Beyond Unimodal: Generalising Neural Processes for Multimodal Uncertainty Estimation Appendix

A Lemma and Proof

For the comprehensiveness of proof, we duplicate Lemma 3.1 here.

Lemma A.1 (Gaussian posterior distribution with factorised prior distribution)**.** *If we have* $p(x_i|\mu) = \mathcal{N}(x_i|\mu, \Sigma_i)$ *and* $p(\mu) = \prod_{i=1}^{n} \mathcal{N}(\mu_{0,i}, \Sigma_{0,i})$ *for* n *i.i.d. observations of* D *dimensional vectors, then the mean and covariance of posterior distribution* $p(\mu|x) = \mathcal{N}(\mu|\mu_n, \Sigma_n)$ *are:*

$$\Sigma_n = \left[\sum_{i=1}^{n} \left(\Sigma_i^{-1} + \Sigma_{0,i}^{-1}\right)\right]^{-1}, \quad \mu_n = \Sigma_n \left[\sum_{i=1}^{n} \left(\Sigma_i^{-1} x_i + \Sigma_{0,i}^{-1} \mu_{0,i}\right)\right] \tag{1}$$

Proof.

$$p(\mu|x) \propto \prod_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^D |\Sigma_i|}} \exp\left(-\frac{1}{2}(x_i - \mu)^T \Sigma_i^{-1} (x_i - \mu)\right) \times \frac{1}{\sqrt{(2\pi)^D |\Sigma_{0,i}|}} \exp\left(-\frac{1}{2}(\mu - \mu_{0,i})^T \Sigma_{0,i}^{-1} (\mu - \mu_{0,i})\right) \tag{2}$$

$$\propto \exp\left[-\frac{1}{2}\left(\sum_{i=1}^{n} (\mu - x_i)^T \Sigma_i^{-1} (\mu - x_i) + (\mu - \mu_{0,i})^T \Sigma_{0,i}^{-1} (\mu - \mu_{0,i})\right)\right] \tag{3}$$

$$\propto \exp\left[-\frac{1}{2}\left(\mu^T \left(\sum_{i=1}^{n} \left(\Sigma_i^{-1} + \Sigma_{0,i}^{-1}\right)\right) \mu - 2\mu^T \left(\sum_{i=1}^{n} \left(\Sigma_i^{-1} x_i + \Sigma_{0,i}^{-1} \mu_{0,i}\right)\right)\right)\right] \tag{4}$$

$$= \frac{1}{\sqrt{(2\pi)^D |\Sigma_n|}} \exp\left(-\frac{1}{2}(\mu - \mu_n)^T \Sigma_n^{-1} (\mu - \mu_n)\right) \tag{5}$$

where we dropped constant terms for clarity. From Equation (4) and (5), we can see that:

$$\Sigma_n^{-1} = \sum_{i=1}^{n} \left(\Sigma_i^{-1} + \Sigma_{0,i}^{-1}\right), \quad \Sigma_n^{-1} \mu_n = \sum_{i=1}^{n} \left(\Sigma_i^{-1} x_i + \Sigma_{0,i}^{-1} \mu_{0,i}\right) \tag{6}$$

$$\Sigma_n = \left[\sum_{i=1}^{n} \left(\Sigma_i^{-1} + \Sigma_{0,i}^{-1}\right)\right]^{-1}, \quad \mu_n = \Sigma_n \left[\sum_{i=1}^{n} \left(\Sigma_i^{-1} x_i + \Sigma_{0,i}^{-1} \mu_{0,i}\right)\right] \tag{7}$$

□

If we use Lemma A.1 with diagonal covariance matrices for $p(r_{*,i}^m|z_i) = \mathcal{N}\left(r_{*,i}^m|z_i, \text{diag}(s_{*,i}^m)\right)$ and $p(z_i) = \prod_{m=1}^{M} \mathcal{N}\left(u^m, \text{diag}(q^m)\right)$, we can obtain the posterior distribution of $\mathcal{N}\left(z_i|\mu_{z_i}, \text{diag}(\sigma_{z_i}^2)\right)$ as follows:

$$\sigma_{z_i}^2 = \left[\sum_{m=1}^{M} \left((s_{*,i}^m)^{\oslash} + (q^m)^{\oslash}\right)\right]^{\oslash}, \mu_{z_i} = \sigma_{z_i}^2 \otimes \left[\sum_{m=1}^{M} \left(r_{*,i}^m \otimes (s_{*,i}^m)^{\oslash} + u^m \otimes (q^m)^{\oslash}\right)\right] \tag{8}$$

where $\oslash$ is the element-wise inversion, and $\otimes$ is the element-wise product.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

B Experimental Details

In this section, we outline additional details of the experimental settings including the datasets (Appendix B.1), hyperparameters of the models used (Appendix B.2), metrics (Appendix B.3), and a brief analysis of computational complexity of MGP and MNPs (Appendix B.4). For all the experiments, we used the Adam optimiser [10] with batch size of 200 and the Tensorflow framework. All the experiments were conducted on a single NVIDIA GeForce RTX 3090 GPU.

B.1 Details of Datasets

Synthetic Dataset Figure 2 in the main paper shows the predictive probability and the attention weight of different attention mechanisms. Here, we describe the dataset and the settings used for the demonstrations.

