Pathologies in Priors and Inference for Bayesian Transformers

Anonymous Author(s)
Affiliation
Address
email

Abstract

In recent years, the transformer has established itself as a workhorse in many applications ranging from natural language processing to reinforcement learning. Similarly, Bayesian deep learning has become the gold-standard for uncertainty estimation in safety-critical applications, where robustness and calibration are crucial. Surprisingly, no successful attempts to improve transformer models in terms of predictive uncertainty using Bayesian inference exist. In this work, we study this curiously underpopulated area of Bayesian transformers. We find that weight-space inference in transformers does not work well, regardless of the approximate posterior. We also find that the prior is at least partially at fault, but that it is very hard to find well-specified weight priors for these models. We hypothesize that these problems stem from the complexity of obtaining a meaningful mapping from weight-space to function-space distributions in the transformer. Therefore, moving closer to function-space, we propose a novel method based on the implicit reparameterization of the Dirichlet distribution to apply variational inference directly to the attention weights. We find that this proposed method performs competitively with our baselines.

1 Introduction

The transformer [57] is a deep learning architecture commonly used to process sequences of data, such as text. Thanks to multi-head self-attention, the transformer builds contextual embeddings by capturing the relationships between the sequence elements. While being most famous for their state-of-the-art performance in natural language processing [5, 14], transformers are also used in computer vision [7, 15, 32, 52, 61], reinforcement learning [6, 37], as well as audio [25, 29, 47] and video [63] processing, yielding impressive results.

The Bayesian learning paradigm provides a theoretical framework to obtain predictive uncertainty, select the optimal model, and improve its calibration. Furthermore, by designing an informative prior for the parameters, Bayesian models offer a principled way to incorporate assumptions about the inferred distribution, thus providing regularization. Finally, recent work [36, 44] has shown that Bayesian neural networks (BNN) are often better calibrated than standard neural networks.

If transformers and Bayesian deep learning are both so popular, why have we not seen any successful Bayesian transformer models? By attempting to implement such models, we make the following contributions: (i) We find that weight space inference in transformers does not provide any improvements over a model trained by maximum likelihood. (ii) We show that the prior is at least partially at fault for this. (iii) We propose to perform inference on the attention weights rather than on the parameters, and present a novel variational method for this using the Dirichlet distribution.

Under review at ICBINB Workshop @ NeurIPS 2021. Do not distribute.

2 Background

2.1 Bayesian deep learning

Bayesian inference computes the posterior distribution as

$$P(\theta \mid y_{1:N}, x_{1:N}) = P(y_{1:N} \mid \theta, x_{1:N}) P(\theta) / P(y_{1:N} \mid x_{1:N}) \tag{1}$$

with neural network parameters θ, training data $\{(x_i, y_i)\}_{i=1}^N$, likelihood function $P(y_{1:N} \mid \theta, x_{1:N})$, prior $P(\theta)$, and evidence $P(y_{1:N} \mid x_{1:N})$. The predictive distribution of a new target y^* given x^* is then obtained by

$$P(y^* \mid x^*, y_{1:N}, x_{1:N}) = \mathrm{E}_{\theta \sim P(\theta \mid y_{1:N}, x_{1:N})}[P(y^* \mid \theta, x^*)] \tag{2}$$

Applied to neural networks, both Eq. (1) and Eq. (2) are intractable and need to be estimated using approximate inference methods, such as variational inference or Monte Carlo sampling.

2.2 Bayesian neural network weight space inference

The most commonly used weight space inference methods in BNNs are variational inference (VI) [3, 16, 23, 40, 41, 43], the Laplace method [12, 31, 36], and Markov Chain Monte Carlo (MCMC) [8, 45, 58]. While MCMC methods directly sample from the (unnormalized) posterior, VI and Laplace approximate the posterior by another distribution. As MCMC methods are expensive computationally and memory-wise, we restrict our focus to VI and Laplace. When applying these methods to transformers, we find that weight space inference fails to improve data fit, calibration and predictive uncertainty compared to a model trained by likelihood maximization (see Fig. 1).

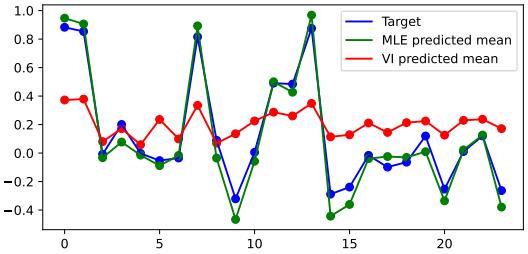


Figure 1: Plot of MLE and VI transformer. MLE captures the mean of the generative process well (MSE: 0.66), but VI does not (MSE: 9.58).

2.3 Empirical weight study

To understand why weight space inference fails, we study the empirical weight distribution of transformers trained with stochastic gradient descent (SGD), hoping to obtain better priors. We follow the framework proposed by Fortuin et al. [21]. We first examine the marginal weight distribution where we especially study the tailedness and modality. We also identify the best-fitting distribution and its parameters within the Gaussian, Student, Logistic, Cauchy, and Laplace families. Furthermore, we investigate the correlation among layer weights by comparing the empirical covariance matrix and the distribution of off-diagonal covariance elements against samples from an isotropic Gaussian.

3 Methods

3.1 Variational attention

As as alternative to weight-space inference in transformers, we propose to treat self-attention weights as random variables and approximate their posterior distribution using VI. Previous attention weight inference methods focus on sampling [1], while others explicitly parameterize the attention weights with a particular distribution [2, 13, 17]. Parameters of explicitly reparameterizable distributions such as the Gaussian [2], Weibull, and Lognormal distributions [17] are learned via VI, while others such as the Dirichlet [13] require using REINFORCE gradient estimators [53].

We implement two baselines for our comparison: 1. *Gaussian attention* where the attention logits are parameterized with a Gaussian distribution and parameters are inferred via VI and 2. *DD*, a data dependent configuration where the variational variances of the Gaussian distribution are amortized in order to support input-dependent (i.e., *heteroscedastic*) uncertainties.

3.2 Implicitly reparameterized Dirichlet attention

Alternatively, we propose to directly parameterize the attention weights of each position i by a Dirichlet distribution with parameter $\boldsymbol{\alpha} = a\mathbf{A_i}$, where a is the sharpness parameter and $\mathbf{A_i}$ the ith row

Table 1: VI and Laplace inference in weight-space compared to maximum likelihood models and concrete dropout. We see that the Bayesian transformers do not outperform the baselines.

