are underrepresented zones in the train data. Having such zones is difficult to fully avoid, but may lead to fairness and ethic issues, for instance when these zones relate to ethnicity. Besides, some ground-truth labels can be imperfect, due to wrong manual annotation or noisy measurement device, so that the model might learn to reproduce this noise. The model capacity can also limit its capability to learn enough patterns on the training data-set. Last, some prediction errors find their only cause at inference time, typically because of poor input quality. For instance, in the case of pictures, estimation can be impacted by bad lighting conditions, blurry picture or improper framing.

Since it is impossible or very costly to avoid such errors, we would like to spot potentially wrong predictions by estimating their uncertainties. Modeling this uncertainty would provide tools to reduce or at least control the risks associated with high errors, for example by requesting a human validation in uncertain cases. Such validation would improve the overall user experience and make it more robust, fair and inclusive. For neural networks, the most common approaches for uncertainty modeling are aleatoric and epistemic, as explained by [23]. In practice, epistemic uncertainty relies on sampling multiple predictions by leveraging dropout randomness, while for aleatoric uncertainty, the model learns to predict from the input patterns a distribution of the prediction. We focus in this paper on the second approach, since [23] found it to be more relevant for real-time applications and large datasets with few outliers. Ideally, such uncertainty shall express the *orientation* in which the ground truth stands from the actual prediction, thus reducing the cost of manual labelling [41]. In the case of a smartphone application, such orientation enables the user to easily refine a poor prediction. More generally, in the context of Active Learning [14] or Active Acquisition [47], uncertainty's orientation helps to better target additional data annotation or acquisition.

To accurately estimate this oriented uncertainty in real-time, we propose a parameterization of a full covariance matrix. When compared to the state-of-the-art uncertainty methods, it offers the best trade-off between performance, color accuracy and uncertainty orientation. We propose in this paper the following contributions. First, we present an uncertainty model based on a specific decomposition of the covariance matrix efficiently predictable by a neural network.

Second, we propose to extract relevant information from this covariance, such as the uncertainty magnitude and orientation. Last, we performed experiments on the task of in-the-wild skin color estimation from selfie pictures, with several applications of uncertainty for the scenario of online make-up assistant.

2 Related work

Errors detection in Machine Learning Detecting and understanding errors has long been a hot topic for machine learning, partly due to the cost or risk induced by wrong predictions [21]. For error detection in classification, probabilistic models are natively providing a probability that indicates how sure the model is for the predicted class, as [13] described. Similarly, Gaussian Processes are popular for regression problems and natively provide a variance for each prediction, as depicted by [45]. On the other hand, non-probabilistic classification models only predict raw scores, which take values of any magnitude and are thus hard to interpret for error detection. This score can however be converted to a probability, as [34] proposed for binary Support Vector Machines classifier, that was extended for multi-class models by [46]. [30] proposes a posterior probability estimation for the best outcome of a ranking system, with industrial applications in Natural Language Processing. These approaches all rely on a cross-validation performed during the training, in order to calibrate the scores conversion on unseen samples. Despite their efficacy, these posterior probability estimations remain conversion of scores in a discrete output space - classes, recommendation objects - and do not apply to regression tasks.

Uncertainty in Deep Learning More recently, [28] described how Deep Learning introduced models with higher representation capabilities, which can be leveraged to estimate an uncertainty of the prediction. As [14] explains, a first approach is to consider the model as Bayesian, whose weights follow a random distribution obtained after the training. This uncertainty is denoted epistemic, as by [22, 8]. In practice, [15] proposed to approximate it with one model by performing multiple predictions with stochastic drop-out, without changing the training procedure. Given an input, it simulates thus a Monte Carlo sampling among the possible models induced by dropout. This iteration leads to much higher inference time, that [35] proposes to reduce by approximating the sampling with analytical formulas, starting from dropout underlying Bernoulli distribution. They get thus reduced run times, while results tend to be similar to slightly worse than with sampling. [8] describes another approach denoted as aleatoric uncertainty, where the model learns to predict the uncertainty from the input. To do so, the model is trained to predict a probability distribution instead of the ground truth only, as formalized by [22]. In practice, this distribution is generally Gaussian, which captures the most likely output value and the covariance around it. According to [23], aleatoric is more relevant than epistemic in the case of large datasets as well as real-time applications.

Oriented Aleatoric Uncertainty Methods While previous cited approaches focused on a single-valued aleatoric uncertainty, [33] and [10] proposed to predict the full covariance matrix using Cholesky decomposition, proposed by [5]. The matrix is built as $\hat{L}^T\hat{L}$ where $\hat{L}$ is predicted as a lower triangular matrix with positive elements in its diagonal. However, they do not extract nor leverage any *orientation* information induced by the covariance, that we focus on in this paper. One reason is that their output spaces are high dimensional, where the orientation is hard to interpret and exploit. In 3D space, [37] obtained promising results by combining aleatoric uncertainty and Kalman filter for tracking object location in a video. However, the covariance decomposition they propose for pure aleatoric uncertainty model is only valid for 2 dimensional output space. They only rely on their final activation functions to reduce risk of non-positive definite matrix, which would not work on any data-set. More recently, [29] proposes a full rank aleatoric uncertainty to visualize detected keypoints areas in the 2D image. This interesting usage of uncertainty is limited to 2D in practice since each keypoint has its own uncertainty area.

3 Problem

3.1 Color Estimation: a Continuous 3D Output

We consider as our real world use-case the problem of skin color estimation from a selfie picture. In this scenario, the user takes a natural picture, from any smartphone and under unknown lighting condition. This picture's pixels are represented in standard RGB, the output color space for most smartphones. We want to estimate the user's skin color as a real color measured by a device, and not a self-declared skin type nor a-posteriori manual annotation. To do so, we consider the skin color measured by a spectrocolorimeter, whose spectrum is converted into the L*a*b* color space as defined by the CIE (Commission Internationale de l'Eclairage), as done by [44]. Compared to the hardware-oriented standard RGB space, this 3 dimensional space is built so that the Euclidean distance between two colors approaches the human perceived difference. The ground truth y is thus represented as 3 continuous values, and the mean squared error $\mathcal{L}_{\text{MSE}}$ approximates the perceived difference between colors y and $\hat{y}$:

$$y = (L^*, a^*, b^*)^T \in \mathbb{R}^3, \qquad \mathcal{L}_{\text{MSE}}(y, \hat{y}) = \|y - \hat{y}\|^2 \tag{1}$$

where $\hat{y}$ is the model prediction for input picture x and $\|.\|$ denotes the $L2$ norm.