We generated 1,000 synthetic training samples (i.e., $N_{train} = 1,000$) for binary classification by using the Scikit-learn's moon dataset [1] with zero-mean Gaussian noise ($std = 0.15$) added. The test samples were generated as a mesh-grid of 10,000 points (i.e., 100×100 grid with $N_{test} = 10,000$). The number of points in the context memory N^m was set to 100. In this demonstration, we simplified the problem by setting $M = 1$ which is equivalent to the unimodal setting and illustrated the difference in attention mechanisms.

Robustness to Noisy Samples Dataset In Section 5.1, we evaluated the models' robustness to noisy samples with the six multimodal datasets. The details of each dataset are outlined in Table 1. These datasets lie within a feature space where each feature extraction method can be found in [5].

Table 1: Multimodal datasets used for evaluating robustness to noisy samples.

	Dataset					
	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
# of modalities	6	2	3	2	3	2
Types of modalities	Images	Image&Text	Images	Images	Images	Images
# of samples	2,000	11,788	680	8,677	4,485	6,718
# of classes	10	10	68	101	15	51

OOD Detection Dataset We used CIFAR10-C [6] which consists of corrupted images of CIFAR10 [11]. 15 types of corruptions and five levels of corruption for each type are available for the dataset. Following [8], we used the first three types as multimodal inputs with different levels of corruption (1, 3, and 5).

B.2 Details of Models

In our main experiments, four unimodal baselines with the early fusion (EF) method [1] (MC Dropout, Deep Ensemble (EF), SNGP, and ETP) and three multimodal baselines with the late fusion (LF) method [1] (Deep Ensemble (LF), TMC, and MGP) were used. In this section, we describe the details of the feature extractors and each baseline.

Feature Extractors We used the same feature extractor for all the methods to ensure fair comparisons of the models. For the synthetic dataset, the 2D input points were projected to a high-dimensional space ($d^m = 128$) with a feature extractor that has 6 residual fully connected (FC) layers with the ReLU activation. For the OOD detection experiment, the Inception v3 [14] pretrained with ImageNet was used as the feature extractor. Note that the robustness to noisy samples experiment does not require a separate feature extractor as the dataset is already in a feature space.

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html`

MC Dropout Monte Carlo (MC) Dropout [3] is a well-known uncertainty estimation method that leverages existing dropout layers of DNNs to approximate Bayesian inference. In our experiments, the dropout rate was set to 0.2 with 100 dropout samples used to make predictions in the inference stage. The predictive uncertainty was quantified based on the original paper [3].

Deep Ensemble Deep ensemble [12] is a powerful uncertainty estimation method that trains multiple independent ensemble members. In the case of the unimodal baseline, we employed five ensemble members, whereas for the multimodal baseline, a single classifier was trained independently for each modality input. In both scenarios, the unified predictions were obtained by averaging the predictions from the ensemble members, while the predictive uncertainty was determined by calculating the variance of those predictions.

SNGP Spectral-normalized Neural Gaussian Process (SNGP) [13] is an effective and scalable uncertainty estimation method that utilises Gaussian process (GP). It consists of a feature extractor with spectral normalisation and a GP output layer. Since we used the identical feature extractor for all the baselines, we only used the GP layer in this work. Following [8], the model's covariance matrix was updated without momentum with $\lambda = \pi/8$ for the mean-field approximation. As the original authors proposed, we quantified the predictive uncertainty based on the Dempster-Shafer theory [2] defined as $u(x) = K/(K + \sum_{k=1}^{K} \exp(\text{logit}_k(x)))$ where $\text{logit}_k(\cdot)$ is the k^{th} class of output logit with the number of classes K.

ETP Evidential Turing Processes (ETP) [9] is a recent variant of NPs for uncertainty estimation of image classification. Since none of the existing NPs can be directly applied to multimodal data, there are several requirements to utilise them for multimodal classification: 1) a context set in the inference stage (e.g., context memory) and 2) a method of processing multimodal data. ETP was selected due to its inclusion of the original context memory, requiring minimal modifications to be applicable to our task. We used the memory size of 200 and quantified the predictive uncertainty with entropy as proposed by the original paper [9].

TMC Trusted Multi-view Classification (TMC) [5] is a simple multimodal uncertainty estimation based on the Subjective logic [7]. We used the original settings of the paper with the annealing epochs of ten for the balancing term. TMC explicitly quantifies its predictive uncertainty based on the Dempster-Shafer theory [2].

MGP Multi-view Gaussian Process (MGP) [8] is the current SOTA multimodal uncertainty estimation method that combines predictive posterior distributions of multiple GPs. We used the identical settings of the original paper with the number of inducing points set to 200 and ten warm-up epochs. Its predictive uncertainty was quantified by the predictive variance as proposed by the original paper [8].

MNPs (Ours) The encoders and decoder in Multimodal Neural Processes (MNPs) consist of two FC layers with the Leaky ReLU activation [16] after the first FC layer. A normalisation layer is stacked on top of the second FC layer for the encoders. For $\text{enc}_\psi^m(\cdot)$ and $\text{enc}_\omega^m(\cdot)$ that approximate the variance of distributions, we ensure positivity by transforming the outputs as $h_+ = 0.01 + 0.99 * \text{Softplus}(h)$ where h is the raw output from the encoders. l^m of the adaptive RBF attention was initialised as $10 * \mathbb{1} \in \mathbb{R}^{d^m}$, and DCM was initialised by randomly selecting training samples. We used five samples for the Monte Carlo method to approximate the integrals in Equations (11)-(13) in the main paper, which we found enough in practice. Refer to Table 2 for the hyperparameters of MNPs. We provide the impact of N^m on the model performance in Appendix C.1.