Dataset	Metric	MLE	Ensemble	Gaussian VI	Laplace	Final Laplace	Concrete DP	Gauss. Attention	Dir. Attention
M1	Log-like.	-26.206 $\pm$ 0.000	-26.011 $\pm$ 0.007	-27.23 $\pm$ 0.01	-26.282 $\pm$ 0.014	-26.219 $\pm$ 0.003	-25.767 $\pm$ 0.008	-26.1623 $\pm$ 0.0006	**-22.04 $\pm$ 0.01**
	Var. MSE	0.014 $\pm$ 0.000	0.0081 $\pm$ 0.0002	0.082 $\pm$ 0.004	0.021 $\pm$ 0.002	0.020 $\pm$ 0.003	**0.007 $\pm$ 0.000**	0.029 $\pm$ 0.000	0.430 $\pm$ 0.002
	MSE	**0.996 $\pm$ 0.000**	1.0143 $\pm$ 0.0002	1.078 $\pm$ 0.001	1.0432 $\pm$ 0.0009	1.043 $\pm$ 0.002	1.0175 $\pm$ 0.0001	1.007 $\pm$ 0.000	1.0263 $\pm$ 0.0002
M2	Log-like.	-26.5670 $\pm$ 0.000	-28.592 $\pm$ 0.009	-35.43 $\pm$ 0.03	-32.92 $\pm$ 0.05	-32.469 $\pm$ 0.01	-27.11 $\pm$ 0.04	-26.374 $\pm$ 0.002	**-24.841 $\pm$ 0.007**
	Var. MSE	**16.943 $\pm$ 0.000**	23.45 $\pm$ 0.09	110.57 $\pm$ 3.25	47.56 $\pm$ 0.06	47.07 $\pm$ 0.04	21.85 $\pm$ 0.08	20.9010 $\pm$ 0.0007	17.93 $\pm$ 0.03
	MSE	**1.170 $\pm$ 0.000**	1.3552 $\pm$ 0.0003	2.95 $\pm$ 0.02	1.9943 $\pm$ 0.0008	1.972 $\pm$ 0.002	1.192 $\pm$ 0.001	1.2015 $\pm$ 0.0002	1.1928 $\pm$ 0.0006
POS	Log-like.	**-3.707 $\pm$ 0.000**	-4.240 $\pm$ 0.006	-17.86 $\pm$ 0.03	-4.539 $\pm$ 0.000	-4.539 $\pm$ 0.000	-8.2004 $\pm$ 0.0001	-3.9692 $\pm$ 0.0008	-3.9682 $\pm$ 0.0003
	Acc.	0.9706 $\pm$ 0.0000	**0.9708 $\pm$ 0.0001**	0.871 $\pm$ 0.002	0.959 $\pm$ 0.000	0.958 $\pm$ 0.000	0.964 $\pm$ 0.000	0.969 $\pm$ 0.000	0.968 $\pm$ 0.000
	F1	**0.971 $\pm$ 0.000**	**0.971 $\pm$ 0.000**	0.852 $\pm$ 0.000	0.959 $\pm$ 0.000	0.959 $\pm$ 0.000	0.964 $\pm$ 0.000	0.969 $\pm$ 0.000	0.968 $\pm$ 0.000
	ECE	0.03 $\pm$ 0.00	**0.0261 $\pm$ 0.0001**	0.052 $\pm$ 0.001	0.048 $\pm$ 0.000	0.048 $\pm$ 0.000	0.031 $\pm$ 0.000	0.0271 $\pm$ 0.0000	0.0287 $\pm$ 0.0000
MNIST	Log-like.	-0.074 $\pm$ 0.000	-0.1133 $\pm$ 0.0008	-3.18 $\pm$ 0.04	-0.088 $\pm$ 0.000	-0.09 $\pm$ 0.00	**-0.064 $\pm$ 0.000**	-0.0720 $\pm$ 0.0001	-0.1045 $\pm$ 0.0005
	Acc.	0.979 $\pm$ 0.000	**0.9825 $\pm$ 0.0003**	0.101 $\pm$ 0.002	0.972 $\pm$ 0.000	0.972 $\pm$ 0.000	0.981 $\pm$ 0.000	0.9790 $\pm$ 0.0002	0.9738 $\pm$ 0.0003
	F1	0.979 $\pm$ 0.000	**0.982 $\pm$ 0.000**	0.092 $\pm$ 0.000	0.972 $\pm$ 0.000	0.972 $\pm$ 0.000	0.981 $\pm$ 0.000	0.9786 $\pm$ 0.0000	0.9736 $\pm$ 0.0000
	ECE	0.022 $\pm$ 0.000	0.0326 $\pm$ 0.0004	0.097 $\pm$ 0.009	0.035 $\pm$ 0.000	0.038 $\pm$ 0.000	**0.020 $\pm$ 0.000**	0.0227 $\pm$ 0.0002	0.0305 $\pm$ 0.0003

of the scaled dot-product attention weights. We then infer a using VI. Samples are obtained by drawing from independent Gamma distributions $X_k \sim \text{Gamma}(\alpha_k, 1)$ and normalizing $(\sum_{k=1}^{K} X_k)^{-1}\mathbf{X} \sim \text{Dirichlet}(\boldsymbol{\alpha})$. We further use contextual Gamma priors such that $\hat{\boldsymbol{\alpha}} \propto \mathbf{A_i}$, yielding an analytical KL divergence as done by Joo et al. [33]. To obtain gradients of a Gamma random variable with respect to $\boldsymbol{\alpha}$, we use the implicit gradient reparametrization [18]:

$$\nabla_\alpha z = -(q_\alpha(z))^{-1}\nabla_\alpha F(z|\alpha) \tag{3}$$

where $q_\alpha(z)$ is the Gamma density function and $F(z|\alpha)$ its CDF. Like Gaussian attention, we consider a variation where the sharpness parameter depends on the input, referred to as data dependent.

4 Experiments

We run experiments using the transformer [57] and vision transformer [15] on MNIST image classification [39], Universal Dependencies part-of-speech (POS) tagging [46] and on synthetic datasets (M1, M2). We evaluate our models using test log-likelihood, predicted variance mean squared error and the expected mean square error on the synthetic dataset. The test log-likelihood, accuracy, F1-score, and expected calibration error (ECE) [26] are used for experiments on the POS tagging and MNIST datasets. We compare the results obtained by our methods with a transformer (MLE) and an ensemble of 30 transformers both trained by maximum likelihood. Further details are given in Appendix A.2.

