Diverse, Global and Amortised Counterfactual Explanations for Uncertainty Estimates

Dan Ley[1], **Umang Bhatt**[1,2], **Adrian Weller**[1,2]
[1]University of Cambridge, UK
[2]The Alan Turing Institute, UK
dwl36@cantab.ac.uk, {usb20, aw665}@cam.ac.uk

Abstract

To interpret uncertainty estimates from differentiable probabilistic models, recent work has proposed generating a single Counterfactual Latent Uncertainty Explanation (CLUE) for a given data point where the model is uncertain, identifying a single, on-manifold change to the input such that the model becomes more certain in its prediction. We broaden the exploration to examine δ-CLUE, the set of potential CLUEs within a δ ball of the original input in latent space. We study the diversity of such sets and find that many CLUEs are redundant; as such, we propose DIVerse CLUE (∇-CLUE), a set of CLUEs which each propose a distinct explanation as to how one can decrease the uncertainty associated with an input. We then further propose GLobal AMortised CLUE (GLAM-CLUE), a distinct and novel method which learns amortised mappings on specific groups of uncertain inputs, taking them and efficiently transforming them in a single function call into inputs for which a model will be certain. Our experiments show that δ-CLUE, ∇-CLUE, and GLAM-CLUE all address shortcomings of CLUE and provide beneficial explanations of uncertainty estimates to practitioners.

Introduction

For models that provide uncertainty estimates alongside their predictions, explaining the source of this uncertainty reveals important information. For instance, determining the features responsible for predictive uncertainty can help to identify in which regions the training data is sparse, which may in turn implicate under-represented sub-groups (by age, gender, race etc). In sensitive settings, domain experts can use uncertainty explanations to appropriately direct their attention to the specific features the model finds anomalous.

In prior work, Adebayo et al. (2020) touch on the unreliability of saliency maps for uncertain inputs, and Tsirtsis, De, and Gomez-Rodriguez (2021) observe that high uncertainty can result in vast possibilities for counterfactuals. Additionally, when models are uncertain, their predictions may be incorrect. We thus consider uncertainty explanations an important precedent for model explanations; only once uncertainty has been explained can state-of-the-art methods be deployed to explain the model's prediction. However, there has been little work in explaining predictive uncertainty.

Depeweg et al. (2017) introduce decomposition of uncertainty estimates, though recent work (Antorán et al. 2021) has demonstrated further leaps, proposing to find an explanation of a model's predictive uncertainty for a given input by searching in the latent space of an auxiliary deep generative model (DGM): they identify a single possible change to the input such that the model becomes more certain in its prediction. Termed CLUE (Counterfactual Latent Uncertainty Explanations), this method aims to generate counterfactual explanations (CEs) on-manifold that reduce the uncertainty of an uncertain input $\mathbf{x}_0$. These changes are distinct from adversarial examples, which find nearby points that change the label (Goodfellow, Shlens, and Szegedy 2015).

CLUE introduces a latent variable DGM with decoder $\mu_\theta(\mathbf{x}|\mathbf{z})$ and encoder $\mu_\phi(\mathbf{z}|\mathbf{x})$. $\mathcal{H}$ refers to any differentiable uncertainty estimate of a prediction $\mathbf{y}$. The pairwise distance metric takes the form $d(\mathbf{x}, \mathbf{x}_0) = \lambda_x d_x(\mathbf{x}, \mathbf{x}_0) + \lambda_y d_y(f(\mathbf{x}), f(\mathbf{x}_0))$, where $f(\mathbf{x}) = \mathbf{y}$ is the model's mapping from an input $\mathbf{x}$ to a label, thus encouraging similarity in input space and/or prediction space. CLUE minimises:

$$\mathcal{L}(\mathbf{z}) = \mathcal{H}\left(\mathbf{y}|\mu_\theta(\mathbf{x}|\mathbf{z})\right) + d\left(\mu_\theta(\mathbf{x}|\mathbf{z}), \mathbf{x}_0\right) \tag{1}$$

to yield $\mathbf{x}_{\text{CLUE}} = \mu_\theta(\mathbf{x}|\mathbf{z}_{\text{CLUE}})$ where $\mathbf{z}_{\text{CLUE}} = \arg\min_{\mathbf{z}} \mathcal{L}(\mathbf{z})$. There are however limitations to CLUE, including the lack of a framework to deal with a diverse set of possible explanations and the lack of computational efficiency. Although finding multiple explanations was suggested, we find the proposed technique to be incomplete.

We start by discussing the multiplicity of CLUEs. Providing practitioners with many explanations for why their input was uncertain can be helpful if, for instance, they are not in control of the recourse suggestions proposed by the algorithm; advising someone to change their age is less actionable than advising them to change a mutable characteristic (Poyiadzi et al. 2020). Specifically, we develop a method to generate a set of possible CLUEs within a δ ball of the original point in the latent space of the DGM used: we term this δ-CLUE. We then introduce metrics to measure the diversity in sets of generated CLUEs such that we can optimise directly for it: we term this ∇-CLUE. After dealing with CLUE's multiplicity issue, we consider how to make computational improvements. As such, we propose a distinct method, GLAM-CLUE (GLobal AMortised CLUE), which serves as a summary of CLUE for practitioners to audit their

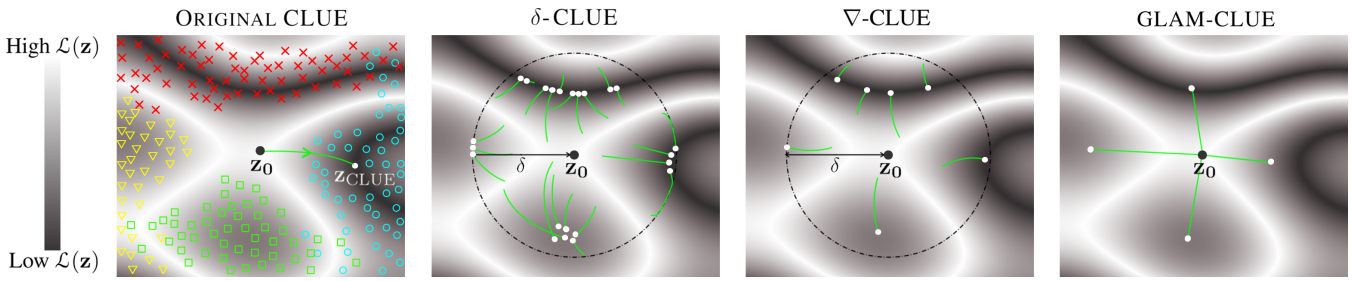


Figure 1: Conceptual colour map of $\mathcal{L}(z)$ with high cost $\mathbf{z}_0$. White circles indicate CLUEs found. Left: Gradient descent to low cost region (original CLUE). Training data shown in colour. Left Centre: Gradient descent constrained to δ-ball. Diverse starting points yield diverse local minima, albeit with redundant solutions (δ-CLUE). Right Centre: Direct optimisation for diversity (∇-CLUE). Right: Efficient, unconstrained mappings without gradient descent (GLAM-CLUE), allowing computational speedups.

