Bayesian Attention Belief Networks

Shujian Zhang [*1] **Xinjie Fan** [*1] **Bo Chen** [2] **Mingyuan Zhou** [1]

Abstract

Attention-based neural networks have achieved state-of-the-art results on a wide range of tasks. Most such models use deterministic attention while stochastic attention is less explored due to the optimization difficulties or complicated model design. This paper introduces Bayesian attention belief networks, which construct a decoder network by modeling unnormalized attention weights with a hierarchy of gamma distributions, and an encoder network by stacking Weibull distributions with a deterministic-upward-stochastic-downward structure to approximate the posterior. The resulting auto-encoding networks can be optimized in a differentiable way with a variational lower bound. It is simple to convert any models with deterministic attention, including pretrained ones, to the proposed Bayesian attention belief networks. On a variety of language understanding tasks, we show that our method outperforms deterministic attention and state-of-the-art stochastic attention in accuracy, uncertainty estimation, generalization across domains, and robustness to adversarial attacks. We further demonstrate the general applicability of our method on neural machine translation and visual question answering, showing great potential of incorporating our method into various attention-related tasks.

1. Introduction

Attention-based architectures were originally proposed to induce useful inductive biases by aggregating features with learnable weights for sequence models (Sutskever et al., 2014; Bahdanau et al., 2015). Since the introduction of the attention-based Transformer (Vaswani et al., 2017), attention has become the foundation for many state-of-the-art models. Due to the computational efficiency and scalability of the Transformer structure, it becomes possible to train unprecedented large models on big datasets (Devlin et al., 2018), which stimulates a great amount of research to pretrain models on large unlabeled datasets. In an unsupervised manner, this approach learns useful representations that benefit downstream tasks, achieving tremendous success in natural language processing (Devlin et al., 2018; Lan et al., 2019; Liu et al., 2019; Joshi et al., 2020; Radford et al., 2018; Yang et al., 2019), compute vision (Dosovitskiy et al., 2020; Chen et al., 2020), and multi-modal tasks (Chen et al., 2019; Lu et al., 2019).

Most of the attention networks treat attention weights as deterministic rather than random variables, leading to the whole networks mostly composed of deterministic mappings. Such networks, although simple to optimize, are often incapable of modeling complex dependencies in data (Chung et al., 2015). By contrast, stochastic belief networks (Neal, 1992; Hinton et al., 2006; Gan et al., 2015; Zhou et al., 2016; Zhang et al., 2018; Fraccaro et al., 2016; Fan et al., 2021; Bayer & Osendorfer, 2014; Bowman et al., 2016), stacking stochastic neural network layers, have shown great advantages over deterministic networks in not only modeling highly structured data but also providing uncertainty estimation.

This paper proposes Bayesian attention belief networks (BABN), where we build deep stochastic networks by modeling unnormalized attention weights as random variables. First, we construct the generative (decoder) network with a hierarchy of gamma distributions. Second, the inference (encoder) network is a stack of Weibull distributions with a deterministic-upward and a stochastic-downward path. Third, we leverage the efficient structure of existing deterministic attention networks and use the keys and queries of current attention networks to parameterize the distributions of BABN. This efficient architecture design enables us to easily convert any existing deterministic attention networks, including pretrained ones, to BABN. Meanwhile, it imposes natural parameter and computational sharing within the networks, maintaining computation efficiency and preventing overfitting. Finally, we optimize both the decoder and encoder networks with an evidence lower bound. As the encoder network is composed of a reparameterizable distribution, $i.e.$, Weibull distribution, the training objective is differentiable. Further, leveraging the fact that the Kullback–

[*]Equal contribution [1]The University of Texas at Austin [2]Xidian University. Correspondence to: Mingyuan Zhou <mingyuan.zhou@mccombs.utexas.edu>.

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

Leibler (KL) divergence from the gamma to Weibull distribution is analytic, we can efficiently reduce the gradient estimation variance.

The proposed BABN has a generic architecture so that any existing deterministic attention models, including pretrained ones, can be converted to BABN while maintaining the inherent advantages of conventional attention, such as efficiency and being simple to optimize. Our proposed method is generally simple to implement and boosts the performance while only slightly increasing the memory and computational cost. On various natural language understanding tasks, neural machine translation, and visual question answering, our method outperforms vanilla deterministic attention and state-of-the-art stochastic attentions, in terms of accuracy and uncertainty estimation. We further demonstrate that BABN achieves strong performance in domain generalization and adversarial robustness.

2. Background on Attention Networks

Most attention structures can be unified with the key, query and value framework, where keys and queries are used to calculate attention weights and values are aggregated by the weights to obtain the final output. Formally, given n key-value pairs and m queries, we denote keys, values, and queries by $K \in \mathbb{R}^{n\times d_k}$, $V \in \mathbb{R}^{n\times d_v}$, and $Q \in \mathbb{R}^{m\times d_k}$. Note that the second dimension of K and Q are often equal because we usually need to compute scaled dot-product between key and query (Vaswani et al., 2017) as

$$\Phi = f_{\text{dot}}(Q, K) = QK^T/\sqrt{d_k} \in \mathbb{R}^{m\times n}.$$

To ensure that the attention weights are positive and sum up to one across keys, f_{dot} is often followed by a softmax function to obtain the final attention weights $W = \text{softmax}(f_{\text{dot}}(Q, K))$. In detail, first we obtain positive unnormalized weights S with the exponential function: $S = \exp(\Phi)$, then we normalize S across the key dimension with f_{norm} as

$$W_{i,j} = f_{\text{norm}}(S)_{i,j} := \frac{S_{i,j}}{\sum_{j'=1}^{n} S_{i,j'}},$$

for $i = 1, ..., m, j = 1, ..., n$. Finally, the output of attention is $O = WV \in \mathbb{R}^{m\times d_v}$, aggregating the values according to the attention weights.

This generic architecture can be used in many different models and applications. More interestingly, attention layers can be stacked on top of each other to build a deep neural network that is capable of modeling complicated deterministic functions. For example, in self-attention, denote the input of the lth attention layer by I^l, then we can obtain the key K^l, query Q^l, and value V^l by linearly projecting I^l to different spaces: $K^l = I^l M_K^l, Q^l = I^l M_Q^l, V^l = I^l M_V^l$, where M's are parametric matrices to learn. The output of this attention layer, O^l, can be fed as next layer's input $I^{l+1} = O^l$, and we can iterate the above process to obtain a deep self-attention-based neural network. Note that other structure details (Vaswani et al., 2017), such as residual structure (He et al., 2016), feed forward networks, and layer normalization (Ba et al., 2016), are also indispensable for the network but it would not affect the general framework we describe here.

3. BABN: Bayesian Attention Belief Networks

We introduce an efficient solution for deep attention belief networks: (a) build a hierarchical distribution to model unnormalized attention weights as the generative model, (b) develop an inference network with a deterministic-upward-stochastic-downward structure, and (c) leverage existing attention architectures and a few light-weight linear layers to parameterize the distributions. The resulting architecture can be efficiently learned with variational inference.

