δ-CLUE: DIVERSE SETS OF EXPLANATIONS FOR UNCERTAINTY ESTIMATES

Dan Ley
University of Cambridge
dwl36@cam.ac.uk

Umang Bhatt
University of Cambridge
usb20@cam.ac.uk

Adrian Weller
University of Cambridge
The Alan Turing Institute
aw665@cam.ac.uk

ABSTRACT

To interpret uncertainty estimates from differentiable probabilistic models, recent work has proposed generating Counterfactual Latent Uncertainty Explanations (CLUEs). However, for a single input, such approaches could output a variety of explanations due to the lack of constraints placed on the explanation. Here we augment the original CLUE approach, to provide what we call δ-CLUE. CLUE indicates *one* way to change an input, while remaining on the data manifold, such that the model becomes more confident about its prediction. We instead return a *set* of plausible CLUEs: multiple, diverse inputs that are within a δ ball of the original input in latent space, all yielding confident predictions.

1 INTRODUCTION

For models that provide uncertainty estimates alongside their predictions, explaining the source of this uncertainty reveals important information. Antorán et al. (2021) propose a method for finding an explanation of a model's predictive uncertainty of 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, while keeping it in distribution, such that the model becomes more certain in its prediction. Termed CLUE (Counterfactual Latent Uncertainty Explanation), this method is effective for generating plausible changes to an input that reduce uncertainty. These changes are distinct from adversarial examples, which instead find nearby points that change the label (Goodfellow et al., 2015). However, there are limitations to CLUE, including the lack of a framework to deal with a potential diverse set of plausible explanations (Russell, 2019), despite proposing methods to generate them.

CLUE introduces a latent variable DGM: $p_\theta(\mathbf{x}) = \int (p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$, with encoder $q_\phi(\mathbf{z}|\mathbf{x})$. The predictive mean of the DGM is $\mathbb{E}_{p_\theta(\mathbf{x}|\mathbf{z})}[\mathbf{x}] = \mu_\theta(\mathbf{x}|\mathbf{z})$ and of the encoder is $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\mathbf{z}] = \mu_\phi(\mathbf{z}|\mathbf{x})$ respectively. $\mathcal{H}$ refers to any differentiable uncertainty estimate of a prediction $\mathbf{y}$. CLUE minimises:

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

$$\text{to yield} \quad \mathbf{x}_{\text{CLUE}} = \mu_\theta\left(\mathbf{x}|\mathbf{z}_{\text{CLUE}}\right) \quad \text{where} \quad \mathbf{z}_{\text{CLUE}} = \arg\min_{\mathbf{z}} \mathcal{L}(\mathbf{z}). \tag{2}$$

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 $f(\mathbf{x})$ is the model's mapping from an input x to a label, thus encouraging similarity between uncertain points and CLUEs in both input and prediction space.

In this paper, we tackle the problem of finding multiple, diverse CLUEs. Providing practitioners with many explanations for why their input was uncertain can be helpful if, for instance, they are not

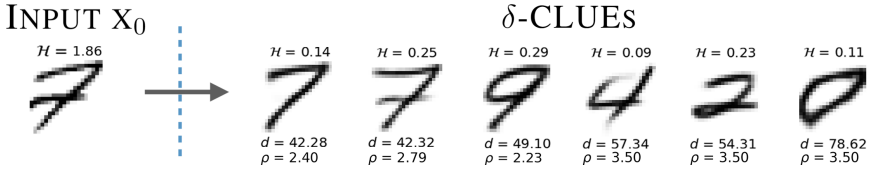

Figure 1: We produce a **diverse set** of candidate explanations that show how to reduce predictive uncertainty while still remaining close to x_0 in both input and latent space ($\mathcal{H}$ is uncertainty, d is input space distance, ρ is latent space distance). We see that the left image might easily be resolved into a confident 7 or 9.

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).

2 METHODOLOGY

We propose to modify the original method to generate 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 space representation of the uncertain input $\mathbf{x}_0$ being explained. We achieve multiplicity by initialising the search in different areas of latent space using varied initialisation methods $\mathcal{S}_i$. Experiments are performed on the MNIST dataset (LeCun, 1998), where finding diverse CLUEs amounts to maximising the number of class labels we converge to in the search. Figure 2 contrasts the original and proposed objectives.

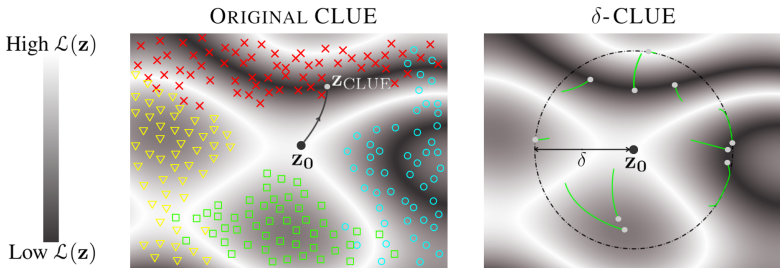

Figure 2: Conceptual colour map of objective function $\mathcal{L}(z)$ with $\mathbf{z}_0$ located in high cost region. Left: Gradient descent to region of low cost (original CLUE algorithm). Training points are shown in colour. Right: Gradient descent constrained to δ-ball at every step. Diverse starting points yield diverse local minima. White circles indicate CLUEs found.