4.1 Result 1: Weight-space inference does not improve over MLE

Different posteriors do not help. We find that all weight-space VI methods are outperformed by both maximum-likelihood baselines with respect to all metrics and on all datasets (see Table 1). Interestingly, changing the posterior distribution does not significantly influence the performance, considering the large gap between the scores of the VI methods and baselines (see also Table 3 in the appendix). Furthermore, no variational posterior systematically outperforms the others.
Linearized Laplace inference (either on all parameters or just the final layer) shows much better results than VI. However, it still underperforms our baselines. Finally, even concrete dropout improves over VI and Laplace inference and is more competitive with our baselines.

The prior is (at least partially) at fault. In our attempt to understand the poor performance of weight-space VI in transformers, we conduct an empirical weight distribution study. We find that the marginal weight distributions are essentially uni-modal, except for some embedding and projection layers which tend to have two or three less significant modes. Furthermore, other than a decrease in tailedness (high degree of freedom) of the last layer, no recurrent pattern in the tailedness of the weight distribution appears across the considered datasets (Fig. 4 in the appendix). Likewise, no single distribution seems to universally fit the empirical distributions of the weights across all datasets (see Fig. 5 and Fig. 6 in the appendix). This suggests that the shape of the weight distribution strongly depends on the considered dataset. Using the observations from this weight distribution study, we choose more appropriate ("improved") priors. Table 2 shows systematic likelihood improvements. Moreover, we find that the performance of VI critically depends on the prior parameters (see Fig. 3 in the appendix).

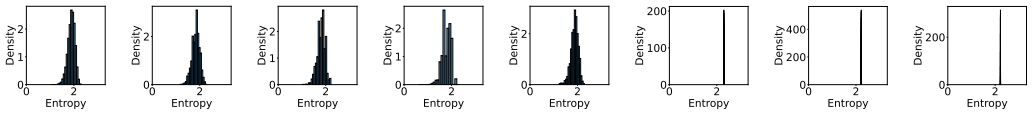
Table 2: Relative improvement of VI with improved priors relative to Gaussian priors.

Dataset	Gauss. VI	Laplace VI	Logistic VI	Cauchy VI	Student VI
M1	1.40%	3.80%	4.12%	1.85%	2.79%
M2	2.85%	3.06%	2.76%	4.36%	2.70%
POS	0.12%	2.05%	2.16%	0.87%	-0.32%
MNIST	26.95%	33.31%	31.36%	5.66%	26.94%

4.2 Results 2: Variational attention is better than weight-space inference

Dirichlet attention works well. Unlike weight-space inference, we find that inference on the attention weights works competitively. Indeed, Dirichlet attention strongly outperforms our baselines in terms of likelihood on the synthetic data and lies between both baselines on the POS tagging and MNIST (Table 1). However, the data dependent configuration does not systematically outperform its standard counterpart (see Table 5 in the appendix). Moreover, Dirichlet attention outperforms Gaussian attention in terms of log-likelihood on the toy data and POS tagging, but not on MNIST.

Variational attention leads to more consistent prior entropies. While investigating the entropy of the predictive distribution when sampling weights from the priors, we find that non-improved priors yield highly variable entropy distributions, ranging from low values around 1 to higher values around 2.3 bits. However, when sampling from improved priors selected by our weight distribution analysis, the entropy distribution concentrates very strongly around a high value of 2.3 bits. This same behavior is observed when sampling from the Gaussian and our proposed Dirichlet attention. This is desirable as the prior predictive should show high uncertainty in function space.



(a) Gaussian (b) Laplace (c) Logistic (d) Cauchy (e) Student (f) Improved (g) Gauss. att. (h) Dir. att.

Figure 2: Prior predictive entropy distributions on MNIST train data. Improving the weight-space priors and using variational attention both lead to more consistently high entropies.

5 Related work

Bayesian Transformers. Previous attempts to make the transformer Bayesian have used VI to perform inference on a subset of layers [56, 62]. While both methods claim state-of-the-art performance on their respective benchmarks, Tran et al. [56] do not provide any quantitative results and Xue et al. [62] initialize their priors to a maximum estimate of the weights which is not strictly Bayesian. Alternatively, Fan et al. [17] parameterize the attention logits of a transformer by a Gaussian distribution and finetune the deterministic self-attention of language models pretrained on large corpora. They however only consider finetuning and not full training using variational attention. Orthogonally, Martin et al. [42] consider attention keys, queries, values, and weights as unobserved random variables and use sequential Monte Carlo methods to sample them.

Bayesian neural network inference. BNN inference has recently advanced in terms of VI methods with more expressive posteriors [16, 40, 41, 43, 55], more efficient inference [22, 23, 54], and greater stability [35, 59]. Likewise, the Laplace inference for BBNs has improved in scalability using further GGN approximations [30, 31, 36, 50, 51] and sub-network inference [12, 36]. Orthogonally, MCMC methods for BNNs have been improved [20, 24, 60, 64], better BNN priors have been studied [19, 21], and even deep ensembles [38] have been cast as approximate inference [9–11, 48, 49].

6 Conclusion

We have shown that weight space inference in Bayesian transformers does not work well, regardless of the choice of posterior. We also found that choosing priors according to an empirical weight distribution analysis improved the performance, suggesting that priors are at least partially at fault. However, we have not found the right priors to make the method competitive. Moreover, we found evidence that naïve weight-space priors lead to low prior predictive entropy, and therefore do not reflect our true beliefs about the output distribution. In order to move closer to the function-space distribution, we suggested to perform inference on the attention weights rather than on parameters. We proposed a novel method based on the implicit reparameterization of the Dirichlet distribution to apply variational inference on the attention weights, which performed competitively with respect to our baselines.

References

[1] Bang An, Jie Lyu, Zhenyi Wang, Chunyuan Li, Changwei Hu, Fei Tan, Ruiyi Zhang, Yifan Hu, and Changyou Chen. Repulsive attention: Rethinking multi-head attention as bayesian inference, 2020.

[2] Hareesh Bahuleyan, Lili Mou, Olga Vechtomova, and Pascal Poupart. Variational attention for sequence-to-sequence models, 2018.

[3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks, 2015.

[4] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[6] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling, 2021.

[7] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/chen20s.html`.

[8] Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo, 2014.

[9] Kamil Ciosek, Vincent Fortuin, Ryota Tomioka, Katja Hofmann, and Richard Turner. Conservative uncertainty estimation by fitting prior networks. In *International Conference on Learning Representations*, 2019.

[10] Francesco D'Angelo and Vincent Fortuin. Repulsive deep ensembles are bayesian. *arXiv preprint arXiv:2106.11642*, 2021.

[11] Francesco D'Angelo, Vincent Fortuin, and Florian Wenzel. On stein variational neural network ensembles. *arXiv preprint arXiv:2106.10760*, 2021.