3.1. Deep Gamma Decoder Attention Networks

Denoting a supervised learning problem with training data $\mathcal{D} := \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$, the conditional probability for conventional attention-based model is $p_{\boldsymbol{\theta}}(\boldsymbol{y}_i \,|\, \boldsymbol{x}_i, W_i)$, where $W_i := f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$, $f_{\boldsymbol{\theta}}(\cdot)$ is a deterministic transformation, and $\boldsymbol{\theta}$ is the neural network parameter that includes the attention projections M's. For notational convenience, below we drop the data index i. Even though the deterministic attention mechanism is easy to implement and optimize, it often fails to capture complex dependencies or provide uncertainty estimation (Fan et al., 2020).

To remedy such issues, we construct deep stochastic attention networks by treating attention weights as latent variables. Instead of directly modeling the normalized attention weights $W = \{W^l\}_{l=1}^L$ on the simplex, we find it easier to model the unnormalized weights $S = \{S^l\}_{l=1}^L$ on the positive real line. We model the distribution of S with a product of gamma distributions:

$$p_{\boldsymbol{\eta}}(S \,|\, \boldsymbol{x}) = \textstyle\prod_{l=1}^L \text{Gamma}(S^l \,|\, \boldsymbol{\alpha}^l = f_{\boldsymbol{\eta}}^l(S^{1:l-1}, \boldsymbol{x}), \boldsymbol{\beta}),$$

where the shape parameter $\boldsymbol{\alpha}^l$ at the lth layer is the output of a neural network $f_{\boldsymbol{\eta}}^l$ parameterized by $\boldsymbol{\eta}$, and the rate parameter is a positive constant $\boldsymbol{\beta}$. The gamma distribution has been widely used for modeling positive real variables and is known to be capable of capturing sparsity and skewness. It is particularly attractive for modeling unnormalized attention weights because normalizing the gamma distributions with the same rate parameter leads to a Dirichlet distribution, which is commonly used for modeling variables on the simplex (Blei et al., 2003; Zhou et al., 2016; Deng et al., 2018; Fan et al., 2020). In this way, the whole generative

process can be expressed as:

$$S \sim p_{\boldsymbol{\eta}}(\cdot \,|\, \boldsymbol{x}), \;\; \boldsymbol{y} \sim p_{\boldsymbol{\theta}}(\cdot \,|\, \boldsymbol{x}, f_{\text{norm}}(S)).$$

Remark 1. *Bayesian inference via Gibbs sampling is available when $\{f^l_{\boldsymbol{\eta}}\}_{l=1}^L$ are simple linear projections and $p_{\boldsymbol{\theta}}$ is the Poison distribution (Zhou et al., 2016):*

$$\begin{aligned} &f^l_{\boldsymbol{\eta}}(S^{1:l-1}, \boldsymbol{x}) = W^l S^{l-1}, \text{ for } l = 1, ..., L, \\ &\boldsymbol{y} \sim \text{Poisson}(W^{L+1} S^L). \end{aligned} \tag{1}$$

We sketch the Gibbs sampler (see Zhou et al. (2016) for details) in Fig. 1, whose upward and downward structure motivates the design of our encoder (inference) network architecture which we will discuss in detail in Section 3.2.

Efficient and Expressive Structures for $\boldsymbol{\alpha}^l$. To be able to model complicated dependencies, we use neural networks to model the mapping $\{f^l_{\boldsymbol{\eta}}\}_{l=1}^L$ from $S^{1:l-1}$ and $\boldsymbol{x}$ to S^l. However, having separate neural networks for each $f^l_{\boldsymbol{\eta}}$ would lead to memory and computation redundancy as it does not exploit the hierarchical relationships among $\{f^l_{\boldsymbol{\eta}}\}_{l=1}^L$. Therefore, we leverage the current attention's efficient structure, and note that the key K^l at layer l is a function output of previous attention weights $S^{1:l-1}$ and input $\boldsymbol{x}$. This motivates us to make use of the key K^l at layer l to construct $f^l_{\boldsymbol{\eta}}$. In particular, we apply a two-layer MLP to transform key K^l to obtain $\boldsymbol{\alpha}^l$:

$$\boldsymbol{\alpha}^l = \text{softmax}(f^l_{\boldsymbol{\eta},2}(\text{ReLU}(f^l_{\boldsymbol{\eta},1}(K^l)))),$$

where $f^l_{\boldsymbol{\eta},1}, f^l_{\boldsymbol{\eta},2}$ are two linear layers connected by the nonlinear activation function, ReLU (Nair & Hinton, 2010). This architecture imposes natural parameter and computation sharing in a hierarchical way, which could not only improve efficiency but also prevent overfitting.

3.2. Deep Weibull Encoder Attention Networks

Due to the nonlinear structure of the decoder attention network, deriving the Gibbs sampler is not feasible and its scalability is also a concern. In this regard, we propose an encoder network to learn a variational distribution $q_{\boldsymbol{\phi}}$ to approximate the posterior distribution of unnormalized attention weights S.

We model the variational distribution $q_{\boldsymbol{\phi}}$ with a product of Weibull distributions:

$$q_{\boldsymbol{\phi}}(S \,|\, \boldsymbol{x}, \boldsymbol{y}) = \textstyle\prod_{l=1}^L \text{Weibull}(S^l \,|\, \boldsymbol{k}^l, \boldsymbol{\lambda}^l),$$

where $\boldsymbol{k}^l, \boldsymbol{\lambda}^l$ are the Weibull shape and scale parameters, respectively. The reason for choosing the Weibull distribution is threefold (Zhang et al., 2018): First, the Weibull is similar to gamma distribution, capable of modeling sparse, skewed, and positive distributions. Second, unlike the gamma distribution, the Weibull distribution has a simple reparameterization so that it is easier to optimize. That is, to sample $s \sim \text{Weibull}(k, \lambda)$ with probability density function (PDF) $p(s \,|\, k, \lambda) = \frac{k}{\lambda^k} s^{k-1} e^{-(s/\lambda)^k}$, it is equivalent to letting $S = g(\epsilon) := \lambda(-\log(1-\epsilon))^{1/k}$, $\epsilon \sim \text{Unif}(0, 1)$. Third, there exists an analytic KL divergence as $\text{KL}(\text{Weibull}(k, \lambda) || \text{Gamma}(\alpha, \beta)) = \frac{\gamma\alpha}{k} - \alpha \log \lambda + \log k + \beta\lambda\Gamma(1 + \frac{1}{k}) - \gamma - 1 - \alpha \log \beta + \log \Gamma(\alpha)$, where γ denotes the Euler–Mascheroni constant and Γ is the gamma function. This provides an efficient way to estimate the training objective which we will discuss in detail in Section 3.3.

