Diverse and Amortised Counterfactual Explanations for Uncertainty Estimates

Dan Ley [1] **Umang Bhatt** [1] **Adrian Weller** [1 2]

Abstract

To interpret uncertainty estimates from differentiable probabilistic models, Antorán et al. (2021) proposed generating a single Counterfactual Latent Uncertainty Explanation (CLUE) for a given data point where the model is uncertain. Ley et al. (2021) formulated δ-CLUE, the set of CLUEs within a δ ball of the original input in latent space – however, we find that many CLUEs generated by this method are very similar, hence redundant. Here we propose DIVerse CLUEs (∇-CLUEs), a set of CLUEs which each provide a distinct explanation as to how one can decrease the uncertainty associated with an input. We further introduce GLobal AMortised CLUEs (GLAM-CLUEs), which represent amortised mappings that apply to specific groups of uncertain inputs, taking them and efficiently transforming them in a single function call into inputs that a model will be certain about. Our experiments show that ∇-CLUEs and GLAM-CLUEs both address shortcomings of CLUE and provide beneficial explanations of uncertainty estimates to practitioners.

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 Explanations), this method is effective for generating plausible changes to an input that reduce uncertainty, distinct from adversarial examples, which instead find nearby points that change the label (Goodfellow et al., 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 $f(\mathbf{x})$ is the model's mapping from an input x to a label. CLUE minimises: $\mathcal{L}(\mathbf{z}) = \mathcal{H}\left(\mathbf{y}|\mu_\theta(\mathbf{x}|\mathbf{z})\right) + d\left(\mu_\theta(\mathbf{x}|\mathbf{z}), \mathbf{x}_0\right)$ to yield $\mathbf{x}_{\text{CLUE}} = \mu_\theta\left(\mathbf{x}|\mathbf{z}_{\text{CLUE}}\right)$ where $\mathbf{z}_{\text{CLUE}} = \arg\min_{\mathbf{z}} \mathcal{L}(\mathbf{z})$.

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 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). With δ-CLUE, Ley et al. (2021) introduce a method for generating a set of CLUEs. However, many CLUEs therein are redundant. We start by introducing metrics to measure the diversity in sets of CLUEs such that we can optimise directly for it: we term this ∇-CLUE. We then consider how to make computational improvements to CLUE, proposing GLAM-CLUE, GLobal AMortised Counterfactual Latent Uncertainty Explanations, which serves as a summary of CLUE for practitioners to audit their model's behavior on uncertain inputs. It does so by finding global translations between certain and uncertain groups in a computationally efficient manner.

2. Diversity Metrics for Counterfactuals

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.

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	$\frac{1}{d'} \sum_{i=1}^{d'} \left(\max_j (\mathbf{x}_j - \mathbf{x}_0)_i + \max_j (\mathbf{x}_0 - \mathbf{x}_j)_i \right)$
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. Where necessary, we define $D = 0$ for $k = 1$ and take d to be some arbitrary distance metric.

We consider similarity between counterfactuals in predic-

[1]University of Cambridge, UK [2]The Alan Turing Institute, UK. Correspondence to: Dan Ley <dwl36@cam.ac.uk>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

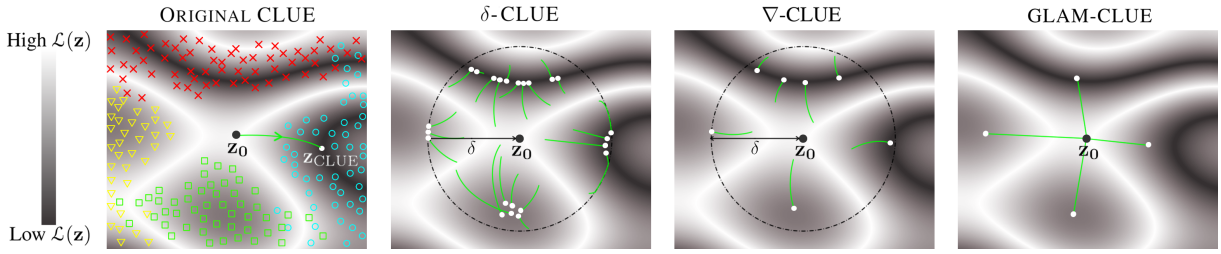

Figure 1. Conceptual colour map of objective function $\mathcal{L}(z)$ with $\mathbf{z}_0$ located in high cost region. White circles indicate explanations found. Left: Gradient descent to region of low cost (Antorán et al., 2021). Training points in colour. Left Centre: Gradient descent constrained to δ-ball. Diverse starting points yield diverse local minima, albeit with many redundant solutions (Ley et al., 2021). Right Centre: Direct optimisation for diversity (∇-CLUE). Right: Efficient mappings without gradient descent (GLAM-CLUE).

tion, input or latent space. Each diversity function D can be applied to a set of $k > 0$ counterfactuals appropriately 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 $y_i = \max_j(\mathbf{y}_i)_j$). We summarise these metrics in Table 1, and provide additional detail in Appendix B.