In the original CLUE objective, the DGM and neural networks used are VAEs (Ivanov et al., 2018) and BNNs (Gal, 2016) respectively. The uncertainty of the BNN for a point 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 $d_x(\mathbf{x}, \mathbf{x}_0) = \|\mathbf{x} - \mathbf{x}_0\|_1$: see Appendix A for trade-offs. Figure 2 (left) shows a conceptual path taken by this optimisation. In our proposed δ-CLUE method, the loss function is the same as in Eq 1, with the additional δ requirement as:

$$\mathbf{x}_{\delta-\text{CLUE}} = \mu_\theta\left(\mathbf{x}|\mathbf{z}_{\delta-\text{CLUE}}\right) \text{ where } \mathbf{z}_{\delta-\text{CLUE}} = \underset{\mathbf{z}:\ \rho(\mathbf{z},\mathbf{z}_0)\leq\delta}{\arg\min}\ \mathcal{L}(\mathbf{z}) \text{ and } \mathbf{z_0} = \mu_\phi(\mathbf{z}|\mathbf{x_0}). \quad (3)$$

We choose $\rho(\mathbf{z}, \mathbf{z}_0) = \|\mathbf{z} - \mathbf{z}_0\|_2$ (the Euclidean ℓ_2 norm) in this paper, as shown in the 2D depiction in Figure 2. 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 $\lambda_x \approx 0.03$. The δ constraint can be applied either throughout each stage of the optimisation as in Projected Gradient Descent (Boyd et al., 2004) (Figure 2, right) or post optimisation (Appendix B). The optimal δ value(s) can be determined through experimentation (Figure 4), although Appendix B discusses other potential methods.

For each uncertain input x_0, we exploit the non-convexity of CLUE's objective to generate diverse δ-CLUEs by initialising gradient descents in different regions of latent space to converge to different local minima (Figure 2). We propose multiple initialisation schemes, $\mathcal{S}_i$; some may randomly initialise within the δ-ball, while others could use training data or class boundaries to determine starting points (shown in dark blue in Figure 3). We describe the δ-CLUE method in Algorithm 1.

3 EXPERIMENTS

We perform constrained optimisation during gradient descent (Figure 2, right). Appendix B provides justification for this decision. In our experiments, we search in the latent space of a VAE to generate δ-CLUEs for the 8 most uncertain digits in the MNIST test set, according to our trained BNN.

We trial this over **a)** a range of several δ values from 0.5 to 3.5, **b)** two latent space loss functions: **Uncertainty** $\mathcal{L}_\mathcal{H} = \mathcal{H}$ and **Distance** $\mathcal{L}_{\mathcal{H}+d} = \mathcal{H} + d$ and **c)** two initialisation schemes as depicted

in Figure 3. Initialisation scheme $\mathcal{S}_1$ picks a random direction at a uniform random radius within the delta ball, while the other scheme $\mathcal{S}_2$ is along paths determined by the nearest neighbours (**NN**) for each class in the training data. We label these experiment variants as: **Uncertainty Random**: [$\mathcal{L}_{\mathcal{H}}$, $\mathcal{S}_1$], **Uncertainty NN**: [$\mathcal{L}_{\mathcal{H}}$, $\mathcal{S}_2$], **Distance Random**: [$\mathcal{L}_{\mathcal{H}+d}$, $\mathcal{S}_1$] and **Distance NN**: [$\mathcal{L}_{\mathcal{H}+d}$, $\mathcal{S}_2$].

Algorithm 1 : δ-CLUE

Inputs: radius of search δ, number of explanations n, initialisation scheme $\mathcal{S}_i$, original datapoint $\mathbf{x}_0$, input space distance function d, latent space distance function ρ, BNN uncertainty estimator $\mathcal{H}$, DGM decoder $\mu_\theta(\cdot)$, DGM encoder $\mu_\phi(\cdot)$

Set δ-ball centre of $\mathbf{z}_0 = \mu_\phi(\mathbf{z}|\mathbf{x}_0)$;
for all explanations $i \leq n$ **do**
  Set initial value of $\mathbf{z}_i = \mathcal{S}(\mathbf{z}_0, \delta, i, n)$;
  while *loss $\mathcal{L}$ is not converged* **do**
    Decode: $\mathbf{x} = \mu_\theta(\mathbf{x}|\mathbf{z}_i)$;
    Use BNN to obtain $\mathcal{H}(\mathbf{y}|\mathbf{x})$;
    $\mathcal{L} = \mathcal{H}(\mathbf{y}|\mathbf{x}) + d(\mathbf{x}, \mathbf{x}_0)$;
    Update $\mathbf{z}_i$ with $\nabla_{\mathbf{z}}\mathcal{L}$;
    if $\rho(\mathbf{z}_i, \mathbf{z}_0) > \delta$ **then**
      Project $\mathbf{z}_i$ onto the surface of the δ-ball as $\mathbf{z}_i = \delta \times \frac{\mathbf{z}_i - \mathbf{z}_0}{\rho(\mathbf{z}_i, \mathbf{z}_0)}$;
    end if
  end while
  Decode explanation: $\mathbf{x}_{\delta-\text{CLUE}} = \mu_\theta(\mathbf{x}|\mathbf{z}_i)$;
  Accept if $\mathcal{H}(\mathbf{y}|\mathbf{x}_{\delta-\text{CLUE}}) < \mathcal{H}_{\text{threshold}}$;
end for

