Learning Global Pairwise Interactions with Bayesian Neural Networks

Tianyu Cui, Pekka Marttinen, Samuel Kaski
Finnish Center for Artificial Intelligence
Department of Computer Science, Aalto University
firstname.lastname@aalto.fi

Abstract

Estimating global pairwise interaction effects with uncertainty properly quantified is centrally important in science discovery applications. We propose a non-parametric probabilistic method for detecting interaction effects of unknown form. First, the relationship between the features and the output is modelled using a Bayesian neural network, capable of representing complex interactions. Second, interaction effects and their uncertainties are estimated from the trained model. For the second step, we propose an intuitive global interaction measure: Bayesian Group Expected Hessian (GEH), which aggregates information of local interactions as captured by the Hessian. GEH provides a natural trade-off between type I and type II error and, moreover, comes with theoretical guarantees ensuring that the estimated interaction effects and their uncertainties can be improved by training a more accurate BNN. The method also empirically outperforms available non-probabilistic alternatives on simulated data.

1 Introduction

Estimating interactions between variables, and the uncertainty of the interactions, is a challenge common to many data science tasks. For example in $y = \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + e$, where β_{12} is the strength of the *interaction*. Here the shape of the interaction is known (multiplicative), but this is not true in general. Estimating the uncertainty is equally important, to assess the statistical significance of the detected interactions. Traditional methods include two general approaches: 1) conducting tests for each combination of features, such as ANOVA based methods [5, 17] and information gains [9, 18]. These method usually require a polynomial number of tests, and lack statistical power; 2) interactions are first learned by 'white-box' machine learning models, and then recover the interaction effects from the trained model. For example, Lasso based methods [2, 10, 11], and Gaussian processes [1]. But all possible interactions have to be pre-specified, which restricts the type of interactions that can be considered.

Here we extend the second approach, by using Bayesian neural network (BNN) to model interactions without any specified form directly from the data. Intuitively, we first train a BNN on the data, then find the encoded interactions by interpreting the trained model. However, the currently available algorithms that both are *interpretable* and aim to recover all kinds of interactions [7, 12, 14, 16] neglect uncertainty. In this work, we propose Bayesian Group Expected Hessian (GEH) to estimate global interactions by aggregating local interactions from a trained BNN. The posterior distribution of GEH represents the uncertainty of the interaction measure, and it can be seen as a non-parametric analogy to the posterior distribution of $|\beta_{12}|$ in previous linear model. Moreover it provides a natural trade-off between type I and type II error by tuning the number of groups.

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

2 Modeling Interactions and their Uncertainty

To learn interactions and their uncertainty, we first train a BNN, which both capable model all kinds of interactions, and captures uncertainty. In practice it is also beneficial to model the main effects separately from the interactions. We will use linear regression, $\mathbf{y}^m = \beta^T \mathbf{x}$, for the main effects, and a BNN, $\mathbf{y}^{in} = g^{\mathbf{W}}(\mathbf{x})$, to capture the interactions. A prediction from this *hybrid* model is the sum of the two components $\hat{\mathbf{y}} = \mathbf{y}^m + \mathbf{y}^{in}$, and given a dataset $\{\mathbf{X}, \mathbf{Y}\}$, training a BNN through variational inference is equivalent to optimize:

$$\mathcal{L}(\theta, \beta) = \int q_\theta(\mathbf{W}) \log p(\mathbf{Y}|\beta^T \mathbf{X} + g^{\mathbf{w}}(\mathbf{X}))\mathrm{d}\mathbf{W} + \mathrm{KL}(q_\theta(\mathbf{W})||p(\mathbf{W})). \tag{1}$$

We use concrete dropout [6] per node in this case, i.e. $q_{\mathbf{p},\mathbf{M}}(\mathbf{W}) = \prod_{l=1}^{L} \prod_{k=1}^{K_l} \mathbf{m}_{l,k}\text{Bernoulli}(1 - p_{l,k})$, where $p_{l,k}$ is the dropout probability for node k of layer l, and $\mathbf{m}_{l,k}$ is a vector of outgoing weights from node k in layer l.

3 Detecting Interactions

One way to measure interactions from a trained BNN $g^{\mathbf{W}}(\mathbf{x})$ is to use the *Hessian* of $g^{\mathbf{W}}(\mathbf{x})$ w.r.t. the input. For the multiplicative case in Section 1 this recovers the coefficient β_{12}. However, for other interactions, Hessian is not constant and represents interaction only at the point which Hessian was calculated, making it a *local interaction* measure. To estimate interaction *globally*, an intuitive way is to aggregate local effects into one global effect, and we propose below three ways of doing this.

Expected Absolute Hessian (EAH) and **Absolute Expected Hessian (AEH)** are the first two intuitive ways. EAH aggregates point-wise Hessian by calculating the expectation of its absolute value, while AEH is defined as the absolute value of expected Hessian, such as:

$$\text{EAH}_g^{i,j}(\mathbf{W}) = \mathbb{E}_{p(\mathbf{x})}\Big[\Big|\frac{\partial^2 g^{\mathbf{W}}(\mathbf{x})}{\partial x_i \partial x_j}\Big|\Big], \text{AEH}_g^{i,j}(\mathbf{W}) = \Big|\mathbb{E}_{p(\mathbf{x})}\Big[\frac{\partial^2 g^{\mathbf{W}}(\mathbf{x})}{\partial x_i \partial x_j}\Big]\Big|. \tag{2}$$