Color and skin tone can be efficiently estimated by regression Convolutional Neural Network, as done by [3, 7, 31, 27, 26]. Beyond predicting y, we focus in this paper in estimating the oriented uncertainty, as described in the next part.

3.2 Oriented Uncertainty

Following the prediction space described above, we now discuss what form of uncertainty could be predicted. The simplest form would be a simple real value

estimating the *magnitude* of the prediction error, measuring thus our level of uncertainty. This enables to apply a threshold on this estimated value for filtering unsure cases and has been widely studied in literature (see Section 2). However, for the case of multi-dimensional predictions, a single-value uncertainty treats each output dimension equally, in an *isotropic* manner. We focus instead on a full rank uncertainty, which is *oriented* since it expresses the most likely orientation of prediction errors.

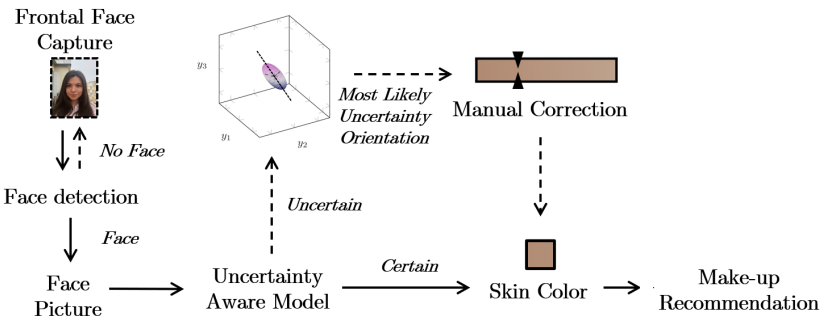

Fig. 1. Pipeline for our color estimation use-case, where the uncertainty is used for filtering uncertain predictions and defining 1-D color bar correction.

The epistemic uncertainty can also provide orientation information, by providing points cloud in the prediction space (see Section 2). However, we focus in this paper on aleatoric uncertainty, for the following reasons. First, [23] advised it for large datasets, and when there is a need for *real-time* application. Second, epistemic uncertainty is mostly advised for detecting inputs out of training data distribution, because model presents higher variability for such data. This appears useless in the case of selfie pictures as input, since face detectors (as the one of [24]) easily ensure to detect outliers before feeding the neural network. In the following, we detail how our model estimates oriented aleatoric uncertainty.

4 Model

4.1 Aleatoric Uncertainty Principle

The aleatoric uncertainty approach is to consider that the model no longer predicts a single value $\hat{y}$ but instead a distribution of random value. In practice, we assume this distribution to be a multivariate normal law, meaning $y \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$ where $\hat{\mu}$ and $\hat{\Sigma}$ are typically estimated by a neural network from the input. The loss to be optimized by our model then relates to the likelihood of the ground truth y with respect to this distribution, written $p_{\hat{\mu},\hat{\Sigma}}(y)$. For a multi-dimensional

output space, $y \in \mathbb{R}^d$, $\hat{\mu} \in \mathbb{R}^d$, $\hat{\Sigma} \in \mathbb{R}^{d \times d}$, and the likelihood is written:

$$p_{\hat{\mu},\hat{\Sigma}}(y) = \frac{1}{(2\pi)^{d/2}|\hat{\Sigma}|^{1/2}} e^{-\frac{1}{2}(y-\hat{\mu})^T \hat{\Sigma}^{-1}(y-\hat{\mu})} \tag{2}$$

where $|\hat{\Sigma}|$ is the determinant of matrix $\hat{\Sigma}$. Practically we minimize an affine transformation of the log-likelihood $log(p_{\hat{\mu},\hat{\Sigma}}(y))$, discarding the constant terms:

$$\mathcal{L}_p(y, \hat{\mu}, \hat{\Sigma}) = (y - \hat{\mu})^T \, \hat{\Sigma}^{-1} \, (y - \hat{\mu}) + \log\left(|\hat{\Sigma}|\right) \tag{3}$$

This loss is similar to the MSE (Equation 1) where the model still learns to predict the most likely value $\hat{\mu}$, which is equivalent to $\hat{y}$. Besides, it learns to predict a matrix $\hat{\Sigma}$ that represents a rich form of uncertainty.

A straightforward choice for covariance $\hat{\Sigma}$ is to consider isotropic noise, like in [23]. In this case, $\hat{\Sigma} = \hat{\sigma}^2 I_d$, where $\hat{\sigma} \in \mathbb{R}+$ is the estimated standard deviation in every direction of the output space. This corresponds to assuming $y_j \sim \mathcal{N}(\hat{\mu}_j, \hat{\sigma}^2)\ \forall j = 1..d$ and only provides information on uncertainty magnitude. In the following, we focus on a richer representation of $\hat{\Sigma}$.

4.2 Covariance for Oriented Uncertainty

To capture the uncertainty in any direction, we need to estimate the covariance matrix $\hat{\Sigma}$ as a symmetric positive definite matrix (SPD), meaning it verifies:

$$\hat{\Sigma}^T = \hat{\Sigma} \quad \text{and} \quad v^T \hat{\Sigma} v > 0 \quad \forall v \neq \vec{0} \in \mathbb{R}^d$$

While the symmetry of $\hat{\Sigma}^T$ can be easily ensured by construction, the positive semi-definiteness property is not straightforward to satisfy when $\hat{\Sigma}$ is the output of an uncontrolled neural network. To build $\hat{\Sigma}$, [33] and [10] propose to use Cholesky decomposition or equivalently LDL decomposition $\hat{\Sigma} = \hat{L}\hat{D}\hat{L}^T$, where $\hat{L}$ is a lower unit triangular matrix and $\hat{D}$ a diagonal matrix (as formalized by [19]). The independent components $\hat{D}_{i,i} > 0$ and $\hat{L}_{i,j}$ $(i < j)$ are produced by a regression layer. However, we propose not to use such decomposition. Indeed, we observed an difficult optimization of such model, and poor results for the main prediction task. We explain this phenomenon by the intuition that $\hat{L}_{i,j}$, are *low lever features* in the sense that they are hard to interpret, contrary to standard deviations $\hat{\sigma}_j$ for instance. Our main proof remains experimental and further theoretical explanation goes beyond the scope of this paper.

