PATH INTEGRALS FOR THE ATTRIBUTION OF MODEL UNCERTAINTIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding model uncertainties is of key importance in Bayesian machine learning applications. This often requires to meaningfully attribute a model's predictive uncertainty to its input features, however, popular attribution methods are primarily targeted at model scores for classification and regression tasks. Thus, in order to explain uncertainties, state-of-the-art alternatives commonly procure *counterfactual* feature vectors associated with low uncertainty and proceed by making direct comparisons. Here, we present a novel algorithm for uncertainty attribution in differentiable models, via path integrals which leverage *in-distribution* curves connecting feature vectors to counterfactual counterparts. We validate our method on benchmark image data sets with varying resolution, and demonstrate that (i) it produces meaningful attributions that significantly simplify interpretability over the existing alternatives and (ii) retains desirable properties from popular attribution methods.

1 INTRODUCTION

Estimating and understanding model uncertainties is of key importance in Bayesian inferential settings, which find applications in domains as diverse as natural language processing (Xiao & Wang, 2019), stochastic processes (Rao & Teg, 2013), network analysis (Perez et al., 2018) or image processing (Kendall & Gal, 2017), to name only a few. In contrast with model scores, model uncertainties manifest aspects of a system or *data generating process* that are not exactly known (Hüllermeier & Waegeman, 2021), and can be decomposed across *aleatoric* and *epistemic* components that help scrutinize different aspects in the functioning of a model, and can facilitate interpretability or fairness assessments in important machine learning applications (Awasthi et al., 2021).

Recently, there has been a growing interest in the study of methods for uncertainty estimation and decomposition (e.g. Depeweg et al., 2018; Smith & Gal, 2018; Van Amersfoort et al., 2020; Tuna et al., 2021) for purposes such as procuring adversarial examples, active learning or *out-of-distribution* detection. Most importantly, recent work has proposed *counterfactual* mechanisms for the interpretability of model uncertainties (Van Looveren & Klaise, 2019; Antoran et al., 2021; Schut et al., 2021), as well as their attribution to individual input features, such as pixels in an image. These methods proceed by identifying *small* adversarial or *in-distribution* variations in the raw input, s.t. predictive uncertainties in a model output are reduced. Then, attributions to individual pixels, words or categories are commonly assigned by direct comparison. This can facilitate the understanding of the strengths and weaknesses of varied probabilistic models, however, the optimization task to produce such *counterfactuals* requires a good balance between reducing uncertainties and minimising changes to original features, which is hard to achieve in practice. Most importantly, these methods do not satisfy commonly desired properties associated with modern importance attribution techniques (Sundararajan et al., 2017), such as *completeness* or *implementation invariance*.

In this paper, we present a novel framework for the attribution of predictive uncertainties, applicable to Bayesian differentiable models. We leverage path integrals (Sundararajan et al., 2017) along with *in-distribution* curves (Jha et al., 2020), and we propose aggregating attributions over paths starting at a reference *counterfactual* which bears no predictive uncertainty. We ensure that *completeness* and additional desirable properties are satisfied, hence, uncertainties are completely explained by (and decomposed over) pixels in an image. We validate our approach by direct comparison with recently introduced *counterfactual CLUE* explanations (Antoran et al., 2021), as well as popular interpretability methods, such as integrated gradients (Sundararajan et al., 2017), *LIME* (Ribeiro et al., 2016) or *kernelSHAP* (Lundberg & Lee, 2017), which we adapt for the attribution of predictive uncertainties. Experiments[1] on benchmark image data sets show that, in comparison to competing alternatives, our proposed method offers sparse and easily interpretable attributions, always limited to relevant *super-pixels*. Thus, we offer improved means to understand the interplay between raw inputs and aleatoric/epistemic uncertainties in deep models.

2 UNCERTAINTY ATTRIBUTIONS

We focus our presentation on a classification task with a neural classifier $f : \mathbb{R}^n \times \mathcal{W} \to \Delta^{|\mathcal{C}|-1}$ of a fixed architecture. The classifier maps feature vectors $\boldsymbol{x} \in \mathbb{R}^n$ along with network weights $\boldsymbol{w} \in \mathcal{W}$ to an element in the standard $(|\mathcal{C}| - 1)$-simplex, which represents membership probabilities across classes in a set $\mathcal{C}$. On training f within an (approximate) Bayesian setting, we commonly obtain a *posterior* over the hypothesis space of models, i.e. a distribution $\pi(\boldsymbol{w}|\mathcal{D})$ over weights conditioned on the available train data $\mathcal{D} = \{\boldsymbol{x}_i, c_i\}_{i=1,2,\ldots}$. Popular approaches to procure such posterior often differ in their approach to incorporate *prior* knowledge and include *dropout* (Srivastava et al., 2014), *Bayes-by-Backprop* (Blundell et al., 2015) or SG-HMC (Springenberg et al., 2016).

A model score for classification with a new data point $\boldsymbol{x}^\star \in \mathbb{R}^n$ is derived from the *posterior predictive distribution* by marginalising over posterior weights, i.e.

$$\pi(\boldsymbol{x}^\star|\mathcal{D}) = \int_{\mathcal{W}} f(\boldsymbol{x}^\star, \boldsymbol{w})\pi(\boldsymbol{w}|\mathcal{D})d\boldsymbol{w} = \mathbb{E}_{\boldsymbol{w}|\mathcal{D}}[f(\boldsymbol{x}^\star, \boldsymbol{w})], \tag{1}$$

and is easily approximated as $\frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}^\star, \boldsymbol{w}_i)$, with weight samples $\boldsymbol{w_i} \sim \pi(\boldsymbol{w}|\mathcal{D})$, $i = 1, \ldots, N$. In the following, we are concerned with the *entropy* as a measure of uncertainty:

$$H(\boldsymbol{x}|\mathcal{D}) = -\sum_{c\in\mathcal{C}} \mathbb{E}_{\boldsymbol{w}|\mathcal{D}}[f_c(\boldsymbol{x}, \boldsymbol{w})] \cdot \log \mathbb{E}_{\boldsymbol{w}|\mathcal{D}}[f_c(\boldsymbol{x}, \boldsymbol{w})] \tag{2}$$

where $f_c(\boldsymbol{x}, \boldsymbol{w})$ represents the probability of class-c membership.

Remark. *Concepts in this paper trivially extend to varied representations of uncertainty in classification and regression settings. Details are omitted for simplicity in the presentation.*

The entropy term in (2) may further be decomposed through the law of iterated variances (Kendall & Gal, 2017) so as to yield an *aleatoric* term

$$H_a(\boldsymbol{x}|\mathcal{D}) = \mathbb{E}_{\boldsymbol{w}|\mathcal{D}}[H(\boldsymbol{x}, \boldsymbol{w})] = -\sum_{c\in\mathcal{C}} \mathbb{E}_{\boldsymbol{w}|\mathcal{D}}\big[f_c(\boldsymbol{x}, \boldsymbol{w}) \cdot \log f_c(\boldsymbol{x}, \boldsymbol{w})\big],$$

which measures the mean predictive entropy across models in the posterior hypothesis space, as well as the *mutual information* or *epistemic* term, $H_e(x|\mathcal{D}) = H(\boldsymbol{x}|\mathcal{D}) - H_a(\boldsymbol{x}|\mathcal{D})$ that represents model uncertainty projected into the latent membership vector $\pi(\boldsymbol{x}|\mathcal{D})$. Intuitively, aleatoric uncertainty represents natural stochastic variation in the observations over repeated experiments; on the other hand, epistemic uncertainty is descriptive of model unknowns due to inadequate data or inappropriate modelling choices.

2.1 PATH INTEGRATED GRADIENTS

Path *integrated gradients* (IG) (Sundararajan et al., 2017) is a simple and popular method for importance attributions that differs from conventional feature removal and permutation techniques (Covert et al., 2020), and is primarily targeted at image processing tasks. It is a practical and easy to implement alternative to layer-wise relevance propagation Montavon et al. (2019) or DeepLift (Shrikumar et al., 2017); it retains desired properties including *sensitivity* and *implementation invariance*, and has been extended to several adaptations (Smilkov et al., 2017; Jha et al., 2020).