where $p(\mathbf{x})$ is the empirical distribution of $\mathbf{x}$. EAH has the **lowest FNR** (False Negative Rate), because if there is any region in $\text{dom}(\mathbf{x})$ where the Hessian is non-zero, $\text{EAH}_g^{i,j}$ will also be non-zero. However, EAH has the **highest FPR** (False Positive Rate), since even when x_i and x_j do not interact in the data generating process, the input Hessian between x_i and x_j will not be exactly 0 due to noise inherent in the data set. This noisy effect will be further aggregated into the global interaction effect and thus false interactions may be detected. In contrast with EAH, AEH has the **lowest FPR** but the **highest FNR**. If we assume that noise is distributed with zero mean independently of the location in $\text{dom}(\mathbf{x})$, the noise will cancel when we take the expectation over $\text{dom}(\mathbf{x})$. Consequently, FPR will be low. On the other hand, AEH may also cancel out some true interactions, if the signs of Hessian are different for different subregions of $\text{dom}(\mathbf{x})$, leading to a high FNR.

Group Expected Hessian (GEH) provides a natural trade-off between EAH and AEH, and thus between FPR and FNR, by clustering $\text{dom}(\mathbf{x})$ into *subregions*, calculating AEH for each subregion, and then computing their weighted average. For M groups, $\text{M-GEH}_g^{i,j}$ is

$$\text{M-GEH}_g^{i,j}(\mathbf{W}) = \sum_{m=1}^{M} \frac{|A_m|}{\sum_{k=1}^{M} |A_k|} \Big|\mathbb{E}_{p(\mathbf{x}|\mathbf{x}\in A_m)}\Big[\frac{\partial^2 g^{\mathbf{W}}(\mathbf{x})}{\partial x_i \partial x_j}\Big]\Big|. \tag{3}$$

where $|A_m|$ is the size of the mth subregion A_m, and $\bigcup_{m=1}^{M} A_m = \text{dom}(\mathbf{x})$. By choosing the subregions A_m properly, GEH has the potential to aggregate only interactions while canceling out the noise. To see this, assume that the noise is independent in $\text{dom}(\mathbf{x})$, but interaction effects are similar for close-by points in $\text{dom}(\mathbf{x})$. Then, a partition can be estimated by a clustering algorithm, such as k-means, with M clusters. Consequently, datapoints within A_m are close to each other (with almost same signs), which GEH will aggregate similarly to EAH. On the other hand, GEH will act like AEH canceling out the noise when integrating over the subregion. When $M = 1$, 1-GEH reduces to AEH, and when $M = N$, N-GEH becomes EAH, where N is the number of data. In Eq.3, $\mathbf{W}$ is the weight in a BNN, i.e. a random variable. Therefore, $\text{M-GEH}_g^{i,j}(\mathbf{W})$ is also a random variable whose distribution follows from the posterior distribution $q_\theta(\mathbf{W})$ of $\mathbf{W}$, thus we call it *Bayesian Group Expected Hessian*. Unbiased estimators for the mean and variance of the Bayesian GEH can

be obtained through Monte Carlo integration. If we further assume that $f(\cdot)$ and $g(\cdot)$ are L-Lipschitz functions, we can prove M-GEH satisfies following property (details in Appendix 4):

Accuracy Improvement Property: *We can reduce the estimation error of GEH by reducing the prediction error of $g^{\mathbf{W}}(\cdot)$, and make the uncertainty of the Bayesian GEH arbitrarily well-calibrated by improving the calibration of the distribution of predictions from $g^{\mathbf{W}}(\cdot)$.*

Determining the Number of Clusters: By increasing M, GEH can detect more complex interactions, but also lead to a higher FPR. An ideal $M^\star$ is the smallest number that can capture rich enough interactions for a specific problem, which means the detected interactions should not change significantly by further increasing M. Based on [8], we propose rank weighted distance to compare two interaction effect vectors corresponding to consecutive numbers of clusters: $\Delta_M^2 = \sum_i (w_M(i) - w_{M-1}(i))^2 (\pi_M(i) - \pi_{M-1}(i))^2$. Here, $w_M(i)$ is the ith interaction effect with M ($M \geq 2$) clusters, $\pi_M(i)$ is the rank of $w_M(i)$. The contribution to Δ_M^2 of those interactions whose relative rank does not change is equal to 0. Otherwise it will be proportional to the squared Euclidean distance of the effect sizes. One way to determine the number of clusters is to inspect values of Δ_M^2, plotted as a function of M, and choose M when Δ_M^2 approximately converges to 0.

4 Experiments

We apply our approach to simulated toy data sets with 8 features with 7 interaction pairs, using model: $y_i = \sum_{j=1}^{8} \beta_j^m x_j + \sum_{k=1}^{7} \beta_k^i h_k(x_k, x_{k+1}) + \epsilon$, where $h_k(\cdot)$ is the functional form of the kth interaction (specified in Appendix) with weight β_k^i. We compare the performance of interaction detection of our approach (Bayesian M-GEH with $M = 11$ determined by Δ_M^2) with other non-probabilistic alternatives (NID [16], SHAP [12] and Lasso [11]) for different signal to noise ratios.