Outputs: A set of $m \leq n$ δ-CLUEs $\mathbf{x}_{\delta-\text{CLUE}}$

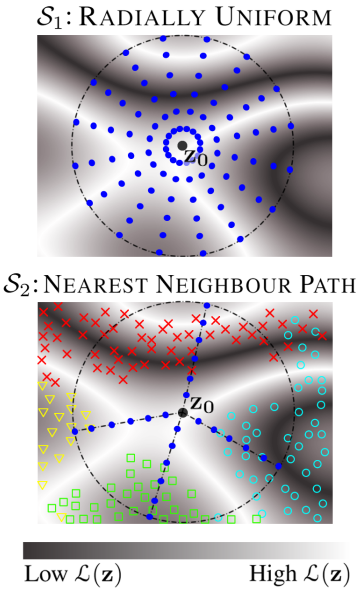

Figure 3: Two possible initialisation schemes $\mathcal{S}_i$ to yield diverse minima. One is random, the other deterministic. Details are provided in Appendix C.

In Figure 4, the $\mathcal{L}_{\mathcal{H}}$ experiments (blue and orange) demonstrate how the best CLUEs found improve as the δ ball expands, at the cost of increased distance from the original input. The $\mathcal{L}_{\mathcal{H}+d}$ experiments (green and red) suggest that the $\mathcal{L}_{\mathcal{H}+d}$ objective can vastly improve performance when it comes to distance (right), at the expense of higher (but acceptable) uncertainty.

Takeaway 1: as δ increases, using either loss $\mathcal{L}_{\mathcal{H}}$ or $\mathcal{L}_{\mathcal{H}+d}$, we reduce the uncertainty of our CLUEs at the expense of greater distance d. Loss $\mathcal{L}_{\mathcal{H}+d}$ experiences larger performance gains in the distance curves (green and red, Figure 4, right).

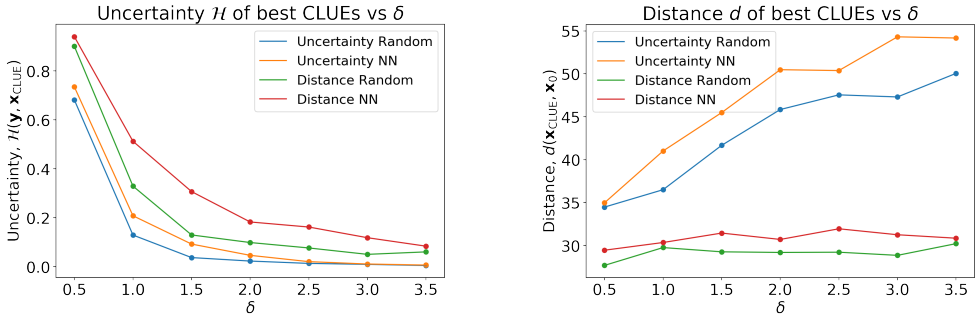

Figure 4: Left: Increasing the size of the δ ball yields lower uncertainty CLUEs. Right: The average distance of CLUEs from $\mathbf{x}_0$ increases with δ. Note that scheme $\mathcal{S}_1$ (blue and green) outperforms scheme $\mathcal{S}_2$ (orange and red) for this dataset.

We demonstrate that δ-CLUEs are successful in converging sufficiently to all local minima within the ball, given large enough n (Figure 5, left). Additionally, as the size of the δ ball increases, the random generation scheme $\mathcal{S}_1$ used in experiments **Uncertainty Random** and **Distance Random**

converge to the highest numbers of diverse CLUEs (Figure 5, right, blue and green). In both loss function landscapes ($\mathcal{L}_{\mathcal{H}}$ and $\mathcal{L}_{\mathcal{H}+d}$), we obtain similarly high levels of diversity as δ increases.

Takeaway 2: we can achieve a diverse plethora of high quality CLUEs when it comes to both class labels and modes of change within classes, permitting a full summary of uncertainty.

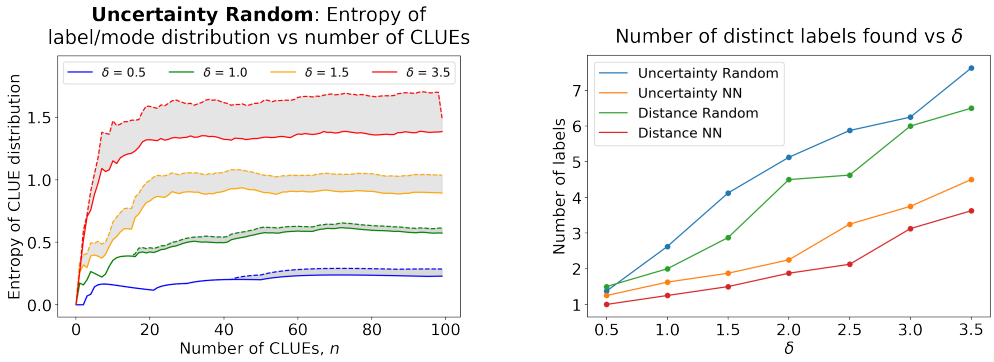

Figure 5: Left: Entropy of the distribution of class labels (solid) and different modes (dashed) found as number of CLUEs increases. Labels vary from 0 to 9 in MNIST whilst there exist multiple modes within each label. Observe the entropy saturating as we converge to all minima within the δ ball. Right: Average number of distinct labels found by sets of 100 CLUEs as δ increases. For small δ, typically only 1 class exists (low diversity). The random search $\mathcal{S}_1$ (blue and green) achieves the greatest diversity.