Given a classifier $f(\cdot, \boldsymbol{w})$ along with a feature vector $\boldsymbol{x}$, path IG *explains* a model score $f(\boldsymbol{x}, \boldsymbol{w})$ using an alternative *fiducial* vector $\boldsymbol{x}^0$ as a reference, which is presumably not associated with any

[1] Source code for reproducing our results can be found at the following link: *[removed for blind review]*

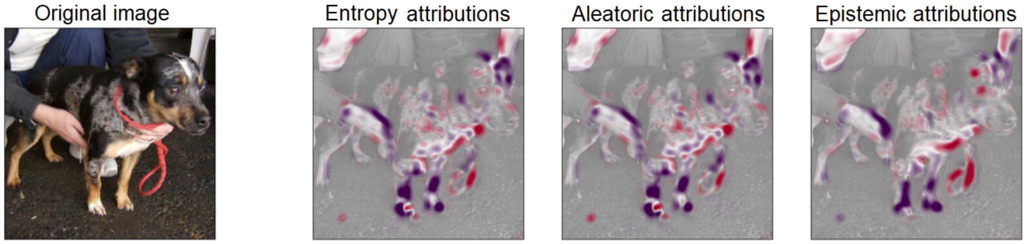

Figure 1: Example contributions to predictive uncertainty for a classification task in *dogs versus cats* data. The image is compared to a fiducial black screen with entropy of 0.49 (aleatoric 0.39, epistemic 0.1). Importances are *smoothed* with a Gaussian filter ($\Sigma = 3I$). Red importances represent contributions towards *increasing* uncertainty, while purple importances contribute towards *decreasing* uncertainty.

class observed in the training data. The attributed importance at index or *pixel* i is given by

$$\text{IG}_i^{\delta}(\boldsymbol{x}|\boldsymbol{w}) = \int_0^1 \frac{\partial f(\delta(\alpha), \boldsymbol{w})}{\partial \delta_i(\alpha)} \frac{\partial \delta_i(\alpha)}{\partial \alpha} d\alpha$$

so that $\sum_i \text{IG}_i(\boldsymbol{x}|\boldsymbol{w}) = f(\boldsymbol{x}, \boldsymbol{w}) - f(\boldsymbol{x}^0, \boldsymbol{w})$. The result follows from the fundamental theorem of calculus for line integrals known as the *gradient theorem*. Here, $\delta : [0, 1] \to \mathbb{R}^n$ represents a curve with endpoints at $\delta(0) = \boldsymbol{x}^0$ and $\delta(1) = \boldsymbol{x}$.

In *vanilla* IG, δ is parametrized as a straight path between fiducial and image feature vectors, i.e. $\delta(\alpha) = \boldsymbol{x}^0 + \alpha(\boldsymbol{x} - \boldsymbol{x}^0)$, and the above simplifies to

$$\text{IG}_i(\boldsymbol{x}|\boldsymbol{w}) = (x_i - x_i^0) \times \int_0^1 \frac{\partial f(\boldsymbol{x}^0 + \alpha(\boldsymbol{x} - \boldsymbol{x}^0), \boldsymbol{w})}{\partial x_i} d\alpha,$$

so that importances are heavily influenced by differences in pixel values between $\boldsymbol{x}$ and $\boldsymbol{x}^0$. However, a straight line often transitions the path $\boldsymbol{x}^0 \rightsquigarrow \boldsymbol{x}$ *out-of-distribution* or outside the *data-manifold* (Jha et al., 2020; Adebayo et al., 2020). Also, the fiducial choice is considered problematic (Sundararajan et al., 2017) and generally defaults to a black background.

2.2 Integrated Gradients with Uncertainty

Commonly, the classifier $f(\cdot, \boldsymbol{w})$ is presumed to be binary (Sundararajan et al., 2017) with model scores constrained to the interval $[0, 1]$. However, the above logic for importance attribution does easily generalize to multi-class Bayesian settings in the presence of uncertainty. Here, the *posterior predictive classifier* $\pi(\boldsymbol{x}|\mathcal{D})$ introduced in (1) accepts a path IG importance at index i given by

$$\text{IG}_i^{\delta}(\boldsymbol{x}) = \int_0^1 \mathbb{E}_{\boldsymbol{w}|\mathcal{D}}\left[\frac{\partial f(\delta(\alpha), \boldsymbol{w})}{\partial \delta_i(\alpha)}\right] \frac{\partial \delta_i(\alpha)}{\partial \alpha} d\alpha,$$

which represents a *mean-average* trajectory over the curve δ and follows from *dominated convergence*. Most importantly, we may employ path IG to *explain* univariate measures of uncertainty from the posterior predictive distribution over classes, as observed in Figure 1. Here,

$$\text{IG-H}_i^{\delta}(\boldsymbol{x}) = -\sum_{c \in \mathcal{C}} \int_0^1 \Delta_i(\alpha) \frac{\partial \delta_i(\alpha)}{\partial \alpha} d\alpha \tag{3}$$

captures variations in predictions covering multiple classes, and is defined s.t.

$$\Delta_i(\alpha) = \big(1 + \log \mathbb{E}_{\boldsymbol{w}|\mathcal{D}}[f_c(\delta(\alpha), \boldsymbol{w})]\big) \cdot \mathbb{E}_{\boldsymbol{w}|\mathcal{D}}\Big[\frac{\partial f_c(\delta(\alpha), \boldsymbol{w})}{\partial \delta_i(\alpha)}\Big]$$

attributes importances for the change in entropy between a fiducial point and a feature vector, and

$$\Delta_i(\alpha) = \mathbb{E}_{\boldsymbol{w}|\mathcal{D}}\Big[\big(1 + \log f_c(\delta(\alpha), \boldsymbol{w})\big) \cdot \frac{\partial f_c(\delta(\alpha), \boldsymbol{w})}{\partial \delta_i(\alpha)}\Big]$$

is the analogue representation restricted to the aleatoric term. Any variation in epistemic uncertainty is readily shown to be *explained* as the difference in importances between the above two terms. In Figure 1, the goal is not to understand why the classifier suggests this picture refers to a dog; instead, we comprehend why the model *struggles* to predict any single class with confidence, and we notice that the leash and a human hand are problematic. Further examples may be found in Appendix A; in all cases, importances have been *smoothed* with a Gaussian filter, averaging over *positive* (increasing uncertainty) and *negative* (decreasing uncertainty) contributions. The attributions are easily computed by standard Bayesian procedures, approximating the inner expectations with simulations, however, the choice of fiducial (black screen) and out-of-distribution path remain controversial and a significant challenge in order to enable an intuitive understanding of predictive uncertainties in our model, which may represent a barrier in applications (Antoran et al., 2021).

3 METHODOLOGY

Next, we describe the computational process summarized in Algorithm 1, which produces novel *in-distribution* attributions of uncertainty for a feature vector $\boldsymbol{x}$ and predictive posterior $\pi(\boldsymbol{x}|\mathcal{D})$ in (1). We do so through the use of a *counterfactual* fiducial bearing no relation to causal inference (Pearl, 2010). This counterfactual is an alternative vector $\boldsymbol{x}^0$ defined similarly to *CLUE*s in Antoran et al. (2021), i.e. (i) *in distribution* and (ii) close to $\boldsymbol{x}$ according to some arbitrary distance metric. However, we furthermore require that the class distribution $\pi(\boldsymbol{x}^0|\mathcal{D})$ bears close to 0 predictive uncertainty. Intuitively, we construct IG attributions using finely tuned fiducial points, by comparing ambiguous images to easily predicted counterparts that bear a significant resemblance.

Algorithm 1: Uncertainty attributions

---

input : Feature vector $\boldsymbol{x}$, predictive posterior $\pi(\cdot|\mathcal{D})$ and uncertainty estimator $H(\cdot)$.
Distance metric $d(\cdot,\cdot)$, VAE encoder $\phi(\cdot)$ and decoder $\psi(\cdot)$.
Penalty $\lambda >> 0$ and learning rate $\nu > 0$.
output: Uncertainty Attributions IG-H$_i(\boldsymbol{x})$, $i = 1, \ldots, n$.
Initialise $\boldsymbol{z}^0 = \boldsymbol{z} = \phi_\mu(\boldsymbol{x})$;
Compute predicted class $\hat{c} = \arg\max_i \pi_i(\boldsymbol{x}|\mathcal{D})$;
while $\mathcal{L}$ *not converged* **do**
  $\mathcal{L} \leftarrow d(\psi(\boldsymbol{z}^0), \boldsymbol{x}) + \frac{1}{2m}\sum_j z_j^2 + \lambda \log \pi_{\hat{c}}(\psi(\boldsymbol{z})|\mathcal{D})$ and $\boldsymbol{z}^0 \leftarrow \boldsymbol{z}^0 - \nu \nabla_{\boldsymbol{z}} \mathcal{L}$;
end
while $\mathcal{L}$ *not converged* **do**
  $\mathcal{L} \leftarrow d(\psi(\boldsymbol{z}), \boldsymbol{x}) + \frac{1}{2m}\sum_j z_j^2$ and $\boldsymbol{z} \leftarrow \boldsymbol{z} - \nu \nabla_{\boldsymbol{z}} \mathcal{L}$;
end
Approximate IG-H$_i^\delta(\boldsymbol{x})$, $i = 1, \ldots, n$ in (5) along $\delta_{\boldsymbol{z}^0 \to \boldsymbol{z}}$ through trapezoidal integration.

To begin with, we assume the existence of a generative *variational auto-encoder* (VAE) composed of an encoder $\phi : \mathbb{R}^n \to \mathbb{R}^m$ and decoder $\psi : \mathbb{R}^m \to \mathbb{R}^n$. As customary, the data-generating process in $\mathbb{R}^m$ is unit-Gaussian with an arbitrary dimensionality $m << n$.