[12] Erik Daxberger, Eric Nalisnick, James Urquhart Allingham, Javier Antorán, and José Miguel Hernández-Lobato. Bayesian deep learning via subnetwork inference, 2021.

[13] Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander M. Rush. Latent alignment and variational attention, 2018.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[16] Michael W. Dusenberry, Ghassen Jerfel, Yeming Wen, Yi-An Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable bayesian neural nets with rank-1 factors, 2020.

[17] Xinjie Fan, Shujian Zhang, Bo Chen, and Mingyuan Zhou. Bayesian attention modules, 2020.

[18] Michael Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients, 2019.

[19] Vincent Fortuin. Priors in bayesian deep learning: A review. *arXiv preprint arXiv:2105.06868*, 2021.

[20] Vincent Fortuin, Adrià Garriga-Alonso, Mark van der Wilk, and Laurence Aitchison. Bnnpriors: A library for bayesian neural network inference with different prior distributions. *Software Impacts*, 9:100079, 2021.

[21] Vincent Fortuin, Adrià Garriga-Alonso, Florian Wenzel, Gunnar Rätsch, Richard Turner, Mark van der Wilk, and Laurence Aitchison. Bayesian neural network priors revisited. *arXiv preprint arXiv:2102.06571*, 2021.

[22] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, 2016.

[23] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout, 2017.

[24] Adrià Garriga-Alonso and Vincent Fortuin. Exact langevin dynamics with stochastic gradients. *arXiv preprint arXiv:2102.01691*, 2021.

[25] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer, 2021.

[26] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017.

[27] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2020.

[28] Tom Hennigan, Trevor Cai, Tamara Norman, and Igor Babuschkin. Haiku: Sonnet for JAX, 2020. URL `http://github.com/deepmind/dm-haiku`.

[29] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer, 2018.

[30] Alexander Immer, Matthias Bauer, Vincent Fortuin, Gunnar Rätsch, and Mohammad Emtiyaz Khan. Scalable marginal likelihood estimation for model selection in deep learning. *arXiv preprint arXiv:2104.04975*, 2021.

[31] Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of bayesian neural nets via local linearization, 2021.

[32] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up, 2021.

[33] Weonyoung Joo, Wonsung Lee, Sungrae Park, and Il-Chul Moon. Dirichlet variational autoencoder, 2019.

[34] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[35] Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick, 2015.

[36] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks, 2020.

[37] Shakti Kumar, Jerrod Parker, and Panteha Naderian. Adaptive transformers in rl, 2020.

[38] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

[39] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[40] Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors, 2016.

[41] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks, 2017.

[42] Alice Martin, Charles Ollion, Florian Strub, Sylvain Le Corff, and Olivier Pietquin. The monte carlo transformer: a stochastic self-attention model for sequence prediction, 2020.

[43] Aaron Mishkin, Frederik Kunstner, Didrik Nielsen, Mark Schmidt, and Mohammad Emtiyaz Khan. Slang: Fast structured covariance approximations for bayesian deep learning with natural gradient, 2019.

[44] John Mitros and Brian Mac Namee. On the validity of bayesian neural networks for uncertainty estimation, 2019.

[45] Radford Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 06 2012. doi: 10.1201/b10905-6.

[46] Joakim Nivre, Željko Agić, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Cristina Bosco, Sam Bowman, Giuseppe G. A. Celano, Miriam Connor, Marie-Catherine de Marneffe, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Daniel Galbraith, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Berta Gonzales, Bruno Guillaume, Jan Hajič, Dag Haug, Radu Ion, Elena Irimia, Anders Johannsen, Hiroshi Kanayama, Jenna Kanerva, Simon Krek, Veronika Laippala, Alessandro Lenci, Nikola Ljubešić, Teresa Lynn, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Shunsuke Mori, Hanna Nurmi, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Prokopis Prokopidis, Sampo Pyysalo, Loganathan Ramasamy, Rudolf Rosa, Shadi Saleh, Sebastian Schuster, Wolfgang Seeker, Mojgan Seraji, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Kiril Simov, Aaron Smith, Jan Štěpánek, Alane Suhr, Zsolt Szántó, Takaaki Tanaka, Reut Tsarfaty, Sumire Uematsu, Larraitz Uria, Viktor Varga, Veronika Vincze, Zdeněk Žabokrtský, Daniel Zeman, and Hanzhi Zhu. Universal dependencies 1.2, 2015. URL `http://hdl.handle.net/11234/1-1548`. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[47] OpenAI Payne, Christine. "musenet.", 2019. URL `openai.com/blog/musenet`.

[48] Tim Pearce, Nicolas Anastassacos, Mohamed Zaki, and Andy Neely. Bayesian inference with anchored ensembles of neural networks, and application to exploration in reinforcement learning, 2018.

[49] Tim Pearce, Felix Leibfried, Alexandra Brintrup, Mohamed Zaki, and Andy Neely. Uncertainty in neural networks: Approximately bayesian ensembling, 2020.

[50] Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning, 2018.

[51] Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Skdvd2xAZ`.

[52] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation, 2021.

[53] Richard Sutton, David Mcallester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Adv. Neural Inf. Process. Syst*, 12, 02 2000.

[54] Jakub Swiatkowski, Kevin Roth, Bastiaan S. Veeling, Linh Tran, Joshua V. Dillon, Jasper Snoek, Stephan Mandt, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. The k-tied normal distribution: A compact parameterization of gaussian mean field posteriors in bayesian neural networks, 2020.

[55] Marcin Tomczak, Siddharth Swaroop, and Richard Turner. Efficient low rank gaussian variational inference for neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4610–4622. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/310cc7ca5a76a446f85c1a0d641ba96d-Paper.pdf`.

[56] Dustin Tran, Michael W. Dusenberry, Mark van der Wilk, and Danijar Hafner. Bayesian layers: A module for neural network uncertainty, 2019.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[58] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

[59] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches, 2018.

[60] Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020.

[61] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020.

[62] Boyang Xue, Jianwei Yu, Junhao Xu, Shansong Liu, Shoukang Hu, Zi Ye, Mengzhe Geng, Xunying Liu, and Helen Meng. Bayesian transformer language models for speech recognition, 2021.

[63] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers, 2021.

[64] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient mcmc for bayesian deep learning. *arXiv preprint arXiv:1902.03932*, 2019.

A Appendix

A.1 Additional experimental results

A.1.1 Weight space inference

We report in Table 3 the full weight space inference results table discussed in Section 4.1.