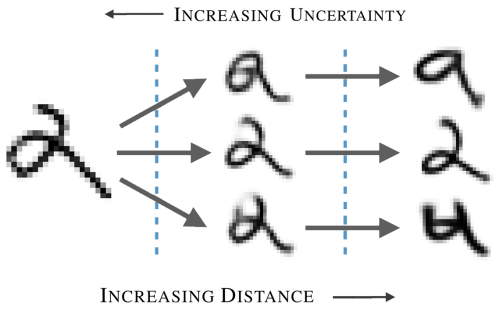

Figure 6: MNIST visualisation of the trade off between uncertainty $\mathcal{H}$ and distance d (example of 3 diverse labels discovered by δ-CLUE).

Given a diverse set of proposed δ-CLUEs (Figure 6), the performances of each class can be ranked by choosing an appropriate δ value and loss $\mathcal{L}$ for the mentioned trade offs (see Appendix E). Here, the digit 2 achieves lower uncertainty for a given distance, whilst the 9 and 4 require higher distances to achieve the same uncertainty. Without a δ constraint, we can move far from the original input and obtain a CLUE from any class that is certain to the BNN.

Takeaway 3: we can produce a **label distribution** over the δ-CLUEs to better summarise the diverse changes that could be made to reduce uncertainty.

4 Conclusion

We propose δ-CLUE, a method for suggesting multiple and diverse changes to an uncertain input that (i) are local to the input and (ii) reduce the uncertainty of the input with respect to the probabilistic model. We can effectively control the trade-off between uncertainty reduction and distance by a) constraining the search within a hypersphere of radius δ and/or b) introducing a distance penalty to the objective function $\mathcal{L}(\mathbf{z})$. We demonstrate diversity in the CLUEs found on MNIST. Diversity arises via convergence to multiple class labels and to different modes of changes within these labels. Practitioners can use δ-CLUE to understand the ambiguity of an input to a probabilistic model by suggesting a set of nearby points in the latent space of a DGM where the model is certain. For example, an uncertain 7 might be "close" to a certain 7 but also "close" to a certain 9, as seen in Figure 1. While we manually assess mode diversity, future work could deploy a clustering algorithm for automatic assessment of various modes (i.e., different forms of the digit 7). As recent work considered specifying the exact level of uncertainty desired in a sample (Booth et al., 2020) and has considered using DGMs to find counterfactual explanations though not for uncertainty (Joshi et al., 2018), we posit that leveraging DGMs to study the *diversity* of plausible explanations is a promising direction to pursue. δ-CLUE is just one step towards realising this goal.

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 under EPSRC grant EP/N510129/1 and TU/B/000074, and the Leverhulme Trust via CFI. The authors thank Javier Antorán for his helpful comments and pointers.

REFERENCES

Javier Antorán, 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.

Serena Booth, Yilun Zhou, Ankit Shah, and Julie Shah. Bayes-TrEx: Model transparency by example. *arXiv e-prints*, pp. arXiv–2002, 2020.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Yarin Gal. Uncertainty in deep learning, 2016.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Divas Grover and Behrad Toghi. MNIST dataset classification utilizing k-nn classifier with modified sliding-window metric. In *Science and Information Conference*, pp. 583–591. Springer, 2019.

Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary conditioning. *arXiv preprint arXiv:1806.02382*, 2018.

Shalmali Joshi, Oluwasanmi Koyejo, Been Kim, and Joydeep Ghosh. xGEMs: Generating examplars to explain black-box models. *arXiv preprint arXiv:1806.08867*, 2018.

Yann LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 344–350, 2020.

Chris Russell. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 20–28, 2019.

Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

A Distance Metrics

In this work, we take $d_x(\mathbf{x}, \mathbf{x}_0) = \|\mathbf{x} - \mathbf{x}_0\|_1$ to encourage sparse explanations. The original CLUE paper found that for regression, $d_y(f(\mathbf{x}), f(\mathbf{x}_0))$ is mean squared error, and for classification, cross-entropy is used, noting that the best choice for $d(\cdot, \cdot)$ will be task-specific.

In some applications, these simple metrics may be insufficient, and recent work by Zhang et al. (2018) alludes to the shortcomings of even more complex distance metrics such as PSNR and SSIM. For MNIST digits (28x28 pixels), *Mahanalobis distance* has been shown to be effective (Weinberger & Saul, 2009), as well as other methods that achieve translation invariance (Grover & Toghi, 2019).

For instance, the experiment in Figure 7 details how simple distance norms (either in input space and latent space) lack robustness to translations of even 5 pixels.