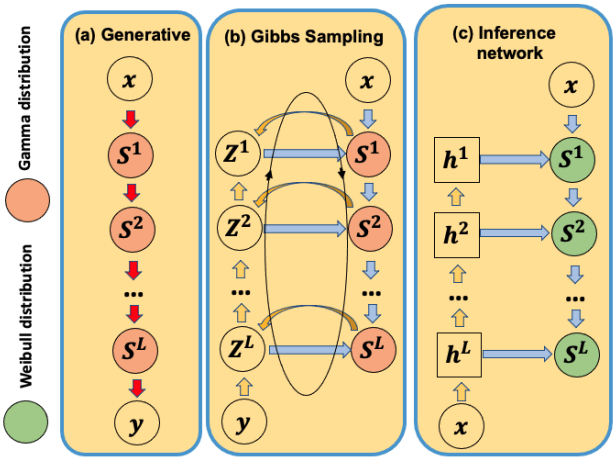

Figure 1. (a) The structure of the generative model that models unnormalized attention weights with a hierarchy of gamma distributions. (b) A sketch of an upward-downward Gibbs sampler mimicking that of the gamma belief network (Zhou et al., 2016), whose generative model is similarly structured as in (a). Z are augmented latent counts that facilitate the derivation of close-form Gibbs sampling update equations. (c) Motivated by the Gibbs sampler's structure, we design the inference network in a similar upward-downward way, where $\boldsymbol{h}$ represents a deterministic upward path and S represents a stochastic downward path. Note that our inference network is not conditioned on $\boldsymbol{y}$ as we are dealing with a supervised problem. Conditioned on $\boldsymbol{y}$ would prevent directly using the inference network for new data points.

Deterministic-upward and Stochastic-downward Structure. Inspired by the upward-downward Gibbs sampler sketched in Fig. 1, we mimic the structure to construct an inference network as:

$$\begin{aligned} \boldsymbol{k}^l &= f^l_{\boldsymbol{k},\boldsymbol{h}}(\boldsymbol{h}^l) + f^l_{\boldsymbol{k},S}(S^{1:l-1}, \boldsymbol{x}), \\ \boldsymbol{\lambda}^l &= f^l_{\boldsymbol{\lambda},\boldsymbol{h}}(\boldsymbol{h}^l) + f^l_{\boldsymbol{\lambda},S}(S^{1:l-1}, \boldsymbol{x}), \\ \boldsymbol{h}^l &= f^l_{\boldsymbol{h}}(\boldsymbol{h}^{l+1}), \end{aligned}$$

where $\{\boldsymbol{h}^l\}_{l=1}^{L+1}$ serve as the augmented latent variables passing the information from data upwards and complement the downward information from attention variables S. A similar bottom-up and top-down structure was proposed in the

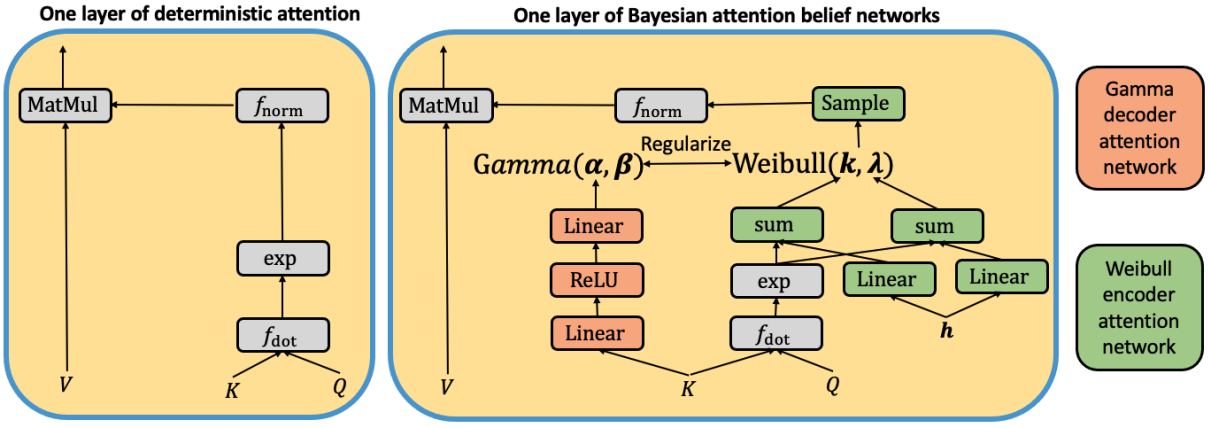

Figure 2. Illustration of the difference and similarity between the vanilla deterministic attention and one layer of our Bayesian attention belief networks. Bayesian attention belief networks (BABN) share the same architecture as the deterministic attention before obtaining key, query, and value. Then BABN adds light-weight linear layers to construct the gamma and Weibull distributions to model unnormalized attention weights, which are used after normalization to obtain the layer output as in the vanilla deterministic attention.

ladder VAE (Sønderby et al., 2016) and was found to help the optimization. In our experiments (section 5.3), we also found that the upward and downward structure plays an important role as the downward path delivers the prior information and the upward path delivers the likelihood information. Without the upward path of $\boldsymbol{h}$, the model often has unstable performances. We note that although $q_{\boldsymbol{\phi}}$ is independent of $\boldsymbol{y}$ during testing, it is possible for $q_{\boldsymbol{\phi}}$ to depend on part of $\boldsymbol{y}$ that has already been observed by the model during training in sequence generation tasks, such as neural machine translation, where the queries come from $\boldsymbol{y}$. Further, we think it is possible for $q_{\boldsymbol{\phi}}$ to approximate $p(S|\boldsymbol{x}, \boldsymbol{y})$ even without conditioning on $\boldsymbol{y}$ as $\boldsymbol{x}$ conveys information of $\boldsymbol{y}$. Formally, we define $f^l_{\boldsymbol{k},\boldsymbol{h}}, f^l_{\boldsymbol{k},S}, f^l_{\boldsymbol{\lambda},\boldsymbol{h}}, f^l_{\boldsymbol{\lambda},S}, f^l_{\boldsymbol{h}}$ as follows:

$$\begin{aligned}
\boldsymbol{k}^l &= \rho * \ln\left[1 + \exp\left(f^l_{\boldsymbol{\phi},1}(\boldsymbol{h}^l)\right)\right] + \exp(\Phi^l),\\
\boldsymbol{\lambda}^l &= \sigma * \ln\left[1 + \exp\left(f^l_{\boldsymbol{\phi},2}(\boldsymbol{h}^l)\right)\right] + \frac{\exp(\Phi^l)}{\Gamma(1+1/\boldsymbol{k}^l)},\\
\boldsymbol{h}^l &= \ln\left[1 + \exp\left(f^l_{\boldsymbol{\phi},3}(\boldsymbol{h}^{l+1})\right)\right],
\end{aligned}$$