3.1 DOMAIN OF INTEGRATION

The domain of integration must be an *in-distribution* curve across end-points $\boldsymbol{x}^0 \rightsquigarrow \boldsymbol{x}$. We select the fiducial as a decoded image $\boldsymbol{x}^0 = \psi(\boldsymbol{z}^0)$, where $\boldsymbol{z}^0$ is the solution to the constrained optimization problem

$$\boldsymbol{z}^0 = \underset{\boldsymbol{z} \in \mathbb{R}^m_{\boldsymbol{x}}}{\arg\min} \left[d(\psi(\boldsymbol{z}), \boldsymbol{x}) + \frac{1}{2m} \sum_j z_j^2 \right] \tag{4}$$

where $\mathbb{R}^m_{\boldsymbol{x}} = \{\boldsymbol{z} \in \mathbb{R}^m \ : \ \|e_{\hat{c}} - \pi(\psi(\boldsymbol{z})|\mathcal{D})\| < \varepsilon\}$ for a small $\epsilon > 0$. Here, $\hat{c} = \arg\max_i \pi_i(\boldsymbol{x}|\mathcal{D})$ is the predicted class by our posterior predictive classifier, and e_i is the unit indicator vector at index i. The metric $d(\cdot,\cdot)$ may take multiple forms, such as the cross-entropy or mean absolute difference over pixel values. The right-most term is the negative log-density (up to proportionality) of $\boldsymbol{z}$ in latent space; this ensures computational stability and restricts the search to be *in-distribution*. Hence, we retrieve a counterfactual fiducial which (i) corresponds to the class prediction by $\pi(\boldsymbol{x}|\mathcal{D})$ and (ii)

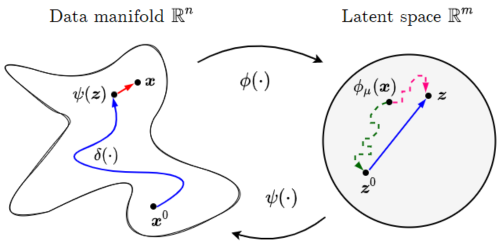

Figure 2: Procedural sketch to generate a path of integration. Here, *fiducial* $\boldsymbol{z}^0$ and *reconstruction* $\boldsymbol{z}$ points are optimized in latent space by gradient descent, starting initially from the encoding of $\boldsymbol{x}$ (dashed lines). A connecting straight path (in blue) is projected to the data-manifold and augmented with an interpolating component (in red).

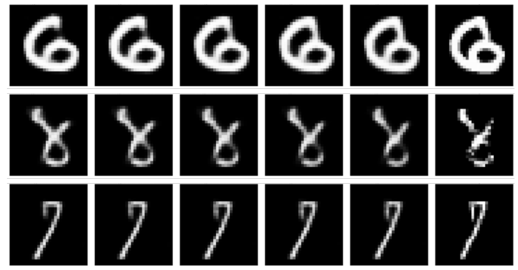

Figure 3: An example of *in-distribution* curves connecting fiducial (left-most) and real (right-most) data points, on MNIST digits data. Digits on the left bear no model uncertainty in classification.

bears close to zero predictive uncertainty. In practice, we approximate (4) through an unconstrained search with a large penalty on

$$d_{\mathcal{X}}(e_{\hat{c}}, \pi(\psi(\boldsymbol{z})|\mathcal{D})) = -\log \pi_{\hat{c}}(\psi(\boldsymbol{z})|\mathcal{D}),$$

i.e. the cross-entropy between the predicted class $\hat{c}$ and the membership vector $\pi(\psi(\boldsymbol{z})|\mathcal{D})$ given a decoding $\psi(\boldsymbol{z})$. We proceed by gradient descent initialised at $\phi_\mu(\boldsymbol{x})$ (the encoder's mean).

We next parametrize a curve $\delta : [0,1] \to \mathbb{R}^n$ by following the steps displayed in Figure 2, s.t. $\delta(\alpha) = \psi(\boldsymbol{z}^0 + \alpha(\boldsymbol{z} - \boldsymbol{z}^0))$ where

$$\boldsymbol{z} = \arg\min_{\boldsymbol{z} \in \mathbb{R}^m} \left[d(\psi(\boldsymbol{z}), \boldsymbol{x}) + \frac{1}{2m} \sum_j z_j^2 \right]$$

is also optimised by gradient descent initialised at a starting point $\phi_\mu(\boldsymbol{x})$. Note that this is the same optimization problem as in (4) but without a constraint imposing a reduction in predictive uncertainty. Consequently, the curve δ offers an in-distribution trajectory (Jha et al., 2020) between $\delta(0) = \psi(\boldsymbol{z}^0) = \boldsymbol{x}^0$ and a reconstruction $\delta(1) = \psi(\boldsymbol{z})$ of $\boldsymbol{x}$. If the auto-encoder does not provide an efficient reconstruction, a two-level auto-encoder (Dai & Wipf, 2019) can offer a viable alternative; however, the domain of integration may be easily augmented through *vanilla* (straight path) IG between the end-points $\psi(\boldsymbol{z}) \rightsquigarrow \boldsymbol{x}$, and we display a few examples on MNIST digits within Figure 3. Overall, changes in predictive entropy and model scores between a reconstruction $\psi(\boldsymbol{z})$ and the original counterpart $\boldsymbol{x}$ are not observed to be significant within our experiments.

3.2 LINE INTEGRAL FOR IMPORTANCE ATTRIBUTION

For simplicity, we restrict the formulae to the *in-distribution* component along the curve $\delta : [0,1] \to \mathbb{R}^n$ defined in Subsection 3.1, and we ignore the trivial *straight path* connecting $\psi(\boldsymbol{z}) \rightsquigarrow \boldsymbol{x}$. Following (3), we now require the *total* differential of the entropy $H(\cdot)$ wrt $\boldsymbol{z}$ in latent space; however, we wish to retrieve importances only for features $\boldsymbol{x}$ in the *original* data manifold within $\mathbb{R}^n$. To this end, the attribution at index $i = 1, \ldots, n$ is given by

$$\text{IG-H}_i^\delta(\boldsymbol{x}) = -\sum_{c \in \mathcal{C}} \sum_{j=1}^{m} (z_j - z_j^0) \int_0^1 \Delta_i(\alpha) \frac{\partial \psi_i(\boldsymbol{z}^0 + \alpha(\boldsymbol{z} - \boldsymbol{z}^0))}{\partial z_j} d\alpha \tag{5}$$

where $\Delta_i(\alpha)$ follows the definitions in (3) for both the entropy in (2) and its aleatoric term. In Figure 4, we show an example that compares attributions in (5) versus a *vanilla* variant of integrated gradients previously introduced in (3). There, we find a *CelebA* image (Liu et al., 2015) with tags for the presence of a *smile*, *arched eyebrows* and *no bags under the eyes*, and notice a significant reduction of noise and improvements in interpretability.

Finally, we note that the attribution method we present in Algorithm 1 is similarly defined for any generic uncertainty term $H(\cdot)$, whether in regression or classification settings. Intuitively, our

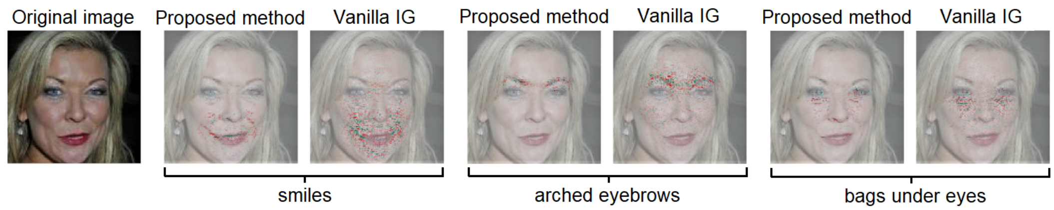

Figure 4: Comparison of uncertainty attributions for individual pixels on a *CelebA* image. We compare predictive uncertainties for three Bayesian classifiers, which measure the presence (or lack) of *smiles* (left), *arched eyebrows* (centre), and *bags under eyes* (right). Red pixels contribute by *increasing uncertainties*, in green we find contributions towards *decreasing uncertainties*.

method computes the total derivative of $H(\cdot)$ wrt $\boldsymbol{z}$ through the original feature space, using the chain rule, and later undertakes summation over contributions in latent space, which is easily managed through vectorized operations.

3.3 Properties

Most importantly, due to *path independence* and noting that $H(\boldsymbol{x}^0|\mathcal{D}) \approx 0$ (by definition), importances drawn from any trajectory $\delta(\cdot)$ parametrized as in Subsection 3.1 will approximately account for **all** of the uncertainty in a posterior predictive task, i.e.

$$H(\boldsymbol{x}|\mathcal{D}) \approx \int_0^1 \nabla H(\delta(\alpha)|\mathcal{D}) d\alpha = \sum_{i=1}^{n} \text{IG-H}_i^\delta(\boldsymbol{x}),$$

and this is commonly referred to as *completeness*. Additionally, the reliance on path derivatives along with the rules of composite functions ensure that both fundamental axioms of *sensitivity(b)* (i.e. *dummy property*) along with *implementation invariance* are inherited (see Friedman, 2004; Sundararajan et al., 2017). Specifically, the attribution will be zero for any index which does not mathematically influence the posterior classifier. Also, given a fixed VAE architecture, uncertainty attributions are identical for distributionally equivalent posterior predictive classifiers. However, the departure from *vanilla* straight paths in the data-manifold means that our proposed method is no longer *symmetry preserving*, i.e. any symmetric variables in $\boldsymbol{x}$ (according to the classifier $\pi(\boldsymbol{x}|\mathcal{D})$ in (1)) are not guaranteed to receive identical attributions.