3. Optimizing for Diversity: ∇-CLUE

The diversity metrics defined in Section 2 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 "Div-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. By optimising simultaneously over k counterfactuals (Mothilal et al., 2020) 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 $\mathcal{L}(\mathbf{z}_i) = \mathcal{H}\left(\mathbf{y}|\mu_\theta(\mathbf{x}|\mathbf{z}_i)\right) + d\left(\mu_\theta(\mathbf{x}|\mathbf{z}_i), \mathbf{x}_0\right)$, to yield $X_{\text{CLUE}} = \mu_\theta\left(X|Z_{\text{CLUE}}\right)$ 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. This is described in Algorithm 1. Appendix C details a sequential procedure, analogous to a greedy optimisation.

We denote an initialisation scheme $\mathcal{S}$ of radius r to generate starting points for the gradient descent. The ∇-CLUE algorithm is equivalent to δ-CLUE from Ley et al. (2021) when $\lambda_D = 0$, which is itself equivalent to the original CLUE algorithm when $\delta = \infty$, $r = 0$ and $k = 1$.

4. GLAM-CLUE: Global Amortised CLUE

CLUE primarily focuses on local explanations of uncertainty estimates, as Antorán et al. (2021) propose a method for finding just a single change for a specified uncertain input. Such local explanations are 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.

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* $\mathcal{L}$ *is 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

We desire a computationally efficient method that only requires a finite portion of the dataset (or set of counterfactuals) from which global properties of uncertainty can be learnt, in the hope that we could apply these properties to unseen test data with a high degree of reliability. Therefore, we propose GLAM-CLUE (GLobal AMortised CLUE), which achieves such levels of reliability with considerable speedups. Summarising global properties of uncertainty can be important too in identifying areas in which the model does not perform as expected or the training data is sparse.

GLAM-CLUE takes groups of high and low certainty points and learns mappings of arbitrary complexity between them in either latent or input space. It can be seen as a global equivalent to CLUE. High certainty points are taken from the training data to learn such mappings, but we demonstrate 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: $\mathbf{z}_{\text{certain}} = G(\mathbf{z}_{\text{uncertain}})$.

Algorithm 2 GLAM-CLUE

Inputs: Inputs $X_{\text{uncertain}}$, X_{certain}, groups $Y_{\text{uncertain}}$, Y_{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 $X_{\text{uncertain}}$, $Y_{\text{uncertain}}$;
  Select X_j from X_{certain}, Y_{certain};
  Encode: $Z_i = \mu_\phi(Z|X_i)$;
  while *loss $\mathcal{L}$ is 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 simple latent space translation from an uncertain class i to a certain class j: $\mathbf{z}_j = G_{i\to j}(\mathbf{z}_i + \boldsymbol{\theta}_{i\to j})$. Mappers should reduce the uncertainty of points while keeping them close to the original when run on test data. To train the parameters of the translation $\boldsymbol{\theta}$, we use the loss function detailed in Equation 1. We learn separate mappers for each pair of groups defined by the practitioner; Algorithm 2 partitions these groups accordingly, and returns distinct parameters $\boldsymbol{\theta}_{i\to j}$ for each case. We posit that more complex models could improve the performance of the mappings at the risk of losing the global sense of an explanation.

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

There exist multiple baselines against which we can test performance. Firstly, we can perform Difference Between Means (DBM) of uncertain data to certain data in either input or latent space. This can be added to uncertain test data and reconstructed in the case of input space, or decoded in the case of latent space. Another baseline is the Nearest Neighbours (NN) in high certainty training data, in either input or latent space (these are visualised in Appendix D).

5. Experiments

We perform experiments to validate our methods on 2 datasets: UCI Credit classification (Dua and Graff, 2017) and MNIST image classification (LeCun, 1998). We train VAEs as our DGMs (Kingma and Welling, 2014) and BNNs for classification (MacKay, 1992). We demonstrate that our

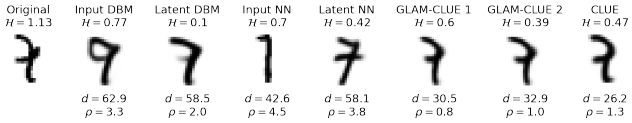

Figure 2. Comparison of explanations for an uncertain input (left) by the baselines, GLAM-CLUE, and CLUE. $\mathcal{H}$ is uncertainty, d is input space distance, ρ is latent space distance. Low uncertainties in baseline schemes have unrealistic distances from the original.