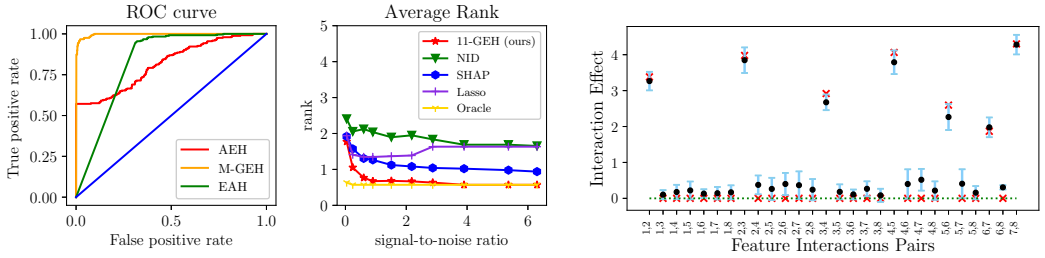

Figure 1: Experiment results on simulation data

Figure 1 (a) shows the ROC curves for the three global interaction methods introduced above, and M-GEH has the highest AUC. Specifically, AEH failed to detect two symmetric interactions and thus a high FNR. EAH failed to reject any false interactions on the basis of its 95% CI, so a high FPR . In (b) we increase the signal-to-noise ratio (S/N) gradually from 0 to see which method can recover the ground-truth interactions with the smallest S/N. It shows that 11-GEH identifies on average the true interactions with a smaller S/N ratio than the other methods. Figure 1 (c) shows the estimated uncertainties of interaction effects according to Bayesian 11-GEH. We observe that almost all true interaction effects (red crosses) are covered by the corresponding 95% credible intervals (blue bars), centered on the point estimates (black dots). See details and more experiments in Appendix 1-3.

5 Conclusion

We presented a novel method to learn global pairwise interactions with uncertainty using Bayesian neural networks. We proposed a flexible global interaction measure, Bayesian Group Expected Hessian, to detect interactions with uncertainty from a trained BNN. The method comes with appealing theoretical properties, which ensure that by improving the underlying BNN, interaction detection can be improved, and it provides a natural trade-off between FPRs and FNRs by tuning the number of groups, which is important in critical fields. Our results provide meaningful uncertainty estimations, and also empirically outperformed several non-probabilistic state-of-the-art baselines. We also demonstrate its ability to detect interactions between higher-level features in Appendix 3.

References

[1] R. Agrawal, B. Trippe, J. Huggins, and T. Broderick. The kernel interaction trick: Fast bayesian discovery of pairwise interactions in high dimensions. In *International Conference on Machine Learning*, pages 141–150, 2019.

[2] J. Bien, J. Taylor, and R. Tibshirani. A lasso for hierarchical interactions. *Annals of Statistics*, 41(3):1111, 2013.

[3] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[4] H. Fanaee-T and J. Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2(2-3):113–127, 2014.

[5] R. Fisher. *Statistical Methods For Research Workers*. Oliver And Boyd: Edinburgh, 1936.

[6] Y. Gal, J. Hron, and A. Kendall. Concrete dropout. In *Advances in Neural Information Processing Systems*, pages 3581–3590, 2017.

[7] P. Greenside, T. Shimko, P. Fordyce, and A. Kundaje. Discovering epistatic feature interactions from neural network models of regulatory DNA sequences. *Bioinformatics*, 34(17):i629–i637, 2018.

[8] M. Ibrahim, M. Louie, C. Modarres, and J. Paisley. Global explanations of neural networks: Mapping the landscape of predictions. *arXiv preprint arXiv:1902.02384*, 2019.

[9] X. Jiang, J. Jao, and R. Neapolitan. Learning predictive interactions using information gain and bayesian network scoring. *PloS one*, 10(12):e0143247, 2015.

[10] Y. Kong, D. Li, Y. Fan, J. Lv, et al. Interaction pursuit in high-dimensional multi-response regression via distance correlation. *The Annals of Statistics*, 45(2):897–922, 2017.

[11] M. Lim and T. Hastie. Learning interactions via hierarchical group-lasso regularization. *Journal of Computational and Graphical Statistics*, 24(3):627–654, 2015.

[12] S. M. Lundberg, G. G. Erion, and S.-I. Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[13] R. K. Pace and R. Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

[14] D. Sorokina, R. Caruana, M. Riedewald, and D. Fink. Detecting statistical interactions with additive groves of trees. In *Proceedings of the 25th International Conference on Machine learning*, pages 1000–1007. ACM, 2008.

[15] A. Tsanas and A. Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49:560–567, 2012.

[16] M. Tsang, D. Cheng, and Y. Liu. Detecting statistical interactions from neural network weights. *arXiv preprint arXiv:1705.04977*, 2017.

[17] T. H. Wonnacott and R. J. Wonnacott. *Introductory statistics*, volume 5. Wiley New York, 1990.

[18] Z. Zeng, X. Jiang, and R. Neapolitan. Discovering causal interactions using bayesian network scoring and information gain. *BMC Bioinformatics*, 17(1):221, 2016.

Appendix 1: Detail of experiment on simulation datasets

Data generating: x_j is drawn uniformly from $(0.5, 1.5)$ when $j = 1, 3, 5, 6, 8$, and from $(-0.5, 0.5)$ when $j = 2, 4, 7$. The functional form of the kth interaction is specified above the panels in Figure 2. Noise ϵ is Gaussian with zero mean and variance adjusted to a specified signal-to-noise ratio. Each simulated dataset includes 20000 samples for training, 5000 for validation, and 5000 for testing.

Bayesian NN setting: To model interactions, we use a concrete dropout Bayesian neural network with 3 hidden layers of sizes 100, 100, and 100 nodes for $g^{\mathbf{w}}(\mathbf{x})$. During training, we set the length-scale of the prior distribution l to 10^{-4}, temperature of the Sigmoid function in the concrete distribution to 0.1, and the learning rate of Adam to 10^{-3}.