4 Experiments

We presents results from applying our proposed methodology for uncertainty attributions to benchmark image data sets, including the repositories *MNIST handwritten digits* (LeCun & Cortes, 2010), *fashion-MNIST* (Xiao et al., 2017) and *CelebA* (Liu et al., 2015). In all cases, we use *MC dropout* in order to procure approximate posteriors $\pi(\boldsymbol{w}|\mathcal{D})$ for network weights in our (approximate) Bayesian predictive models. Details on architectural choices, hyper-parameters, training regimes and preprocessing may be found within Appendix B.

We compare our method to alternatives including *vanilla* integrated gradients (Sundararajan et al., 2017) and *CLUE* (Antoran et al., 2021); as well as adaptations of *LIME* (Ribeiro et al., 2016) and *kernelSHAP* (Lundberg & Lee, 2017), which we fine-tune in order to measure variances in uncertainty instead of model scores. We use the following settings with the aforementioned algorithms:

- Vanilla IG is implemented with a black fiducial and a straight line as domain of integration.
- *CLUE* attributions are derived as the differential between a real image and its decoded CLUE counterpart (cf. Antoran et al., 2021, Appendix F). The cost function weighs for reconstruction and uncertainty terms are tuned on a validation set.
- *LIME* is implemented through *quickshift* segmentation, with kernel 1, maximum distance 5 and ratio of 0.2. We use a binomial mask with deactivation probability 0.2, and *Lasso* regression to attribute importances.

- *SHAP* proceeds through pixel/index coalitions of varying size; masked index points do not default to "black-background" or mean values, but are instead re-sampled from their corresponding marginal distributions. We use *Lasso* regression to attribute importances.

In all experiments, we first produce scores for the data using the posterior predictive distribution $\pi(\boldsymbol{x}|\mathcal{D})$, next, we compute the entropy $H(\boldsymbol{x}|\mathcal{D})$. We rank data points by degree of uncertainty and apply the attribution methods to the highest-ranking images or feature vectors. We show that, in comparison to the alternatives, our proposed approach offers sparse and easily interpretable uncertainty attributions limited to relevant *super-pixels*. Our attributions are furthermore dominated by *positive* importances, i.e. pixels that contribute only to an *increase* in uncertainty and add up to the total predictive uncertainty in classification.

4.1 MNIST HANDWRITTEN DIGITS

In Figure 5a we observe attributions of uncertainty on a selection of high-entropy MNIST digits. These attributions are further decomposed across both aleatoric and epistemic terms, and additional examples may be found in Appendix C. We display positive importances for pixels which contribute by *increasing* the uncertainty in classification according to our model. We note that our attributions are humanly interpretable and isolated to very few pixels that confuse the predictive model; commonly, these represent small regions where dark ink is either *missing* or in *excess*.

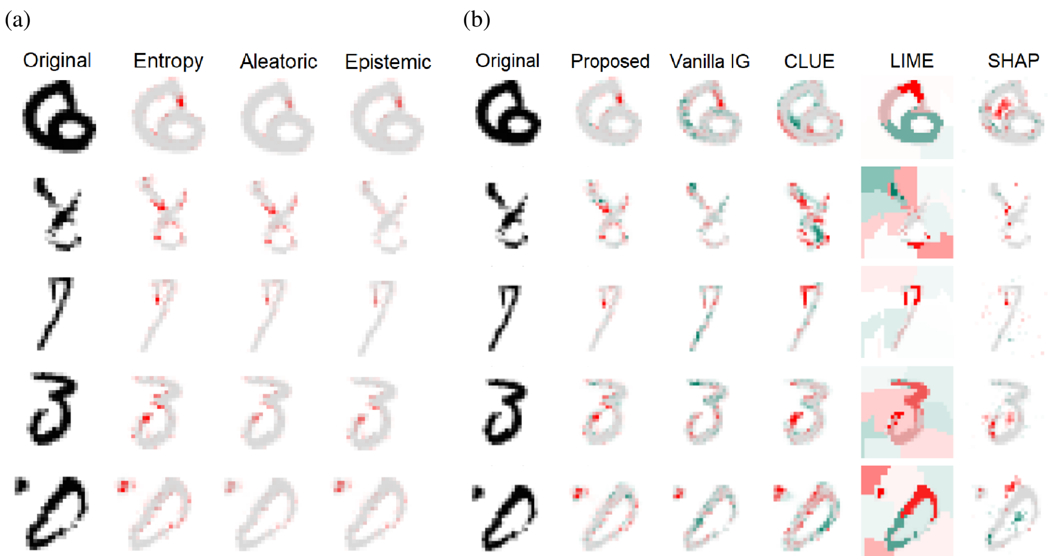

Figure 5: (a) Uncertainty attributions with decomposition across aleatoric/epistemic components and (b) comparison against popular methods, on MNIST digits.

In Figure 5b, we find a comparison of uncertainty attribution methods applied to the same digits, with further examples also found in Appendix C. There, we only display attributions for the entropy, and show both positive (in red) and negative (in green) contributions. Noticeably, our attributions are strongly dominated by the few pixels that contribute by *increasing uncertainty*. Comparatively, *LIME* importances are highly restricted by the tuning of its segmentation algorithm. Similarly, *kernelSHAP* requires significant re-sampling from the joint distribution of pixel *coalitions*, and struggles to identify small super-pixels that are primary drivers of uncertainty. The *CLUE* methodology is targeted (and efficient) at identifying interpretable *in-distribution* counterfactuals, hence, the importances derived commonly suggest significantly re-drawing the original image, and this easily overestimates the minimal changes required to facilitate predictions without uncertainty. Finally, vanilla IG cannot associate importances with background pixels, as these share values with the fiducial image. Also, the fiducial is associated with high predictive uncertainty, thus, the attributions offer a confusing mix of pixels that both decrease and increase uncertainties.

4.2 FASHION-MNIST DATASET

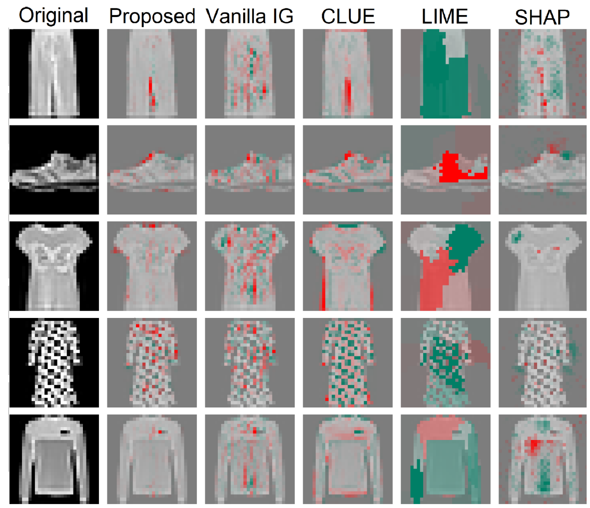

Figure 6: Comparison of uncertainty attribution methods on fashion-MNIST images.

In Figure 6, we observe a similar comparison of uncertainty attribution methods applied to a selection of high-entropy *Fashion-MNIST* images, with further examples also found in Appendix C. We note that our attributions are again interpretable and restricted to *missing* or *excess* pixels, s.t. visual representations of clothing items would be classified without uncertainty. Similarly to the previous example with digits, *LIME* importances are severely restricted by the segmentation algorithm, additionally, *kernelSHAP* is inconsistent and often struggles to identify small and meaningful super-pixels that are drivers of uncertainty. *CLUE* importances suggest a significant re-drawing of the original clothing pieces to make them more similar to an easily *classifiable* counterfactual, which again overestimates the extent of changes required in order to mitigate predictive uncertainty. Finally, attributions through vanilla IG are confusing for the same reasons as outlined in our prior example.

4.3 CELEBA DATASET

Finally, we compute similar uncertainty attributions for classification scores with facial attributes on *CelebA* images. To improve our presentation, we omit *LIME* importances here, however, these may be found among further examples in Appendix C. In Figure 7a we find attributions for the class label *smile*, whereas Figure 7b shows results for the class label *arched eyebrows*. In both cases, we notice that our attributions are comparatively neat, interpretable and always restricted to facial features around the mouth, cheeks or eyebrows, depending on the classification task. For instance, our attributions can complement the difference between an uncertain smirk and an easily classifiable smile; in non-smiling people, the attributions may bring attention to features around the cheek that are commonly associated with smiles, and thus confuse the model.