model's behavior on uncertain inputs. It does so by finding translations between certain and uncertain groups in a computationally efficient manner. Such efficiency is, amongst other factors, a function of the dataset, the model, and the number of CEs required; there thus exist applications where either ∇-CLUE or GLAM-CLUE is most appropriate.

Multiplicity in Counterfactuals

Constraining CLUEs: δ-CLUE

We propose δ-CLUE (Ley, Bhatt, and Weller 2021), which generates a set of solutions that are all within a specified distance δ of $\mathbf{z}_0 = \mu_\phi(\mathbf{z}|\mathbf{x_0})$ in latent space: $\mathbf{z}_0$ is the latent representation of the uncertain input $\mathbf{x}_0$ being explained. We achieve multiplicity by initialising the search randomly in different areas of latent space. While CLUE suggests this, its random generation method and lack of constraint are prone to a) finding minima in a limited region of the space or b) straying far from this region without control over the proximity of CEs (Appendix B). Figure 1 contrasts the original and proposed objectives (left and left centre respectively).

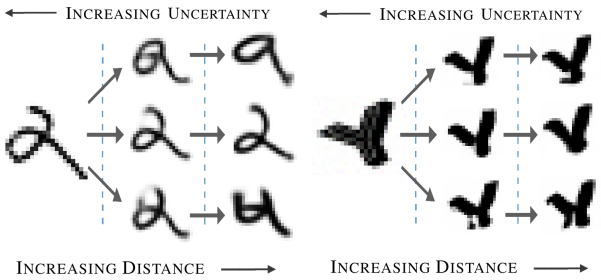
The original CLUE objective uses VAEs (Kingma and Welling 2013) and BNNs (MacKay 1992) as the DGMs and classifiers respectively. The predictive uncertainty of the BNN is given by the entropy of the posterior over the class labels; we use the same measure. The hyperparameters (λ_x, λ_y) control the trade-off between producing low uncertainty CLUEs and CLUEs which are close to the original inputs. To encourage sparse explanations, we take $d_x(\mathbf{x}, \mathbf{x}_0) = \|\mathbf{x} - \mathbf{x}_0\|_1$. We find this to suffice for our datasets, though other metrics such as FID scores (Heusel et al. 2018) could be used in more complex vision tasks for both evaluation (as in Singla et al. (2020)) and optimisation of CEs (see Appendix B). In our proposed δ-CLUE method, the loss function matches Eq 1, with the additional δ requirement as $\mathbf{x}_{\delta-\text{CLUE}} = \mu_\theta(\mathbf{x}|\mathbf{z}_{\delta-\text{CLUE}})$ where $\mathbf{z}_{\delta-\text{CLUE}} = \arg\min_{\mathbf{z}:\ \rho(\mathbf{z},\mathbf{z}_0)\leq\delta} \mathcal{L}(\mathbf{z})$ and $\mathbf{z_0} = \mu_\phi(\mathbf{z}|\mathbf{x_0})$. We choose $\rho(\mathbf{z}, \mathbf{z}_0) = \|\mathbf{z} - \mathbf{z}_0\|_2$ (the ℓ_2 norm) in this paper, as shown in Figure 1. We first set $\lambda_x = \lambda_y = 0$ to explore solely the uncertainty landscape, given that the size of the δ-ball removes the strict need for the distance component in $\mathcal{L}(z)$ and grants control over the locality of solutions, before trialling $\lambda_x = 0.03$. We apply the δ constraint at each stage of the optimisation (Figure 1, left centre), as in Projected Gradient Descent (Boyd, Boyd, and Vandenberghe 2004).

For each uncertain input, we exploit the non-convexity of CLUE's objective to generate diverse δ-CLUEs by initialising in different regions of latent space (Figure 1). While previous work has considered sampling the latent space around an input (Pawelczyk, Broelemann, and Kasneci 2020a), we find that subsequent gradient descent yields improvements. Example results are in Figure 2. δ-CLUE is a special case of Algorithm 1, or explicitly Algorithm 3 (Appendix B).

Diversity Metrics for Counterfactual Explanations

Once we have generated a set of viable CLUEs, we desire to measure the diversity within the set; as such, we require candidate convex similarity functions between points, which could be applied either pairwise or over all counterfactuals. We consider these between counterfactual labels (prediction space) or between counterfactuals themselves (input or latent space). A given diversity function D can be applied to a set of $k > 0$ counterfactuals in an appropriate space i.e. $D(\mathbf{x}_1, ..., \mathbf{x}_k)$, $D(\mathbf{z}_1, ..., \mathbf{z}_k)$ or $D(\mathbf{y}_1, ..., \mathbf{y}_k)$ where $\mathbf{x}_i \in \mathbb{R}^{d'}$, $\mathbf{z}_i \in \mathbb{R}^{m'}$ and $\mathbf{y}_i \in \mathbb{R}^{c'}$ (we define the hard prediction $y_i = \max_j(\mathbf{y}_i)_j$). Table 1 summarises the metrics.

Leveraging Determinantal Point Processes: We build on Mothilal, Sharma, and Tan (2020) to leverage determinantal point processes, referred to as DPPs (Kulesza 2012),



(a) Uncertain $2 \rightarrow 9, 2, 4$. (b) Uncertain Y $\rightarrow$ Y, V, X.

Figure 2: Visualisation of the trade-off between uncertainty $\mathcal{H}$ and distance d. Left: MNIST. Right: Synbols.