Table 3: Weight space inference results in the transformer vs baselines

Dataset	Metric	MLE	Ensemble	Gaussian VI	Laplace VI	Logistic VI	Cauchy VI	Student VI	Concrete Dropout	Laplace	Final Laplace
M1	Log-likelihood	-26.2075 ± 0.0000	-26.011 ± 0.007	-27.23 ± 0.01	-28.01 ± 0.16	-28.08 ± 0.11	-27.61 ± 0.02	-27.68 ± 0.10	**-25.767 ± 0.008**	-26.282 ± 0.014	-26.219 ± 0.003
	Variance MSE	0.0137 ± 0.0000	0.0081 ± 0.0002	0.082 ± 0.004	0.24 ± 0.05	0.226 ± 0.066	0.225 ± 0.009	0.161 ± 0.046	**0.0066 ± 0.0000**	0.021 ± 0.002	0.020 ± 0.003
	MSE	**0.9963 ± 0.0000**	1.0143 ± 0.0002	1.078 ± 0.001	1.174 ± 0.016	1.181 ± 0.014	1.133 ± 0.001	1.128 ± 0.015	1.0175 ± 0.0001	1.0432 ± 0.0009	1.043 ± 0.002
M2	Log-likelihood	**-26.5670 ± 0.0000**	-28.592 ± 0.009	-35.427 ± 0.034	-35.86 ± 0.11	-35.72 ± 0.08	-37.15 ± 0.03	-35.43 ± 0.61	-27.11 ± 0.04	-32.92 ± 0.05	-32.469 ± 0.01
	Variance MSE	**16.9430 ± 0.0000**	23.45 ± 0.09	110.57 ± 3.25	125.08 ± 14.73	121.90 ± 16.25	130.11 ± 1.89	82.47 ± 15.27	21.85 ± 0.08	47.56 ± 0.06	47.07 ± 0.04
	MSE	**1.1700 ± 0.0000**	1.3552 ± 0.0003	2.95 ± 0.02	3.10 ± 0.07	3.03 ± 0.09	3.45 ± 0.01	2.82 ± 0.09	1.192 ± 0.001	1.9943 ± 0.0008	1.972 ± 0.002
POS	Log-likelihood	**-3.7066 ± 0.0000**	-4.240 ± 0.006	-17.86 ± 0.03	-17.69 ± 0.01	-18.02 ± 0.04	-17.07 ± 0.10	-22.91 ± 0.05	-8.2004 ± 0.0001	-4.5388 ± 0.0000	-4.5391 ± 0.0000
	Accuracy	0.9706 ± 0.0000	**0.9708 ± 0.0001**	0.871 ± 0.002	0.8694 ± 0.0003	0.8689 ± 0.0004	0.878 ± 0.002	0.824 ± 0.001	0.9636 ± 0.0000	0.9585 ± 0.0000	0.9584 ± 0.0000
	F1	**0.9707 ± 0.0000**	0.9706 ± 0.0000	0.8524 ± 0.0000	0.8531 ± 0.0000	0.8535 ± 0.0000	0.8594 ± 0.0000	0.7980 ± 0.0000	0.9637 ± 0.0000	0.9585 ± 0.0000	0.9585 ± 0.0000
	ECE	0.0302 ± 0.0000	**0.0261 ± 0.0001**	0.052 ± 0.001	0.0477 ± 0.0007	0.0498 ± 0.0007	0.0543 ± 0.0007	0.050 ± 0.001	0.0314 ± 0.0000	0.0481 ± 0.0000	0.0481 ± 0.0000
MNIST	Log-likelihood	-0.0739 ± 0.0000	-0.1133 ± 0.0008	-3.179 ± 0.038	-3.490 ± 0.125	-3.385 ± 0.126	-2.636 ± 0.016	-3.183 ± 0.009	**-0.0642 ± 0.0000**	-0.0879 ± 0.0000	-0.0903 ± 0.0000
	Accuracy	0.9786 ± 0.0000	**0.9825 ± 0.0003**	0.101 ± 0.002	0.099 ± 0.003	0.099 ± 0.003	0.1024 ± 0.0002	0.099 ± 0.002	0.9807 ± 0.0000	0.9720 ± 0.0000	0.9720 ± 0.0000
	F1	0.9786 ± 0.0000	**0.9820 ± 0.0000**	0.0923 ± 0.0000	0.0173 ± 0.0000	0.0173 ± 0.0000	0.0961 ± 0.0000	0.0173 ± 0.0000	0.9807 ± 0.0000	0.9719 ± 0.0000	0.9720 ± 0.0000
	ECE	0.0218 ± 0.0000	0.0326 ± 0.0004	0.097 ± 0.009	0.108 ± 0.010	0.117 ± 0.012	0.064 ± 0.001	0.110 ± 0.034	**0.0200 ± 0.0000**	0.0354 ± 0.0000	0.0377 ± 0.0000

A.1.2 Sub-network variational inference

In addition to performing inference on the entire set of model parameters presented in Section 4.1, we experiment sub-network variational inference. We find that this method performs better than full inference while still under-performing our baselines. Interestingly, we observe the same behavior as in full network VI, where the posterior distribution does not significantly change the result.