B.3 Details of Metrics

Apart from test accuracy, we report the expected calibration error (ECE) [4] and the area under the receiver operating characteristic curve (AUC). ECE is defined as:

$$\text{ECE} = \frac{1}{n} \sum_{i=1}^{b} |B_i||\text{acc}(B_i) - \text{conf}(B_i)|$$

Table 2: Hyperparameters of MNPs.

	Dataset						
Parameter	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB	CIFAR10-C
N^m	100	200	300	700	300	400	200
α	1	0.03	1	1	0.0001	1	1
β	1	1	1	1	1	1	1
τ	0.25	0.01	0.1	0.01	0.5	0.01	0.01

where n is the number of testing samples, B_i is a bin with partitioned predictions with the number of bins b, $|B_i|$ is the number of elements in B_i, acc(B_i) is the accuracy of predictions in B_i, and conf(B_i) is the average predictive confidence in B_i. Following [13] and [8], we set $b = 15$. AUC was used for the OOD detection experiment with ground truth labels of class 0 being the ID samples and class 1 being the OOD samples. Each model's predictive uncertainty was used as confidence score to predict whether a test sample is a ID or OD sample.

B.4 Computational Complexity of MGP and MNPs

In addition to the empirical difference of wall-clock time per epoch in Table 5 in the main paper, we provide computational complexity of the two models in Table 3. We assume that the number of inducing points in MGP equals to the number of context points in MNPs. During training of MNPs, each modality requires a cross-attention ($\mathcal{O}(N^m N_T)$) and a contrastive learning ($\mathcal{O}((N_T)^2)$) that sum to $\mathcal{O}(M(N^m N_T + (N_T)^2))$ with M being the number of modalities, whereas during inference, each modality only requires the cross-attention which results in $\mathcal{O}(MN^m N_T)$.

Table 3: Computational complexity of MGP and MNPs.

	Training	Inference
MGP	$\mathcal{O}(M(N^m)^3)$	$\mathcal{O}(M(N^m)^3)$
MNPs (Ours)	$\mathcal{O}(M(N^m N_T + (N_T)^2))$	$\mathcal{O}(MN^m N_T)$

C Ablation Studies

In this section, we analyse MNPs' performance with different settings and show the effectiveness of the proposed framework.

C.1 Context Memory Updating Mechanisms

We compare the updating mechanism of DCM based on MSE in Equation (2)-(3) in the main paper with three other baselines: random sampling, first-in-first-out (FIFO) [15], and cross-entropy based (CE). Random sampling bypasses DCM and randomly selects training samples during inference. For FIFO, we follow the original procedure proposed by [15] that updates the context memory during training and only uses it during inference. CE-based mechanism replaces j^* in Equation (3) in the main paper with $j^* = \underset{j \in \{1,\dots,N_T\}}{\text{argmax}} \frac{1}{K} \sum_{k=1}^{K} -T_Y[j,k] \log(\widehat{T}_Y^m[j,k])$.

We provide experimental results for all the experiments outlined in Section 5. We highlight that random sampling and FIFO achieve high accuracy both without noise and with noise as shown in Table 4 and 6. However, MSE and CE outperform the others in terms of ECE in Table 5 and OOD AUC in Table 7. As MSE and CE select the new context points based on classification error, the selected context points tend to be close to decision boundary, which is the most difficult region to classify. We believe this may contribute to the lower calibration error, suppressing overconfident predictions. The MSE and CE mechanisms show comparable overall results, but we selected MSE for its lower ECE. In terms of time efficiency, Table 8 shows that random sampling is slower than the other three methods.

For DCM updated by MSE, we also provide difference in performance for a range of number of context points N^m in Figure 1-7. For every figure, the bold line indicates the mean value, and the shaded area indicates 95% confidence interval. Unsurprisingly, the training time and the testing time increase with respect to N^m. The general trend in test accuracy across the datasets shows the benefit of increasing the number of context points. However, the performance gain in ECE and OOD AUC is ambivalent as different patterns are observed for different datasets. We leave an in-depth analysis of this behaviour for our future study.

Table 4: Test accuracy with different context memory updating mechanisms (↑).

Updating Mechanism	Dataset					
	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
Random	99.40±0.14	88.50±5.12	94.85±0.90	90.38±1.38	76.03±2.96	68.42±0.53
FIFO	99.30±0.11	90.33±3.26	**95.29±1.85**	91.09±0.97	76.08±1.92	69.65±0.66
CE	99.40±0.14	**93.67±2.25**	95.00±1.43	**93.59±0.27**	77.40±0.73	70.77±1.11
MSE	**99.50±0.00**	93.50±1.71	95.00±0.62	93.46±0.32	**77.90±0.71**	**71.97±0.43**

Table 5: Test ECE with different context memory updating mechanisms (↓).