Diversity Metric	Function (D)
Determinantal Point Processes	$\det(\mathbf{K})$ where $\mathbf{K}_{i,j} = \frac{1}{1 + d(\mathbf{x}_i, \mathbf{x}_j)}$
Average Pairwise Distance	$\frac{1}{\binom{k}{2}} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(\mathbf{x}_i, \mathbf{x}_j)$
Coverage	$\frac{1}{d'} \sum_{i=1}^{d'} (\max_j(\mathbf{x}_j - \mathbf{x}_0)_i + \max_j(\mathbf{x}_0 - \mathbf{x}_j)_i)$
Prediction Coverage	$\frac{1}{c'} \sum_{i=1}^{c'} \max_j[(\mathbf{y}_j)_i]$
Distinct Labels	$\frac{1}{c'} \sum_{j=1}^{c'} \mathbf{1}_{[\exists i \,:\, y_i = j]}$
Entropy of Labels	$-\frac{1}{\log c'} \sum_{j=1}^{c'} p_j(k) \log p_j(k)$

Table 1: Diversity metrics, D. If necessary, we define $D = 0$ for $k = 1$ and take d to be some arbitrary distance metric.

as $\det(\mathbf{K})$ in Table 1. DPPs implicitly normalise to $0 \leq D \leq 1$. This metric is effective overall and achieves diversity by diverting attention away from the most popular (or salient) points to a diverse group of points instead. However, matrix determinants are computationally expensive for large k.

Diversity as Average Pairwise Distance: We can calculate diversity as the average distance between all distinct pairs of counterfactuals (as in Bhatt et al. (2021)). While we can adjust for the number of pairs (accomplishing invariance to k), this metric does not satisfy $0 \leq D \leq 1$, scaling instead with the pairwise distances characterised by the dataset.

Coverage as a Diversity Metric: Previous work in interpretability has leveraged the notion of coverage as a measure of the quality of sets of CEs. Ribeiro, Singh, and Guestrin (2016) define coverage to be the sum of distinct features contained in a set, weighted by feature importance: this could be applied to CEs to suggest a way of optimally choosing a subset from a full set of CEs. Plumb et al. (2020) introduce coverage as a measure of the quality of global CEs. Herein, we interpret coverage as a measure of diversity, using it directly for optimisation and evaluation of CEs. The metric, as given in Table 1, rewards changes in both positive and negative directions separately (though penalises a lack of changes in positive/negative directions). See Appendix C.

Prediction Coverage: Since rewarding negative changes in $\mathbf{y}$-space is redundant (maximising the prediction of one label implicitly minimises the others), we adjust the coverage metric in $\mathbf{y}$-space to be the maximum prediction for a particular label found in a set of CEs, averaged over all predictions. This satisfies $\frac{1}{c'} \leq D \leq 1$, where we require at least $k = c'$ CEs to achieve $D = 1$, equivalent to finding at least one fully confident prediction for each label.

Targeting Diversity of Class Labels: While recent work focuses on producing diverse explanations for binary classification problems (Russell 2019) and others summarise current methods therein (Pawelczyk, Broelemann, and Kasneci 2020b), these metrics perform well in applications rich in class labels, and conversely are likely ineffective in binary

Algorithm 1: ∇-CLUE (**simultaneous**)

Inputs: δ, k, $\mathcal{S}$, r, $\mathbf{x}_0$, d, ρ, $\mathcal{H}$, μ_θ, μ_ϕ, D, λ_D

Initialise $\varnothing$ of CLUEs: $X_{\text{CLUE}} = \{\}$;
Set δ-ball centre of $\mathbf{z}_0 = \mu_\phi(\mathbf{z}|\mathbf{x}_0)$;
for $1 \leq i \leq k$ **do**
  Set initial value of $\mathbf{z}_i = \mathcal{S}(\mathbf{z}_0, r, i, k)$;
end for
while *loss $\mathcal{L}$ has not converged* **do**
  for $1 \leq i \leq k$ **do**
    Decode: $\mathbf{x}_i = \mu_\theta(\mathbf{x}|\mathbf{z}_i)$;
    Use predictor to obtain $\mathcal{H}(\mathbf{y}|\mathbf{x}_i)$;
    $\mathcal{L}(\mathbf{z}_i) = \mathcal{H}(\mathbf{y}|\mathbf{x}_i) + d(\mathbf{x}_i, \mathbf{x}_0)$;
  end for
  $\mathcal{L}(\mathbf{z}_1, ..., \mathbf{z}_k) = -\lambda_D D(\mathbf{z}_1, ..., \mathbf{z}_k) + \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}(\mathbf{z}_i)$;
  Update $\mathbf{z}_1, ..., \mathbf{z}_k$ with $\nabla_{\mathbf{z}_1, ..., \mathbf{z}_k} \mathcal{L}(\mathbf{z}_1, ..., \mathbf{z}_k)$;
  for $1 \leq i \leq k$ **do**
    Constrain $\mathbf{z}_i$ to δ ball using $\rho(\mathbf{z}_i, \mathbf{z}_0)$;
  end for
end while
for $1 \leq i \leq k$ **do**
  Decode explanation: $\mathbf{x}_i = \mu_\theta(\mathbf{x}|\mathbf{z}_i)$;
  if $\mathcal{H}(\mathbf{y}|\mathbf{x}_i) < \mathcal{H}_{\text{threshold}}$ **then**
    $X_{\text{CLUE}} \leftarrow X_{\text{CLUE}} \cup \mathbf{x}_i$;
  end if
end for

Outputs: X_{CLUE}, a set of $n \leq k$ diverse CLUEs

tasks. Posterior probabilities are defined as $\mathbf{y} \in \mathbb{R}^{c'}$ and $y_i = \arg\max_i \mathbf{y}_i$. We define the probability of class j as $p_j(k) = \frac{\sum_{i=1}^{k} \mathbf{1}_{[y_i = j]}}{k} = \frac{\text{number of counterfactuals in class } j}{\text{number of counterfactuals}}$. Using this, we suggest diversity through the **Number of Distinct Labels** found, as well as the **Entropy of the Label Distribution**. The former metric loses its effect once all labels are found, whereas the latter does not. The former satisfies $0 \leq D \leq 1$, and given that the maximum entropy of a c' dimensional distribution is $\log(c')$, so too does the latter.