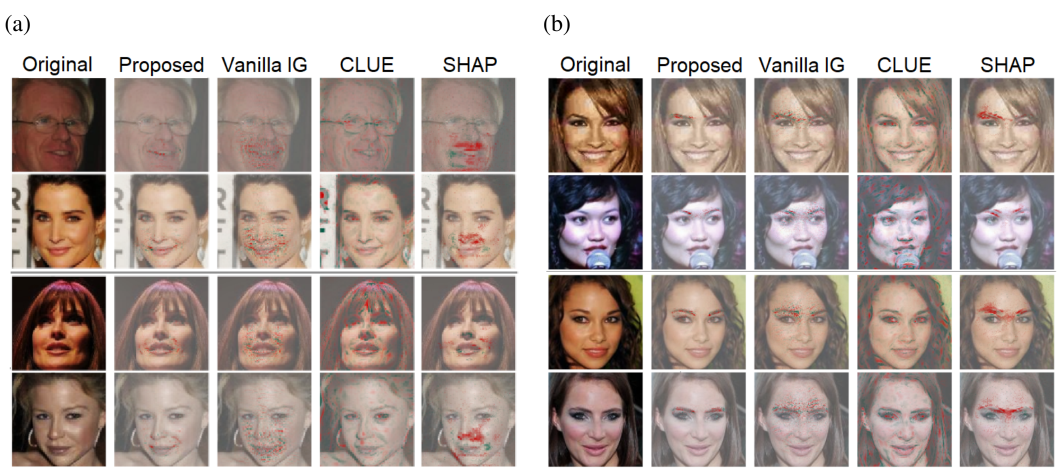

Figure 7: Uncertainty attributions on CelebA images. (a) Pictures with the *smile* attribute labelled as *positive* (top two pictures) or *negative* (bottom two pictures). (b) Pictures with the *arched eyebrows* attribute labelled as *negative* (top two pictures) or *positive* (bottom two pictures)

Similarly to digits and fashion items in previous examples, our importances isolate the pixels for facial features that seemingly contradict the predicted class by the Bayesian classifier. In comparison, attributions through vanilla integrated gradients identify multiple artefacts, present a mix of positive and negative importances, and are hard to interpret. *KernelSHAP* offers inconsistent results that commonly highlight wide areas around the region of interest. Finally, *CLUE* importances visibly struggle with higher resolution images, the reason for this being reliance on direct comparisons between an image and its counterfactual reference. We also note that redrawing a high-fidelity face reconstruction with an autoencoder is considerably more difficult than drawing digits or clothes.

5 Conclusion

In this paper, we have introduced a novel computational framework for the attribution of model uncertainties in approximate Bayesian settings. Our experimental results show potential for facilitating interpretations of the interplay between raw features and predictive uncertainties in complex deep models, and can thus contribute to improved transparency and interpretability in deep learning applications. Our method shows considerable improvements over both simple baselines and existing alternatives for uncertainty attribution in image processing tasks. Alternative methods are generally based on direct comparisons of images versus adversarial or in-distribution counterfactuals, along with subjective assessments. In our case, we leverage traditional path integral methods in order to ensure that uncertainties are attributed respecting the desirable properties of *completeness*, *sensitivity(b)*, and *implementation invariance*.

Acknowledgements

text removed for the blind review process.

References

Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 700–712, 2020.

Javier Antoran, Umang Bhatt, Tameem Adel, Adrian Weller, and José Miguel Hernández-Lobato. Getting a CLUE: A method for explaining uncertainty estimates. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=XSLF1XFq5h`.

Pranjal Awasthi, Alex Beutel, Matthäus Kleindessner, Jamie Morgenstern, and Xuezhi Wang. Evaluating fairness of machine learning models under uncertain and incomplete information. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 206–214, 2021.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pp. 1613–1622, 2015.

Ian Covert, Scott Lundberg, and Su-In Lee. Feature removal is a unifying principle for model explanation methods. *arXiv preprint arXiv:2011.03623*, 2020.

Bin Dai and David Wipf. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=B1e0X3C9tQ`.

Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*, pp. 1184–1193. PMLR, 2018.

Eric J Friedman. Paths and consistency in additive cost sharing. *International Journal of Game Theory*, 32(4):501–518, 2004.

Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1133–1141, 2017. doi: 10.1109/WACV.2017.131.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

Anupama Jha, Joseph K Aicher, Matthew R Gazzara, Deependra Singh, and Yoseph Barash. Enhanced integrated gradients: improving interpretability of deep learning models using splicing codes as a case study. *Genome biology*, 21(1):1–22, 2020.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yann LeCun and Corinna Cortes. MNIST handwritten digits database, 2010. `http://yann.lecun.com/exdb/mnist/`.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4768–4777, 2017.

Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 193–209, 2019.

Judea Pearl. Causal inference. In *Proceedings of Workshop on Causality: Objectives and Assessment at NIPS 2008*, pp. 39–58, 2010.

Iker Perez, David Hodge, and Theodore Kypraios. Auxiliary variables for bayesian inference in multi-class queueing networks. *Statistics and Computing*, 28(6):1187–1200, 2018.

Vinayak Rao and Yee Whte Teg. Fast mcmc sampling for markov jump processes and extensions. *Journal of Machine Learning Research*, 14(11), 2013.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Lisa Schut, Oscar Key, Rory Mc Grath, Luca Costabello, Bogdan Sacaleanu, Yarin Gal, et al. Generating interpretable counterfactual explanations by implicit minimisation of epistemic and aleatoric uncertainties. In *International Conference on Artificial Intelligence and Statistics*, pp. 1756–1764. PMLR, 2021.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pp. 3145–3153, 2017.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, 2017.

Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI*, 2018.

Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust bayesian neural networks. *Advances in neural information processing systems*, 29:4134–4142, 2016.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15:1929–1958, 2014.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

Omer Faruk Tuna, Ferhat Ozgur Catak, and M Taner Eskil. Exploiting epistemic uncertainty of the deep learning models to generate adversarial samples. *arXiv preprint arXiv:2102.04150*, 2021.

Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pp. 9690–9700. PMLR, 2020.

Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. *arXiv preprint arXiv:1907.02584*, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. Data available at `https://github.com/zalandoresearch/fashion-mnist`.

Yijun Xiao and William Yang Wang. Quantifying uncertainties in natural language processing tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 7322–7329, 2019.

A VANILLA IG FOR UNCERTAINTY ATTRIBUTION

In Figure 8 we find further examples of *vanilla* IG attributions of uncertainty using a straight path between a black background and *Dogs versus Cats* pictures. Importances are *smoothed* with a Gaussian filter ($\Sigma = 3I$) to average over positive and negative contributions. Red importances represent contributions towards *increasing* uncertainty, in purple, contributions towards *decreasing* uncertainty

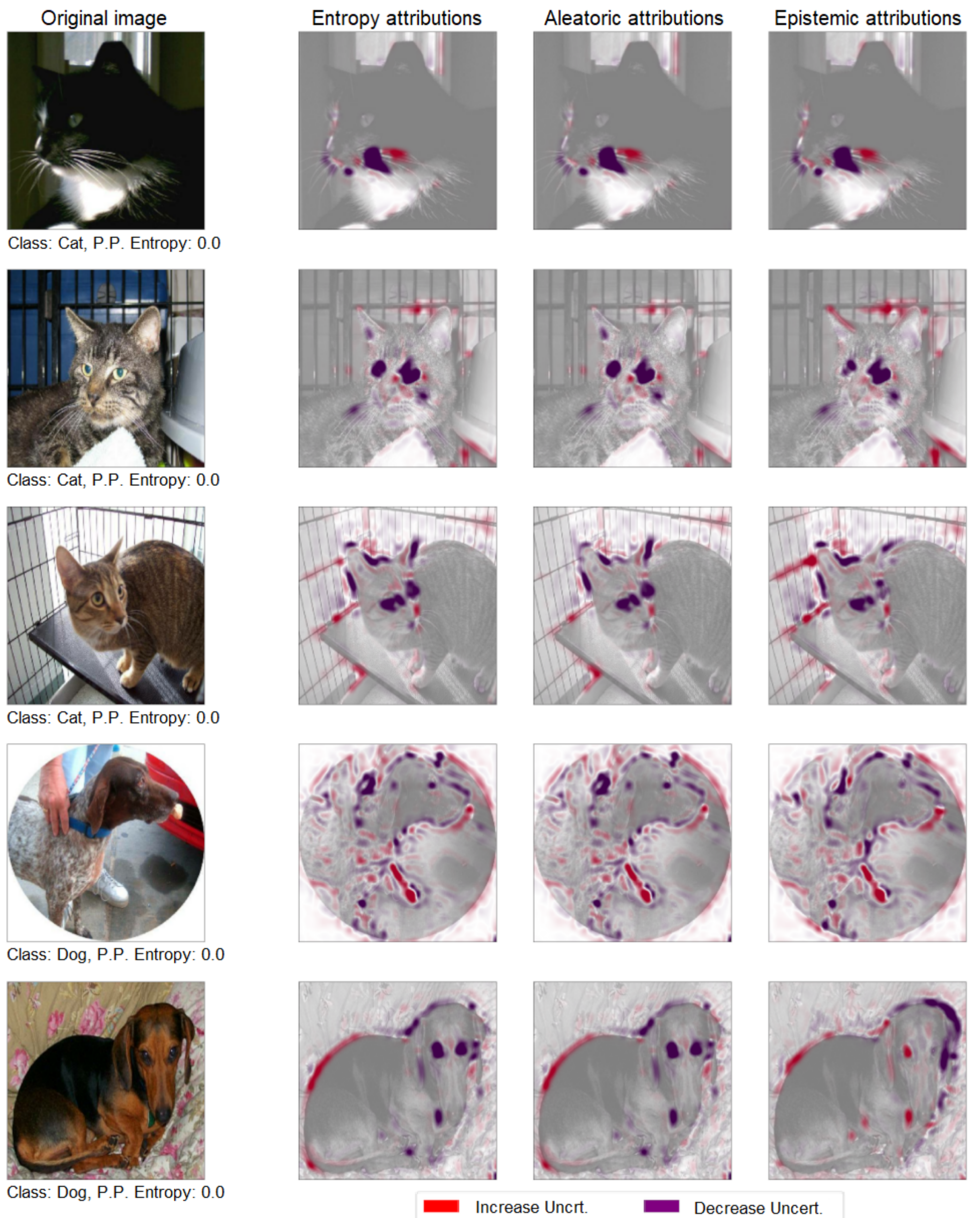

Figure 8: Vanilla uncertainty attributions using a straight path. Classes predicted with 0 posterior predictive entropy. Importances reflect a reduction in entropy from a black background fiducial.