Updating Mechanism	Dataset					
	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
Random	0.007±0.001	0.069±0.029	0.050±0.009	0.043±0.005	0.059±0.061	0.052±0.006
FIFO	0.007±0.001	0.067±0.021	0.057±0.016	0.027±0.004	0.056±0.048	0.032±0.007
CE	0.006±0.001	0.050±0.016	0.041±0.009	**0.017±0.003**	**0.038±0.010**	0.034±0.008
MSE	**0.005±0.001**	**0.049±0.008**	**0.040±0.005**	**0.017±0.003**	**0.038±0.010**	**0.028±0.006**

Table 6: Average test accuracy across 10 noise levels with different context memory updating mechanisms (↑).

Updating Mechanism	Dataset					
	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
Random	98.39±0.21	83.11±4.08	92.55±0.55	89.36±1.18	72.85±2.30	62.10±0.44
FIFO	98.51±0.11	85.86±2.87	**93.81±0.67**	89.59±1.02	72.59±1.82	63.00±0.89
CE	98.49±0.13	88.80±1.57	93.75±0.72	**92.87±0.21**	73.98±0.41	63.97±0.71
MSE	**98.58±0.10**	**88.96±1.98**	93.80±0.49	92.83±0.18	**74.14±0.35**	**64.11±0.15**

Table 7: Test accuracy (↑), ECE (↓), and OOD detection AUC (↑) with different context memory updating mechanisms.

Updating Mechanism	Test accuracy ↑	ECE ↓	OOD AUC ↑	
			SVHN	CIFAR100
Random	74.61±0.22	0.073±0.005	0.860±0.003	0.777±0.002
FIFO	74.82±0.11	0.073±0.006	0.862±0.007	0.778±0.005
CE	74.70±0.19	0.013±0.002	0.871±0.004	**0.789±0.004**
MSE	**74.92±0.07**	**0.011±0.001**	**0.872±0.002**	0.786±0.005

Table 8: Wall-clock inference time (ms/epoch) with different context memory updating mechanisms.

Updating Mechanism	Dataset						
	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB	CIFAR10-C
Random	31.80±3.68	8.15±2.80	12.14±3.09	255.37±13.25	33.73±3.56	79.19±5.95	710.48±8.58
FIFO	24.91±0.68	5.87±3.06	7.20±2.77	101.02±2.90	**25.04±3.35**	**41.50±2.74**	496.23±10.85
CE	25.00±0.28	5.61±1.56	6.85±1.04	101.10±2.59	25.47±3.77	43.45±3.85	500.79±7.04
MSE	**22.53±1.88**	**5.57±1.5**8	**6.70±0.92**	**101.01±2.38**	26.60±10.37	41.87±2.10	**493.18±9.91**

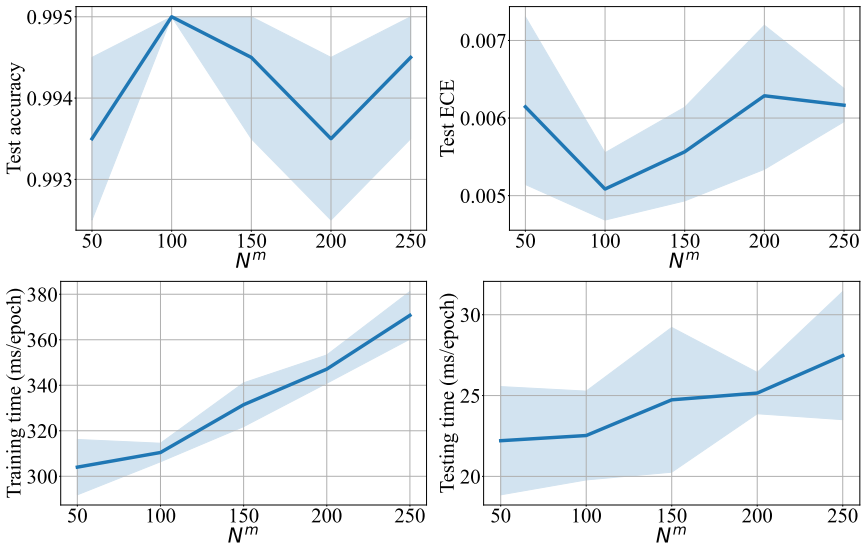

Figure 1: Test accuracy, ECE, average training time, and average testing time with different N^m for the Handwritten dataset.

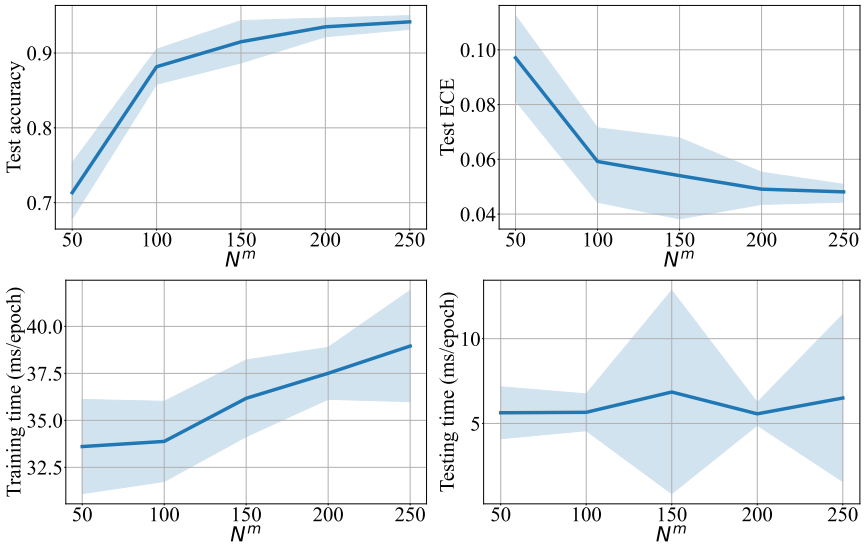

Figure 2: Test accuracy, ECE, average training time, and average testing time with different N^m for the CUB dataset.