Optimizing for Diversity: ∇-CLUE

The diversity metrics defined in Table 1 find utility in the optimisation of a set of k counterfactuals. We optimise for diversity in the CLUEs we generate through an explicit diversity term in our objective for the CLUEs found. We call this DIVerse CLUE or ∇-CLUE. We posit that whilst some aforementioned metrics may perform poorly during optimisation, we retain them for evaluation.

Once the diversity metric is selected, the optimisation of k counterfactuals can be performed **simultaneously** (Algorithm 1) in latent space (Mothilal, Sharma, and Tan 2020), or **sequentially** (Appendix D), where the approach is analogous to a greedy algorithm of the former approach. The notation $X_{\text{CLUE}} = \{\mathbf{x}_1, ..., \mathbf{x}_k\}$ is adopted to represent a set of k counterfactuals (similarly Z_{CLUE} and Y_{CLUE}).

We denote an initialisation scheme $\mathcal{S}$ of radius r to generate starting points for the gradient descent. Note that the removal of the δ constraint or the initialisation may be

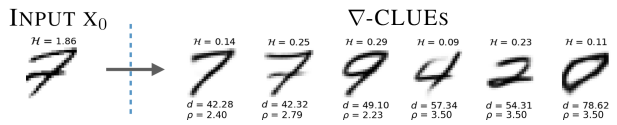


Figure 3: We produce a diverse set of candidate explanations that show how to reduce uncertainty while remaining close to x_0 in input/latent space ($\mathcal{H}$ is uncertainty, d is input distance, ρ is latent distance). We see that x_0 might most easily be resolved into a confident 7 or 9. Results are taken from a larger ∇-CLUE set and are not exemplary of setting $k = 5$.

achieved at $\delta = \infty$ and $r = 0$ respectively (although the latter yields the same counterfactual k times as a result of symmetry). Thus, **the ∇-CLUE algorithm is equivalent to δ-CLUE when $\boldsymbol{\lambda_D = 0}$**, which is itself equivalent to the original CLUE algorithm when $\delta = \infty$, $r = 0$ and $k = 1$. Example results are in Figure 3.

Simultaneous Diversity Optimisation (Algorithm 1): By optimising simultaneously over k counterfactuals in latent space, issues with how the diversity metric D might scale with k can be avoided. We have the simultaneous optimisation problem of minimising $\mathcal{L}(\mathbf{z}_1, ..., \mathbf{z}_k) = -\lambda_D D(\mathbf{z}_1, ..., \mathbf{z}_k) + \frac{1}{k}\sum_{i=1}^{k} \mathcal{L}(\mathbf{z}_i)$ where $\mathcal{L}(\mathbf{z}_i) = \mathcal{H}(\mathbf{y}|\mu_\theta(\mathbf{x}|\mathbf{z}_i)) + d(\mu_\theta(\mathbf{x}|\mathbf{z}_i), \mathbf{x}_0)$, to yield $X_{\text{CLUE}} = \mu_\theta(X|Z_{\text{CLUE}})$ where $Z_{\text{CLUE}} = \arg\min_{\mathbf{z}_1,...,\mathbf{z}_k} = \mathcal{L}(\mathbf{z}_1, ..., \mathbf{z}_k)$. Note that we apply the diversity function in latent space; it could equally be applied in input space.

Sequential Diversity Optimisation (Appendix D): Given a set of counterfactuals Z_{CLUE} (initially the empty set $\varnothing$), we can apply ∇-CLUE sequentially, appending each new counterfactual to the set. At each iteration, we minimise $\mathcal{L}(\mathbf{z}) = \lambda_D D(Z_{\text{CLUE}} \cup \mathbf{z}) + \mathcal{H}(\mathbf{y}|\mu_\theta(\mathbf{x}|\mathbf{z})) + d(\mu_\theta(\mathbf{x}|\mathbf{z}), \mathbf{x}_0)$ to yield $\mathbf{z}_{\text{CLUE}}$ which we append to the set.

Global and Amortised Counterfactuals

CLUE primarily focuses on local explanations of uncertainty estimates, as Antorán et al. (2021) propose a method for finding a single, small change to an uncertain input that takes it from uncertain to certain with respect to a classifier. Such local explanations can be computationally expensive to apply to large sets of inputs. Large sets of counterfactuals are also difficult to interpret. We thus face challenges when using them to summarise global uncertainty behaviour, which is important in identifying areas in which the model does not perform as expected or the training data is sparse.

We desire a computationally efficient method that requires a finite portion of the dataset (or a finite set of CEs) from which global properties of uncertainty can be learnt and applied to unseen test data with high reliability. We propose GLAM-CLUE (GLobal AMortised CLUE), which achieves such reliability with considerable speedups.

Proposed Method: GLAM-CLUE

GLAM-CLUE takes groups of high/low certainty points and learns mappings of arbitrary complexity between them in latent space (**training step**). Mappers are then applied to generate CEs from uncertain inputs (**inference step**). It can be seen as a global equivalent to CLUE. Initially, inputs are taken from the training data to learn such mappings, but we demonstrate that we can make improvements by instead using CLUEs generated from uncertain points in the training data. Algorithm 2 defines a mapper of arbitrary complexity from uncertain groups to certain groups in latent space: $\mathbf{z}_{\text{certain}} = G(\mathbf{z}_{\text{uncertain}})$. These mappers have parameters $\boldsymbol{\theta}$.

Algorithm 2: GLAM-CLUE (**Training Step**)

Inputs: $X_{\text{uncertain}}, X_{\text{certain}}$, groups $Y_{\text{uncertain}}, Y_{\text{certain}}$, DGM encoder μ_ϕ, loss $\mathcal{L}$, trainable parameters $\boldsymbol{\theta}$

for all groups $(i \to j)$ in $(Y_{\text{uncertain}}, Y_{\text{certain}})$ **do**
  Select X_i from $X_{\text{uncertain}}$ given $Y_{\text{uncertain}}$;
  Select X_j from X_{certain} given Y_{certain};
  Encode: $Z_i = \mu_\phi(Z|X_i)$;
  while *loss $\mathcal{L}$ has not converged* **do**
    Update $\boldsymbol{\theta}_{i\to j}$ with $\nabla_{\boldsymbol{\theta}_{i\to j}} \mathcal{L}(\boldsymbol{\theta}_{i\to j}|Z_i, X_j)$;
  end while
end for

Outputs: A collection of mapping parameters $\boldsymbol{\theta}_{i\to j}$ for given mappers $G_{i\to j}$ that take uncertain inputs from group i and produce nearby certain outputs in group j