where $f^l_{\boldsymbol{\phi},1}, f^l_{\boldsymbol{\phi},2}$, and $f^l_{\boldsymbol{\phi},3}$ are linear layers that preserve the dimension of $\boldsymbol{h}^l$, and $\boldsymbol{h}^{L+1}$ is initialized as a function of $\boldsymbol{x}$: $\boldsymbol{h}^{L+1} = f_{\boldsymbol{\phi},0}(\boldsymbol{x})$. The structure involves the following parts. 1) For $\boldsymbol{k}^l, \boldsymbol{\lambda}^l$, we introduce weights ρ, σ to balance the importance of the two parts in $\boldsymbol{k}^l, \boldsymbol{\lambda}^l$. 2) We leverage the efficient deterministic attention architecture to construct the functions $f^l_{\boldsymbol{k},S}$ and $f^l_{\boldsymbol{\lambda},S}$, where $\Phi^l = f(Q^l, K^l)$ is the function of $S^{1:l-1}$ and $\boldsymbol{x}$. Using Φ^l to construct the inference network is an efficient way to introduce parameter and computation sharing between the layers of the encoder and decoder. 3) For $\boldsymbol{\lambda}^l$, we rescale $\exp(\Phi^l)$ with $\Gamma(1+1/\boldsymbol{k}^l)$ so that the expectation of the Weibull distribution is $\exp(\Phi^l)$ when $\sigma = 0$, which corresponds to the deterministic attention before normalization. 4) In addition, we model the functions $f^l_{\boldsymbol{k},\boldsymbol{h}}, f^l_{\boldsymbol{\lambda},\boldsymbol{h}}, f^l_{\boldsymbol{h}}$ with linear layers coupled with $\ln[1 + \exp(\cdot)]$ to obtain positive outputs. We need to point out that both Φ^l and $\boldsymbol{h}^l$ are functions of only $\boldsymbol{x}$ but not $\boldsymbol{y}$, which enables us to directly use the variational distribution $q_{\boldsymbol{\phi}}$ during testing for new data points (Wang & Zhou, 2020; Fan et al., 2021). 5) We leverage the key and query of the *first* attention layer to initialize hidden states $\boldsymbol{h}^{L+1}$. In particular, we let $\boldsymbol{h}^{L+1} = \text{softmax}(\Phi^1)$. As there is yet no randomness introduced to Φ^1, this mapping from $\boldsymbol{x}$ to $\boldsymbol{h}^{L+1}$ is still deterministic. By sharing the parameter and computation with the main network, $f_{\boldsymbol{\phi},0}$ does not add any memory or computation cost.

Remark 2. *As our model leverages the efficient structure of the existing deterministic attention module and uses keys and queries to construct the prior and variational distribution for unnormalized attention weights, it is simple to convert existing deterministic attention networks to BABN. Fig. 2 shows that BABN shares parts of architecture with the deterministic attention. BABN adds a few light-weight linear layers to construct the gamma prior and Weibull variational distribution with the upward-downward structure. More importantly, we note that we can use pretrained deterministic attention model checkpoints to initialize BABN, and then finetune the stochastic neural network.*

Remark 3. *BABN can be easily extended to multi-head attention, where queries, keys, and values are projected H times linearly with H different learned projections, and the outputs of H heads are concatenated as the final output. Since the unnormalized multi-head attention weights are conditionally independent, we can still model the unnormalized attention weights with the same hierarchical formulation. Specifically, for each layer, conditioned on previous layers, we obtain the queries, keys for multiple heads to construct the distributions for unnormalized attention*

weights of each head separately. Then, we normalize the attention weights for each head so that within each head, the attention weights sum to one across keys, which is the same as the vanilla multi-head attention model.

3.3. Learning Bayesian Attention Belief Networks

Now, we have defined the gamma decoder network and Weibull encoder network. We learn the encoder network $q_{\boldsymbol{\phi}}$ to approximate the posterior distribution $p(S \mid \boldsymbol{x}, \boldsymbol{y})$ by minimizing the KL divergence, $\mathcal{L}_{\text{KL}} = \text{KL}(q_{\boldsymbol{\phi}}(S)||p(S \mid \boldsymbol{x}, \boldsymbol{y}))$, which is equivalent to maximizing,

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) := \mathbb{E}_{q_{\boldsymbol{\phi}}(S)} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}, S)\right] - \text{KL}(q_{\boldsymbol{\phi}}(S)||p_{\boldsymbol{\eta}}(S)),$$

an evidence lower bound (ELBO) (Hoffman et al., 2013; Blei et al., 2017; Kingma & Welling, 2013) of the intractable log marginal likelihood $\log p(\boldsymbol{y} \mid \boldsymbol{x}) = \log \int p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}, S) p_{\boldsymbol{\eta}}(S) dS$. The objective $\mathcal{L}$ consists of two parts: the likelihood part, which maximizes the data likelihood under the encoder network; the regularization part, which enforces the variational distribution to be close to the prior distribution. We also use the same objective $\mathcal{L}$ to learn the decoder networks $p_{\boldsymbol{\eta}}$ and $p_{\boldsymbol{\theta}}$, as the exact marginal likelihood is intractable, and the ELBO is a good approximation when the variational distribution well approximates the true posterior (Kingma & Welling, 2013).

Note that as $q_{\boldsymbol{\phi}}$ is a product of Weibull distributions, it is reparameterizable. In particular, to sample S from $q_{\boldsymbol{\phi}}$, we sequentially sample S^l conditional on previous samples $S^{1:l-1}$, as $S^l \sim \text{Weibull}(S^l \mid \boldsymbol{k}^l, \boldsymbol{\lambda}^l)$. This can be realized by letting $S^l = g^l_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^l) := \boldsymbol{\lambda}^l(-\log(1-\boldsymbol{\epsilon}^l))^{1/\boldsymbol{k}^l}$, where $\boldsymbol{\epsilon}^l$ is a tensor with the same shape as S^l and its elements are $i.i.d$ samples from the uniform distribution. In practice, we found that drawing $\boldsymbol{\epsilon}^l$ from Uniform $(0, 1)$ leads to numerical issues. Therefore, to prevent numeral instability, we choose to draw $\boldsymbol{\epsilon}^l$ from Uniform $(0.1, 0.9)$ as an approximation. Further, we note that at each layer l, the KL between the conditional distribution of encoder and decoder, $\text{KL}(q_{\boldsymbol{\phi}}(S^l \mid S^{1:l-1})||p_{\boldsymbol{\eta}}(S^l \mid S^{1:l-1}))$, is analytical. Therefore, we follow the same way in Fan et al. (2020) to efficiently compute $\text{KL}(q_{\boldsymbol{\phi}}(S)||p_{\boldsymbol{\eta}}(S)$ by decomposing it as $\sum_{l=1}^{L} \mathbb{E}_{q_{\boldsymbol{\phi}}(S^{1:l-1})} \underbrace{\text{KL}(q_{\boldsymbol{\phi}}(S^l|S^{1:l-1})||p_{\boldsymbol{\eta}}(S^l|S^{1:l-1}))}_{\text{analytic}}$, where the integrand is analytic. Putting it all together, we can rewrite the ELBO objective as $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{\boldsymbol{\epsilon}}[\mathcal{L}_{\boldsymbol{\epsilon}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\epsilon})]$, where

$$\mathcal{L}_{\boldsymbol{\epsilon}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\epsilon}) = \log p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}, g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})) - \sum_{l=1}^{L} \underbrace{\text{KL}(q_{\boldsymbol{\phi}}(S^l \mid g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{1:l-1}))||p_{\boldsymbol{\eta}}(S^l \mid g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{1:l-1})))}_{\textit{analytic}}.$$

With the reparameterization, now we can efficiently estimate the gradient of $\mathcal{L}$ with respect to $\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\eta}$ by computing the gradient of $\mathcal{L}_{\boldsymbol{\epsilon}}$ with one sample of $\boldsymbol{\epsilon}$. Both reparameterization and semi-analytic KL (Owen, 2013) reduce the Monte Carlo estimation variance and still keep the estimation unbiased. Finally, following previous work (Bowman et al., 2016), we add a weight λ to the KL term and anneal it from a small value to one.