Table 4: VI on first attention layer with Gaussian priors vs baselines

Dataset	Metric	MLE	Ensemble	Gaussian VI	Laplace VI	Logistic VI	Cauchy VI	Student VI
M1	Log-likelihood	-26.2075 ± 0.0000	**-26.0107 ± 0.0067**	-26.1726 ± 0.0013	-26.1723 ± 0.0002	-26.1724 ± 0.0001	-26.1802 ± 0.0055	-26.1844 ± 0.0007
	Variance MSE	0.0137 ± 0.0000	0.0081 ± 0.0002	0.0083 ± 0.0000	0.0081 ± 0.0003	0.0082 ± 0.0003	**0.0067 ± 0.0000**	0.0081 ± 0.0001
	MSE	**0.9963 ± 0.0000**	1.0143 ± 0.0002	1.0006 ± 0.0000	1.0006 ± 0.0007	1.0006 ± 0.0006	1.0026 ± 0.0001	1.0017 ± 0.0003
M2	Log-likelihood	**-26.5670 ± 0.0000**	-28.5916 ± 0.0085	-30.5040 ± 0.0039	-30.5249 ± 0.0059	-30.5258 ± 0.0069	-30.7045 ± 0.0080	-30.6088 ± 0.0055
	Variance MSE	**16.9430 ± 0.0000**	23.4497 ± 0.0921	61.2115 ± 0.1124	61.4751 ± 0.5851	61.4699 ± 0.5920	61.3169 ± 0.1897	58.3248 ± 0.2691
	MSE	**1.1700 ± 0.0000**	1.3552 ± 0.0003	1.8599 ± 0.0005	1.8637 ± 0.0054	1.8636 ± 0.0051	1.8992 ± 0.0012	1.8695 ± 0.0005
POS	Log-likelihood	**-3.7066 ± 0.0000**	-4.2401 ± 0.0059	-4.3838 ± 0.0003	-4.3836 ± 0.0017	-4.3835 ± 0.0016	-4.3857 ± 0.0002	-4.4075 ± 0.0013
	Accuracy	0.9706 ± 0.0000	**0.9708 ± 0.0001**	0.9589 ± 0.0000	0.9589 ± 0.0000	0.9589 ± 0.0000	0.9588 ± 0.0001	0.9581 ± 0.0000
	F1	**0.9707 ± 0.0000**	0.9706 ± 0.0000	0.9588 ± 0.0000	0.9589 ± 0.0000	0.9589 ± 0.0000	0.9588 ± 0.0000	0.9579 ± 0.0000
	ECE	0.0302 ± 0.0000	**0.0261 ± 0.0001**	0.0336 ± 0.0000	0.0335 ± 0.0001	0.0335 ± 0.0001	0.0415 ± 0.0000	0.0333 ± 0.0000
MNIST	Log-likelihood	**-0.0739 ± 0.0000**	-0.1133 ± 0.0008	-0.1352 ± 0.0001	-0.1298 ± 0.0001	-0.1333 ± 0.0003	-0.1313 ± 0.0002	-0.1370 ± 0.0004
	Accuracy	0.9786 ± 0.0000	**0.9825 ± 0.0003**	0.9592 ± 0.0001	0.9615 ± 0.0002	0.9614 ± 0.0001	0.9594 ± 0.0002	0.9583 ± 0.0000
	F1	0.9786 ± 0.0000	**0.9820 ± 0.0000**	0.9595 ± 0.0000	0.9616 ± 0.0000	0.9615 ± 0.0000	0.9592 ± 0.0000	0.9584 ± 0.0000
	ECE	**0.0218 ± 0.0000**	0.0326 ± 0.0004	0.0411 ± 0.0003	0.0395 ± 0.0002	0.0404 ± 0.0001	0.0276 ± 0.0003	0.0424 ± 0.0001

A.1.3 Variational attention

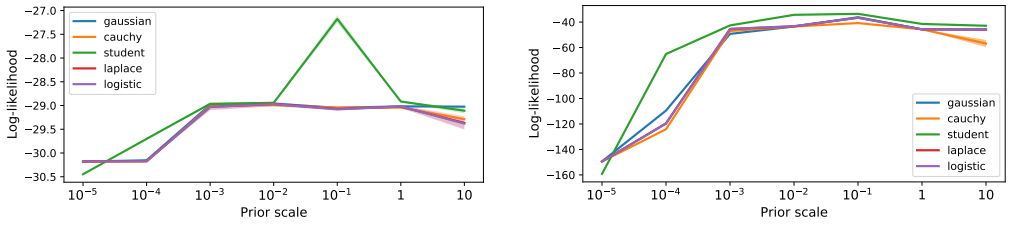
We report in Table 5 the full variational attention results table discussed in Section 4.2 including the data dependent configurations.

Table 5: Variational attention methods vs baselines

Dataset	Metric	MLE	Ensemble	Gauss. Attention	Gauss. DD Attention	Dir. Attention	Dir. DD Attention
M1	Log-likelihood	-26.208 ± 0.000	-26.011 ± 0.007	-26.1623 ± 0.0006	-26.1799 ± 0.0001	**-22.04 ± 0.01**	-25.242 ± 0.006
	Variance MSE	0.014 ± 0.000	**0.0081 ± 0.0002**	0.029 ± 0.000	0.045 ± 0.000	0.430 ± 0.002	0.1012 ± 0.0004
	MSE	**0.996 ± 0.000**	1.0143 ± 0.0002	1.007 ± 0.000	1.007 ± 0.000	1.0263 ± 0.0002	1.0417 ± 0.0003
M2	Log-likelihood	-26.567 ± 0.000	-28.592 ± 0.009	-26.374 ± 0.002	-25.3282 ± 0.0003	**-24.841 ± 0.007**	-26.263 ± 0.004
	Variance MSE	**16.943 ± 0.000**	23.45 ± 0.09	20.9010 ± 0.0007	18.528 ± 0.002	17.93 ± 0.03	20.17 ± 0.02
	MSE	1.17 ± 0.00	1.3552 ± 0.0003	1.2015 ± 0.0002	**1.089 ± 0.000**	1.1928 ± 0.0006	1.3018 ± 0.0001
POS	Log-likelihood	**-3.707 ± 0.000**	-4.240 ± 0.006	-3.9692 ± 0.0008	-4.0934 ± 0.0005	-3.9682 ± 0.0003	-3.859 ± 0.002
	Accuracy	0.9706 ± 0.0000	**0.9708 ± 0.0001**	0.969 ± 0.000	0.969 ± 0.000	0.968 ± 0.000	0.969 ± 0.000
	F1	**0.9707 ± 0.0000**	0.9706 ± 0.0000	0.969 ± 0.000	0.969 ± 0.000	0.968 ± 0.000	0.969 ± 0.000
	ECE	0.0302 ± 0.0000	**0.0261 ± 0.0001**	0.0271 ± 0.0000	0.0270 ± 0.0000	0.0287 ± 0.0000	0.0278 ± 0.0001
MNIST	Log-likelihood	-0.0739 ± 0.0000	-0.1133 ± 0.0008	**-0.0720 ± 0.0001**	-0.0838 ± 0.0001	-0.1045 ± 0.0005	-0.0955 ± 0.0009
	Accuracy	0.9786 ± 0.0000	**0.9825 ± 0.0003**	0.9790 ± 0.0002	0.9769 ± 0.0001	0.9738 ± 0.0003	0.9766 ± 0.0002
	F1	0.9786 ± 0.0000	**0.9820 ± 0.0000**	0.9786 ± 0.0000	0.9769 ± 0.0000	0.9736 ± 0.0000	0.9764 ± 0.0000
	ECE	**0.0218 ± 0.0000**	0.0326 ± 0.0004	0.0227 ± 0.0002	0.0252 ± 0.0001	0.0305 ± 0.0003	0.0281 ± 0.0000

A.1.4 Likelihood sensitivity to the prior

As discussed in Section 4.1, we find that the model test likelihood is very sensitive to the choice of prior (see Figure 3).