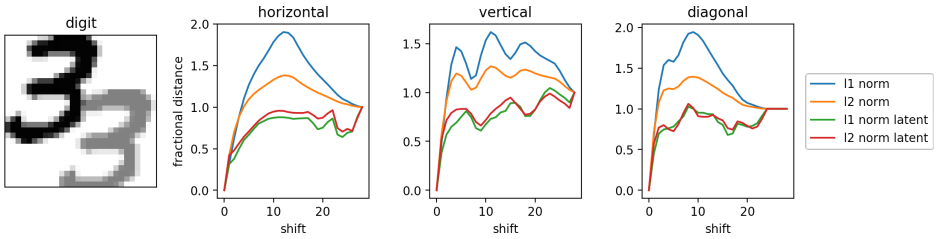

Figure 7: We apply horizontal, vertical and diagonal translations of an MNIST digit (in both input space and latent space for both ℓ_1 and ℓ_2 norms). As we increase the shift (in pixels), we compute the distance between the shifted and original digits, divided by the distance between an empty image and the original (to normalise over different metrics, resulting in convergence to 1.0). For reference, the shaded digit indicates the original digit shifted diagonally by 10 pixels.

B Constrained vs Unconstrained Search

Using the $\mathcal{L}_{\mathcal{H}} = \mathcal{H}$ loss function, finding minima within the δ ball is rare for small δ, and so it is necessary to use a constrained optimisation method in our experiments (Figure 8), to avoid all solutions lying outside of the ball and being rejected.

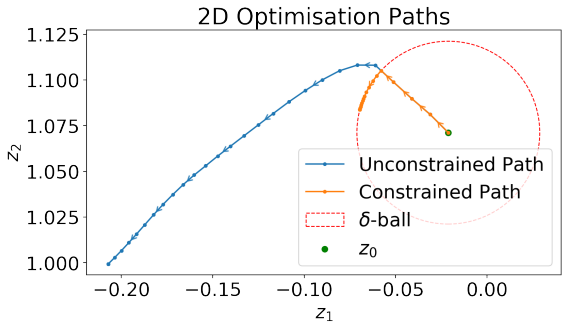

Figure 8: Constrained vs unconstrained gradient descents in a 2D VAE latent space $\mathcal{L}(\mathbf{z}) = \mathcal{H}$. We project values outside of the δ ball onto its surface at each step of the gradient descent.

Thus, we observe in Figure 9, right, that for small δ, virtually all δ-CLUEs lie on the surface of the ball. The left hand figure indicates that average latent space distances $\rho(\mathbf{z}_{\text{CLUE}}, \mathbf{z}_0)$ lie close to the line $\delta = \delta$ (purple, dashed), with the distance weighted loss $\mathcal{L}_{\mathcal{H}+d} = \mathcal{H} + d$ producing more nearby δ-CLUEs, as expected. In either case, the effect of the constraint weakens for larger δ, as more minima exist within the ball instead of on it. Depending on user preference, the optimal δ value represents the trade off between the loss of uncertainty and the distance from the original input.

As suggested in the main text, there may exist methods to determine δ pre-experimentation; the distribution of training data in the latent space of the DGM could potentially uncover relationships between uncertainty and distance, both for individual inputs and on average. For instance, we might

search in latent space for the distance to nearest neighbours within each class to determine δ. In many cases, it could be useful to provide a summary of counterfactuals at various distances and uncertainties, making a range of δ values more appropriate.

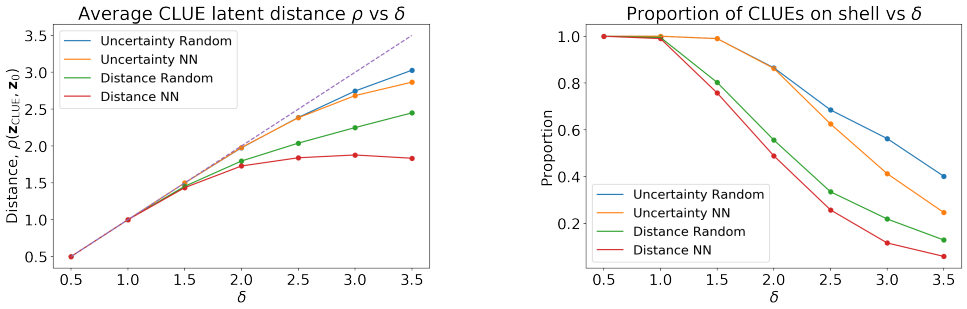

Figure 9: Justification for use of a constrained method. More solutions lie on the ball for a given δ, instead of within it. Left: How the average final distance in latent space varies with δ. Right: proportion of points that lie on the shell as δ increases. At small δ, almost all minima lie on the shell, whereas at larger δ more lie inside.