B SUMMARY OF MODEL ARCHITECTURES AND HYPER-PARAMETERS

The following is a comprehensive summary of model architecture choices, hyper-parameters, training regimes and further details used within the experiments in this paper. All of our models are implemented through Keras, and the source code to reproduce the results may be found at *url removed for the blind review process*

B.1 MNIST HANDWRITTEN DIGITS

Classifier. Our digits classifier is a convolutional neural network with *max-pooling* layers and dropout, structured as:

- Two convolutional layers of kernel size 3×3 and *relu* activation; the *filter counts* are 32 and 64 for the first and second layers. Each convolutional layer uses a *stride* length of 1 and is followed by a *max-pooling* layer of *pool size* 2×2.
- The output above is flattened and fed through a *dense* layer of 128 neurons with *relu* activation, followed by dropout with deactivation rate of 0.5, and a final *softmax* regression layer for categorical outputs.

The classifier is fitted in order to minimize the *categorical cross entropy* wrt the train labels, using the *Adam* optimizer, over 10 epochs, with a constant learning rate of $1e^{-3}$ and with *batch size* of 32.

Autoencoder. The variational autoencoder that facilitates a structured latent representation of digits also relies on convolution and deconvolution layers. The encoder is structured as:

- Two convolutional layers of kernel size 3×3, stride length 2 and *relu* activation; the *filter counts* are 32 and 64 for the first and second layers.
- A *dense* layer of 128 neurons that is fed with the above flattened output; using a *relu* activation.
- Two *linear dense* layers mapping the above 128 neurons to both a distributional mean vector and a log-standard-deviation vector, which define the multivariate distributional mapping in latent space for an image, of dimension 16.
- A *sampling* operation from a normal distribution, used to create a random output from the afore-defined distributional parameters.

In addition, the decoder is defined as:

- A dense layer with *relu* activation, mapping a latent element to a vector of dimensionality $7 \times 7 \times 64$.
- Two deconvolutional layers of kernel size 3×3, stride length 2 and *relu* activation; the *filter counts* are 64 and 32 for the first and second layers.
- An output deconvolutional layer of kernel size 3×3, *filter counts* 1, stride length 1 and *sigmoid* activation for pixel values; which reconstructs a digit image.

The autoencoder is fitted in order to minimize a custom loss, accounting for a digit reconstruction term (through a cross-entropy loss) along with the Kullback-Leibler divergence among latent mappings for each image and a multivariate normal distribution $\mathcal{N}(\mathbf{0}, I)$. We use the *Adam* optimizer, over 50 epochs, with a constant learning rate of $1e^{-3}$ and with *batch size* of 32.

B.2 FASHION-MNIST DATASET

The categorical classifier in this task is defined similarly as in the previous example with MNIST handwritten digits. However, we add two additional *dropout* layers (with dropout probability 0.5) after each of the *max-pooling* operations. The variational autoencoder used in this exercise is identical to the one above. In both cases, training proceeds with the *Adam* optimizer, at a constant learning rate of $1e^{-3}$ with *batch size* 32. The classifier is trained for 10 epochs using the cross-entropy as the cost function, while the autoencoder is trained for 50 epochs using a combination of binary cross-entropy (as reconstruction loss) and the Kullback-Leibler divergence (as a regularisation term for organising latent representations).

B.3 CELEBA DATASET

In this task, images are centred around the face and cropped to size 128×128, further standardized to pixel values in the range $[0, 1]$. During training, we leverage data augmentation with random rotations; we use a *maximum angle* of ± 18 degrees, random translation by a maximum factor of 0.1 and random horizontal flip.

Classifier. The classifier is composed of 6 convolutional blocks followed by a fully connected dense layer with *softmax* activation. Each convolutional block utilizes a *kernel* size of 3 and *stride* size 1, along with *batch normalization*, *dropout* with deactivation probability of 0.2, *relu* activation and *max-pooling* (*pool size* 2 and *stride* 2). The number of channels at the output of each convolutional layer is, respectively, 32, 64, 128, 128, 256 and 256. The last convolutional block is followed by a flattening operation and a *dropout* layer with deactivation probability 0.4.

We train this classifier for 5 epochs using the *Adam* optimizer with batch size 64 and the *cross-entropy* as cost function. The learning rate is decreased after each epoch by a factor of 0.8; starting from $1e^{-4}$ for the *smiling* and *arched eyebrows* classifiers, and $3e^{-5}$ for the *bags under eyes* classifier.

Autoencoder. The encoder within the variational autoencoder is composed by a series of 5 convolutional blocks. Each block shares the same structure, with *kernel* size 3, *stride* 2, *batch normalization* and *leaky-relu* activation with negative slope coefficient of 0.3. The number of filters at the output of each block is 32, 64, 128, 256 and 512. After the last block we insert a flattening layer and two dense layers each with 256 output neurons, which define the multivariate distributional mapping in latent space for an image.

The decoder consists of a fully connected dense layer with 80192 output neurons (reshaped into a $4 \times 4 \times 512$ activation map) followed by 5 up-sampling blocks. Each block up-samples the input by a factor 2 and feeds it into a convolutional layer with kernel size 3 and stride 1, followed by *batch normalisation* and *leaky-relu* activation with 0.3 negative slope coefficient. The number of channels at the output of each block are 256, 128, 64, 32 and 3 respectively. We apply an additional convolutional layer with kernel size 3, stride 1, 3 output channels and *sigmoid* activation for a final reconstructed RGB image with values restricted in the $[0, 1]$ interval.

The variational autoencoder is trained for 100 epochs using the *Adam* optimizer, with batch size 64 and a learning rate of $5e^{-4}$ which is decreased after each epoch by a factor of 0.98. We use a *perceptual loss* function together with the Kullback-Leibler divergence regularisation term, following details on (Hou et al., 2017) (VAE-123 model).

C Additional experimental results

The following are visualizations of importance attributions derived from our work, which complement the experimental results presented within the main body of the paper. In all cases, the results are derived using the same model specifications as described above. Fiducial choices and configurations of competing methods are all implemented as described in Section 4 within the main body of text.

C.1 MNIST handwritten digits

In Figure 9 we find further examples of uncertainty attributions applied to MNIST digits, using the method described in Algorithm 1. The attributions are decomposed across both aleatoric and epistemic terms, and display positive importances for pixels which contribute by *increasing* the uncertainty in classification according to our model. Digits selected represent a range of numbers that have shown high predictive uncertainty according to our model.

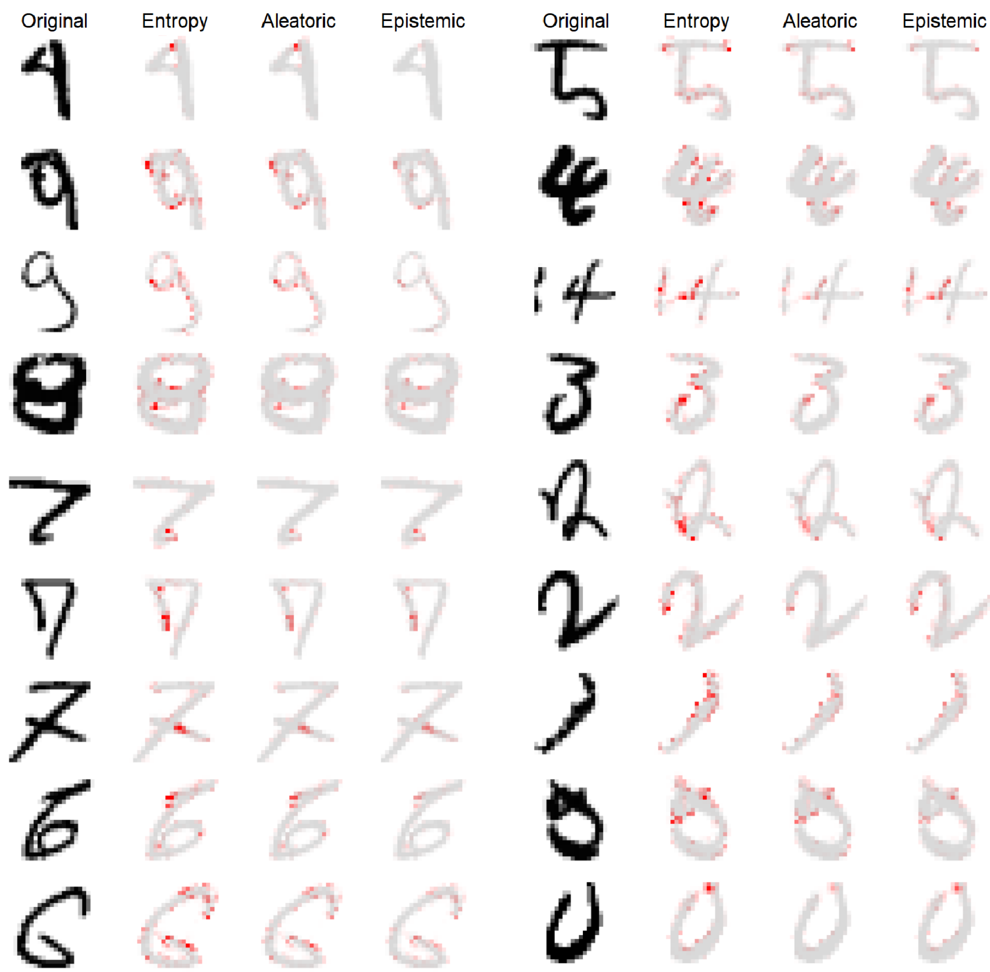

Figure 9: Attributions of uncertainty on MNIST digits.