To strive for global explanations, we restrict each mapper in our experiments to be a single latent translation from an uncertain class i to a certain class j: $\mathbf{z}_j = G_{i\to j}(\mathbf{z}_i) = \mathbf{z}_i + \boldsymbol{\theta}_{i\to j}$. When run on test data, mappers should reduce the uncertainty of points while keeping them close to the original. To train the parameters of the translation $\boldsymbol{\theta}$, we use the loss function detailed in Equation 2, similar to Van Looveren and Klaise (2021), who inspect the k nearest data points (our min operation implies $k = 1$). We infer from Figure 7, right, that regularisation in latent space implies regularisation in input space. We learn separate mappers for each pair of groups defined by the practitioner (Figure 6); Algorithm 2 loops over these groups, partitioning the data accordingly, and returning distinct parameters $\boldsymbol{\theta}_{i\to j}$ for each case.

$$\mathcal{L}(\boldsymbol{\theta}|Z_{\text{uncertain}}, X_{\text{certain}}) =$$
$$\lambda_\theta \|\boldsymbol{\theta}\|_1 + \frac{1}{|Z_{\text{uncertain}}|} \sum_{\mathbf{z}\in Z_{\text{uncertain}}} \min_{\mathbf{x}\in X_{\text{certain}}} \|\mu_\theta(\mathbf{z}+\boldsymbol{\theta}) - \mathbf{x}\|_2^2 \quad (2)$$

Few works in the counterfactual literature address uncertainty explanations; we avoid comparison with state-of-the-art CE methods for the reasons discussed in the introduction, but there do exist standard baselines we can test against. We can perform Difference Between Means (DBM) of uncertain data and certain data in either input or latent space. This can be added to uncertain test data and reconstructed. Another baseline is the Nearest Neighbours (NN) in high certainty training data, in either input or latent space. Figure 5 visualises these baselines in latent space. Our experiments demonstrate that GLAM-CLUE outperforms these baselines significantly, and performs on par with CLUE. Pawelczyk et al. (2021) create a benchmarking tool which shows that CLUE performs on par with the current state-of-the-art. By extension, so too does our scheme, but 200 times faster.

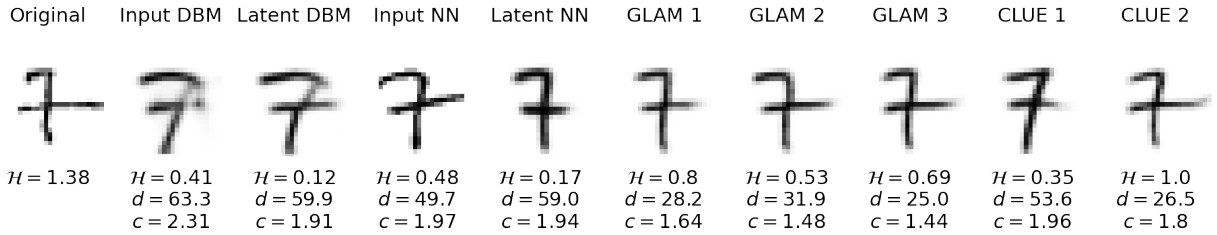


Figure 4: Comparison of CLUEs generated for an uncertain input (left) by the baselines/GLAM-CLUE/CLUE. $\mathcal{H}$ is uncertainty, d is input distance, $c = \mathcal{H} + \lambda_x d$ is cost. Low uncertainties in some baseline schemes are invalidated by unrealistic distances. GLAM 1/2/3 are described in the Experiments section. CLUE 1/2 are generated from $\lambda_x = 0$ and $\lambda_x = 0.03$ respectively.

When the class of uncertain test data is unknown, mappings could be applied over each combination of classes, picking the best performing CEs. When the number of classes is large, a scheme to select a limited number of these (e.g. the top n predictions from the classifier) could be used. Generic mappings from uncertainty to certainty would not require this selection but on the whole would be harder to train (simple translations are likely invalid for the far right case of Figure 6). We posit that more complex models such as neural networks could improve the performance of mappings at the risk of losing the global sense of the explanation.

Grouping Uncertainty

Most counterfactual explanation techniques center around determining ways to change the class label of a prediction; for example, Transitive Global Translations (TGTs) consider each possible combination of classes and the mappings between them (Plumb et al. 2020). We choose here to partition the data into classes, but also into certain and uncertain groups according to the classifier used. By using these partitions, we learn mappings from uncertain points to certain points, either within specific classes or in the general case. While TGTs constrain a mapping G from group i to j to be symmetric ($G_{i\to j} = G^{-1}_{j\to i}$) and transitive ($G_{i\to k} = G_{j\to k} \circ G_{i\to j}$), we see no direct need for the symmetry constraint. There exists an infinitely large domain of uncertain points, unlike the bounded domain for certain points, implying a 'many-to-one' mapping. We also forgo the transitivity constraint: defining direct mappings from uncertain points to specific certain points is sufficient.

Our method is general to all schemes (and more) in Figure 6. Our experiments consider the groups to be class labels, testing against the far left scheme which considers mapping from uncertain points to certain points within a given class. Future work may consider modes within classes, as well as the more general far right scheme of learning mappings from arbitrary uncertain inputs to their certain analogues. The original CLUE method is analogous to the far right scheme, which is agnostic to the particular classes it maps to and from (although struggles with diverse mappings).

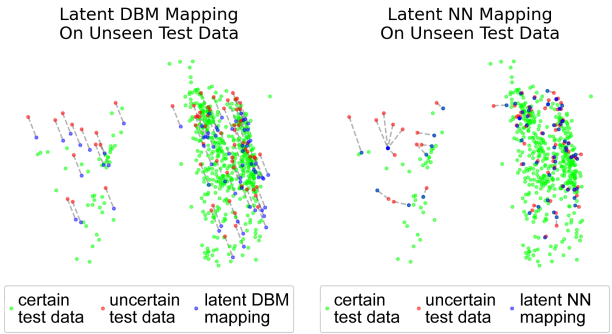


Figure 5: 2D latent space visualisation of DBM/NN baselines (MNIST digit 4). Left/Right: Uncertain points in the test data with their respective latent DBM/latent NN mappings. High certainty test data shown in green throughout.