4.3 Euler Angles Decomposition

To build $\hat{\Sigma}$, we instead rely on the following decomposition of SPD:

$$\hat{\Sigma} = \hat{U}\hat{D}\hat{U}^T \quad \text{with } \hat{D} = Diag(\hat{\sigma}_1^2, \ldots, \hat{\sigma}_d^2)$$

where $\hat{U} \in \mathbb{R}^{d \times d}$ is a unitary matrix with columns being eigenvectors of $\hat{\Sigma}$ and $\hat{\sigma}_1^2, \ldots, \hat{\sigma}_d^2 \in \mathbb{R}^+$ are the corresponding eigenvalues. Each eigenvector $\hat{U}_j \in \mathbb{R}^d$

is an uncertainty orientation associated with a standard deviation σ_j, so that $\hat{\Sigma}$ geometrically corresponds ellipsoidal level sets of the distribution.

To simplify the notations and place ourselves in the color output space, we now consider the 3D space for y ($d = 3$). We propose to express $\hat{U}$ as the multiplication of the rotation matrices around each canonical axis [42]:

$$\hat{U} = R_{y_1}(\hat{\theta}_1)R_{y_2}(\hat{\theta}_2)R_{y_3}(\hat{\theta}_3) \in \mathbb{R}^{3\times 3}$$

where $R_{y_j}(\hat{\theta}_j)$ are the rotation matrices around axis y_j and $\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3 \in [-\pi/4, \pi/4]$ are the rotation angles respectively around y_1, y_2 and y_3 axes, also named Euler angles (see appendix for explicit matrices expressions). This choice of representation enables the model to predict high-level interpretable features, meaning the Euler angles $\hat{\theta}_i$. Euler angles representation usually suffer from periodicity and discontinuity problems, as [9] points out. However, in our problem of covariance matrix $\hat{\Sigma}$ construction, we can enforce a narrow range for the $\hat{\theta}_i$. Indeed, having $\hat{\theta}_i \in [-\pi/4, \pi/4]$ ensures that $\hat{\Sigma}$ covers the whole SPD space, while keeping each $\hat{\sigma}_i$ closely associated to the canonical axe y_i, thus easing the optimization. Without such boundaries, one notes that $\hat{\theta} = (\pi/2, 0, 0),\ \hat{\sigma} = (1, 1, 2)$ and $\hat{\theta} = (0, 0, 0),\ \hat{\sigma} = (1, 2, 1)$ would give the same $\hat{\Sigma}$, which makes optimization difficult since $\hat{\sigma}_i$ values are shifted. While we focus on the 3 dimensional case, this parameterization could be extended to higher dimensions [40].

In practice, to compute the loss of Equation 3 we build $|\hat{\Sigma}|$ and $\hat{\Sigma}^{-1}$ as:

$$|\hat{\Sigma}| = \prod_{j=1}^{d} \hat{\sigma}_j^2, \qquad \hat{\Sigma}^{-1} = \hat{U}Diag(\hat{\sigma}_1^{-2}, \ldots, \hat{\sigma}_d^{-2})\hat{U}^T$$

which are differentiable, so that the loss is minimizable by gradient descent.

To enforce the boundaries on each $\hat{\theta}_i$, we preferred not to use strict clipping of neurons values. Indeed, doing so was leading to numerical issues with vanishing gradient for the $\hat{\theta}_i$ regressions. Instead, we let $\hat{\theta}_i$ as raw outputs of linear regressions, and included a penalization term in the loss:

$$\mathcal{L}_\theta(\hat{\theta}) = \lambda_\theta \sum_{j=1}^{d} \max(0, \hat{\theta}_i - \pi/4) + \max(0, -\hat{\theta}_i - \pi/4) \tag{4}$$

where $\lambda_\theta > 0$ is a hyperparameter. This penalization corresponds to soft boundary constraints, meaning $\hat{\theta}_i$ can take values beyond $\pi/4$ or below $-\pi/4$, but then the loss is increased. The total loss function takes the form:

$$\mathcal{L}(y, \hat{\mu}, \hat{\Sigma}, \hat{\theta}) = \mathcal{L}_p(y, \hat{\mu}, \hat{\Sigma}) + \mathcal{L}_\theta(\hat{\theta})$$

4.4 Oriented Uncertainty Benefits

We now propose to extract information from the matrix $\hat{\Sigma}$ predicted on a new picture. First, we can interpret an uncertainty magnitude. In case of isotropic

uncertainty $\hat{\Sigma} = \hat{\sigma}^2 I_d$, the magnitude is directly given by $\hat{\sigma}$. [14] showed that by applying a threshold on $\hat{\sigma}$, we filter out the most unsure cases. We extend this to oriented uncertainty by considering the determinant of the covariance matrix:

$$|\hat{\Sigma}| = \prod_j \hat{\sigma}_j^2 \in \mathbb{R} \tag{5}$$

We can also extract the most likely orientation of y with respect to predicted $\hat{y}$:

$$\hat{v} = \hat{U}_{j^*} \in \mathbb{R}^d \text{ where } j^* = \underset{j}{\text{argmax}}\ \hat{\sigma}_j \tag{6}$$

which is of norm 1. One notes that error is equally likely to stand in the orientation $\hat{v}$ and $-\hat{v}$ due to the symmetry of the normal distribution.

4.5 Probabilities Re-Scaling for Models Comparison

We now describe an optional step in the training, that do not serve the general purpose of the model and has no impact on $|\hat{\Sigma}|$ and $\hat{v}$ computation. We use it to get unbiased likelihood in order to compare models in our experiments.