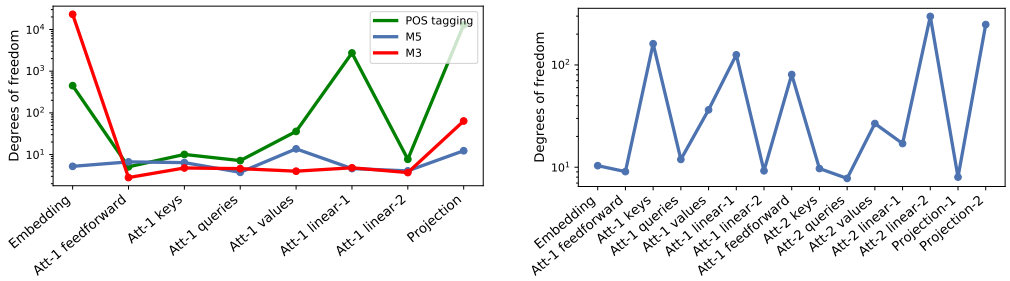


(a) M1 (b) M2

Figure 3: Log-likelihood sensitivity to the choice of prior scale

A.1.5 Weight distribution tailedness

We here present more detailed results regarding the tailedness of the weight distribution. We find that no obvious patterns in the thickness of tails exist across the considered dataset (Figure 4). Q-Q plots of the empirical weight distribution against light-tailed Gaussian and heavier tailed Laplace distribution provide more evidence of this phenomenon. Indeed, the weight distributions of transformers trained on MNIST and POS tagging are well fitted by a Gaussian, while transformers trained on M1 and M2 toy datasets are well fitted by a Laplace (Figure 5 and 6). This suggest that not one universal distribution fits the empirical distribution of the weights across all datasets.



(a) M1 (b) MNIST

Figure 4: Marginal weight distribution tailedness. No patterns in the thickness of the tails, except for a decrease at the last layer, appears across the considered tasks.

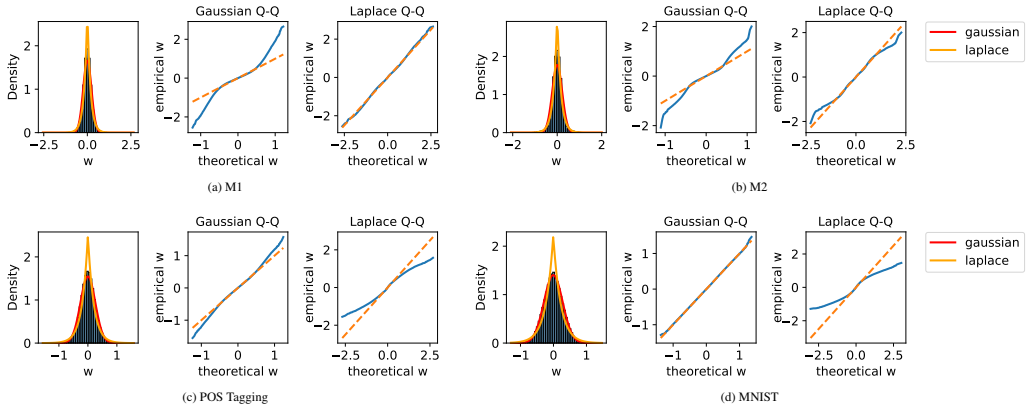


Figure 5: Attention queries matrix weight distribution Q-Q plot.

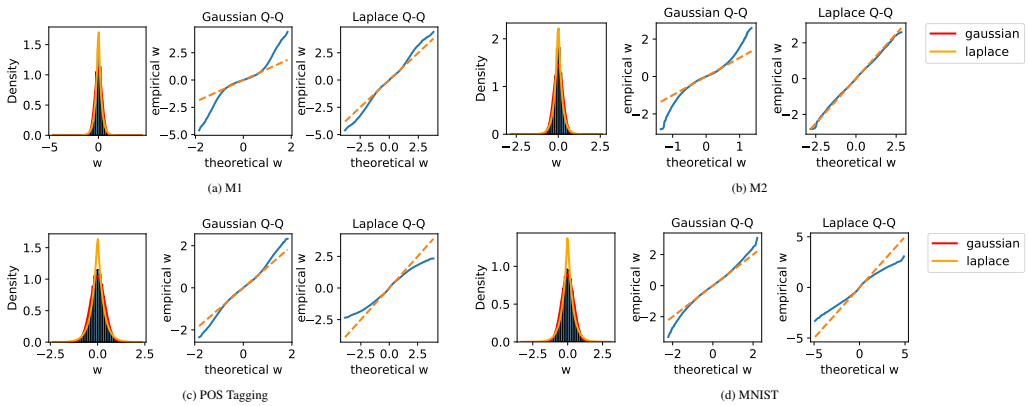
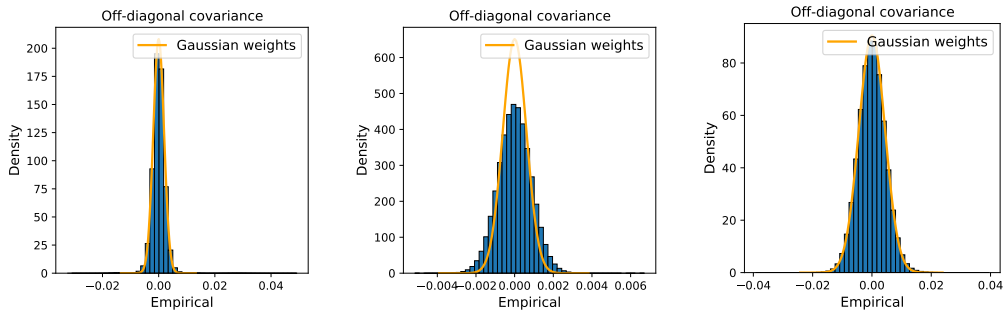


Figure 6: Attention MLP hidden layer empirical weight distribution Q-Q plot.