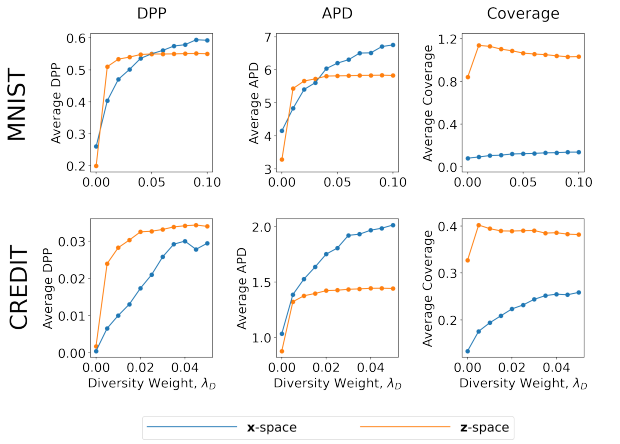

Figure 3. Effect of λ_D on diversity and performance. Row 1: MNIST. Row 2: UCI Credit. Columns 1 to 3: DPP, APD and Coverage diversity metrics applied to the set of $k = 10$ ∇-CLUEs.

constraints allow practitioners to better control the diversity of counterfactuals (∇-CLUE). We then show that we can efficiently generate explanations that apply globally to groups of inputs with our amortised scheme (GLAM-CLUE).

5.1. ∇-CLUE

We perform an ablative study on the effect of increasing the diversity weight λ_D on the CLUEs produced, optimising the DPP diversity metric in $\mathbf{z}$-space and measuring the effect that this has on each other metric, as well the effect on performance (Figure 3). 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. Note that $\lambda_D = 0$ is exactly equivalent to δ-CLUE, and hence it is seen that δ-CLUE yields slightly better sets of CLUEs that are much less diverse.

Takeaway: We observe that, when optimising for one diversity metric, increasing λ_D monotonically improves diversity by almost every other metric. Appendix C shows that this does however come at the expense of some performance.

5.2. GLAM-CLUE

Gradient descent at inference time is computationally 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 counterfactual to a local uncertain input. GLAM-CLUE computes counterfactuals for all uncertain test points in a single, amortised function call, permitting considerable speedups. We demonstrate that the performance of these counterfactual beats mean performance of all the baselines discussed in Section 4, achieving lower variance also.

We train 2 mappers: GLAM-CLUE 1 learns from all certain and uncertain 7s in the MNIST training data; GLAM-CLUE 2 learns from all uncertain 7s in the training data and their corresponding certain CLUEs. Figure 4 shows improvements when using the GLAM-CLUE 2 algorithm, demonstrating that CLUEs capture properties of uncertainty

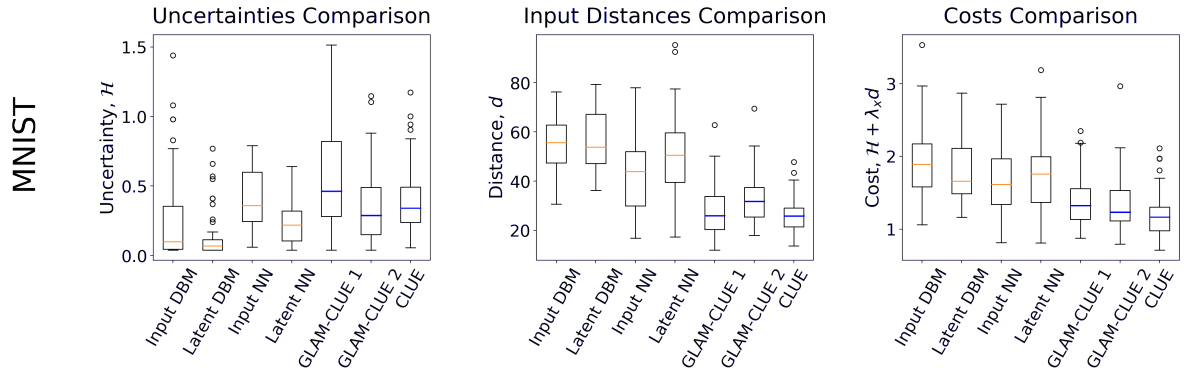

Figure 4. GLAM-CLUE vs baselines when mapping uncertain 7s to certain 7s in MNIST. Left: Distributions of uncertainties, $\mathcal{H}$ (original uncertainties exceed 1.5). Centre: Distributions of input distances, d. Right: Distributions of total costs, $\mathcal{H} + \lambda_x d$ with $\lambda_x = 0.03$.

more reliably than the training data, at the expense of extra computation time to generate the CLUEs used in learning.