When looking at $p_{\hat{\mu},\hat{\Sigma}}(y)$ values (Equation 2) on the test samples during the optimization process, we observed that uncertainty models tend to show *overconfidence*, meaning they predict $\hat{\Sigma}$ with lower volume through the epochs. This phenomenon of overconfidence of neural network has regularly be observed [4], for instance with probabilities inferred after softmax always close to 0% or 100% (see Figure 8 in Appendix for an illustration). To avoid this, we propose to multiply every predicted covariance $\hat{\Sigma}$ by a unique factor $\beta \in \mathbb{R}^+$, in order to adapt the magnitudes of the distributions $\mathcal{N}(\hat{\mu}, \hat{\Sigma})$ while keeping their shape and orientation. Such factor does not change the relative order between the magnitudes $|\hat{\Sigma}|$ among samples, typically when filtering most uncertain cases. We propose to consider these optimal β^* value as:

$$\beta^* = \text{argmax}_\beta \frac{1}{N} \sum_i^N \log\left(p_{\hat{\mu}_i,\beta\hat{\Sigma}_i}(y_i)\right) \tag{7}$$

where the sum covers a *subpart* of the train data, while we estimate all $\hat{\mu}_i$ and $\hat{\Sigma}_i$ by a model trained on the remaining of the train data. This sub-training is necessary for computing $p_{\hat{\mu}_i,\beta\hat{\Sigma}_i}(y_i)$ from unbiased inputs. Such re-scaling is similar to scores conversion for probabilistic SVM [34], where a cross-validation is performed on the training data to learn the conversion. Due to the computation costs, we preferred to split the training data into 2 equal parts reflecting the validation strategy such as group-out or stratification. We can then re-scale the estimated covariances as $\beta^*\hat{\Sigma}$ when computing metrics and comparing models.

5 Experiments

5.1 Dataset

Our dataset is composed of various selfie pictures taken by different people with their own smartphone in indoor and outdoor environments. Besides, they had

their skin color measured in a controlled environment using a spectrophotometer under a specific protocol to reduce measurement noise. In order to face real-world skins diversity and include various race groups, we performed several acquisitions in different countries (see Table 1). For standardization purpose, we pre-processed all pictures by detecting facial landmarks using the face detector of [24] and then placing eyes in a standardized location. Resulting images are thus centered on the face and resized to 128×128. This pre-processing step is identically done at inference time in our real-world application (see Figure 1).

5.2 Evaluation and Implementation Details

To compute all results of this sections, we performed a 5-fold cross validation on our dataset, and evaluated the predictions on the successive test folds. To avoid biased predictions, each volunteer pictures are grouped in the same fold. Furthermore, we stratified the folds with respect to y_1 value, that corresponds to L^* in the color space, that can be interpreted as skin intensity.

Using this process, we compared the following models:

- Regression w/o Uncertainty: pure regression CNN (Section 3.1, [27])
- Aleatoric Isotropic: aleatoric single-valued uncertainty, as proposed by [23]
- Aleatoric (Cholesky): the aleatoric uncertainty model using Cholesky decomposition, as proposed by [10]
- Epistemic via Dropout: epistemic uncertainty model proposed by [15] for 100 sampled predictions
- Epistemic Sampling-free: epistemic uncertainty model with direct covariance estimation proposed by [35]
- Aleatoric (Ours): the uncertainty model described in Section 4.3

For each model, we used the same architecture for the convolutional network, 4 convolutions blocks with skip connections followed by a dropout layer (similar to ResNet [18]). Only the loss and final dense layers have different architecture between models. We used ReLu activations except for oriented uncertainty in which $\hat{\theta}_j$ are linear output of regression. The corresponding soft constraint term (Equation 4) is scaled by $\lambda_\theta = 10$. For aleatoric uncertainty models, the lower bound for $\hat{\sigma}_j$ is set as 10^{-2} to avoid numerical overflow. Contrary to [23], we prefer our model to directly estimate $\hat{\sigma}_j$ instead of $\log(\hat{\sigma}_j)$, for avoiding weights updates to produce too high variation on $\hat{\sigma}_j$. For all models, we used the optimization procedure of [25] on batches of size 16 and a learning rate 10^{-4} during 400 epochs. We implemented our networks and layers using TensorFlow [2].

Table 1. Description of the data used for our experiments. The full data-set is still under collection and will cover more countries.

Country	China	France	India	Japan	USA	Total
# People	53	249	22	32	832	1188
# Pictures	936	3967	368	144	10713	16128

5.3 Metrics for Regression and Uncertainty

In order to evaluate the core regression task, for each sample of the test folds we computed the ΔE^* [44] between the prediction and the ground-truth y:

$$\Delta E^* = \|\hat{\mu} - y\| \tag{8}$$

where $\hat{\mu}$ is replaced by $\hat{y}$ for the regression model.

For uncertainty models, we evaluated the quality of predicted distributions $\mathcal{N}(\hat{\mu}, \hat{\Sigma})$ when compared to the ground truth y. For epistemic uncertainty via dropout, we considered a normal distribution of covariance $\hat{\Sigma}$ computed from sampled predictions. For each model, we computed the scaling factor β^* (Equation 7, Section 4.5) and considered the average likelihood after scaling:

$$\langle p^*_\beta \rangle = \frac{1}{N} \sum_i^N p_{\hat{\mu}_i, \beta^* \hat{\Sigma}_i}(y_i), \qquad \langle \log(p^*_\beta) \rangle = \frac{1}{N} \sum_i^N \log \left(p_{\hat{\mu}_i, \beta^* \hat{\Sigma}_i}(y_i) \right) \tag{9}$$

For orientation-aware models, we computed the angle error α between the orientation $\hat{v}$ (Equation 6) and the actual ground truth y orientation from $\hat{y}$:

$$\alpha = cos^{-1} \left(\hat{v}. \frac{(y - \hat{\mu})}{\|y - \hat{\mu}\|} \right)$$

To assess the quality of estimated errors distributions, we also considered the probability of having an error ΔE below a certain threshold E:

$$\mathbb{P}_{\hat{\mu}, \hat{\Sigma}}(\|\hat{\mu} - y\| < E) = \iiint_{\|e\|^2 < E^2} p_{\hat{\mu}, \hat{\Sigma}}(\hat{\mu} + e) \; d^3e \tag{10}$$

where the integral does not depend on $\hat{\mu}$ (see equation 2). This quantity differs from $|\hat{\Sigma}|$ by having a clear probabilistic interpretation, but is slower to compute. We compared these numerically computed values on test samples to the actual

Table 2. Comparisons of the models. For each metrics, ↑ indicates when higher is better, and ↓ indicates when lower is better. For accuracy metrics, best result is highlighted in italics bold while second best is in bold.