Comparison methods: We apply Bayesian M-GEH and NID on the same trained neural networks, using $M = 11$ which is determined by Δ_M^2. We implement SHAP interaction score with learning rate equal to 0.01, and a Lasso regression containing all pairwise multiplicative interactions with regularization set to 5×10^{-4}. We include a linear regression model with the correct functional forms for the true interactions and the multiplicative form for other interactions as the 'Oracle'. We rank feature pairs according to the *absolute values* of interaction scores from each method from low to high. A good interaction measure should assign the true interactions as small a rank as possible.

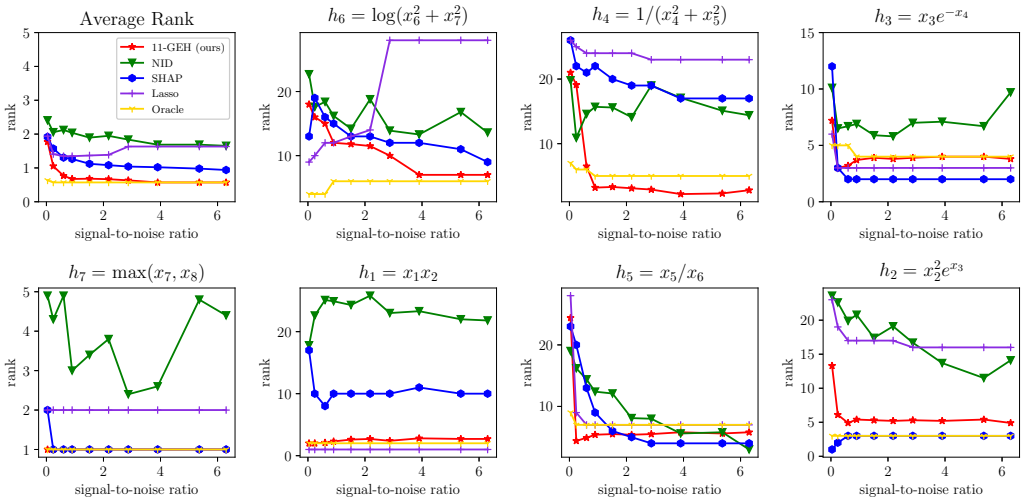

Figure 2: Comparison of detected interactions as a function of S/N. The 11-GEH method detects the ground-truth interactions clearly better than the others (top left panel), with performance very close to the Oracle. The rest of the panels show the ranks assigned to each of the seven true interactions.

Appendix 2: Experiment on public datasets

Datasets: We analyze 3 publicly available regression datasets: California housing prices, Bike sharing, and Energy efficiency datasets. California housing prices dataset [13] aims to predict housing prices using 9 features, such as location, number of rooms, number of people, etc. Bike sharing dataset [4] predicts the hourly bike rental count from environmental and seasonal information. Energy efficiency dataset [15] aims to predict the load of heating and cooling from the shape of a building. We use 70% of data for training, 20% for validation, and 10% for testing.

Experimental setup: We 1) analyse the original datasets and report and interpret the results, and 2) construct null hypothesis by permuting the target variable in each dataset, which allows us to estimate FPRs for the different methods. For each dataset, we construct 500 permutation datasets. Since all the interactions in permutation datasets are false, if any 95% CI excludes 0, it will be considered as a false positive, thus we can calculate the false positive rate for AEH, M-GEH, and EAH on each dataset. The same settings are used as in previous section, except that only one hidden layer with 30 units is used for the third dataset, which has only around 600 data points.

Table 1: Top 3 interactions for real-world datasets without injected interactions

Datasets	Interacting Features	M-GEH	95% CI	P_{Bayes}	$P_{Permute}$
California Housing	total room, population	1.532	$(0.030, 3.034)$	0.026	0.000
	longitude, latitude	0.901	$(0.241, 1.561)$	0.003	0.000
	total room, income	0.531	$(0.065, 0.997)$	0.011	0.000
Bike Sharing	workingday, hour	0.337	$(0.253, 0.421)$	0.001	0.000
	temperature, humidity	0.183	$(0.141, 0.225)$	0.001	0.000
	hour, temperature	0.180	$(0.122, 0.238)$	0.002	0.000
Energy Efficiency	roof area, wall area	1.223	$(0.689, 1.757)$	0.000	0.000
	roof area, height	0.938	$(0.539, 1.336)$	0.001	0.000
	compactness, roof area	0.699	$(0.384, 1.013)$	0.000	0.000

Table 2: Average FPRs for different M on permuted datasets

	AEH	M-GEH	EAH
California Housing	0.015	0.097	1.000
Bike Sharing	0.017	0.042	0.210
Energy Efficiency	0.013	0.051	0.089

Results: The optimal M for original datasets are 10, 9, and 15 for California housing, bike sharing, and energy efficiency respectively. Table 1 shows results for top 3 interactions in the datasets. M-GEH, CI, and P_{Bayes} are the estimated means, credible intervals, and P values from our method. We also show a P value obtained by permuting the target multiple times ($P_{Permute}$), to create an empirical null distribution of the maximum interaction score. **All top interactions are meaningful and statistically significant based on both our CIs and permutation**. Examples are shown in Figure 3. One strong interaction in the California housing data set is between longitude and latitude, which together specify the location that obviously affects the price. As another example, whether the day is a working day or not will affect the peak hours of bike renting.

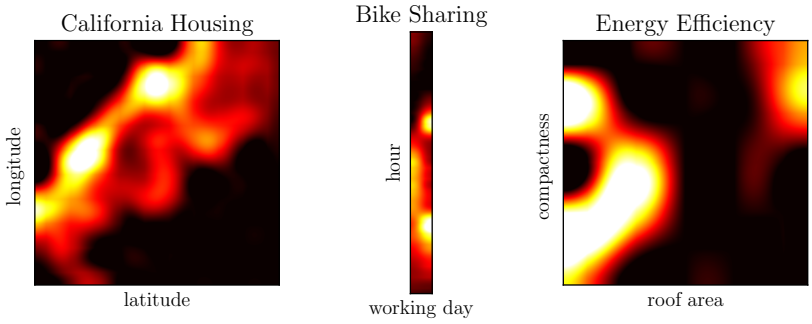

Figure 3: Visualization of one detected interaction for each dataset.