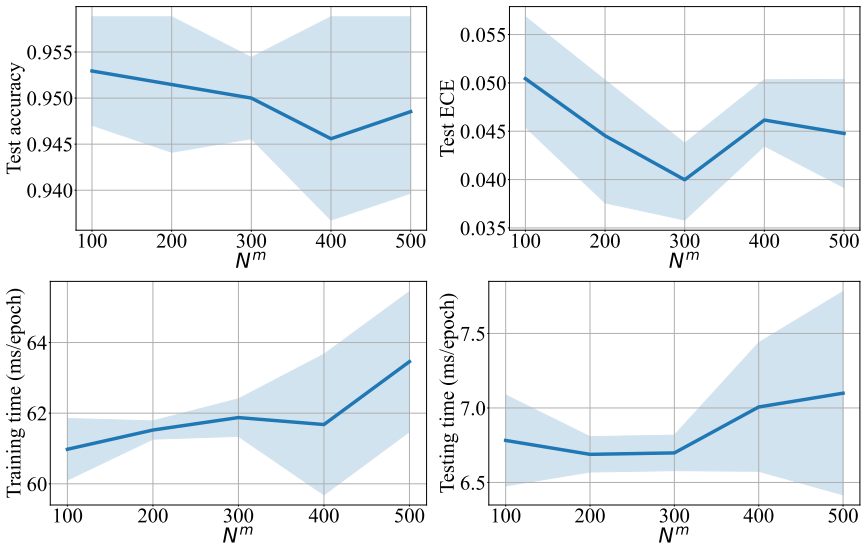

Figure 3: Test accuracy, ECE, average training time, and average testing time with different N^m for the PIE dataset.

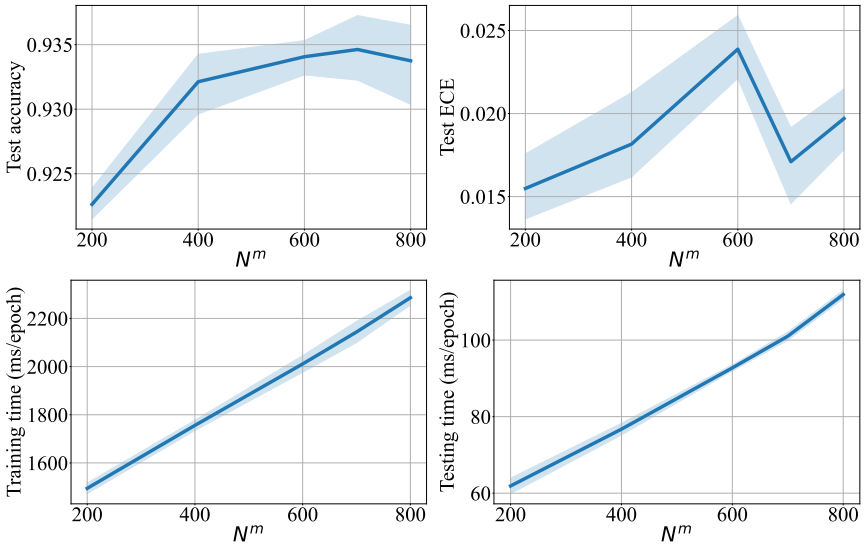

Figure 4: Test accuracy, ECE, average training time, and average testing time with different N^m for the Caltech101 dataset.

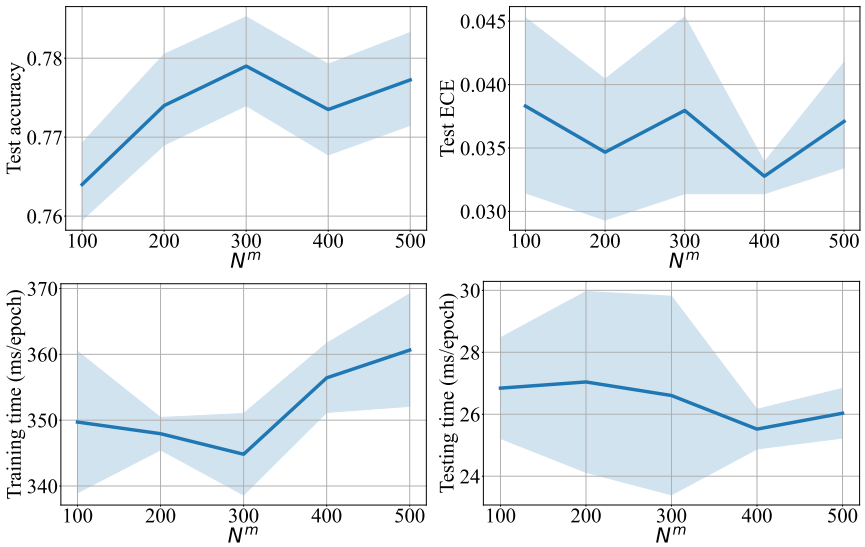

Figure 5: Test accuracy, ECE, average training time, and average testing time with different N^m for the Scene15 dataset.