4. Related Work

Stochastic attentions: Xu et al. (2015), along with several following work (Shankar & Sarawagi, 2018; Deng et al., 2018), proposed hard attention to model attention weights with categorical distributions, which only attends to one subject at a time. The categorical distribution, however, is not reparameterizable and therefore hinders the use of standard backpropagation. REINFORCE gradient estimator makes the optimization possible, but it has high variance and one often needs to carefully design baselines to make the performance comparable to deterministic attention (Xu et al., 2015; Deng et al., 2018). Stochastic soft attention, on the other hand, is less investigated. Deng et al. (2018) proposed modeling attention weights with the Dirichlet distribution, which is not reparameterizable and introduces optimization difficulties. Fan et al. (2020) considered using reparameterizable distributions, such as Lognormal and Weibull distributions, to model unnormalized attention weights, which alleviates the optimization issue of previous stochastic attention. Compared to Fan et al. (2020) who try to convert deterministic attention modules to stochastic ones, our method is motivated from building a deep stochastic network by modeling attention weights as random variables. With a deterministic-upward and stochastic-downward structure, our inference network comprises Weibull distributions, whose scale parameter $\boldsymbol{\lambda}$ and shape parameter $\boldsymbol{k}$ are both sample-dependent. This makes it differ from Fan et al. (2020), where the shape parameter $\boldsymbol{k}$, controlling the uncertainty of distribution, is a hyperparameter and the inference network does not involve a deterministic upward path. The proposed generalization gives us greater flexibility in modeling attention weights. We also conduct more extensive experiments to investigate the domain generalization ability and adversarial robustness of stochastic attentions.

Deep stochastic networks: Augmenting deterministic neural networks with random variables provides us a principled way to capture the randomness in data and estimate uncertainty (Gal & Ghahramani, 2016; Chung et al., 2015; Bowman et al., 2016; Tran et al., 2018). More importantly, stacking stochastic layers into a deep stochastic network instead of a shallow probabilistic model is often preferable due to its capability to model more complicated dependencies (Zhang et al., 2018). For example, Zhang et al. (2018) have applied a gamma belief network for topic modeling, and a deep Weibull network is used to approximate the pos-

terior for scalable inference. We apply a similar structure to the widely used attention models and leverage the existing efficient attention architecture to build scalable networks.

5. Experimental Results

Our method can be straightforwardly deployed wherever the regular attention is utilized. To test its effectiveness and general applicability, we apply our method to a diverse set of tasks, including language understanding, neural machine translation, and visual question answering. For language understanding, we further study a model's generalization across domains and robustness towards adversarial attacks. Meanwhile, we experiment with a diverse set of state-of-the-art models, including, ALBERT (Lan et al., 2019), BERT (Devlin et al., 2018), and RoBERTa (Liu et al., 2019). In the following, we provide the main experimental settings and results, with more details provided in Appendix A.

5.1. Attention in Natural Language Understanding

The self-attention-based Transformer models have become the de-facto standard for NLP tasks. The dominant approach is to first pretrain models on big corpora to learn generic features and then finetune the models on the corresponding datasets for downstream tasks. This approach has constantly been refreshing the state-of-the-art results on various tasks. However, the cost of training such models from scratch is often prohibitive for researchers with limited resources and it also brings burdens to our environment (Strubell et al., 2019). For example, it takes 79 hours to train a BERT-base model on 64 V100 GPUs, which costs about $\$3,751$-$\$12,571$ cloud computations and brings CO_2 emissions of 1438 lbs (Strubell et al., 2019). Considering this, we believe that starting from pretrained models is not only efficient and environmental friendly, but also makes it accessible for researchers with limited computations. As discussed in Remark 2, we can convert a pretrained deterministic attention model to BABN and then finetune it on downstream tasks. Therefore, in this section, we investigate the effectiveness of only applying BABN during the finetuning stage.

5.1.1. In-domain Performance Evaluation

First, we consider the standard setting, $i.e.$, evaluating in-domain accuracies, where both the training and testing data are from the same domain.

Experimental Settings. We include 8 datasets from General Language Understanding Evaluation (GLUE) (Wang et al., 2018) and two versions of Stanford Question Answering Datasets (SQuAD) (Rajpurkar et al., 2016; 2018) as the benchmarks. We build our method on a state-of-the-art model, ALBERT (Lan et al., 2019), which is a memory-efficient version of BERT (Devlin et al., 2018) with parameter sharing and embedding factorization. We leverage the pretrained checkpoint as well as the codebase for finetuing provided by Huggingface PyTorch Transformer (Wolf et al., 2019). We use the base version of ALBERT (Lan et al., 2019). During testing, we obtain point estimates by approximating the posterior means of prediction probabilities by substituting the latent unnormalized attention weights by their posterior expectations (Srivastava et al., 2014).

Results. In Table 1, we compare BABN with the deterministic attention and BAM (Fan et al., 2020), which is the state-of-the-art stochastic attention. BAM is also applied during the finetuning stage, resuming from the same checkpoint. We report the mean accuracies and standard deviations for 5 independent runs. Table 1 shows that BABN outperforms both deterministic attention and BAM, which indicates that stochastic belief networks give better performance than deterministic ones and the more flexible structure of BABN is also preferable to the structure of BAM. We consistently observe clear improvements even though we only apply BABN at the finetuning stage.[1] We leave as future work using BABN at the pretrain stage.

5.1.2. Generalization across Domains

In real applications, it is very likely to apply a deep learning model to the data from a new domain unseen in the training dataset. Therefore, it is important to evaluate a model's generalization ability across domains. In NLP, significant work has studied domain generalization on sentiment analysis (Chen et al., 2018; Peng et al., 2018; Miller, 2019). Recently, Desai & Durrett (2020) studied the cross-domain generalization of pretrained Transformer models on more difficult tasks and found it still challenging for these pretrained models to generalize. In this section, we follow the setting of Desai & Durrett (2020) to study the generalization ability of our method.

Experimental Settings. Following Desai & Durrett (2020), we test domain generalization on three challenging tasks, including natural language inference (NLI), paraphrase detection (PD), and commonsense reasoning (CR). Each task includes both a source domain, used for finetuning the model, and a target domain, used for evaluating the model. Specifically, SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) are the source and target domains for NLI, respectively; QQP and TwitterPPDB (Lan et al., 2017) are the source and target domains for PD, respectively; SWAG (Zellers et al., 2018) and HSWAG (Zellers et al., 2019) are the source and target domains for CR, respectively. These

[1] We provide the parameter sizes and step time for different attention types combined with ALBERT-base, a Transformer-based model, where the attention module constructs the main model in Table 7 in the Appendix.