Experiments

We perform experiments on 3 datasets to validate our methods: UCI Credit classification (Dua and Graff 2017), MNIST image classification (LeCun 1998) and Synbols image classification (Lacoste et al. 2020). On Credit and MNIST, we train VAEs as our DGMs (Kingma and Welling 2013) and BNNs for classification (MacKay 1992). For Synbols, we train Hierarchical VAEs (Zhao, Song, and Ermon 2017) and a resnet deep ensemble, owing to higher dataset complexity (rotations, sizes and obscurity of shapes). We demonstrate that our constraints allow practitioners to better control the uncertainty-distance trade-off of CEs (δ-CLUE) and the diversity of CEs (∇-CLUE). We then show that we can efficiently generate explanations that apply globally to groups of inputs with our amortised scheme (GLAM-CLUE).

δ-CLUE

We learn from the δ-CLUE experiments that the δ value controls the trade-off between the uncertainty of the CLUEs generated and their distance from the original point (Figure 2). Importantly, by tuning λ_x in the distance term d of Equation 1, we achieve lower distances with only small uncertainty increases (Figure 7, right). We observe further in

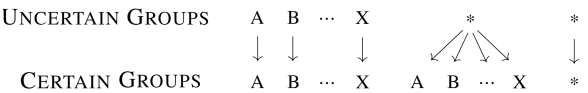


Figure 6: Example mappings from uncertainty to certainty in groups A to X, without necessarily satisfying symmetry or transitivity. Asterisks represent membership to any group.

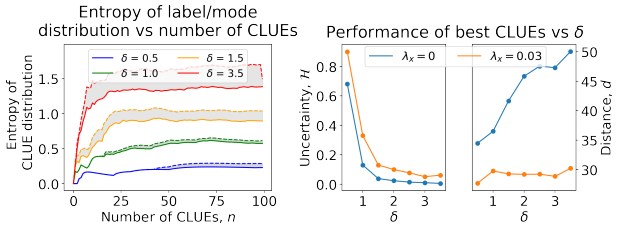


Figure 7: Left: MNIST diversity analysis. Entropy of the label distribution (solid) and modes (dashed) found as number of CLUEs increases. There exist multiple modes within each label of 0 to 9. Observe the entropy saturating as we converge to all minima within the δ ball. Right: Performance of δ-CLUEs (uncertainty, $\mathcal{H}$, distance, d). Batch size: 8 most uncertain MNIST digits. Learning rate: 0.1. Iterations: 30.

Figure 7, left that diversity increases with δ, although a large number of CLUEs can be required before such levels become saturated (left). Modes are defined as groups of points within specific classes. Full analysis in Appendix B.

Takeaway: δ-CLUE produces a high performing set of diverse explanations. However, we require many iterations to achieve such diversity (∇-CLUE addresses this).

∇-CLUE

We perform an ablative study, increasing the diversity weight λ_D and optimising the DPP diversity metric in **z**-space, measuring the effect that this has on each other metric. We use the simultaneous ∇-CLUE scheme in Algorithm 1 for a fixed number of $k = 10$ CLUEs and parameters: $\delta = r = 4$ for MNIST; $\delta = r = 1$ for UCI Credit. The optimal δ value(s) can be determined through experimentation (Figure 7, right), although Appendix B discusses alternative methods such as inspecting nearest neighbours in the data.

Takeaway: When optimising for one diversity metric, increasing λ_D monotonically improves diversity by almost every other metric. Uncertainty suffers minimally relative to the gains we achieve in diversity and ∇-CLUE requires fewer CLUEs to achieve the same diversity level as δ-CLUE.

GLAM-CLUE

Gradient descent at the inference step (generation of CEs) is expensive. Uncertainty estimates, distance metrics, and diversity metrics (notably DPPs, which operate on $k \times k$ matrices) all require evaluation over many iterations, to yield only a single CE. While local CEs have utility in certain settings, GLAM-CLUE returns CEs for all uncertain points in a single, amortised function call, permitting considerable speedups. We demonstrate that these counterfactuals outperform the baselines, achieving lower variance also.

We train 3 mappers: GLAM 1 learns from all certain and uncertain 4s in the MNIST training data; GLAM 2/3 learn from all uncertain 4s in the training data and their corresponding certain CLUEs, for $\lambda_x = 0$ and $\lambda_x = 0.03$ respectively. Figure 9 shows improvements when using GLAM 2 and 3, demonstrating that CLUEs capture properties of uncertainty more reliably than the training data, at the expense of extra computation time to generate the CLUEs used.

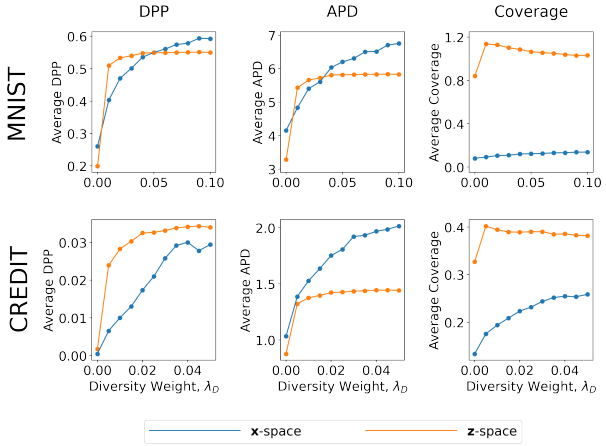


Figure 8: Effect of λ_D on diversity. Row 1: MNIST. Row 2: Credit. Columns 1 to 3: DPP, APD and Coverage metrics on $k = 10$ ∇-CLUEs. $\lambda_D = 0$ is δ-CLUE. Batch size: 8 most uncertain inputs. Learning rate: 0.1. Iterations: 30.

We observe that while the baseline schemes achieve low uncertainties, they do so at the expense of moving much further away from the input (Figure 4), implying infeasible actionability. An advantage to GLAM-CLUE is that the uncertainty-distance trade-off can be tuned with λ_θ in Equation 2: larger λ_θ restricts translations in latent space, thus lowering distances in input space but raising uncertainties. For a given λ_x, GLAM-CLUE's fast learning rate allows for the optimal λ_θ to be determined quickly. Furthermore, 98% of uncertain 4 to certain 4 GLAM-CLUE mappings resulted in a classification of 4 (87% for CLUE which simply minimises uncertainty and is not class specific).