Model	Performance	Regression task		Uncertainty Metrics		
	Inference time (ms) ↓	$\hat{\mu}$ accuracy $\langle \Delta E^* \rangle \downarrow$	vs baseline ↓	Likelihood $\langle log\,(p_{\beta^*}) \rangle \uparrow$	ROC-AUC $\Delta E < 1 \uparrow$	Angle $\langle \alpha \rangle \downarrow$
No uncertainty	68.1 ± 6.1	***3.54***	*baseline*	-	-	-
Aleat. isotropic	70.2 ± 7.3	3.59	+1.5%	−7.86	58.47	-
Aleat. Cholesky	73.2 ± 4.2	4.49	+26.8%	***−7.02***	***62.74***	***36.85°***
Epist. w dropout	7820 ± 19	***3.54***	+0%	−9.07	58.15	40.88°
Epist sample-free	81.2 ± 5.3	***3.54***	+0%	−9.08	57.70	41.55°
Aleat. (ours)	71.4 ± 4.7	3.65	+3.1%	**−7.33**	**59.09**	**39.43°**

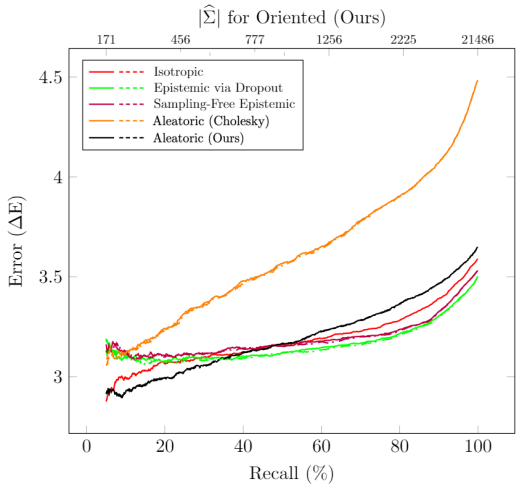

Fig. 2. ΔE with respect to recall for samples with $|\hat{\Sigma}|$ below a threshold (Equation 5). For plain curves, successive thresholds are applied on samples $|\hat{\Sigma}|$ values, for dashed curves on $\mathbb{P}_{\hat{\mu},\hat{\Sigma}}(\|\hat{\mu} - y\| < E)$ values. On top, indicative $|\hat{\Sigma}|$ values for our model are shown. Points with recall below 5%, noiser, are omitted.

realization of $\|\hat{\mu} - y\| < E$ to get the ROC-AUC [11]. We choose as threshold $E = 1$ which corresponds to human perception of color dissimilarities [44].

Table 2 shows the computed metrics. As expected, the color estimation accuracy is unchanged for epistemic uncertainty models. The difference is very little for our model, while the Cholesky uncertainty shows 26% *higher average error*. According to our intuition described in Section 4.1, this degradation seems to be due to the complex optimization process, since $\hat{\Sigma}$ representation relies on low level features (see Figure 7 in Appendix for an illustration). Besides, estimated distributions appears better for the Cholesky, while second best metrics are for our oriented uncertainty model. Using our model, the angle error is reduced by 2° compared to the sampling-free epistemic, and the log-likelihood is significantly higher. Compared to other models, our method gives thus the best trade-off with maintained accuracy, fast computation and accurate uncertainty distribution.

5.4 Uncertainty for Samples Selection

A first benefit of uncertainty-aware models is to detect samples whose error ΔE is likely to be high by applying a threshold on predicted $|\hat{\Sigma}|$ (Equation 5). In our pipeline shown in Figure 1, this conditions the manual color correction step for the uncertain cases. In order to evaluate how efficient this condition is, we applied for each model successive thresholds on test samples $|\hat{\Sigma}|$. In practice, for every threshold T we compute the recall for selected samples as $\frac{1}{N}\sum_i^N \mathbb{1}(|\hat{\Sigma}_i| < T)$, where $\mathbb{1}$ denotes the indicator function and $\hat{\Sigma}_i$ is the i-th test sample estimated covariance. The average ΔE error is similarly computed for samples verifying

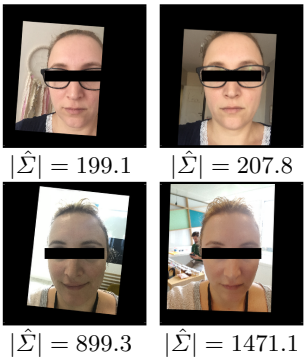

Fig. 3. Pictures with lowest $|\hat{\Sigma}|$ value (top) and highest $|\hat{\Sigma}|$ value (bottom) from same person (/16 pictures).

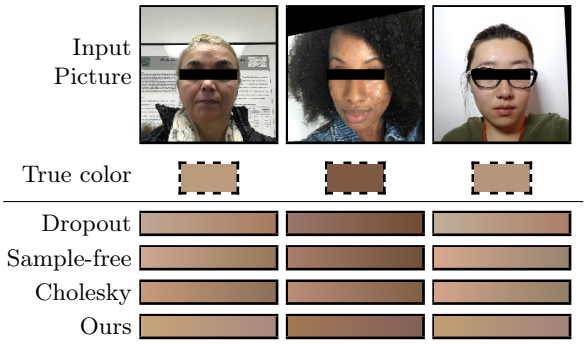

Fig. 4. Examples of input pictures with the 1D correction color bars estimated by each uncertainty model. The bar from Cholesky model suffers from the higher error of the estimated $\hat{\mu}$.

$|\hat{\Sigma}_i| < T$. Plain curves in Figure 2 show both quantities when computed for every threshold T values. This is similar to a precision-recall curves for binary classification, as described by [12], that we extend to regression tasks using estimated $|\hat{\Sigma}|$ (as done by [30, 16]). Equivalently, we computed the dashed line curves by selecting samples verifying $\mathbb{P}_{\hat{\mu},\hat{\Sigma}}(\|\hat{\mu} - y\| < E) > T$. Selection based on this second quantity requires heavier computation (Equation 10 versus Equation 5) and is thus not convenient for real-time application.