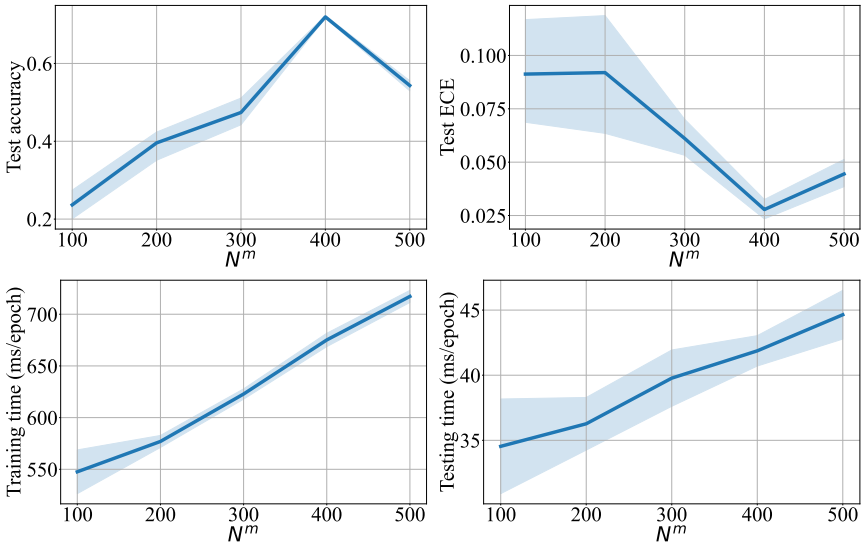

Figure 6: Test accuracy, ECE, average training time, and average testing time with different N^m for the HMDB dataset.

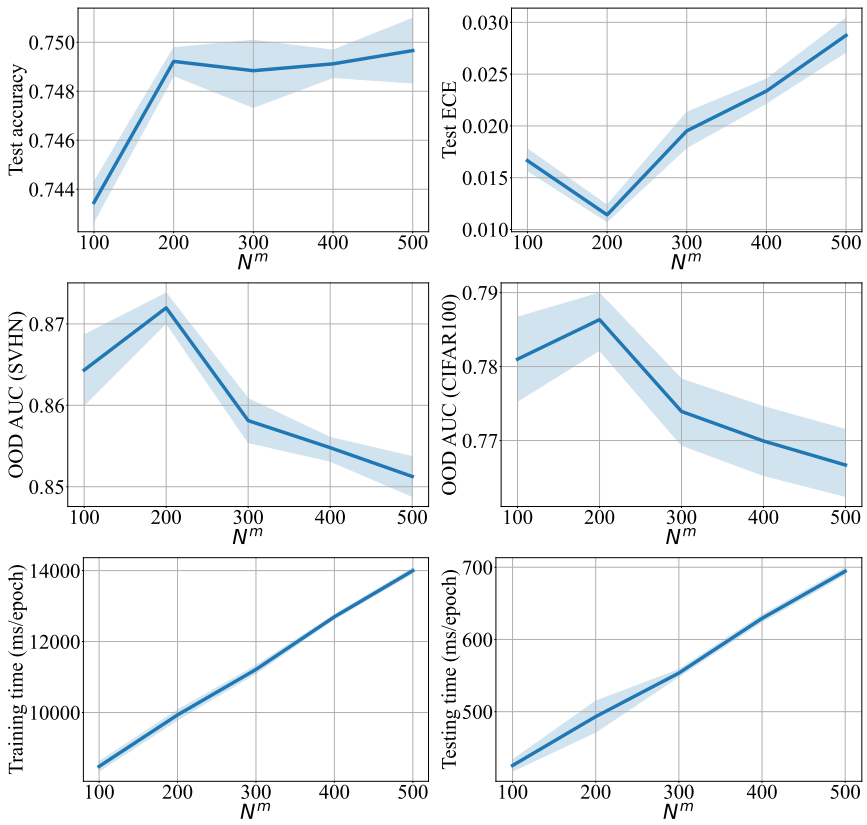

Figure 7: Test accuracy, ECE, OOD AUC (SVHN), OOD AUC (CIFAR100), average training time, and average testing time with different N^m for the CIFAR10-C dataset.

C.2 Multimodal Aggregation Methods

We demonstrate the performance of MBA compared with two other methods namely "Concat" and "Mean". "Concat" bypasses MBA and directly provides r_*^m of multiple modalities to the decoder (see Figure 1 in the main paper) by simple concatenation followed by passing to a MLP which lets $p(f(T_X^M)|C^M, T_X^M)$ in Equation (12) in the main paper be parameterised by a decoder where $\{C^M, T_X^M\} = MLP(Concat(\{r_*^m\}_{m=1}^M))$. $Concat(\cdot)$ represents concatenating multiple vectors along their feature dimension. Similarly,"Mean" also bypasses MBA and simply averages the multiple modalities into single representation. Formally, $p(f(T_X^M)|C^M, T_X^M)$ parameterised by a decoder where $\{C^M, T_X^M\} = \frac{1}{M}\sum_{m=1}^M r_*^m$.