Takeaway: Amortisation of counterfactuals works. A simple global translation for class specific points is shown to produce counterfactuals of comparable quality to CLUE. Notably, performance of GLAM-CLUE is improved when training on CLUEs rather than training data, optimal when we generate CLUEs using $\lambda_x = 0.03$, as used in evaluation.

Computational Speedup

At the inference step, GLAM-CLUE performs significantly faster than CLUE in terms of **average CPU time**, detailed in Table 2. For uncertain 4s in the MNIST test set, CLUE required on average 220 seconds to converge; GLAM-CLUE took around 1 second to compute. The bottleneck in these

Input DBM	Latent DBM	Input NN
0.0306	0.0262	0.0236
Latent NN	**GLAM-CLUE**	**CLUE**
0.0245	0.0238	4.68

Table 2: Avg. time in seconds for 1 MNIST CE (inference step). Credit achieves similar speedups (186 times faster).

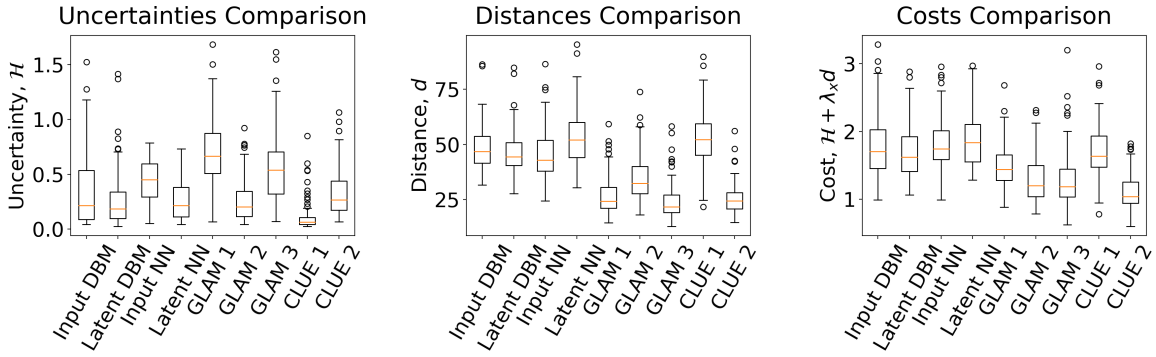


Figure 9: GLAM-CLUE vs baselines when mapping uncertain 4s to certain 4s in MNIST. Left: Distributions of $\mathcal{H}$ (original values exceed 1.5). Centre: Distributions of d. Right: Distributions of total costs, $\mathcal{H} + \lambda_x d$, with $\lambda_x = 0.03$ as used by Antorán et al. (2021). Similar for all classes (Appendix E). CLUE 1/CLUE 2 are generated from $\lambda_x = 0$ and $\lambda_x = 0.03$ respectively. Batch size: 6000 (all 4s in training set). Learning rate: 0.1. Iterations: 30. Multiple random seed runs yield negligible differences.

processes is the uncertainty evaluation of the BNN, and as such these timings are not necessarily representative of all models. A drawback to GLAM-CLUE is that the optimisation required on average 17.6 seconds to train. Should CLUEs be included during training (i.e. GLAM 2 and 3), extra time is required to obtain these. Moving beyond basic mappers to more advanced models, we expect performance to improve at the cost of an increased training step time.

Takeaway: GLAM-CLUE produces explanations around 200 times faster than CLUE. This speedup, alongside the baselines, means that we have the option to take the best performing counterfactual out of GLAM-CLUE and the baselines, without requiring significant computation.

Related and Future Work

The majority of this paper is dedicated to increasing the practical utility of the uncertainty explanations proposed as CLUE in Antorán et al. (2021), and we mitigate CLUE's multiplicity and efficiency issues. Very few works address explaining the uncertainty of probabilistic models. Booth et al. (2020) take a user-specified level of uncertainty for a sample in an auxiliary discriminative model and generate the corresponding sampling using deep generative models (DGM). Joshi et al. (2018) propose xGEMs that use a DGM to find CEs (as we do) though not for uncertainty. Mothilal, Sharma, and Tan (2020) and Russell (2019) use linear programs to find a diverse set of CEs, though also not for uncertainty. Neither paper considers computational advances nor ventures to consider global CEs, as we do. Plumb et al. (2020) define a mapper that transforms points from one low-dimensional group to another. Mahajan, Tan, and Sharma (2020) and Yang et al. (2021) redesign DGMs to generate CEs quickly, similar to GLAM-CLUE. In spirit of such works, we propose amortising CLUE to find a transformation that leads the model to treat the transformed uncertain points from Group A as certain points from Group B. This method could extend beyond CLUE to other classes of CEs.

Future explorations include higher dimensional datasets such as CIFAR10 (Krizhevsky 2012) and CelebA (Liu et al. 2015) that would fully test CLUE and the extensions proposed in this paper, potentially requiring the use of FID scores (Heusel et al. 2018) to replace the simple distance metric in both evaluation (Singla et al. 2020) and optimisation. DGM alternatives such as GANs (Goodfellow et al. 2014) could be explored therein. Further, since Antorán et al. (2021) demonstrate success on human subjects in the use of DGMs for counterfactuals, our reasoning is that we can hope to retain this efficacy with our extensions of CLUE, though ideally additional human experiments would further validate our methods. Multiple runs at various random seeds would also shed light on the sensitivity of the ∇-CLUE algorithm.

Conclusion

Explanations from machine learning systems are receiving increasing attention from practitioners and industry (Bhatt et al. 2020). As these systems are deployed in high stakes settings, well-calibrated uncertainty estimates are in high demand (Spiegelhalter 2017). For a method to interpret uncertainty estimates from differentiable probabilistic models, Antorán et al. (2021) propose generating a Counterfactual Latent Uncertainty Explanation (CLUE) for a given data point on which the model is uncertain. In this work, we examine how to make CLUEs more useful in practice. We devise δ-CLUE, a method to generate a set of potential CLUEs within a δ ball of the original input in latent space, before proposing DIVerse CLUE (∇-CLUE), a method to find a set of CLUEs in which each proposes a distinct explanation for how to decrease the uncertainty associated with an input (to tackle the redundancy within δ-CLUE). However, these methods prove to be potentially computationally inefficient for large amounts of data. To that end, we propose GLobal AMortised CLUE (GLAM-CLUE), which learns an amortised mapping that applies to specific groups of uncertain inputs. GLAM-CLUE efficiently transforms an uncertain input in a single function call into an input that a model will be certain about. We validate our methods with experiments, which show that δ-CLUE, ∇-CLUE, and GLAM-CLUE address shortcomings of CLUE. We hope our proposed methods prove beneficial to practitioners who seek to provide explanations of uncertainty estimates to stakeholders.