Table 1. Results of the in-domain accuracies for different models on GLUE and SQuAD benchmarks.

MODEL	MRPC	COLA	RTE	MNLI	QNLI	QQP	SST-2	STS	SQUAD 1.1	SQUAD 2.0
ALBERT-BASE	86.5	54.5	75.8	85.1	90.9	90.8	92.4	90.3	80.86/88.70	78.80/82.07
ALBERT-BASE+BAM	88.5	55.8	76.2	85.6	91.5	90.7	92.7	91.1	81.40/88.82	78.97/82.23
ALBERT-BASE+**BABN**	**89.2**±0.3	**56.8**±0.5	**77.6**±0.6	**86.2**±0.3	**91.9**±0.3	**91.2** ±0.1	**93.1**±0.2	**91.8**±0.2	**81.81**±0.1/**89.10**±0.1	**79.20**±0.1 / **82.41**±0.1

Table 2. Results of domain generalization. We report the accuracy and ECE of various models on both in-domain data and out-of-domain data for three tasks: natural language inference, paraphrase detection, and commonsense reasoning.

	ACCURACY ↑		ECE ↓	
	ID	OD	ID	OD
NATURAL LANGUAGE INFERENCE	SNLI	MNLI	SNLI	MNLI
DA (PARIKH ET AL., 2016)	84.63	57.12	**1.02**	8.79
ESIM (CHEN ET AL., 2017)	88.32	60.91	1.33	12.78
BERT-BASE (DESAI & DURRETT, 2020)	90.04	73.52	2.54	7.03
BERT-BASE+BAM	90.25	73.81	2.37	6.40
BERT-BASE+**BABN**	**90.63**	**74.32**	1.98	**5.09**
ROBERTA-BASE	91.23	78.79	**1.93**	3.62
ROBERTA-BASE+BAM	91.29	79.11	2.85	2.94
ROBERTA-BASE+**BABN**	**91.70**	**79.86**	2.62	**2.67**
PARAPHRASE DETECTION	QQP	TWITTER	QQP	TWITTER
DA (PARIKH ET AL., 2016)	85.85	83.36	3.37	9.79
ESIM (CHEN ET AL., 2017)	87.75	84.00	3.65	8.38
BERT-BASE (DESAI & DURRETT, 2020)	90.27	87.63	2.71	8.51
BERT-BASE+BAM	90.77	87.14	2.91	9.21
BERT-BASE+**BABN**	**90.84**	**88.32**	**1.42**	**7.43**
ROBERTA-BASE (DESAI & DURRETT, 2020)	91.11	86.72	2.33	9.55
ROBERTA-BASE+BAM	91.24	86.87	2.01	9.50
ROBERTA-BASE+**BABN**	**91.72**	**87.31**	**1.74**	**9.42**
COMMONSENSE REASONING	SWAG	HSWAG	SWAG	HSWAG
DA (PARIKH ET AL., 2016)	46.80	32.48	5.98	40.37
ESIM (CHEN ET AL., 2017)	52.09	32.08	7.01	19.57
BERT-BASE (DESAI & DURRETT, 2020)	79.40	34.48	2.49	12.62
BERT-BASE+BAM	79.44	35.18	2.38	12.49
BERT-BASE+**BABN**	**79.57**	**36.23**	**1.91**	**11.82**
ROBERTA-BASE (DESAI & DURRETT, 2020)	82.45	41.68	1.76	11.93
ROBERTA-BASE+BAM	82.61	42.04	1.66	11.21
ROBERTA-BASE+**BABN**	**83.12**	**43.11**	**1.32**	**9.72**

benchmarks are known to exhibit challenging domain shifts (Desai & Durrett, 2020). For each experiment, we report both the in-domain (ID) accuracy on the source domain and out-of-domain (OD) accuracy on the target domain. As in Desai & Durrett (2020), we also report the expected calibration error (ECE) as a measure of model calibration. To compute ECE, we need to divide the samples into groups with their confidences, defined as the probability of the maximum predicted class. Then, ECE:= $\sum_i \frac{B_i}{N} |\text{acc}(B_i) - \text{conf}(B_i)|$, where B_i, $\text{acc}(B_i)$, and $\text{conf}(B_i)$ are the count, accuracy, and confidence of samples in the ith group, respectively. We set the number of groups to 10 as in Desai & Durrett (2020).

Results. We summarize our results in Table 2. Our baselines include two small-scale and non-pretrained models: Decomposable Attention (DA) (Parikh et al., 2016) and Enhanced Sequential Inference Model (ESIM) (Chen et al., 2017), and two state-of-the-art large-scale and pretrained models with deterministic attention: BERT-base (Devlin et al., 2018) and RoBERTa-base models (Liu et al., 2019). We experiment with adding BABN to both BERT-base and RoBERTa-base models. Table 2 shows that adding BABN consistently improves upon the corresponding deterministic models on not only in-domain, which confirms our results in Section 5.1.1, but also out-of-domain. The performance gains on out-of-domain are often greater than the gains on in-domain, meaning that BABN can significantly help the model to generalize across domains. This gets along with our intuition that deep stochastic models should generalize better than deterministic ones. Further, we note that BABN also improves ECE, meaning that BABN helps to obtain better-calibrated models for uncertainty estimation.

5.1.3. ROBUSTNESS TOWARDS ADVERSARIAL ATTACKS

Neural networks are known to be vulnerable to adversarial examples that have imperceptible perturbations from the original counterparts (Goodfellow et al., 2014). It has been found that even large language models pretrained on large corpora still suffer from the same issue (Jin et al., 2020). Therefore, it is important to evaluate and improve a model's robustness against adversarial attacks. We argue that as our Bayesian attention belief networks are built by stacking probabilistic layers, the stochastic connections would make the model more robust so that it is more difficult to generate perturbations that would fool our model.

Experimental Settings. To compare the adversarial robustness of BABN and the deterministic attention, we first finetune the ALBERT-base models according to the same settings as in Section 5.1.1, and then apply three state-of-the-art untargeted black-box adversarial attacks, including (1) Textfooler (Jin et al., 2020), generating natural looking attacks with rule-based synonym replacement; (2) Textbugger (Li et al., 2019), generating misspelled words by character- and word-level perturbations; (3) BAE (Garg & Ramakrishnan, 2020), generating BERT-based adversarial examples. We implement all the attacks using the NLP attack package, TextAttack (Morris et al., 2020), with the default settings. For each model, we conduct 1000 adversarial attacks and Table 3 reports the percentages of failed adversarial attacks. Higher percentages indicate more robust models.

Results. Table 3 shows that BABN outperforms the deterministic attention baseline on most datasets, and achieves a much better average accuracy. The improvement is consistent across all three different adversarial attacks with different levels of failure rates, with Textfooler being the

Table 3. Results of pretrained large-scale models' robustness against adversarial attacks. For each model, we report the percentages of failed attacks under three adversarial attacks respectively.