C Initialisation Schemes $\mathcal{S}_i$

This appendix details the initialisation schemes $\mathcal{S}_i$ that are used to generate start points for the algorithm. While some schemes may appear preferential in 2 dimensions, the manner at which these scale up to higher dimensions means that we could require an infeasible number of initialisations to cover the appropriate landscape, and so deterministic schemes such as a path towards nearest neighbours within each class ($\mathcal{S}_2$), or a gradient descent into predictions within each class ($\mathcal{S}_5$) might be desirable. The following mathematical analysis applies to an ℓ_2-norm $\rho(\mathbf{z}, \mathbf{z}_0) = \|\mathbf{z} - \mathbf{z}_0\|_2$:

$$\mathcal{S}_1 : \rho(\mathbf{z}, \mathbf{z}_0) \sim \mathcal{U}(0, \delta) \implies \mathbb{E}[\rho(\mathbf{z}, \mathbf{z_0})] = \frac{\delta}{2} \text{ (pick a random radial direction)}$$

$$\mathcal{S}_3 : \rho(\mathbf{z}, \mathbf{z}_0) \sim \mathcal{N}\left(0, \frac{\delta^2}{4}\right) \text{ s.t. } 0 \leq \rho(\mathbf{z}, \mathbf{z}_0) \leq \delta \text{ (pick a random radial direction)}$$

$$\mathcal{S}_4 : [\mathbf{z} - \mathbf{z}_0]_i \sim \mathcal{U}\left(-\delta, \delta\right) \text{ s.t. } \rho(\mathbf{z}, \mathbf{z}_0) \leq \delta$$

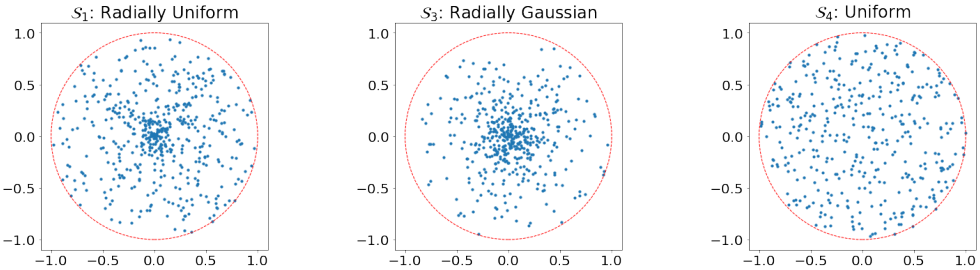

Figure 10: Random generation schemes $\mathcal{S}_1$, $\mathcal{S}_3$ and $\mathcal{S}_4$ depicted in 2D space. In Schemes $\mathcal{S}_3$ and $\mathcal{S}_4$ we reject samples outside of the δ ball (where $\rho(\mathbf{z}, \mathbf{z}_0) > \delta$). Future schemes may generate within a sub-ball that is smaller than the ball with which we constrain, though this may only be effective in specific latent landscapes.

We propose two potential deterministic schemes, that may outperform a random scheme when a) the latent dimension is large, b) δ becomes very large, c) we impose a larger distance weight in the objective function or d) we change datasets. Here $\mathbf{z}_i$ represents the starting point for explanation i, n is the total number of explanations (both used in Algorithm 1), Y represents the total number of class labels y, and $j \in \mathbb{Z}^+$. This produces a total of $Y \times j_{\max} = Y \times \left\lfloor \frac{n}{Y} \right\rfloor = n$ explanations if $Y | n$.

$$\mathcal{S}_2 : \mathbf{z}_i = \mathbf{z}_0 + \delta \times \frac{j}{m} \times \frac{\mathbf{z}_y - \mathbf{z}_0}{\rho(\mathbf{z}_y, \mathbf{z}_0)} \; \forall y$$

$$\mathcal{S}_5 : \mathbf{z}_i = \mathbf{z}_0 + \mathbf{s}_{yj} \; \forall y$$
$$\text{where } 1 \leq j \leq m \text{ and } m = \left\lfloor \frac{n}{Y} \right\rfloor$$

where, for the $\mathcal{S}_5$ scheme, $\mathbf{s}_{yj}$ is defined along a path from $\mathbf{z}_0$ to a radius δ, where at all points the direction of $\mathbf{s}$ is $\nabla_{\mathbf{z}} p(\text{class}(\mathbf{z}) = y)$, and $\frac{j}{m}$ is defined as the fraction travelled along that path.

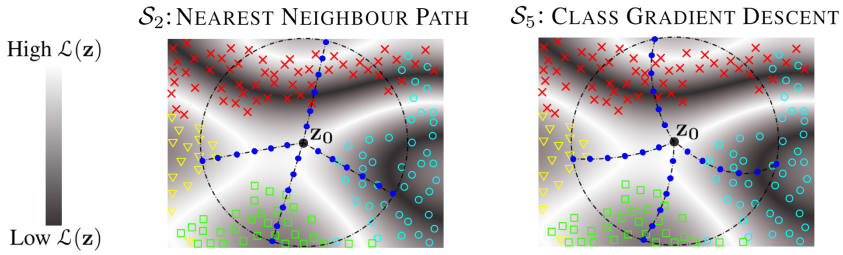

Figure 11: Left: Scheme $\mathcal{S}_2$, nearest neighbour path, searches for the nearest low uncertainty points in training data for each class, before initialising starting points fractionally on the path towards said neighbour. Right: Scheme $\mathcal{S}_5$ performs a gradient descent in the prediction space of the BNN, towards maximising the probability of each class. It too initialises starting points along said path.

A series of modifications to these schemes may improve their performance:

- Generating within small regions around each of the points along the path (in $\mathcal{S}_2$ and $\mathcal{S}_5$).
- Performing a series of further subsearches in latent space around each of the best δ-CLUEs under a particular scheme.
- Combining δ-CLUEs from multiple methods to achieve greater diversity.