The results are shown in Table 9-12. In every case, MBA outperforms both baselines. While similar performance can be observed for Handwritten, Scene15, and Caltech101, large differences are observed in CUB, PIE, and HMDB across different metrics. The test accuracy of CIFAR10 is almost consistent across all methods, but large gaps in ECE and OOD performance are observed. This highlights the importance of MBA, especially in robustness and calibration performance.

Table 9: Test accuracy with different multimodal aggregation methods (↑).

Aggregation Methods	Dataset					
	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
Concat	99.35±0.22	89.00±1.24	89.71±2.49	92.63±0.18	77.18±0.64	56.06±2.13
Mean	99.45±0.11	92.50±2.43	90.88±2.24	93.14±0.25	77.60±0.56	57.80±1.97
MBA	**99.50±0.00**	**93.50±1.71**	**95.00±0.62**	**93.46±0.32**	**77.90±0.71**	**71.97±0.43**

Table 10: Test ECE with different multimodal aggregation methods (↓).

Aggregation Methods	Dataset					
	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
Concat	0.007±0.001	0.109±0.008	0.092±0.020	0.038±0.005	0.061±0.005	0.060±0.017
Mean	0.006±0.001	0.057±0.012	0.059±0.008	0.030±0.004	0.038±0.005	0.117±0.014
MBA	**0.005±0.001**	**0.049±0.008**	**0.040±0.005**	**0.017±0.003**	**0.038±0.009**	**0.028±0.006**

Table 11: Average test accuracy across 10 noise levels with different multimodal aggregation methods (↑).

Aggregation Methods	Dataset					
	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
Concat	97.71±0.46	85.51±1.42	85.94±2.48	89.84±0.17	72.23±0.52	45.22±2.86
Mean	98.42±0.09	88.27±1.83	88.74±2.33	92.07±0.16	74.06±0.28	49.58±2.24
MBA	**98.58±0.10**	**88.96±1.98**	**93.80±0.49**	**92.83±0.18**	**74.14±0.35**	**64.11±0.15**

Table 12: Test accuracy (↑), ECE (↓), and OOD detection AUC (↑) with different multimodal aggregation methods.

Aggregation Methods	Test accuracy ↑	ECE ↓	OOD AUC ↑	
			SVHN	CIFAR100
Concat	74.24±0.27	0.125±0.005	0.781±0.016	0.728±0.004
Mean	74.72±0.24	0.109±0.003	0.803±0.007	0.742±0.003
MBA	**74.92±0.07**	**0.011±0.001**	**0.872±0.002**	**0.786±0.005**

C.3 Attention Types

We decompose the attention weight $A(T_X^m, C_X^m)$ in Equation (9) in the main paper as follows:

$$A(T_X^m, C_X^m) = \text{Norm}(\text{Sim}(T_X^m, C_X^m)) \tag{9}$$

where $\text{Norm}(\cdot)$ is the normalisation function such as Softmax and Sparsemax, and $\text{Sim}(\cdot, \cdot)$ as the similarity function such as the dot-product and the RBF kernel. We provide experimental results of four different combinations of normalisation functions and similarity functions in Table 13-16.

Among the four combinations, the RBF function with Sparsemax outperforms the others in most cases. More importantly, Table 15 shows a large difference in robustness to noisy samples between the RBF function with Sparsemax and the dot-product with Sparsemax, even when a marginal difference in accuracy is shown in Table 13. For instance, for the PIE dataset, the difference in accuracy without noisy samples is 0.3, but the difference increases to 6.0 in the presence of noisy samples. The same pattern is observed with OOD AUC in Table 16. This illustrates the strength of RBF attention that is more sensitive to distribution-shift as shown in Figure 2 in the main paper. Lastly, for both similarity functions, Sparsemax results in superior overall performance.

Table 13: Test accuracy with different attention mechanisms (↑).

Similarity Function	Normalisation Function	Dataset					
		Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
RBF	Softmax	98.80±0.45	87.00±6.42	75.15±3.00	82.95±0.47	69.83±1.41	56.28±1.18
	Sparsemax	**99.50±0.00**	**93.50±1.71**	**95.00±0.62**	**93.46±0.32**	77.90±0.71	**71.97±0.43**
Dot	Softmax	99.00±0.18	79.67±3.94	86.32±2.88	88.90±0.36	74.95±0.33	64.68±0.78
	Sparsemax	98.95±0.11	82.17±2.67	94.26±1.90	92.46±0.26	**78.30±1.06**	63.23±1.89

Table 14: Test ECE with different attention mechanisms (↓).

Similarity Function	Normalisation Function	Dataset					
		Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
RBF	Softmax	0.019±0.005	0.084±0.020	0.100±0.017	0.025±0.004	0.152±0.007	0.202±0.019
	Sparsemax	**0.005±0.001**	**0.049±0.008**	**0.040±0.005**	**0.017±0.003**	0.038±0.009	**0.028±0.006**
Dot	Softmax	0.008±0.003	0.166±0.015	0.373±0.037	0.033±0.007	0.061±0.010	0.175±0.006
	Sparsemax	0.010±0.001	0.131±0.028	0.053±0.010	0.025±0.002	**0.032±0.008**	0.084±0.015

Table 15: Average test accuracy with different attention mechanisms (↑).