An advantage of GLAM-CLUE is that the uncertainty-distance trade-off can be tuned with λ_θ in Equation 1: larger λ_θ restricts translations in latent space to be smaller, thus lowering distances in input space but raising uncertainties. For a given λ_x, GLAM-CLUE's efficiency allows for the optimal λ_θ to be determined quickly. While the baseline schemes achieve lower uncertainties, they do so at the expense of moving further from the original input (Figure 2), reducing the chance of yielding an actionable suggestion.

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 (Figure 4).

5.3. Computational Speedup

At inference time, GLAM-CLUE performs significantly faster than CLUE by **average CPU time** (Table 2). For uncertain 7s in MNIST, CLUE required 220 seconds to converge; GLAM-CLUE computes in around 1 second. The bottleneck in these processes is the uncertainty evaluation of the BNN, and as such these timings are not necessarily representative of all situations. A drawback to GLAM-CLUE is that the optimisation required on average 17.6 seconds to train. Should we use CLUEs during training (i.e. GLAM-CLUE 2), another 200+ seconds are required. Moving beyond basic mappers to more advanced models, we expect performance to improve at the cost of an increase in training time.

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. CPU time in **seconds** to compute 1 MNIST counterfactual.

Takeaway: At inference time, GLAM-CLUE can produce counterfactual explanations 200 times faster than CLUE.

6. Related Work

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 counterfactual explanations though not for uncertainty. Like CLUE and xGEMs, we use a DGM to find a set of viable CLUEs. Though not for uncertainty, Mothilal et al. (2020) and Russell (2019) find a diverse set of counterfactual explanations. Neither of the papers consider computational advances nor do they venture to consider global counterfactual explanations, as we do. Plumb et al. (2020) define a Global Counterfactual Explanation that uses a mapper to transform points from one low-dimensional group to another. In spirit of their work, 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.

7. 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). To interpret such estimates, (Antorán et al., 2021) propose generating a Counterfactual Latent Uncertainty Explanation (CLUE) for a given data point on which the model is uncertain; Ley et al. (2021) extend this work to generate a set of potential CLUEs. We study how to measure the diversity of these sets and find that many CLUEs are similar to each other. We propose 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. However, these methods prove to be 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 ∇-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.

References

J. Antorán, U. Bhatt, T. Adel, A. Weller, and J. M. Hernández-Lobato. Getting a CLUE: A method for explaining uncertainty estimates. In *International Conference on Learning Representations*, 2021.

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

S. Booth, Y. Zhou, A. Shah, and J. Shah. Bayes-TrEx: Model transparency by example. *arXiv e-prints*, pages arXiv–2002, 2020.

A. Dosovitskiy and J. Djolonga. You only train once: Loss-conditional training of deep networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HyxY6JHKwr.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL https://archive.ics.uci.edu/.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

S. Joshi, O. Koyejo, B. Kim, and J. Ghosh. xGEMs: Generating examplars to explain black-box models. *arXiv preprint arXiv:1806.08867*, 2018.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

A. Kulesza. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012. ISSN 1935-8245. doi: 10.1561/2200000044. URL http://dx.doi.org/10.1561/2200000044.

A. Lacoste, P. Rodríguez, F. Branchaud-Charron, P. Atighehchian, M. Caccia, I. H. Laradji, A. Drouin, M. Craddock, L. Charlin, and D. Vázquez. Synbols: Probing learning algorithms with synthetic datasets. *CoRR*, abs/2009.06415, 2020. URL https://arxiv.org/abs/2009.06415.

Y. LeCun. The MNIST database of handwritten digits. 1998. URL http://yann.lecun.com/exdb/mnist/.

D. Ley, U. Bhatt, and A. Weller. δ-CLUE: Diverse sets of explanations for uncertainty estimates. In *ICLR Workshop on Security and Safety in Machine Learning Systems*, 2021.

D. J. MacKay. A practical Bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, Jan 2020. doi: 10.1145/3351095.3372850. URL http://dx.doi.org/10.1145/3351095.3372850.

M. Pawelczyk, K. Broelemann, and G. Kasneci. On counterfactual explanations under predictive multiplicity. In *Conference on Uncertainty in Artificial Intelligence*, pages 809–818. PMLR, 2020.

G. Plumb, J. Terhorst, S. Sankararaman, and A. Talwalkar. Explaining groups of points in low-dimensional representations. In *International Conference on Machine Learning*, pages 7762–7771. PMLR, 2020.

R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, and P. Flach. FACE: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020.

M. T. Ribeiro, S. Singh, and C. 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*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

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

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

Appendix

This appendix is formatted as follows.