ATTACK	ATTENTION	MRPC	COLA	RTE	QQP	SST-2	AVG.
TEXTFOOLER	BASE	**6.5**	2.6	16.2	25.4	7.0	11.5
	BAM	6.2	3.1	**17.8**	28.7	12.5	12.5
	BABN	6.2	**5.1**	17.7	**33.7**	**16.4**	**15.8**
TEXTBUGGER	BASE	**10.6**	16.8	19.9	30.1	40.1	23.5
	BAM	9.9	16.7	21.0	32.5	51.7	26.4
	BABN	9.5	**17.6**	**21.4**	**35.8**	**55.5**	**28.0**
BAE	BASE	44.8	4.9	35.6	**48.8**	13.9	29.6
	BAM	48.6	5.1	**36.3**	42.2	22.8	31.0
	BABN	**50.4**	**7.1**	35.9	42.8	**25.7**	**32.4**

strongest attacker. These results verify our conjecture that by stacking stochastic layers, our Bayesian attention belief networks are more robust than deterministic models due to the stochastic connections. To the best of our knowledge, it is the first time to show that stochastic attention could improve adversarial robustness on large language models.

5.2. Attention in Neural Machine Translation

To show that BABN is generally applicable, we conduct experiments on the task of neural machine translation and compare BABN with SOTA stochastic attentions, including variational attention (VA) based methods (Deng et al., 2018) and BAM (Fan et al., 2020).

Experimental Settings. For fair comparisons, we adapt the deterministic attention model used by Deng et al. (2018) to BABN. The model is very different from the previous models, as it is LSTM-based, where attention is used to connect the encoder and decoder of the translation system (Deng et al., 2018). We follow the experimental settings of Deng et al. (2018). Models are trained from scratch. IWSLT (Cettolo et al., 2014) is used as benchmark. We adopt the widely used BLEU score (Papineni et al., 2002) as the evaluation metric for the translation results. Experimental details are summarized in Appendix A.

Table 4. Results of BLEU scores, parameter size and step time for different attentions on IWSLT.

ATTENTION	BLEU ↑	PARAMS ↓	S/STEP ↓
BASE	32.77	42M	0.08
VA + ENUM (DENG ET AL., 2018)	33.68	64M	0.12
VA + SAMPLE (DENG ET AL., 2018)	33.30	64M	0.15
BAM (FAN ET AL., 2020)	33.81±0.02	42M	0.10
BABN	**34.23**±0.05	42M	0.11

Results. In Table 4, we report the BLEU scores, model parameter sizes, and step time (second/step) for each attention type. It shows that BABN gives the best BLEU score outperforming deterministic attention (base), variational attention (VA), and BAM, while keeping the parameter size at the same level as deterministic attention. The runtime of BABN is on a par with BAM and slightly slower than deterministic attention, but it outruns the variational attention methods.

Table 5. Accuracies and PAvPUs of different attentions on both the original VQA-v2 dataset and the noise ones.

	ACCURACY ↑		PAVPU ↑	
	ORIGINAL	NOISY	ORIGINAL	NOISY
BASE	66.74	63.58	71.96	68.29
BAM	66.82	63.98	72.01	68.58
BABN	**66.92**±0.02	**64.40**±0.03	**72.21**±0.03	**70.43**±0.04

5.3. Attention in Visual Question Answering

We also conduct experiments on a multi-modal learning task, visual question answering (VQA) (Goyal et al., 2017), where the model learns to predict the answer to a given question on a given image. Transformer-like attention architectures have been widely used to learn the multi-modal reasoning between image and language (Yu et al., 2019). We adapt the recently proposed MCAN model (Yu et al., 2019) to BABN and compare with deterministic attention and BAM (Fan et al., 2020).

Experimental Settings. We mainly follow the setting by Yu et al. (2019), and experiment on the VQA-v2 dataset (Goyal et al., 2017). As in Fan et al. (2020), we also include a noisy dataset by perturbing the input with Gaussian noise to the image features (Larochelle et al., 2007) to investigate the model's robustness. We use 4-layer encoder-decoder based MCAN as the baseline model, where the deterministic attention was originally used. We report accuracies as well as uncertainty estimations, which are measured by a hypothesis testing based Patch Accuracy vs Patch Uncertainty (PAvPU) (Fan et al., 2020; Mukhoti & Gal, 2018), reflecting whether the model is uncertain about its mistakes. The higher the PAvPU is, the better the uncertainty estimation is. We set the p-value threshold to be 0.05 (Fan et al., 2020). For uncertainty estimation, we sample 20 unnormalized attention weights from the variational distribution. We provide more detailed experimental settings in Appendix A.

Results. In Table 5, we report the accuracy and PAvPU of different attentions on both original and noisy data. It shows that BABN consistently improve upon the deterministic attention and BAM in terms of both accuracy and PAvPU, meaning that BABN in general is more uncertain on its mistakes and more certain on its correct predictions. Further, we note that the performance gain is more significant on the noisy dataset, indicating that BABN helps to learn a more robust model, which also agrees with our results on domain generalization in Section 5.1.2.

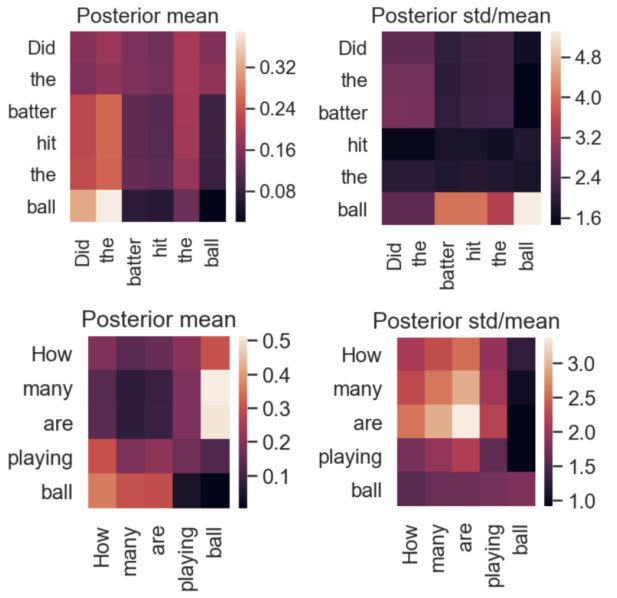

Figure 3. For two questions from VQA, we visualize the posterior mean and std/mean for attention weights of BABN, where each row corresponds to one question. Rows represent queries, and columns represent keys. For example, considering the first question, on the left plot, when the row is "Did" and the column is "hit", the color represents the average attention weight from the query "Did" to the key "hit". On the right plot, the color at the same location represents the uncertainty from the query "Did" to the key "hit". We note that the model is mostly certain except for the query "ball" from the right plot, which is assigning high average attention weights for "Did" and "the" rather than other words as shown on the left.

Results Analysis. *Visualizations.* In Fig. 3, we plot statistics of the posterior distributions for the attention weights of one question in VQA. We visualize the normalized posterior mean (left) as a measure of the average importance of each query-key pair, and posterior standard deviation divided by posterior mean (std/mean on the right) as a measure of uncertainty. The plot shows that BABN is able to learn different uncertainties (std/mean) for each query-key pair in contrast to the fixed std/mean of BAM. This sample-dependent uncertainty of BABN enables the strong capability in modeling attention weights and therefore gives good uncertainty estimation.