In Figure 10 we display further comparisons of uncertainty attributions methods in application to MNIST digits with predictive uncertainty. We only display attributions for the entropy, and show both positive (increase uncertainty, in red) and negative (decrease uncertainty, in green) contributions.

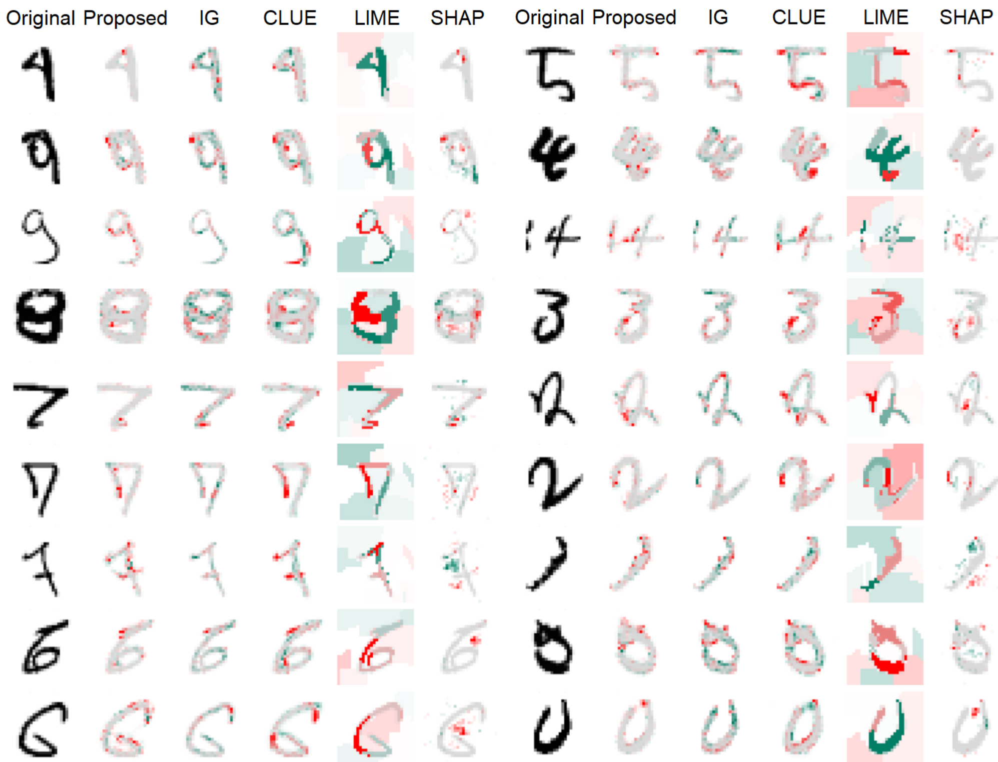

Figure 10: Comparison of uncertainty attributions methods on MNIST digits.

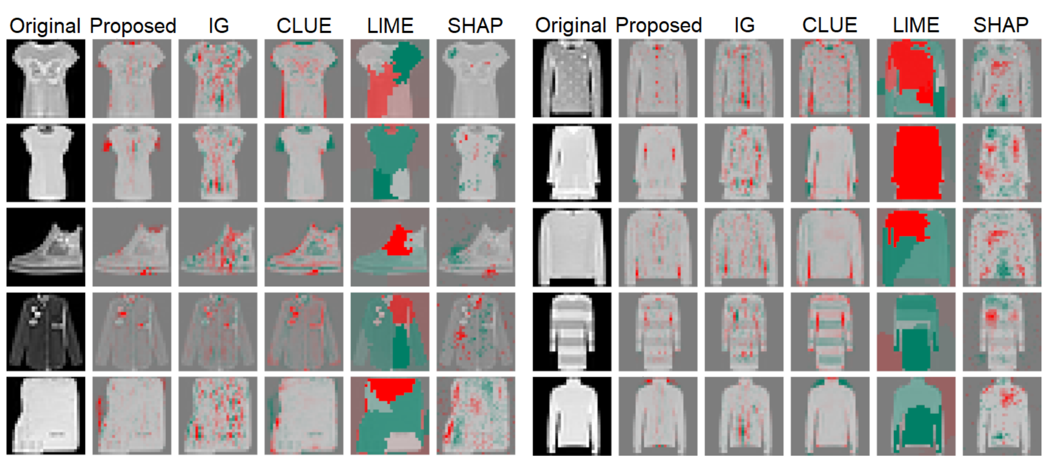

Figure 11: Comparison of uncertainty attributions methods on MNIST-fashion images.

C.2 FASHION-MNIST DATASET

Next, in Figure 11 we display further comparisons of our proposed algorithm along versus competing uncertainty attributions methods, in application to MNIST-fashion images with high predictive uncertainty. Red attributions contribute by increasing uncertainty; green attributions represent contributions towards decreasing uncertainty.

C.3 CELEBA DATASET

Finally, within Figures 12, 13 and 14 we display further comparisons of uncertainty attribution methods in application to CelebA images with high predictive uncertainty. The figures show attributions for models trained (independently) on class labels *smile*, *bags under eyes* and *arched eyebrows*, respectively. Unlike in the main body of the paper, these figures include *LIME* importances along with the rest of the competing approaches for uncertainty attributions.

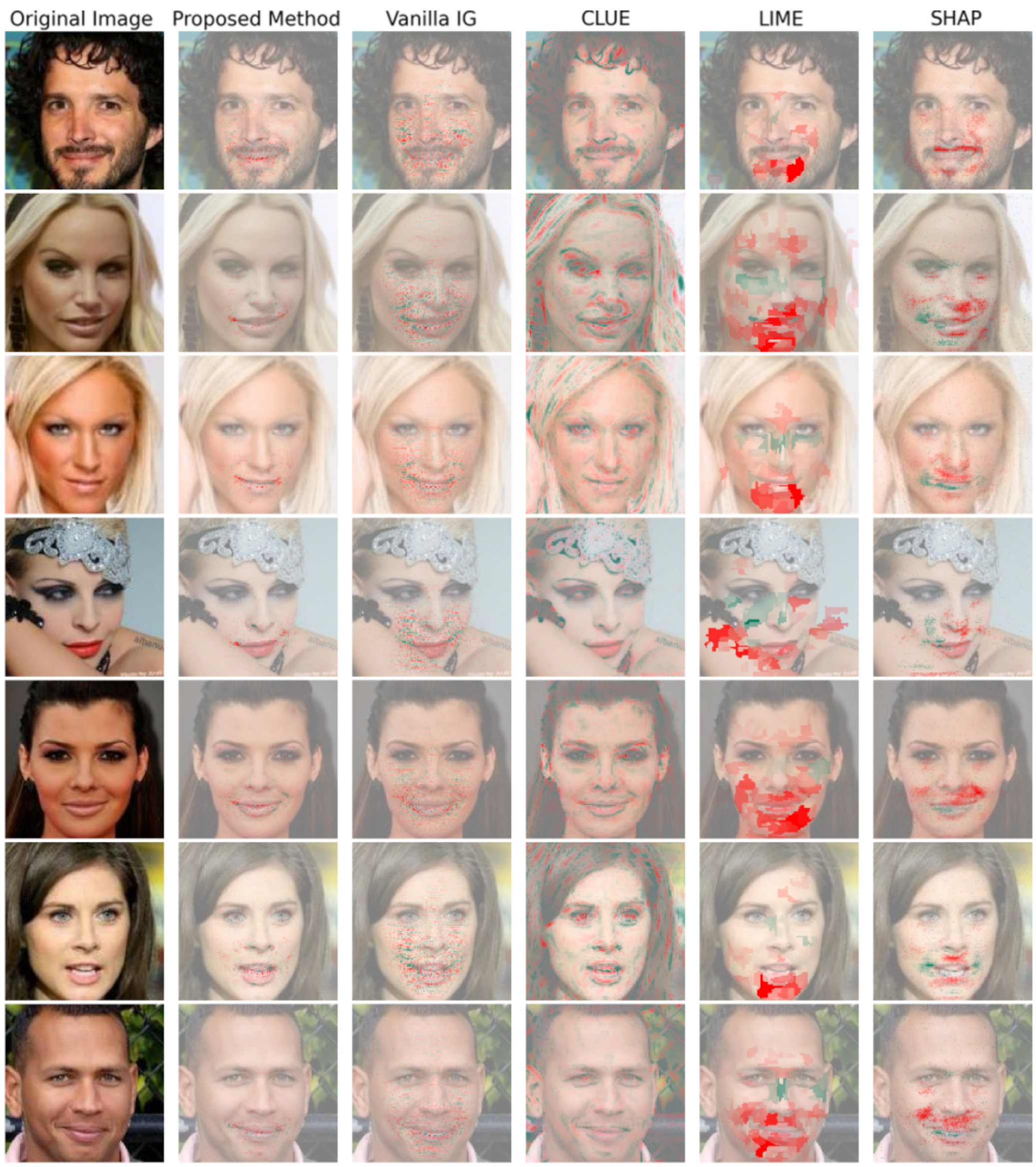

Figure 12: Comparison of uncertainty attributions methods on CelebA images, class label *smile*.