1. We discuss the **Datasets and Models** used in Appendix A.
2. We provide technicalities for **Diversity Metrics for Counterfactuals** in Appendix B.
3. We analyse possible adjustments to **∇-CLUE** in Appendix C.
4. We discuss the technicalities of **GLAM-CLUE** in Appendix D.

Where necessary, we provide discussion of potential limitations to our work and possible areas for future improvement.

A. Datasets and Models

One tabular dataset and two image datasets are employed in our experiments (all publicly available). Details are provided in Table 3.

The default of credit card clients dataset, which we refer to as "Credit" in this paper, can be obtained from `https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients/`. We augment input dimensions by performing a one-hot-encoding over necessary variables (i.e. gender, education). Note that this dataset is different from the also common German credit dataset.

The MNIST handwritten digit image dataset can be obtained from and is described in detail at `http://yann.lecun.com/exdb/mnist/`. For the aforementioned datasets, we thank Antorán et al. (2021) for making their private BNN and VAE models available for use in our work.

B. Diversity Metrics for Counterfactuals

B.1. Technicalities of Metrics

Leveraging Determinantal Point Processes: We build on Mothilal et al. (2020) to leverage determinantal point processes, referred to as DPPs (Kulesza, 2012), as $\det(\mathbf{K})$ in Table 1. DPPs implicitly normalise to $0 \leq D \leq 1$. However, matrix determinants are computationally expensive for large k. 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.

Diversity through Average Pairwise Distance: We can calculate diversity as the average distance between all distinct pairs of counterfactuals. While we can adjust for the number of pairs (thus, accomplishing invariance to the number of counterfactuals 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 a set of counterfactuals. Ribeiro et al. (2016) define coverage to be the sum of distinct features contained in a set, weighted by feature importance: this could be applied to counterfactual explanations to suggest a way of optimally choosing a limited subset from a full set of counterfactuals such that coverage is maximised. Plumb et al. (2020) introduce coverage as a measure of the quality of global counterfactual explanations. Herein, we interpret coverage as a measure of diversity, using it directly for optimisation and evaluation of counterfactuals. The metric, as given in Table 1, rewards changes in both positive and negative directions separately. For each feature, we find the 2 counterfactuals that produce the largest positive and largest negative change for that feature and sum their magnitudes. We repeat this over all features, summing the results.

D is bounded by the scale of the features, and we now determine this bound under the coverage metric. Take $S_+ = \sum_{i=1}^{d'} \max(x_i)$ and $S_- = \sum_{i=1}^{d'} \min(x_i)$ to represent the sum over all features of the maximum and minimum values each feature can take, and that $|\mathbf{x}_0| = \sum_{i=1}^{d'} (\mathbf{x}_0)_i$ (the sum over all features of the uncertain input $\mathbf{x}_0$), where d' is the dimensionality of the feature space. The minimum coverage of a counterfactual ($D = 0$) clearly occurs when the counterfactual is simply the original input. The maximum coverage can be calculated as:

$$D_{\text{max}} = \frac{1}{d'} \left((S_+ - |\mathbf{x}_0|) - (|\mathbf{x}_0| - S_-) \right) \tag{2}$$

$$= \frac{S_+ - S_-}{d'} \text{ (independent of } \mathbf{x}_0) \tag{3}$$

In the MNIST experiments performed, we have $d' = 28 \times 28 = 784$, with the maximum and minimum values of each pixel to be 1 and 0 respectively, thus giving $S_+ = 784$ and $S_- = 0$. This does indeed result in $D_{\text{max}} = 1$. If S_+ and S_- are known, we can guarantee this normalisation by dividing the coverage by D_{max}. In other applications, where features can scale infinitely, normalising D is not possible.

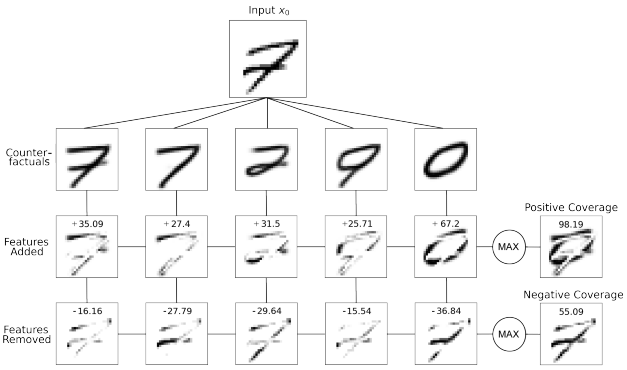

Figure 5. To compute positive and negative coverage, we take the positive and negative differences between counterfactuals and the original input, and further combine these by selecting the maximum change observed in a given feature (pixels in this case). We see that the 5 counterfactual explanations shown demonstrate changes that almost completely remove the original input, whilst adding features across a range of other areas. Total coverage is the sum of the positive and negative coverages.