Table 2 shows the average FPRs of different global interaction measure on 3 datasets. As expected, AEH has the lowest FPRs, but it is unable to detect complex interactions such as the one between *working day* and *hour* in the bike sharing dataset. EAH has the highest FPR, and in the California Housing dataset in particular it considers all false interactions significant. **The FPRs for M-GEH are approximately correct (close to 5% when using 95% CIs)**.

Appendix 3: Detect higher-level feature interactions in MNIST

Motivation: We aim to demonstrate the ability of our method to detect interpretable interactions between higher-level features. For this, we design a classification task where the positive label represents a combination of interpretable characteristics of the input. The classification task here is to identify a given combination of two digits, e.g. (5,3), and the inputs are obtained by concatenating

randomly chosen MNIST digits. Our expectation is that nodes in upper layers represent interpretable properties of the inputs (e.g. "5 on the left"), such that an interaction between two such nodes corresponds to the positive label (e.g. "5 on the left" and "3 on the right").

Datasets: We repeat the experiment twice: the first dataset consists of pairs (7,4), (4,7), (0,4), (4,0), (7,0), (0,7), and the positive label is (7,0); the second dataset consists of pairs (5,4), (4,5), (3,4), (4,3), (5,3), (3,5), and the positive label is (5,3).

Experimental setup: We train a LeNet (2 convolutional layers, and 3 fully connected layers) with concrete dropout, and use M-GEH to detect interactions between nodes in the second top fully connected layer, where nodes can be regarded as some high-level features learned by previous layers. Clustering is also implemented on the same layer, and the optimal M for each task is 4 and 2, respectively. We provide interpretations for these high-level features by finding one-digit image inputs with white on the other side, e.g. (1,-) or (-,6) that, from all possible one-digit images in the MNIST data, maximize the activation of the node. This is the activation maximization with experts technique for interpreting nodes in intermediate NN layers [3], with empirical distribution of digits in MNIST as the expert.

Results: Figure 4 shows the top two interactions in the second-highest layer, and presents inter-

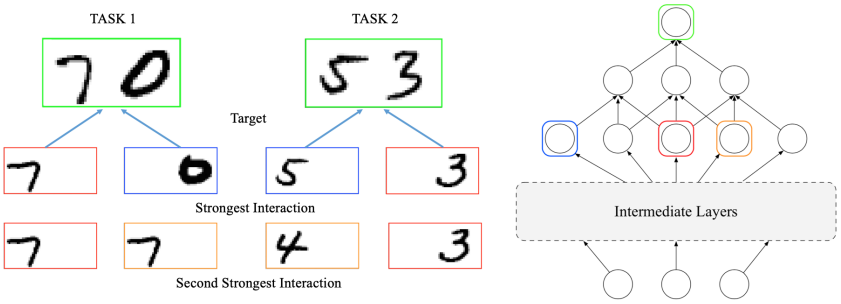

Figure 4: Higher-level feature interactions on MNIST dataset. The structure of the NN is simplified for illustrative purposes.

pretations for the interacting nodes. We see that all the interacting nodes represent digits related to the prediction tasks instead of unrelated digits, such as 3 in the first task or 7 in the second task. Interestingly, **in both tasks the strongest interaction is between nodes whose interpretation matches the human intuition exactly**. In the first task, (7,0) is obtained as an interaction between "7 on the left" and "0 on the right", and similarly for the task of classifying (5,3). The second strongest interaction in the (5,3) classification is between nodes with interpretations (4,-) and (-,3), which may be interpreted as excluding digit 4 on the left, when there is 3 on the right. The interaction between nodes which both have interpretation (7,-) may be related to learning different parts of digit 7.

Appendix 4: Proof of Accuracy Improvement Property

We divide the Accuracy Improvement Property into two parts, and give the proof for each as following:

Property 1 The estimation error in interaction measures, $L = \sum_{i,j} |\text{M-GEH}_g^{i,j} - \text{M-GEH}_f^{i,j}|$, is linearly upper bounded by the prediction error ϵ of $g(\cdot)$.

Property 2 When $g^{\mathbf{W}}(\cdot)$ is a probabilistic model (e.g. BNN), we can make the uncertainty of the Bayesian GEH arbitrarily well-calibrated by improving the calibration of the distribution of predictions from $g^{\mathbf{W}}(\cdot)$.

Proof of Property 1

The estimation error of interaction effect between feature i and j, $L^{i,j} = |\text{M-GEH}_g^{i,j} - \text{M-GEH}_f^{i,j}|$, can be further derived through:

$$\begin{aligned} L^{i,j} &= \sum_{m=1}^{M} \frac{|A_m|}{\sum_{k=1}^{M}|A_k|} \left| \mathbb{E}_{p(\mathbf{x}|\mathbf{x}\in A_m)} \left[\frac{\partial^2 g^{\mathbf{W}}(\mathbf{x})}{\partial x_i \partial x_j} \right] \right| - \sum_{m=1}^{M} \frac{|A_m|}{\sum_{k=1}^{M}|A_k|} \left| \mathbb{E}_{p(\mathbf{x}|\mathbf{x}\in A_m)} \left[\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \right] \right| \\ &\leq \sum_{m=1}^{M} \frac{|A_m|}{\sum_{k=1}^{M}|A_k|} \left| \mathbb{E}_{p(\mathbf{x}|\mathbf{x}\in A_m)} \left[\frac{\partial^2 (g^{\mathbf{W}}(\mathbf{x}) - f(\mathbf{x}))}{\partial x_i \partial x_j} \right] \right| = \sum_{m=1}^{M} \frac{|A_m|}{\sum_{k=1}^{M}|A_k|} l_m^{i,j} \end{aligned} \tag{4}$$

where g is the learned neural network, and f is the underlining data generating process. We denote $l_m^{i,j}$ as the estimation error from mth group.