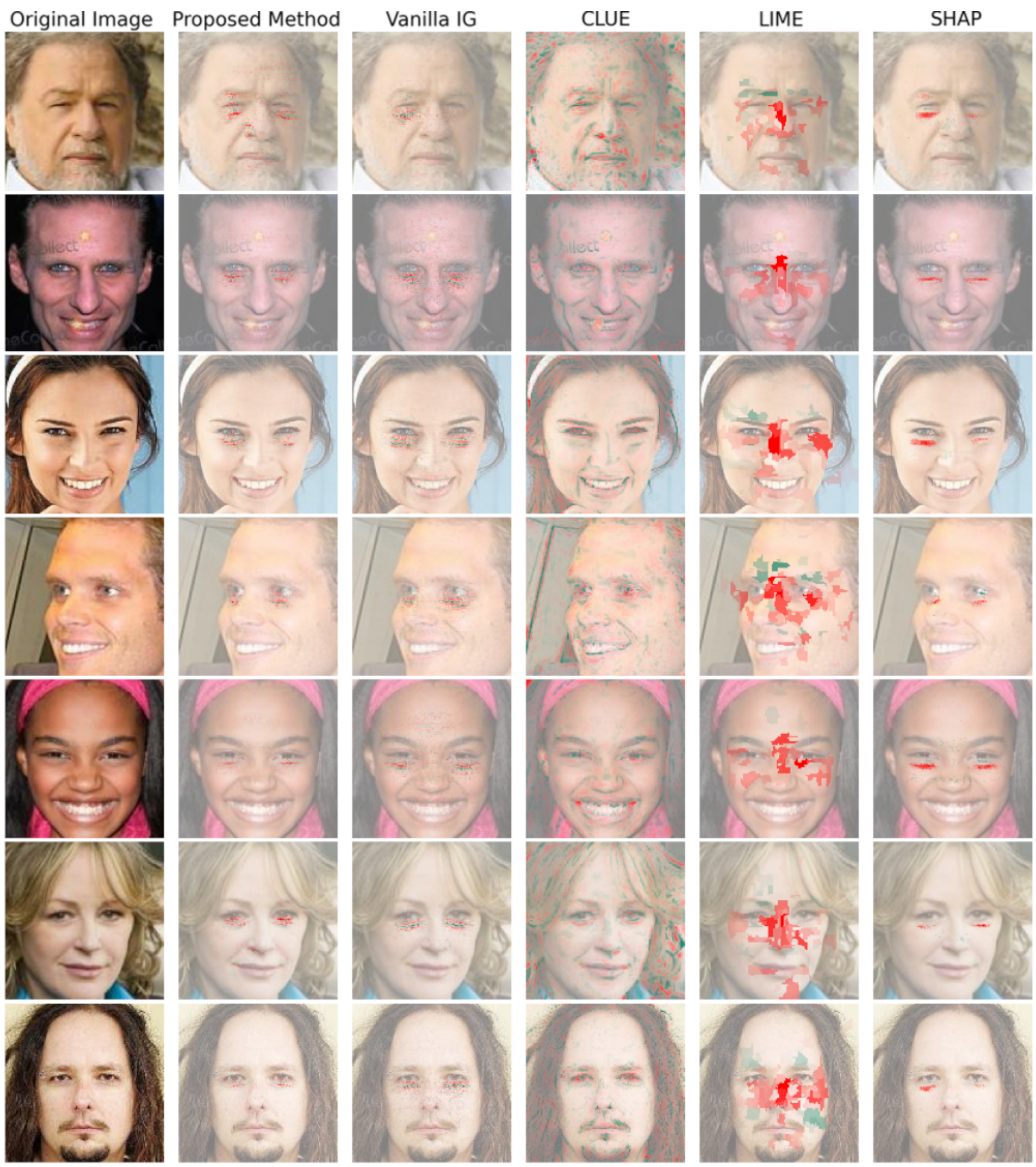

Figure 13: Comparison of uncertainty attributions methods on CelebA images, class label *bags under eyes*.

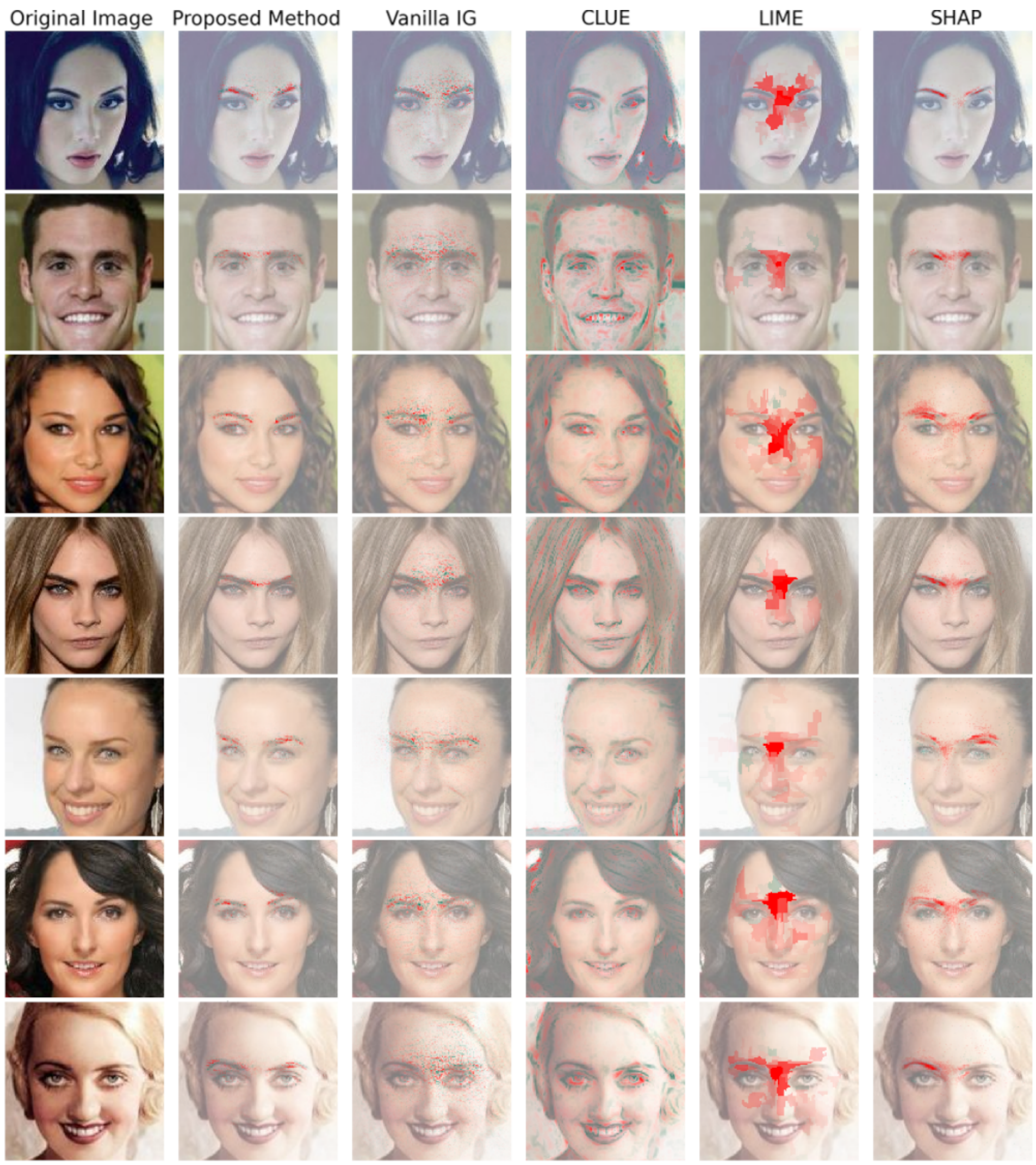

Figure 14: Comparison of uncertainty attributions methods on CelebA images, class label *arched eyebrows*.