In theory, 2 counterfactuals are sufficient to achieve the max-

Name	Targets	Input Type	Input Dimension	No. Train	No. Test
Credit	Binary	Continuous & Categorical	24	27000	3000
MNIST	Categorical	Image (Greyscale)	28×28	60000	10000
Synbols	Categorical	Image (RGB)	$3 \times 32 \times 32$	60000	20000

Table 3. Summary of the datasets used in our experiments.

imum coverage (one counterfactual with all features at their maximum values, and one with all features at their minimum values e.g. one fully black and one fully white image in MNIST). While coverage can never decrease as k increases, the exact nature of this relationship is dependent on the dataset and the counterfactual generation method. This is analytically indeterminate and thus cannot be regularised.

Prediction Coverage: Given that $\mathbf{y}_0$ as an estimate of the true label is inaccurate for uncertain predictions, and that 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 the set of counterfactuals, averaged over all predictions. This satisfies $\frac{1}{c'} \leq D \leq 1$, where we require at least $k = c'$ counterfactuals 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 et al., 2020), these metrics perform well in applications rich in class labels, and conversely are likely ineffective in binary 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.

B.2. Future Work

Future work might include a human subject experiment to determine the metric most aligned with human ideas of diversity; or better still, what each of the metrics represent themselves with regards to human intuition. The set of diversity metrics proposed in this paper are not exhaustive either, and further investigation of other metrics, perhaps with inspiration drawn from said human subject experiments, could provide meaningful insights.

C. ∇-CLUE

C.1. Performance Trade-off

While ∇-CLUE can successfully increase the diversity of counterfactuals through the λ_D hyperparameter, this comes with a trade-off between diversity and performance, **as stated in the main text**. This is a natural result of optimising for DPP diversity in $\mathbf{z}$-space; we achieve sufficient diversity in this space, though diversity in latent space implies some form of diversity in input space, resulting in higher distances from the original input.

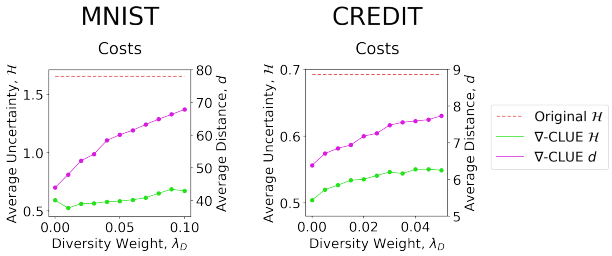

Figure 6. Performance degradation as λ_D increases. Average counterfactual uncertainty $\mathcal{H}$ (green) maintains an acceptably low value with respect to the original uncertainty (red). Distance metric d (purple) suffers to a greater extent.

C.2. Sequential Diversity Optimisation

In replacement of the simultaneous approach proposed in the paper, ∇-CLUE could be applied sequentially, where the approach is analogous to a greedy algorithm of the solution to 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}).

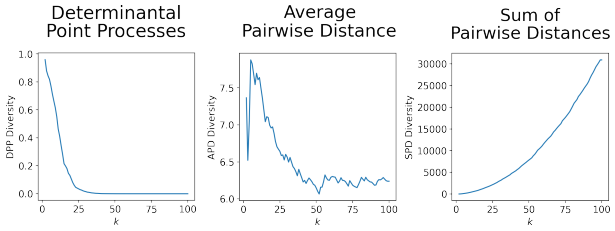

Figure 7. Effect of size k of a set of ∇-CLUEs on diversity under DPP, APD and SPD metrics. Under sequential ∇-CLUE, careful consideration should be taken when considering the effect of k on diversity, since tuning the λ_D hyperparameter after each iteration requires added complexity and is undesirable.

Given a set of counterfactuals Z_{CLUE} (initially the empty set $\varnothing$), we append 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. This is described in Algorithm 3. Possible pitfalls include the manner with which D scales with k (it is undesirable to have to re-tune the hyperparameter λ_D at

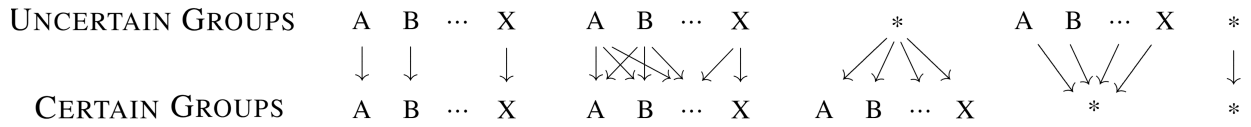

Figure 8. Possible direct mappings from uncertainty to certainty in groups A to X, without necessarily satisfying symmetry or transitivity. Asterisks represent uncertain/certain points that do not belong to a specific group. Correspondence in Appendix D.1.

each iteration). For instance, we might wish to remove the normalisation term $\binom{k}{2}$ of APD diversity when performing sequential ∇-CLUE, resulting in a new metric: the Sum of Pairwise Distances (SPD). Figure 7 details the effect of k under three metrics in input space. With a constant λ_D value, we see that the effect of the diversity term in ∇-CLUE will vary based on the diversity metric that we use.