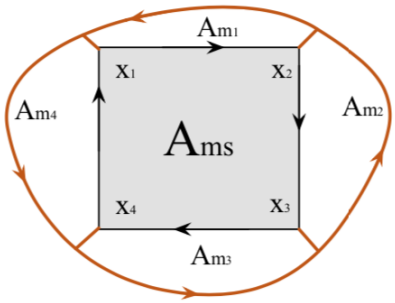

Figure 5: Decomposition of each group

$$\begin{aligned} l_m^{i,j} &= \left| \iint_{A_m} \left[\frac{\partial^2 (g^{\mathbf{W}}(\mathbf{x}) - f(\mathbf{x}))}{\partial x_i \partial x_j} \right] p(\mathbf{x}|\mathbf{x} \in A_m) dx_i dx_j \right| \\ &\leq P_m \left| \iint_{A_m} \frac{\partial^2 (g^{\mathbf{W}}(\mathbf{x}) - f(\mathbf{x}))}{\partial x_i \partial x_j} dx_i dx_j \right| \\ &= P_m \left| \iint_{A_{ms}+A_{m1}+A_{m2}+A_{m3}+A_{m4}} \frac{\partial^2 (g^{\mathbf{W}} - f)(\mathbf{x})}{\partial x_i \partial x_j} dx_i dx_j \right| \end{aligned} \tag{5}$$

where p_m is the highest density of the conditional probability distribution $p(\mathbf{x}|\mathbf{x} \in A_m)$. We divide the domain of A_m into finite subregions, which contains a rectangle subregion A_{ms}, and several non-rectangled subregions (for example $\{A_{mi}\}_{i=1}^{4}$ in Figure 5), and the rectangle subregion does not have to touch the boundary of the group. This is generally true if the domain of each group is compact.

Use Figure 3 as an example, for subregion A_{ms},

$$\begin{aligned} &\left| \iint_{A_{ms}} \frac{\partial^2 (g^{\mathbf{W}}(\mathbf{x}) - f(\mathbf{x}))}{\partial x_i \partial x_j} dx_i dx_j \right| \\ =&\Big| (g^{\mathbf{W}}(\mathbf{x}_2) - f(\mathbf{x}_2)) + (g^{\mathbf{W}}(\mathbf{x}_4) - f(\mathbf{x}_4)) - (g^{\mathbf{W}}(\mathbf{x}_1) - f(\mathbf{x}_1)) + (g^{\mathbf{W}}(\mathbf{x}_3) - f(\mathbf{x}_3)) \Big| \\ \leq& 4\epsilon \end{aligned} \tag{6}$$

where ϵ is the prediction error of $g(\cdot)$. For those non-rectangled subresions, such as A_{m1}, according to Green's theorem:

$$\begin{aligned} &\left| \iint_{A_{m1}} \frac{\partial^2 (g^{\mathbf{W}}(\mathbf{x}) - f(\mathbf{x}))}{\partial x_i \partial x_j} dx_i dx_j \right| \\ =&\left| \oint_{A_{m1}} \frac{\partial (g^{\mathbf{W}}(\mathbf{x}) - f(\mathbf{x}))}{\partial x_j} dx_j \right| \leq L \left| \oint_{A_{m1}} dx_j \right| = L|\Delta x_j|. \end{aligned} \tag{7}$$

Here we assume that $g(\cdot)$ and $f(\cdot)$ are both L-Lipschitz functions, and Δx_j is the maximum difference of feature x_j in subregion A_{m1}.

Based on the above reasoning, we can conclude that:

$$L^{i,j} = \sum_{m=1}^{M} \frac{|A_m|}{\sum_{k=1}^{M} |A_k|} l_m^{i,j} \leq \alpha\epsilon + \beta L, \tag{8}$$

thus we proved property 1.

From Eq.8, we can notice that the upper bound consists of two parts: $\alpha\epsilon$ and βL. If the area of the rectangle subregion A_{ms} is large, β will be small, and the bound will be tighter and also will be dominated by the prediction error. Moreover, if we want to make the bound even much tighter, instead of one rectangle we can use a combination of multiple rectangles inside the region.

Thus training a better BNN, can reduce the upper bound of interaction estimation error, thus can obtain a more stable and accurate estimated interaction effects.