D Further MNIST δ-CLUE Analysis

For an uncertain input $\mathbf{x}_0$, we generate 100 δ-CLUEs and compute the minimum, average and maximum uncertainties/distances from this set, before averaging this over 8 different uncertain inputs. Repeating this over several δ values produces Figures 12 through 14.

Special consideration should be taken in selecting the best method to assess a set of 100 δ-CLUEs: the minimum/average uncertainty/distance δ-CLUEs could be selected, or some form of submodular selection algorithm could be deployed on the set. Figure 13 shows the variance in performance of δ-CLUEs; the worst δ-CLUEs converge to high uncertainties and high distances that are too undesirable (the selection of δ-CLUEs is then a non-trivial problem to solve, and in our analysis we simply select the best cost δ-CLUE for each CLUE, where cost is a combination of uncertainty and distance).

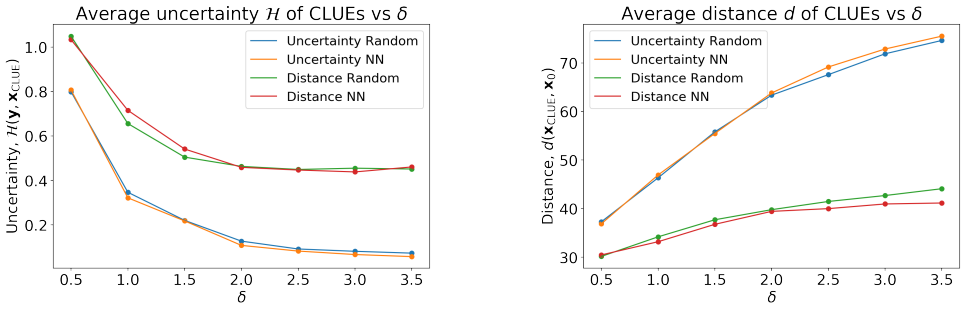

Figure 12: In Figure 4 of the main text, we plot the best (minimum) uncertainties/distances of the δ-CLUEs. Here, we reproduce the plot for average uncertainties/distances and observe that it follows similar trends, shifted vertically, with higher disparity between the $\mathcal{L}_{\mathcal{H}}$ and $\mathcal{L}_{\mathcal{H}+d}$ loss functions.

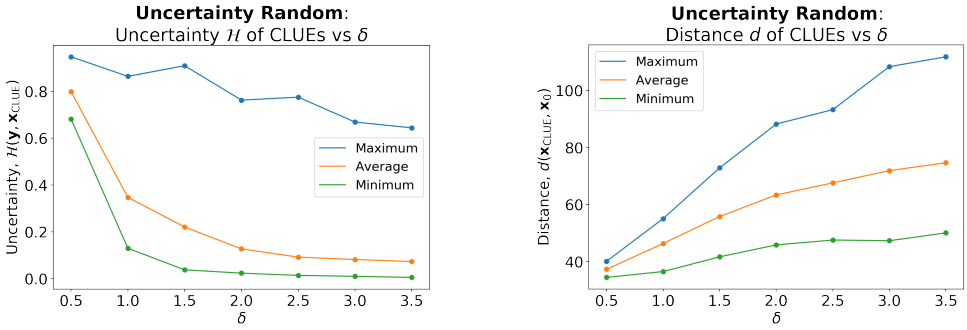

Figure 13: We reproduce Figure 4 for the **Uncertainty Random** experiment ($\mathcal{L}_{\mathcal{H}} = \mathcal{H}$ and $\mathcal{S}_1$), plotting the minimum, average and maximum values found in the set of 100 δ-CLUEs averaged over 8 uncertain inputs.

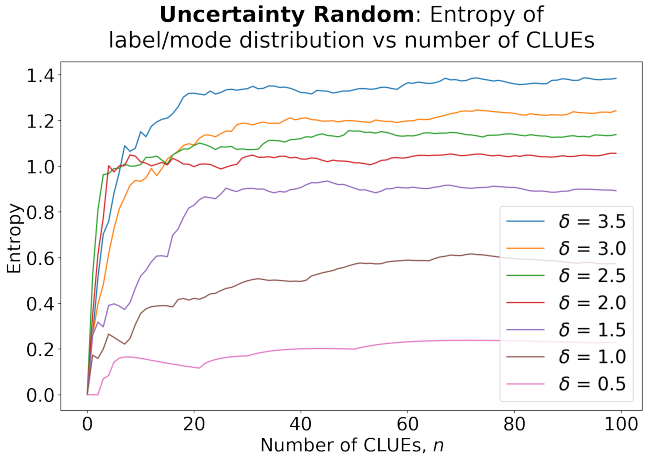

Figure 14: A more refined plot of Figure 5, left, to answer the question: "How many times must we run δ-CLUE in order to saturate the entropy of the label distribution of the δ-CLUEs found?".

In Figure 15, the late convergence of class 2 (green) and the lack of 1s, 3s and 6s suggests that $n > 100$ is required, although under computational constraints $n = 100$ yields good quality CLUEs for the prominent classes (7 and 9).