We see in Figure 2 that most curves are very close for all models, except for the Cholesky model. The accuracy error for Cholesky uncertainty is indeed much higher under all thresholds, which was expected from overall accuracy (Table 2). Besides, we see that for the most certain test samples (for recall $\leq 20\%$), our model indeed selects the samples with *actual* lowest errors. Last, dashed and plain curves are hard to distinguish, which means that thresholding on $|\hat{\Sigma}|$ is a good approximation for thresholding on $\mathbb{P}_{\hat{\mu},\hat{\Sigma}}(\|\hat{\mu} - y\| < E)$.

For a given person, we also looked at rejected and valid pictures based on the $|\hat{\Sigma}|$ criteria (see Figure 3). In general, uncertainty is higher when lighting conditions are bad, such as yellow light, strong back light and shadowed face. These patterns seems to be leveraged by the model for estimating the uncertainty, and are indeed strongly impacting the skin color estimation task.

5.5 Color Control using Uncertainty Orientation

For our use case, another benefit of oriented uncertainty is to enable a better user experience for manual color correction. This correction is requested for users with $|\hat{\Sigma}|$ falling above the operational threshold. Requesting a non-expert user to re-define a color in 3D is practically impossible. Based on the low angle error α obtained, we propose to use the most likely orientation $\hat{v}$ (Equation 6) as the

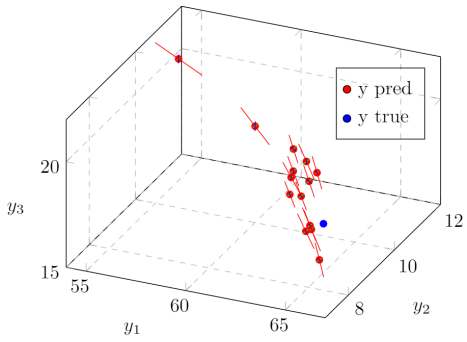

Fig. 5. Predictions and uncertainties for all pictures of a given panelist. We see that uncertainties are generally oriented towards the ground truth (in blue), and that further predictions have higher uncertainty in the main axis (in red).

only degree of freedom for the color control. To confirm this intuition, Figure 5 shows an example of $\hat{v}$ orientations estimated for all pictures of a single volunteer: $\hat{v}$ is generally oriented towards the real color. Figure 4 shows practical examples of color bars. The bars are centered on estimated color $\hat{\mu}$, directed towards $\hat{v}$. The bar length was fixed at $\Delta E^* = 5$ in the color space, after discussion with user experience experts in order to have a standard control bar expressiveness.

5.6 Uncertainty Orientation for Foundation Recommendation

We present here preliminary results that illustrate additional benefits of our oriented uncertainty for online make-up assistant service. For the sub-set of our data-set from acquisition in Japan (see Table 1), a make-up artist have assessed the most suitable foundation shade chosen in a range of 25 shades. We also measured the color f in L*a*b* space for each shade of the range, using same spectrophotometer and comparable protocol than for the skin color. For each participant, we note $f^* \in \mathbb{R}^3$ the color of the best foundation shade chosen by make-up artist. Based on our make-up knowledge, this best foundation should be the closest from the skin tone, meaning the predicted $\hat{\mu}$ color should be close from this best shade color f^*. Based on this, we computed for each picture the difference between the estimated skin color and this foundation color and as a $\Delta E^*_{f^*} = \|\hat{\mu} - f^*\|^2$, as well as the probability distribution value $p_{\hat{\mu},\hat{\Sigma}}(f)$ (Equation 2) for every shade color f of the range. Using those 25 values, we could thus rank the shades in a scenario of product recommendation. For the regression model, we similarly ranked the shades using ΔE^* which can be considered the best products recommendation when not using uncertainty. We display in Table 3 the average rank of the best foundation shade according to make-up artist, as well as the average ΔE^* and $log(p_{\hat{\mu},\beta^*\hat{\Sigma}}(f^*))$ values for this shade color:

$$\langle \log(p^*_\beta(f^*)) \rangle = \frac{1}{N} \sum_i^N \log\left(p_{\hat{\mu}_i,\beta^*\hat{\Sigma}_i}({f_i}^*)\right)$$

Table 3. Metrics on Japan data for foundation recommendation using uncertainty orientation. Reference is ideal foundation color (instead of panelist skintone for Table 2). Best result is highlighted in italics bold, second best in bold.

Model	$\langle \Delta E^*_{f^*} \rangle$	$\langle \log(p^*_\beta(f^*)) \rangle$	Shade Rank
Regression	**3.20**	-	$2.43^{th}/25$
Aleat. Isotropic	3.22	-7.63	$2.54^{th}/25$
Aleat. Cholesky	4.34	**-7.43**	$3.34^{th}/25$
Epist. w Dropout	**3.20**	-12.25	$\mathbf{2.30}^{th}/25$
Epist. Sample-Free	**3.20**	-11.27	$2.82^{th}/25$
Aleat. Ours	3.37	***-7.26***	$\boldsymbol{2.20}^{th}/25$

where scaling factor β^* do not impact the products ranking but helps for models comparison. We get the best rank for our model, even if the raw skin to foundation distance $\Delta E^*_{f^*}$ is higher than with some other models. We emphasize that these results do not include any manual color correction as described in Section 5.5. Details for the products recommendation go beyond the scope of this paper.

6 Conclusion

In this paper, we proposed to estimate multivariate aleatoric uncertainty by using a different parameterization of the covariance matrix based on Euler angles. We experimented our model on a real-world data-set, which addresses skin color estimation from selfie pictures. This use-case is at the core of a make-up online assistant but is very sensitive in terms of ethics and AI fairness. The uncertainty estimation is an answer to reduce ethic risks, among other benefits it brings. Our oriented uncertainty model showed a similar accuracy to pure regression model, contrary to the model using Cholesky decomposition which got 26.8% higher errors on the core diagnosis task. Furthermore, when comparing to other approaches, our model obtained the best metrics about the estimated distributions, such as the angle error for the most likely error orientation. This shows its ability to infer the scale and orientation of the actual prediction error.