Similarity Function	Normalisation Function	Dataset					
		Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
RBF	Softmax	94.56±0.66	82.58±5.98	65.88±2.98	81.23±0.29	67.77±1.05	38.63±0.63
	Sparsemax	**98.58±0.10**	**88.96±1.98**	**93.80±0.49**	**92.83±0.18**	**74.14±0.35**	**64.11±0.15**
Dot	Softmax	77.99±0.32	73.89±1.77	70.80±1.71	63.80±0.12	58.74±0.24	34.28±0.45
	Sparsemax	96.00±0.24	70.30±2.61	87.44±1.44	81.95±1.92	67.84±1.00	40.26±0.56

Table 16: Test accuracy (↑), ECE (↓), and OOD detection AUC (↑) with different attention mechanisms.

Similarity Function	Normalisation Function	Test accuracy ↑	ECE ↓	OOD AUC ↑	
				SVHN	CIFAR100
RBF	Softmax	67.65±0.16	0.080±0.001	0.864±0.006	0.771±0.006
	Sparsemax	74.92±0.07	**0.011±0.001**	**0.872±0.002**	**0.786±0.005**
Dot	Softmax	68.81±0.62	0.130±0.019	0.849±0.009	0.775±0.005
	Sparsemax	**75.07±0.09**	0.055±0.001	0.837±0.004	0.765±0.004

C.4 Adaptive Learning of RBF Attention

We have shown that the effectiveness of learning the RBF attention's parameters with the synthetic dataset in Figure 2 in the main paper. We further provide the ablation studies with the real-world datasets in Table 17-20.

Table 17: Test accuracy with and without $\mathcal{L}_{RBF}$ (↑).

Method	Dataset					
	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
Without $\mathcal{L}_{RBF}$	96.85±0.29	91.17±2.40	93.38±1.27	92.64±0.38	74.45±0.45	48.95±1.70
With $\mathcal{L}_{RBF}$	**99.50±0.00**	**93.50±1.71**	**95.00±0.62**	**93.46±0.32**	**77.90±0.71**	**71.97±0.43**

Table 18: Test ECE with and without $\mathcal{L}_{RBF}$ (↓).

Method	Dataset					
	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
Without $\mathcal{L}_{RBF}$	0.007±0.001	0.078±0.011	0.043±0.007	0.036±0.004	0.054±0.011	0.043±0.008
With $\mathcal{L}_{RBF}$	**0.005±0.001**	**0.049±0.008**	**0.040±0.005**	**0.017±0.003**	**0.038±0.010**	**0.028±0.006**

Table 19: Average test accuracy across 10 noise levels with and without $\mathcal{L}_{RBF}$ (↑).

Method	Dataset					
	Handwritten	CUB	PIE	Caltech101	Scene15	HMDB
Without $\mathcal{L}_{RBF}$	89.44±0.54	86.69±1.65	91.50±0.94	92.32±0.27	71.18±0.38	37.33±0.92
With $\mathcal{L}_{RBF}$	**98.58±0.10**	**88.96±1.98**	**93.80±0.49**	**92.83±0.18**	**74.14±0.35**	**64.11±0.15**

Table 20: Test accuracy (↑), ECE (↓), and OOD detection AUC (↑) with and without $\mathcal{L}_{RBF}$.

			OOD AUC ↑	
Method	Test accuracy ↑	ECE ↓	SVHN	CIFAR100
Without $\mathcal{L}_{RBF}$	**74.96±0.16**	0.019±0.002	0.822±0.004	0.746±0.004
With $\mathcal{L}_{RBF}$	74.92±0.07	**0.011±0.001**	**0.872±0.002**	**0.786±0.005**

D Broader Impacts

As a long-term goal of this work is to make multimodal classification of DNNs more trustworthy by using NPs, it has many potential positive impacts to our society. Firstly, with transparent and calibrated predictions, more DNNs can be deployed to safety-critical domains such as medical diagnosis. Secondly, this work raises awareness to the machine learning society to evaluate and review reliability of a DNN model. Lastly, our study shows the potential capability of NPs in more diverse applications. Nevertheless, a potential negative impact may exist if the causes of uncertain predictions are not fully understood. To take a step further to reliable DNNs, the source of uncertainty should be transparent to non-experts.

References

[1] T. Baltrušaitis, C. Ahuja, and L.-P. Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2019. doi: 10.1109/TPAMI.2018.2798607.

[2] A. Dempster. Upper and lower probabilities induced by a multi- valued mapping. *Annals of Mathematical Statistics*, 38:325–339, 1967.

[3] Y. Gal and Z. Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.

[4] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017.

[5] Z. Han, C. Zhang, H. Fu, and J. T. Zhou. Trusted multi-view classification. In *International Conference on Learning Representations*, 2021.

[6] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[7] A. Jøsang. Belief fusion. In *Subjective Logic*, volume 3, chapter 12, pages 207–236. Springer, 2016.

[8] M. C. Jung, H. Zhao, J. Dipnall, B. Gabbe, and L. Du. Uncertainty estimation for multi-view data: The power of seeing the whole picture. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[9] M. Kandemir, A. Akgül, M. Haussmann, and G. Unal. Evidential turing processes. In *International Conference on Learning Representations*, 2022.

[10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[11] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[12] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[13] J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7498–7512. Curran Associates, Inc., 2020.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] J. Wang, T. Lukasiewicz, D. Massiceti, X. Hu, V. Pavlovic, and A. Neophytou. NP-match: When neural processes meet semi-supervised learning. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 22919–22934. PMLR, 17–23 Jul 2022.

[16] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.