Proof of Property 2

If we denote $\pi(\mathbf{y}, \mathbf{x}) = \pi(\mathbf{y}|\mathbf{x})\pi(\mathbf{x})$ is the true model and $p(g^{\mathbf{W}}(\mathbf{x})|\mathbf{x})$ is the posterior predictive distribution of model $g^{\mathbf{W}}(\mathbf{x})$ given $\mathbf{x}$, we call the uncertainty of $p(g^{\mathbf{W}}(\mathbf{x})|\mathbf{x})$ *perfectly calibrated* when $\pi(\mathbf{y}|\mathbf{x}) = p(g^{\mathbf{W}}(\mathbf{x})|\mathbf{x})$. Another way to define calibration is: if the ξ percentage Bayesian credible interval of the mean of $p(g^{\mathbf{W}}(\mathbf{x})|\mathbf{x})$ is the same as the ξ frequentist confidence interval of the mean of $\pi(\mathbf{y}|\mathbf{x})$ for all $\xi \in [0, 1]$ with an infinite number of data, predictive distribution $p(g^{\mathbf{W}}(\mathbf{x})|\mathbf{x})$ is *perfectly calibrated*. So if the credible interval is closer to the corresponding confidence interval, $p(g^{\mathbf{W}}(\mathbf{x})|\mathbf{x})$ is *better calibrated*. Here we assume that the mean of both $\pi(\mathbf{y}|\mathbf{x})$ and $p(g^{\mathbf{W}}(\mathbf{x})|\mathbf{x})$ are Gaussian distributed, which is generally true according to CLT.

In the rest of this section, we first define the *closeness* between the ξ credible interval and the ξ confidence interval. Then we show that for two predictive model $g^{\mathbf{W}_1}(\cdot)$ and $g^{\mathbf{W}_2}(\cdot)$, if $p(g^{\mathbf{W}_1}(\mathbf{x})|\mathbf{x})$ is better calibrated than $p(g^{\mathbf{W}_2}(\mathbf{x})|\mathbf{x})$, then $p(\sum_{i\in S} \phi_i g^{\mathbf{W}_1}(\mathbf{x}^{(i)})|\mathbf{x}^{(i\in S)})$ is better calibrated than $p(\sum_{i\in S} \phi_i g^{\mathbf{W}_2}(\mathbf{x}^{(i)})|\mathbf{x}^{(i\in S)})$, where $\mathbf{x}^{(i\in S)}$ is the set of $|S|$ data points sampled from data distribution. Thus we have proved property 2, because gradient (or Hessian) can be regarded as a linear combination with infinitesimal changes, thus Eq.**??** can be written in the form $\sum_i \phi_i g^{\mathbf{W}}(\mathbf{x}^{(i)})$ with properly chosen ϕ_i.

Closeness between Two Intervals

We denote CreI_ξ to be the ξ credible interval of $\hat{m}$, the mean of $p(g^{\mathbf{W}}(\mathbf{x})|\mathbf{x})$, and ConI_ξ to be the ξ confidence interval of m, the mean of $\pi(\mathbf{y}|\mathbf{x})$. Then we define the closeness of CreI_ξ and ConI_ξ to be:

$$\delta(\xi) = \frac{|\text{CreI}_\xi \cap \text{ConI}_\xi|}{|\text{CreI}_\xi \cup \text{ConI}_\xi|}.$$

When $\delta(\xi) = 1$, two intervals perfectly match, and when $\delta(\xi) = 0$, two intervals have no intersection.

Calibration Improvement Preserved under Linear Combination

We first prove that if $p(g^{\mathbf{W}_1}(\mathbf{x})|\mathbf{x})$ is better calibrated than $p(g^{\mathbf{W}_2}(\mathbf{x})|\mathbf{x})$, then $p(g^{\mathbf{W}_1}(\mathbf{x}^{(i)}) + g^{\mathbf{W}_1}(\mathbf{x}^{(j)})|\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is better calibrated than $p(g^{\mathbf{W}_2}(\mathbf{x}^{(i)}) + g^{\mathbf{W}_2}(\mathbf{x}^{(j)})|\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$.

We denote that $\hat{m}_i \sim N(\hat{\mu}_i, \hat{\sigma}_i^2)$, $\hat{m}_j \sim N(\hat{\mu}_j, \hat{\sigma}_j^2)$ where $\hat{m}_i$ and $\hat{m}_j$ are the mean of $p(g^{\mathbf{W}}(\mathbf{x}^{(i)}))$ and $p(g^{\mathbf{W}}(\mathbf{x}^{(j)}))$ respectively. And $m_i \sim N(\mu_i, \sigma_i^2)$, $m_j \sim N(\mu_j, \sigma_j^2)$, where m_i and m_j are the mean of $\pi(\mathbf{y}^{(i)}|\mathbf{x}^i)$ and $\pi(\mathbf{y}^{(j)}|\mathbf{x}^j)$.

We only present the case[1] when $\text{CreI}_\xi \cap \text{ConI}_\xi$ is different from CreI_ξ or ConI_ξ. We can calculate the intervals based on Gaussian: $\text{CreI}_\alpha^{(i)} = [\hat{\mu}_i - \alpha\hat{\sigma}_i, \hat{\mu}_i + \alpha\hat{\sigma}_i]$ and $\text{ConI}_\alpha^{(i)} = [\mu_i - \alpha\sigma_i, \mu_i + \alpha\sigma_i]$, where α is the value of percent point function for ξ. Then for data $\mathbf{x}^{(i)}$, the closeness of interval is

$$\delta_i(\alpha) = \frac{\alpha(\sigma_i + \hat{\sigma}_i) + \mu_i - \hat{\mu}_i}{\alpha(\sigma_i + \hat{\sigma}_i) + \hat{\mu}_i - \mu_i} = 1 - 2\frac{\hat{\mu}_i - \mu_i}{\alpha(\sigma_i + \hat{\sigma}_i) + \hat{\mu}_i - \mu_i}, \tag{9}$$

[1] It is easy to prove when one interval contains another interval.