Algorithm 3 ∇-CLUE (**sequential**)

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

Initialise $\varnothing$ of CLUEs: $X_{\text{CLUE}} = \{\}$;
Initialise $\varnothing$ of latent space CLUEs: $Z_{\text{CLUE}} = \{\}$;
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 $\mathbf{z}_i = \mathcal{S}(\mathbf{z}_0, r, i, k)$;
  while *loss* $\mathcal{L}$ *is not converged* **do**
    Decode: $\mathbf{x}_i = \mu_\theta(\mathbf{x}|\mathbf{z}_i)$;
    Use predictor to obtain $\mathcal{H}(\mathbf{y}|\mathbf{x}_i)$;
    $\mathcal{L}(\mathbf{z}_i) = \lambda_D D(Z_{\text{CLUE}} \cup \mathbf{z}_i) + \mathcal{H}(\mathbf{y}|\mathbf{x}_i) + d(\mathbf{x}_i, \mathbf{x}_0)$;
    Update $\mathbf{z}_i$ with $\nabla_{\mathbf{z}} \mathcal{L}(\mathbf{z}_i)$;
    Constrain $\mathbf{z}_i$ to δ ball using $\rho(\mathbf{z}_i, \mathbf{z}_0)$;
  end while
  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**
    $X_{\text{CLUE}} \leftarrow X_{\text{CLUE}} \cup \mathbf{x}_i$;
    $Z_{\text{CLUE}} \leftarrow Z_{\text{CLUE}} \cup \mathbf{z}_i$;
  end if
end for

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

C.3. Future Work

We devote this section to performing a full ablative analysis over all diversity metrics, since only DPP diversity in $\mathbf{z}$-space was trialled in the main paper. This includes a more thorough trial of sequential ∇-CLUE also. Some preliminary experiments have also shown that the strategy of finding diverse initialisations as opposed to optimising for diversity can also be effective; the latter method, used in this paper, has been shown to compromise the performance of the CLUEs found, and thus finding the best starting initialisations and performing δ-CLUE (where $\lambda_D = 0$) might yield equally diverse sets that perform better overall.

D. GLAM-CLUE

D.1. 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 not only partition the data into classes, but 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 detailed in Figure 8, and our experiments consider these groups to be class labels, testing against the far left scheme, considering mappings 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 local CLUE method is analogous to the far right scheme, which is agnostic to the particular classes it maps to and from. As a result of this, the original method also struggles with diverse mappings.

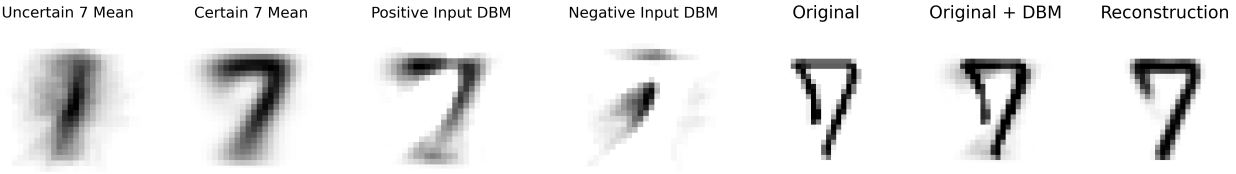

Figure 9. Visualisation of the input DBM baseline. The mean of all uncertain 7s in the MNIST training data is taken, followed by the mean of all certain 7s is shown in the 1st and 2nd plots. The 3rd and 4th plots show the positive and negative changes made when moving from uncertainty to certainty. The 5th to 7th plots illustrate how a final, certain counterfactual explanation is produced using this baseline (by adding the difference between means in input space and reconstructing the result).

D.2. Baseline Visualisation

Although drawing inspiration from Transitive Global Translations (TGTs), as proposed by Plumb et al. (2020), our method performs a different operation; instead of learning translations in input space that result in high quality mappings in a lower dimensional latent space, we find that results are best when learning translations in latent space, as described in the main text. This is seen also in the fact that the latent space DBM baseline outperforms input DBM; the difference between means translation is a special case of the GLAM-CLUE translation that we propose, and is the value we use as an initialisation during gradient descent. We also provide a visualisation for the input space DBM baseline in Figure 9. In the case of image data, the resulting image when DBM was added to the original input had to be clipped to match the scale of the data (in our case, between 0 and 1). Figure 10 displays the latent space equivalent (latent DBM) in a 2 dimensional latent space, as well as the Nearest Neighbour (NN) baseline.