The proposed model can be used for real-time color adjustment. Users with uncertain predictions are requested to make manual correction via a simplified UX with 1D color bar whose orientation is given by the uncertainty. Besides, we experimented another benefit of our model in the case of foundation recommendation, where the orientation helps to recommend the best product. In our future work, we will evaluate the benefits of the uncertainty bar for the end user. To do so, we are currently conducting a study were panelist are asked to correct their diagnosed skin tone using the uncertainty-aware color bar. Beyond, we plan to use our uncertainty model as domain discriminator of a conditional generative adversarial network [32], in order to less penalize generated pictures whose predicted label lies in the most likely orientation of uncertainty.

References

1. Aarabi, P.: Method, system and computer program product for generating recommendations for products and treatments (Sep 12 2017), uS Patent 9,760,935
2. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). pp. 265–283 (2016)
3. Bokaris, P.A., Malherbe, E., Wasserman, T., Haddad, M., Perrot, M.: Hair tone estimation at roots via imaging device with embedded deep learning. Electronic Imaging **2019**(6), 483–1 (2019)
4. Bulatov, K.B., Polevoy, D.V., Mladenov, V., Spasov, G., Georgieva, P., Petrova, G.: Reducing overconfidence in neural networks by dynamic variation of recognizer relevance. In: ECMS. pp. 488–491 (2015)
5. Cholesky, A.L.: Sur la résolution numérique des systèmes d'équations linéaires. Bulletin de la Sabix. Société des amis de la Bibliothèque et de l'Histoire de l'École polytechnique (39), 81–95 (2005)
6. Coudray, N., Ocampo, P.S., Sakellaropoulos, T., Narula, N., Snuderl, M., Fenyö, D., Moreira, A.L., Razavian, N., Tsirigos, A.: Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. Nature medicine **24**(10), 1559 (2018)
7. Das, A., Dantcheva, A., Bremond, F.: Mitigating bias in gender, age and ethnicity classification: a multi-task convolution neural network approach. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 0–0 (2018)
8. Der Kiureghian, A., Ditlevsen, O.: Aleatory or epistemic? does it matter? Structural Safety **31**(2), 105–112 (2009)
9. Diebel, J.: Representing attitude: Euler angles, unit quaternions, and rotation vectors. Matrix **58**(15-16), 1–35 (2006)
10. Dorta, G., Vicente, S., Agapito, L., Campbell, N.D., Simpson, I.: Structured uncertainty prediction networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5477–5485 (2018)
11. Fawcett, T.: An introduction to roc analysis. Pattern recognition letters **27**(8), 861–874 (2006)
12. Flach, P., Kull, M.: Precision-recall-gain curves: Pr analysis done right. In: Advances in neural information processing systems. pp. 838–846 (2015)
13. Friedman, J., Hastie, T., Tibshirani, R.: The elements of statistical learning, vol. 1. Springer series in statistics New York (2001)
14. Gal, Y.: Uncertainty in deep learning. Ph.D. thesis, PhD thesis, University of Cambridge (2016)
15. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059 (2016)
16. Gurevich, P., Stuke, H.: Learning uncertainty in regression tasks by artificial neural networks. arXiv preprint arXiv:1707.07287 (2017)
17. Hazirbas, C., Bitton, J., Dolhansky, B., Pan, J., Gordo, A., Canton Ferrer, C.: Towards measuring fairness in AI: the Casual Conversations dataset. arXiv e-prints arXiv:2104.02821 (Apr 2021)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 770–

778. IEEE Computer Society (2016). https://doi.org/10.1109/CVPR.2016.90, https://doi.org/10.1109/CVPR.2016.90
19. Higham, N.J.: Analysis of the Cholesky decomposition of a semi-definite matrix. Oxford University Press (1990)
20. Holder, C.J., Obara, B., Ricketts, S.: Visual siamese clustering for cosmetic product recommendation. In: Asian Conference on Computer Vision. pp. 510–522. Springer (2018)
21. Holzinger, A.: From machine learning to explainable ai. In: 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA). pp. 55–66. IEEE (2018)
22. Kendall, A., Cipolla, R.: Modelling uncertainty in deep learning for camera relocalization. In: 2016 IEEE international conference on Robotics and Automation (ICRA). pp. 4762–4769. IEEE (2016)
23. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in neural information processing systems. pp. 5574–5584 (2017)
24. King, D.E.: Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research **10**(Jul), 1755–1758 (2009)
25. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
26. Kips, R., Tran, L., Shakhmametova, A., Malherbe, E., Askenazi, B., Perrot, M.: Toward online foundation shade recommendation: Skin color estimation from smartphone images through deep learning. In: Proceedings of the 31st IFSCC Congress, Yokohama, Japan (2020)
27. Kips, R., Tran, L., Malherbe, E., Perrot, M.: Beyond color correction : Skin color estimation in the wild through deep learning. Electronic Imaging **2020** (2020)
28. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436 (2015)
29. Lu, C., Koniusz, P.: Few-shot keypoint detection with uncertainty learning for unseen species. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19416–19426 (2022)
30. Malherbe, E., Vanrompay, Y., Aufaure, M.A.: From a ranking system to a confidence aware semi-automatic classifier. Procedia Computer Science **60**, 73–82 (2015)
31. Masood, S., Gupta, S., Wajid, A., Gupta, S., Ahmed, M.: Prediction of human ethnicity from facial images using neural networks. In: Data Engineering and Intelligent Computing, pp. 217–226. Springer (2018)
32. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
33. Peretroukhin, V., Wagstaff, B., Kelly, J.: Deep probabilistic regression of elements of so (3) using quaternion averaging and uncertainty injection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 83–86 (2019)
34. Platt, J., et al.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers **10**(3), 61–74 (1999)
35. Postels, J., Ferroni, F., Coskun, H., Navab, N., Tombari, F.: Sampling-free epistemic uncertainty estimation using approximated variance propagation. arXiv preprint arXiv:1908.00598 (2019)
36. Rafiei, M.H., Adeli, H.: Novel machine-learning model for estimating construction costs considering economic variables and indexes. Journal of construction engineering and management **144**(12), 04018106 (2018)
37. Russell, R.L., Reale, C.: Multivariate uncertainty in deep learning. IEEE Transactions on Neural Networks and Learning Systems (2021)