where we assume $\hat{\mu}_i + \alpha\hat{\sigma}_i$ is greater than $\mu_i + \alpha\sigma_i$ (another case can be shown in the same way). And also for data $\mathbf{x}^{(j)}$:

$$\delta_j(\alpha) = \frac{\alpha(\sigma_j + \hat{\sigma}_j) + \mu_j - \hat{\mu}_j}{\alpha(\sigma_j + \hat{\sigma}_j) + \hat{\mu}_j - \mu_j} = 1 - 2\frac{\hat{\mu}_j - \mu_j}{\alpha(\sigma_j + \hat{\sigma}_j) + \hat{\mu}_j - \mu_j}. \tag{10}$$

The mean of $p(g^{\mathbf{W}_2}(\mathbf{x}^{(i)}) + g^{\mathbf{W}_2}(\mathbf{x}^{(j)})|\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is equal to $\hat{m}_i + \hat{m}_j$, because $\mathbf{x}^i$ and $\mathbf{x}^j$ are independent. Thus $\hat{m}_i + \hat{m}_j \sim N(\hat{\mu}_i + \hat{\mu}_j, \hat{\sigma}_i^2 + \hat{\sigma}_j^2)$, and also $m_i + m_j \sim N(\mu_i + \mu_j, \sigma_i^2 + \sigma_j^2)$. Here we consider the intervals with percent point function equals to $\sqrt{2}\alpha$, then $\text{CreI}^{(i,j)}_{\sqrt{2}\alpha} = \big[\hat{\mu}_i + \hat{\mu}_j - \sqrt{2}\alpha\sqrt{\hat{\sigma}_i^2 + \hat{\sigma}_j^2}, \hat{\mu}_i + \hat{\mu}_j + \sqrt{2}\alpha\sqrt{\hat{\sigma}_i^2 + \hat{\sigma}_j^2}\big]$ and $\text{ConI}^{(i,j)}_{\sqrt{2}\alpha} = \big[\mu_i + \mu_j - \sqrt{2}\alpha\sqrt{\sigma_i^2 + \sigma_j^2}, \mu_i + \mu_j + \sqrt{2}\alpha\sqrt{\sigma_i^2 + \sigma_j^2}\big]$. Thus the closeness of these two intervals is:

$$\begin{aligned}
\delta_{i,j}(\sqrt{2}\alpha) &= \frac{\sqrt{2}\alpha(\sqrt{\sigma_i^2 + \sigma_j^2} + \sqrt{\hat{\sigma}_i^2 + \hat{\sigma}_j^2}) + (\mu_i + \mu_j) - (\hat{\mu}_i + \hat{\mu}_j)}{\sqrt{2}\alpha(\sqrt{\sigma_i^2 + \sigma_j^2} + \sqrt{\hat{\sigma}_i^2 + \hat{\sigma}_j^2}) + (\hat{\mu}_i + \hat{\mu}_j) - (\mu_i + \mu_j)} \\
&= 1 - 2\frac{(\hat{\mu}_i + \hat{\mu}_j) - (\mu_i + \mu_j)}{\sqrt{2}\alpha(\sqrt{\sigma_i^2 + \sigma_j^2} + \sqrt{\hat{\sigma}_i^2 + \hat{\sigma}_j^2}) + (\hat{\mu}_i + \hat{\mu}_j) - (\mu_i + \mu_j)} \\
&\geq 1 - 2\frac{(\hat{\mu}_i + \hat{\mu}_j) - (\mu_i + \mu_j)}{\alpha(\sigma_i + \sigma_j + \hat{\sigma}_i + \hat{\sigma}_j) + (\hat{\mu}_i + \hat{\mu}_j) - (\mu_i + \mu_j)} \geq \delta_i(\alpha) + \delta_j(\alpha) - 1
\end{aligned} \tag{11}$$

So when $p(g^{\mathbf{W}_1}(\mathbf{x})|\mathbf{x})$ is better calibrated than $p(g^{\mathbf{W}_2}(\mathbf{x})|\mathbf{x})$, we have $\delta_i^1(\alpha) > \delta_i^2(\alpha)$ and $\delta_j^1(\alpha) > \delta_j^2(\alpha)$. Then the lower bound of $\delta_{i,j}^1(\sqrt{2}\alpha)$ will be greater than $\delta_{i,j}^2(\sqrt{2}\alpha)$, and this applies for all α, so $p(g^{\mathbf{W}_1}(\mathbf{x}^{(i)}) + g^{\mathbf{W}_1}(\mathbf{x}^{(j)})|\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is better calibrated than $p(g^{\mathbf{W}_2}(\mathbf{x}^{(i)}) + g^{\mathbf{W}_2}(\mathbf{x}^{(j)})|\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$. When $p(g^{\mathbf{W}_1}(\mathbf{x})|\mathbf{x})$ is perfectly calibrated, we have $\delta_{i,j}^1(\sqrt{2}\alpha) \geq \delta_i^1(\alpha) + \delta_j^1(\alpha) - 1 = 1$, thus $p(g^{\mathbf{W}_1}(\mathbf{x}^{(i)}) + g^{\mathbf{W}_1}(\mathbf{x}^{(j)})|\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is also perfectly calibrated.

It can be generalized to the distribution of all possible linear combinations of predictions trivially, since the linear combination of independent Gaussian distributions are also Gaussian distribution with linearly combined mean and standard deviation. So $p(\sum_{i\in S}\phi_i g^{\mathbf{W}_1}(\mathbf{x}^{(i)})|\mathbf{x}^{(i\in S)})$ is also better calibrated than $p(\sum_{i\in S}\phi_i g^{\mathbf{W}_2}(\mathbf{x}^{(i)})|\mathbf{x}^{(i\in S)})$ if $p(g^{\mathbf{W}_1}(\mathbf{x})|\mathbf{x})$ is better calibrated than $p(g^{\mathbf{W}_2}(\mathbf{x})|\mathbf{x})$.