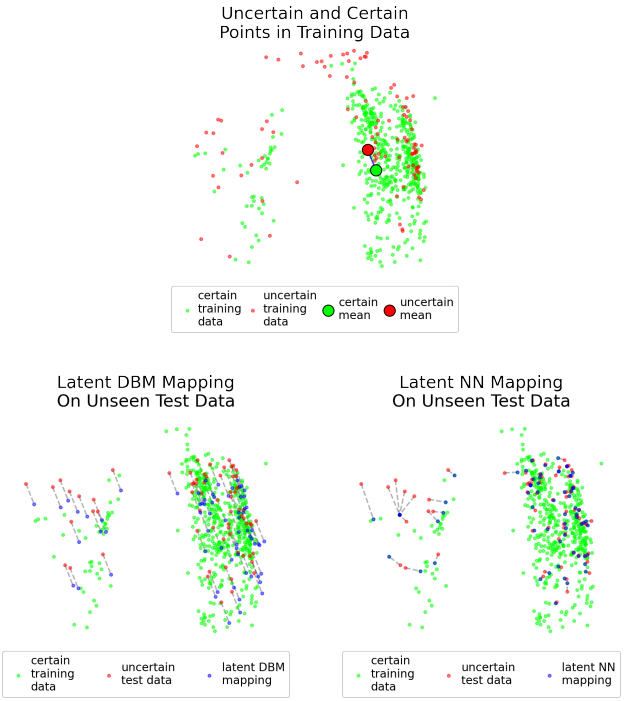

Figure 10. 2D latent space visualisation of DBM/NN baselines for MNIST digit 4. Top: Uncertain and certain points in the training data with their mean values. Left: Uncertain points in the test data with their latent DBM mappings. Right: Uncertain points in the test data with their NN mappings. High certainty training data shown in green throughout. Latent dimension is higher in practice.

D.3. Future Work

While GLAM-CLUE shows very good performance in the experiments demonstrated in the main text, it is not clear that performance would be maintained in all situations. Concerns include the performance on more complex datasets, or simply the performance in cases where a group of uncertain points are not easily separated from a group of certain points by a simple translation (as in Figure 11).

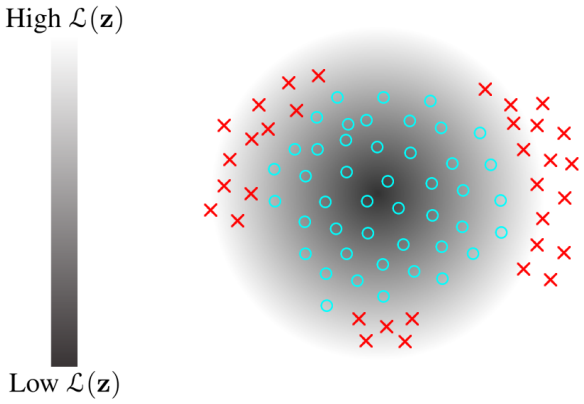

Figure 11. 2D visualisation of a shortcoming of GLAM-CLUE. The group of uncertain points (red) is not easily mapped onto the group of certain points (blue) by a single translation, unless further division of the uncertain group (into 3 clusters for instance) is performed, or a more complex mapper is learnt.

To this end, there are two further avenues to explore: the use of more complex mapping functions, or the potential to split the uncertain groups into groups that translations perform well on (in the Figure 11 example, this might entail clustering the uncertain points further into 3 groups). This latter approach would maintain GLAM-CLUE's utility in computational efficiency, as we demonstrate that learning simple translations is extremely fast, and a promising route for further research in this area.

We have the additional issue of selecting an appropriate λ_θ parameter in the algorithm to best tune the trade-off between uncertainty and distance. Dosovitskiy and Djolonga (2020) propose a method that replaces multiple models trained on one loss function each by a single model trained on a distribution of losses. A similar approach could be taken by using a distribution over individual terms of our objective and varying the hyperparameter weight at **inference time**. This could yield a powerful technique for minimising uncertainty and distance but allowing the trade-off between the two to be selected **post-training**.

As far as more complex datasets are concerned, preliminary trials on the black and white Synbols dataset (Lacoste et al., 2020) showed that the DBM baselines produced incoherent results. Our understanding is that, in input space, taking the mean of a particular class that contains an equal distribution of points with white backgrounds and points with black backgrounds will result in a cancellation between the two, such that the mean vector is close to zero. The same analogy in latent space might be that black points within a particular class may not be clustered in a similar region to those of white points for the same class. As such, further clustering, as alluded to above and in Figure 11 is probably necessary.