38. Sabbeh, S.F.: Machine-learning techniques for customer retention: A comparative study. International Journal of Advanced Computer Science and Applications **9**(2) (2018)
39. Sallab, A.E., Abdou, M., Perot, E., Yogamani, S.: Deep reinforcement learning framework for autonomous driving. Electronic Imaging **2017**(19), 70–76 (2017)
40. Schaub, H., Tsiotras, P., Junkins, J.L.: Principal rotation representations of proper n× n orthogonal matrices. International Journal of Engineering Science **33**(15), 2277–2295 (1995)
41. Settles, B.: Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2009)
42. Stuelpnagel, J.: On the parametrization of the three-dimensional rotation group. SIAM review **6**(4), 422–430 (1964)
43. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1701–1708 (2014)
44. Tkalcic, M., Tasic, J.F.: Colour spaces: perceptual, historical and applicational background, vol. 1. IEEE (2003)
45. Williams, C.K., Rasmussen, C.E.: Gaussian processes for machine learning, vol. 2. MIT press Cambridge, MA (2006)
46. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 694–699. ACM (2002)
47. Zhang, Z., Romero, A., Muckley, M.J., Vincent, P., Yang, L., Drozdzal, M.: Reducing uncertainty in undersampled mri reconstruction with active acquisition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2049–2058 (2019)

A Rotation Matrices

$$R_{y_1}(\hat{\theta}_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\hat{\theta}_1 & -\sin\hat{\theta}_1 \\ 0 & \sin\hat{\theta}_1 & \cos\hat{\theta}_1 \end{bmatrix}$$

$$R_{y_2}(\hat{\theta}_2) = \begin{bmatrix} \cos\hat{\theta}_2 & 0 & \sin\hat{\theta}_2 \\ 0 & 1 & 0 \\ -\sin\hat{\theta}_2 & 0 & \cos\hat{\theta}_1 \end{bmatrix}$$

$$R_{y_3}(\hat{\theta}_3) = \begin{bmatrix} \cos\hat{\theta}_3 & -\sin\hat{\theta}_3 & 0 \\ \sin\hat{\theta}_3 & \cos\hat{\theta}_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

B Neural Network Architecture

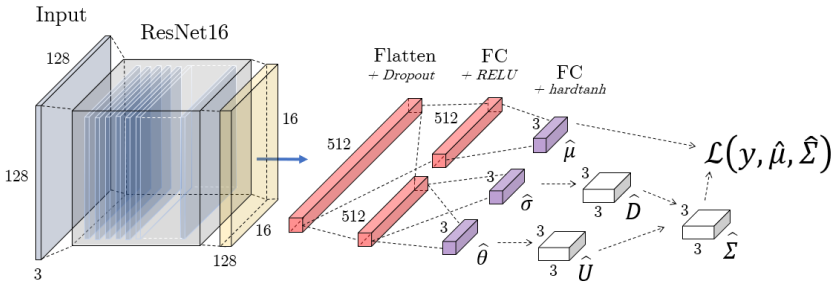

Fig. 6. Architecture for our neural network. The ResNet-16 is a simplified version of convolution blocks of ResNet [18] and the same convolution architecture was used for all compared models.

C Behavior during Training

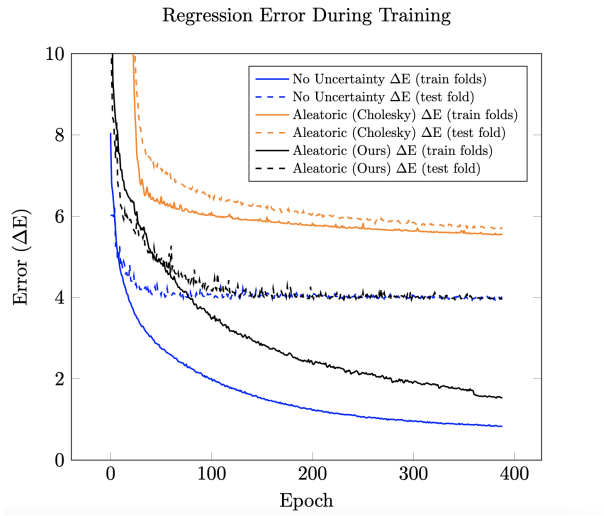

Fig. 7. Evolution of ΔE (Equation 8) through epochs of the 1st fold of our evaluation. Our model learns slower compared to the pure regression model, but converges to the same value on test fold. On the contrary, Cholesky uncertainty really starts optimization at around 30 epochs, and saturates to high ΔE values for both train and test data.

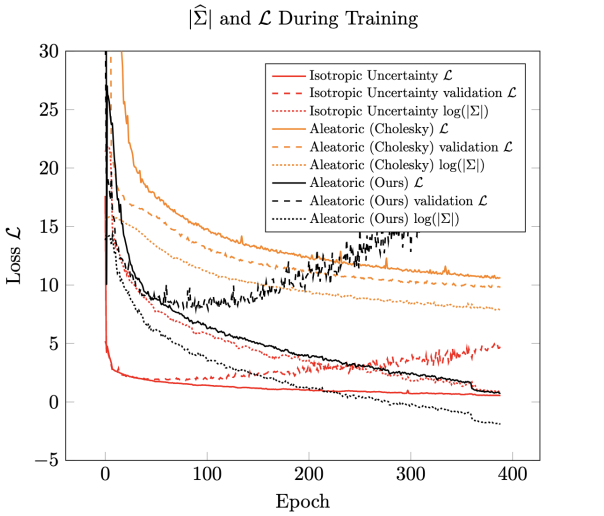

Fig. 8. Evolution of log-likelihood loss $\mathcal{L}$ (Equation 3) through epochs of the 1st fold of our evaluation. Similar overfitting behavior for $\mathcal{L}$ occurs for isotropic and our uncertainty model, that is explained by the decreasing volume of covariance matrix $|\Sigma|$ to very low values during epochs. Besides, Cholesky model keeps a larger volume for matrix Σ with no overfiting, but we interpret the higher values for $\mathcal{L}$ as a more difficult gradient descent, which seems also visible from Table 1 and Figure 5. Those behaviors motivates our re-scaling of Section 4.5 for comparing $\mathcal{L}$ between models.