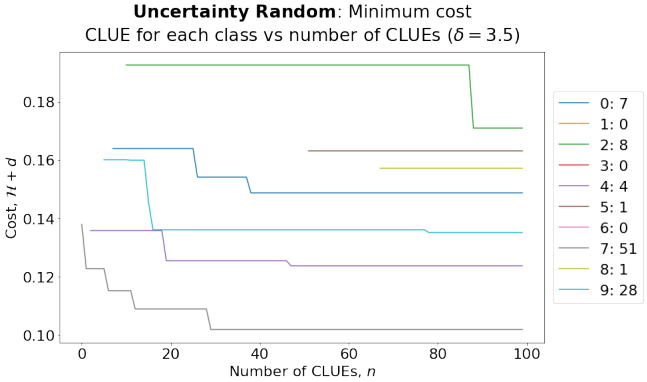

Figure 15: For a single uncertain input $\mathbf{x}_0$, we generate n δ-CLUEs and observe how the minimum cost (a combination of uncertainty and distance) of δ-CLUEs for each class converges. Legend shows class labels 0 to 9, and the final number of each discovered by δ-CLUE (summing to 100).

Figure 16 demonstrates how convergence of the δ-CLUE set is a function, not only of the class labels found, but also of the different mode changes that result within each class (alternative forms of each label). In the main text (Figure 5), we count manually the mode changes within each class; in future, clustering algorithms such as Gaussian Mixture Models could be deployed to automatically assess these. The concept of modes is important when a low number of classes exists, such as in binary classification tasks, where we may require multiple ways of answering the question: "what possible mode change could an end user make to modify their classification from a no to a yes?".

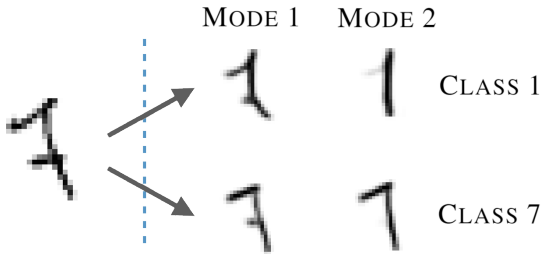

Figure 16: MNIST: 10 class labels exist (0 to 9), whereas an undefined number of modes within each class also exist. These modes are counted manually in this paper.

E COMPUTING A LABEL DISTRIBUTION FROM δ-CLUES

This final appendix addresses the task of computing a label distribution from a set of δ-CLUEs, as suggested by takeaway 3 of the main text. We use $\delta = 3.5$ and analyse one uncertain input $\mathbf{x}_0$ under the experiment **Distance Random** where $\mathcal{L}_{\mathcal{H}+d} = \mathcal{H} + d$ and $\mathcal{S}_1$ are used.

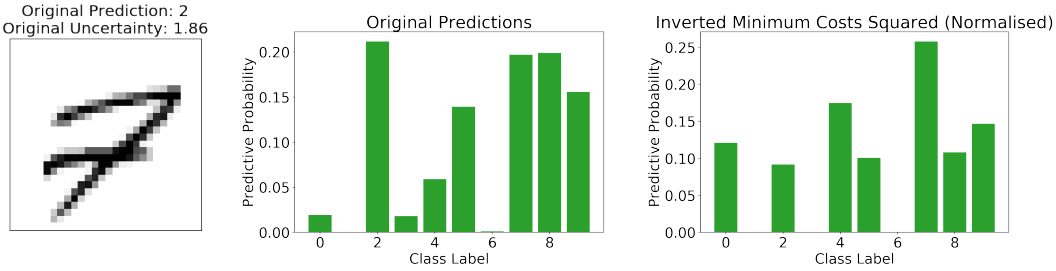

Figure 17: Left: An original uncertain input that is incorrectly classified. Centre: The original predictions from the BNN. Right: The new **label distribution** based off of the δ-CLUEs found.

For (Figure 17, right), we take the minimum costs from (Figure 18, right) and take the inverse square.

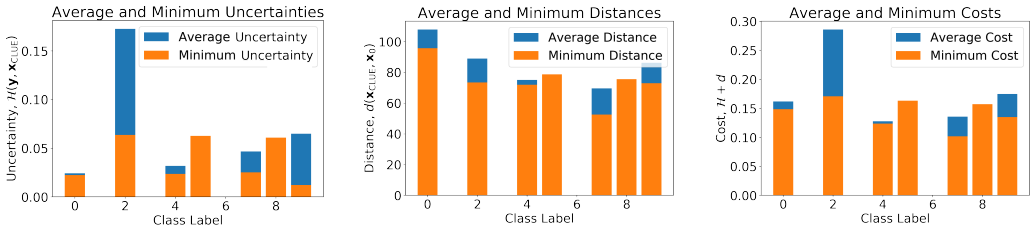

Figure 18: Left: Average and minimum uncertainties $\mathcal{H}$ for each class in the δ-CLUE set. Centre: Average and minimum distances d. Right: Average and minimum costs, where the weight λ_x is multiplied by the distance function and added to the uncertainty.

Figure 19: The 100 δ-CLUEs yielded in this experiment (**Distance Random** with $\delta = 3.5$). Above digits: Label prediction and uncertainty. Below: Distance from original in input space. Low uncertainty CLUEs may be found at the expense of a greater distance from the original input.