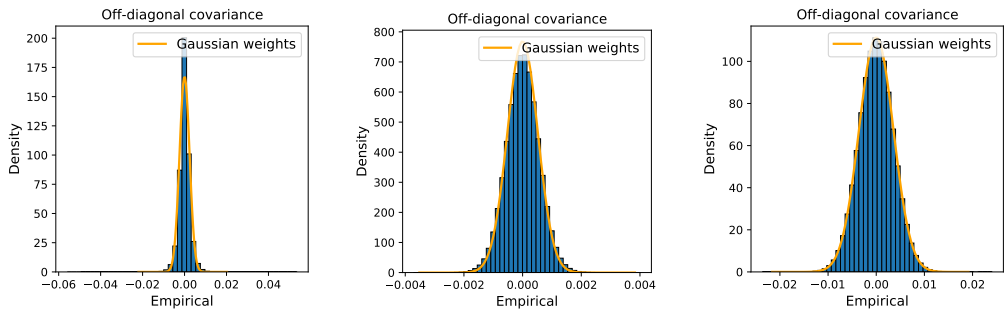
A.1.6 Off-diagonal covariance value distributions

We here present histograms of the off-diagonal empirical covariance elements. Covariance values have small magnitude and concentrate strongly in distribution around 0. Depending on the dataset, off-diagonal covariance elements are slightly larger than samples from an isotropic Gaussian as shown in Figure 7 and 10.



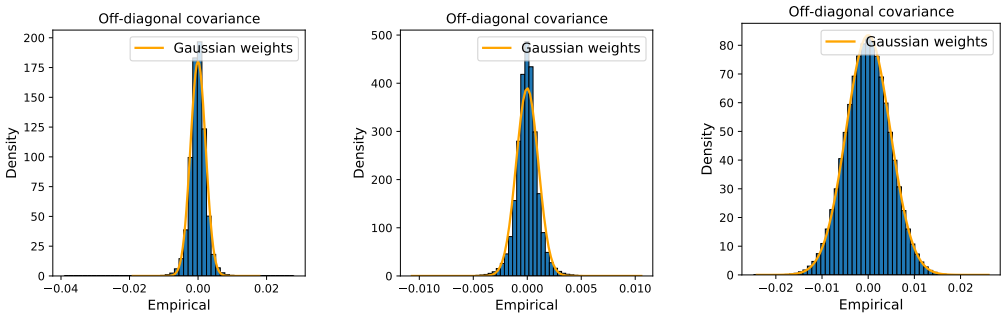
(a) Attention queries (b) Attention feedforward (c) Attention MLP hidden layer

Figure 7: M1 off-diagonal covariance matrix value histograms



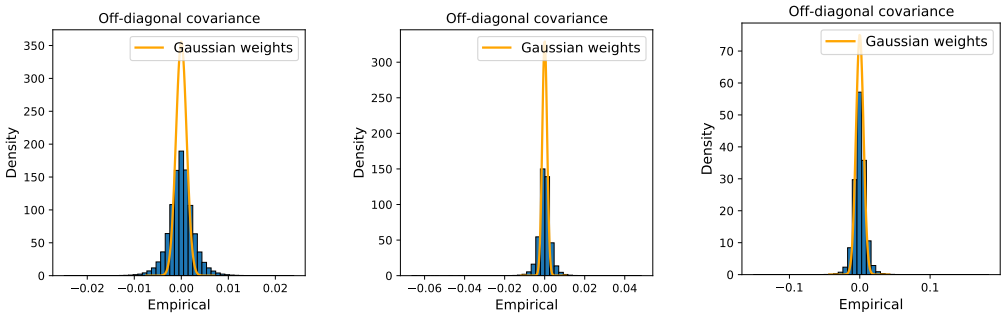
(a) Attention queries (b) Attention feedforward (c) Attention MLP hidden layer

Figure 8: M2 off-diagonal covariance matrix value histograms



(a) Attention queries (b) Attention feedforward (c) Attention MLP hidden layer

Figure 9: POS tagging off-diagonal covariance matrix value histograms



(a) Attention queries (b) Attention feedforward (c) Attention MLP hidden layer

Figure 10: MNIST off-diagonal covariance matrix value histograms

A.1.7 Concrete dropout high/low entropy samples

Here we present samples from the MNIST test set which have high (respectively low) predictive distribution entropy. By visual inspection, we find that high predictive entropy samples have an ambiguous labeling while low entropy predictive sample labels are much easier to identify.

1st 2nd 3rd 4th 1st 2nd 3rd 4th

(e) Highest entropy (j) Lowest entropy

Figure 11: Concrete Dropout : highest/lowest entropy MNIST test samples

A.2 Implementation details

In the following section, we present more detailed implementation specifications.

A.2.1 Architecture & Datasets

Toy Data We generate synthetic data from the two following models:

- Model M1: $X_{t+1} = \sum_{i=0}^{4} 0.2 \cos(0.4\pi i X_t + 1/(i+1)) + \sqrt{0.5}\epsilon_{t+1}$

- Model M2: $X_{t+1} = \sum_{i=0}^{4} \sum_{j=0}^{4} 0.5 \cos(0.8\pi j X_{t-i}) + \sqrt{0.1}\epsilon_{t+1}$

where $(X_i)_{i=0}^{4}$ and $(\epsilon_t)_{t=1}^{T}$ are i.i.d standard Gaussian random variables. We use 800 training, 80 validation and 80 testing sequences of length 24. For experiments with Toy Data, we use the transformer from [57] with GeLU activation [27], a unique attention block, one attention head and a hidden size of 64. We train until convergence for 100 epochs. We draw 30 posterior samples to compute the predictive distribution and KL divergence. We evaluate our model with the test data log-likelihood, mean squared error of the predicted variance and the expected mean square error given the previous value X_t.

Part of Speech Tagging We use the English split of the Part of Speech tagging dataset from the Universal Dependencies v1.2 corpus [46]. This dataset contains 204'586 train, 25'148 validation and 25'096 test tokens. We use a maximum length of 40 tokens, pad the shorter sequences and split the sentences which exceed the text length. For experiments with this dataset, we use a transformer from [57] with GeLU activation, a unique attention block, one attention head and a hidden size of 32. We train until convergence for 100 epochs. We draw 10 posterior samples to compute the predictive distribution and KL divergence. We evaluate our model with the token level test data log-likelihood, token level accuracy, token level F1-score and expected calibration error (ECE) [26].

MNIST We experiment with the MNIST image classification dataset [39]. We split the original dataset into 48'000 training, 12'000 validation and 9'984 testing samples. For experiments with this dataset, we use a Vision transformer from [15] with GeLU activation, two attention block, one attention head, a hidden size of 32 and a patch size of 4. We train until convergence for 150 epochs. We draw 10 posterior samples to compute the predictive distribution and KL divergence. We evaluate our model with the test data log-likelihood, accuracy, F1-score and ECE.

A.2.2 Training setup

Our models are trained using the Adam optimizer [34] with the triangular learning rate schedule from [57]. Our weight distribution experiments are conducted using 900 transformers trained by likelihood maximization with SGD as done in [21].

A.2.3 Software packages

We implement and train our models using the JAX [4] and Haiku [28] Python libraries.