Acknowledgments

UB acknowledges support from DeepMind and the Leverhulme Trust via the Leverhulme Centre for the Future of Intelligence (CFI) and from the Mozilla Foundation. AW acknowledges support from a Turing AI Fellowship under grant EP/V025379/1, The Alan Turing Institute, and the Leverhulme Trust via CFI. The authors thank Javier Antorán and Gregory Plumb for their helpful comments and pointers.

References

Adebayo, J.; Muelly, M.; Liccardi, I.; and Kim, B. 2020. Debugging Tests for Model Explanations. In *Advances in Neural Information Processing Systems*.

Antorán, J.; Bhatt, U.; Adel, T.; Weller, A.; and Hernández-Lobato, J. M. 2021. Getting a CLUE: A Method for Explaining Uncertainty Estimates. In *International Conference on Learning Representations*.

Bhatt, U.; Chien, I.; Zafar, M. B.; and Weller, A. 2021. DIVINE: Diverse Influential Training Points for Data Visualization and Model Refinement. arXiv:2107.05978.

Bhatt, U.; Xiang, A.; Sharma, S.; Weller, A.; Taly, A.; Jia, Y.; Ghosh, J.; Puri, R.; Moura, J. M.; and Eckersley, P. 2020. Explainable Machine Learning in Deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 648–657.

Booth, S.; Zhou, Y.; Shah, A.; and Shah, J. 2020. BayesTrEx: Model Transparency by Example. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*.

Boyd, S.; Boyd, S. P.; and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge university press.

Depeweg, S.; Hernández-Lobato, J. M.; Doshi-Velez, F.; and Udluft, S. 2017. Uncertainty Decomposition in Bayesian Neural Networks with Latent Variables. arXiv:1706.08495.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml. Accessed: 2022-03-30.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Networks. In *Advances in Neural Information Processing Systems*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2018. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*.

Joshi, S.; Koyejo, O.; Kim, B.; and Ghosh, J. 2018. xGEMs: Generating Examplars to Explain Black-Box Models. *arXiv preprint arXiv:1806.08867*.

Kingma, D. P.; and Welling, M. 2013. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*.

Krizhevsky, A. 2012. Learning Multiple Layers of Features from Tiny Images. *University of Toronto*.

Kulesza, A. 2012. Determinantal Point Processes for Machine Learning. *Foundations and Trends® in Machine Learning*, 5(2-3): 123–286.

Lacoste, A.; Rodríguez, P.; Branchaud-Charron, F.; Atighehchian, P.; Caccia, M.; H. Laradji, I.; Drouin, A.; Craddock, M.; Charlin, L.; and Vázquez, D. 2020. Synbols: Probing Learning Algorithms with Synthetic Datasets. In *Advances in Neural Information Processing Systems*.

LeCun, Y. 1998. The MNIST Database of Handwritten Digits. *http://yann.lecun.com/exdb/mnist/*.

Ley, D.; Bhatt, U.; and Weller, A. 2021. δ-CLUE: Diverse Sets of Explanations for Uncertainty Estimates. In *ICLR Workshop on Security and Safety in Machine Learning Systems*.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

MacKay, D. J. 1992. A Practical Bayesian Framework for Backpropagation Networks. *Neural computation*, 4(3): 448–472.

Mahajan, D.; Tan, C.; and Sharma, A. 2020. Preserving Causal Constraints in Counterfactual Explanations for Machine Learning Classifiers. In *NeurIPS Workshop on CausalML: Machine Learning and Causal Inference for Improved Decision Making*.

Mothilal, R. K.; Sharma, A.; and Tan, C. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*.

Pawelczyk, M.; Bielawski, S.; van den Heuvel, J.; Richter, T.; and Kasneci, G. 2021. CARLA: A Python Library to Benchmark Algorithmic Recourse and Counterfactual Explanation Algorithms. In *Advances in Neural Information Processing Systems (Benchmark & Data Set Track)*.

Pawelczyk, M.; Broelemann, K.; and Kasneci, G. 2020a. Learning Model-Agnostic Counterfactual Explanations for Tabular Data. In *Proceedings of The Web Conference 2020*, 3126–3132.

Pawelczyk, M.; Broelemann, K.; and Kasneci, G. 2020b. On Counterfactual Explanations under Predictive Multiplicity. In *Conference on Uncertainty in Artificial Intelligence*, 809–818. PMLR.

Plumb, G.; Terhorst, J.; Sankararaman, S.; and Talwalkar, A. 2020. Explaining Groups of Points in Low-Dimensional Representations. In *International Conference on Machine Learning*, 7762–7771. PMLR.

Poyiadzi, R.; Sokol, K.; Santos-Rodriguez, R.; De Bie, T.; and Flach, P. 2020. FACE: Feasible and Actionable Counterfactual Explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 344–350.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. 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*, 1135–1144. ACM.

Russell, C. 2019. Efficient Search for Diverse Coherent Explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 20–28.

Singla, S.; Pollack, B.; Chen, J.; and Batmanghelich, K. 2020. Explanation by Progressive Exaggeration. In *International Conference on Learning Representations*.

Spiegelhalter, D. 2017. Risk and Uncertainty Communication. *Annual Review of Statistics and Its Application*, 4: 31–60.

Tsirtsis, S.; De, A.; and Gomez-Rodriguez, M. 2021. Counterfactual Explanations in Sequential Decision Making Under Uncertainty. In *Advances in Neural Information Processing Systems*.

Van Looveren, A.; and Klaise, J. 2021. Interpretable Counterfactual Explanations Guided by Prototypes. In *Machine Learning and Knowledge Discovery in Databases. Research Track*, 650–665.

Yang, F.; Alva, S. S.; Chen, J.; and Hu, X. 2021. Model-Based Counterfactual Synthesizer for Interpretation. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

Zhao, S.; Song, J.; and Ermon, S. 2017. Learning Hierarchical Features from Deep Generative Models. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 4091–4099. PMLR.