Ablation Study. We also conduct ablation study to exam the role of the upward-downward structure by turning the weight parameters ρ and σ to zeros. We found that tuning either parameter to zero would lead to performance drop, especially the parameter ρ, which demonstrates the necessity and effectiveness of the upward-downward structure. Please see detailed results in Table 8 in Appendix.

6. Conclusion

We propose Bayesian attention belief network (BABN), a deep stochastic network by modeling attention weights as hierarchically dependent random variables. A multi-stochastic-layer generative model and a deterministic-upward-stochastic-downward inference network are constructed by leveraging the existing attention architecture. This generic and efficient architecture design enables us to easily convert existing deterministic attention models, including pretrained ones, to BABN, while only slightly increasing memory and computational cost. On various language understanding tasks, BABN exhibits strong performance in accuracy, uncertainty estimation, domain generalization, and adversarial robustness. Interestingly, clear improvement in performance has already been achieved by adding BABN only during the finetuning stage. We further demonstrate the general applicability of BABN on additional tasks, including neural machine translation and visual question answering, where BABN consistently outperforms corresponding baselines and shows great potential to be an efficient alternative to many existing attention models.

Acknowledgements

S. Zhang, X. Fan, and M. Zhou acknowledge the support of Grants IIS-1812699 and ECCS-1952193 from the U.S. National Science Foundation, the APX 2019 project sponsored by the Office of the Vice President for Research at The University of Texas at Austin, the support of a gift fund from ByteDance Inc., and the Texas Advanced Computing Center (TACC) for providing HPC resources that have contributed to the research results reported within this paper.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Bahdanau, D., Cho, K. H., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

Bayer, J. and Osendorfer, C. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In *EMNLP*, 2015.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. In

Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pp. 10–21, 2016.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., and Federico, M. Report on the 11th iwslt evaluation campaign, iwslt 2014. 2014.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *International Conference on Machine Learning*, pp. 1691–1703. PMLR, 2020.

Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., and Inkpen, D. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1657–1668, 2017.

Chen, X., Sun, Y., Athiwaratkun, B., Cardie, C., and Weinberger, K. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570, 2018.

Chen, Y.-C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. Uniter: Learning universal image-text representations. 2019.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pp. 2980–2988, 2015.

Dagan, I., Glickman, O., and Magnini, B. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pp. 177–190. Springer, 2005.

Deng, Y., Kim, Y., Chiu, J., Guo, D., and Rush, A. Latent alignment and variational attention. In *Advances in Neural Information Processing Systems*, pp. 9712–9724, 2018.

Desai, S. and Durrett, G. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dolan, W. B. and Brockett, C. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Edunov, S., Ott, M., Auli, M., Grangier, D., and Ranzato, M. Classical structured prediction losses for sequence to sequence learning. *arXiv preprint arXiv:1711.04956*, 2017.

Fan, X., Zhang, S., Chen, B., and Zhou, M. Bayesian attention modules. *Advances in Neural Information Processing Systems*, 33, 2020.

Fan, X., Zhang, S., Tanwisuth, K., Qian, X., and Zhou, M. Contextual dropout: An efficient sample-dependent dropout module. *arXiv preprint arXiv:2103.04181*, 2021.

Fraccaro, M., Sønderby, S. K., Paquet, U., and Winther, O. Sequential neural models with stochastic layers. In *Advances in neural information processing systems*, pp. 2199–2207, 2016.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Gan, Z., Henao, R., Carlson, D., and Carin, L. Learning deep sigmoid belief networks with data augmentation. In *Artificial Intelligence and Statistics*, pp. 268–276. PMLR, 2015.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, M., and Zettlemoyer, L. S. A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*, 2017.

Garg, S. and Ramakrishnan, G. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6174–6181, 2020.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Iyer, S., Dandekar, N., and Csernai, K. First quora dataset release: Question pairs. *data. quora. com*, 2017.

Jin, D., Jin, Z., Zhou, J. T., and Szolovits, P. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8018–8025, 2020.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Lan, W., Qiu, S., He, H., and Xu, W. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1224–1234, 2017.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio, Y. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pp. 473–480, 2007.

Li, J., Ji, S., Du, T., Li, B., and Wang, T. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium*, 2019.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv e-prints*, pp. arXiv–1907, 2019.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Lu, J., Batra, D., Parikh, D., and Lee, S. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

Miller, T. Simplified neural unsupervised domain adaptation. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, volume 2019, pp. 414. NIH Public Access, 2019.

Morris, J., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., and Qi, Y. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 119–126, 2020.

Mukhoti, J. and Gal, Y. Evaluating Bayesian deep learning methods for semantic segmentation. *arXiv preprint arXiv:1811.12709*, 2018.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Neal, R. M. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113, 1992.

Owen, A. B. *Monte Carlo Theory, Methods and Examples*, chapter 8 Variance Reduction. 2013.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. A decomposable attention model for natural language inference. In *EMNLP*, 2016.

Peng, M., Zhang, Q., Jiang, Y.-g., and Huang, X.-J. Cross-domain sentiment classification with target domain specific information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2505–2513, 2018.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. 2018.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. SQuAD: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822*, 2018.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Shankar, S. and Sarawagi, S. Posterior attention models for sequence to sequence learning. 2018.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. Ladder variational autoencoders. In *NIPS*, pp. 3738–3746, 2016.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650, 2019.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Teney, D., Anderson, P., He, X., and Van Den Hengel, A. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4223–4232, 2018.

Tran, D., Dusenberry, M. W., van der Wilk, M., and Hafner, D. Bayesian layers: A module for neural network uncertainty. *arXiv preprint arXiv:1812.03973*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Wang, Z. and Zhou, M. Thompson sampling via local uncertainty. In *International Conference on Machine Learning*, pp. 10115–10125. PMLR, 2020.

Warstadt, A., Singh, A., and Bowman, S. R. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Williams, A., Nangia, N., and Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, 2018.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pp. 2048–2057, 2015.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

Yu, Z., Yu, J., Cui, Y., Tao, D., and Tian, Q. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6281–6290, 2019.

Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *EMNLP*, 2018.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Zhang, H., Chen, B., Guo, D., and Zhou, M. WHAI: Weibull hybrid autoencoding inference for deep topic modeling. In *International Conference on Learning Representations*, 2018.

Zhou, M., Cong, Y., and Chen, B. Augmentable gamma belief networks. *The Journal of Machine Learning Research*, 17(1): 5656–5699, 2016.

A. Experimental details

A.1. Natural Language Understanding

A.1.1. Model Specifications for In-domain Evaluation

ALBERT (Lan et al., 2019) is used as the pretrained model on large corpora to extract the context embeddings. ALBERT is a memory-efficient version of BERT with parameter sharing and embedding factorization. In our experiments, we use the ALBERT-base model with 12 attention layers and hidden dimension 768. The embedding dimension for factorized embedding is 128.

A.1.2. Experimental Settings for In-domain Evaluation

Our experiments are conducted on both the General Language Understanding Evaluation (GLUE) and Stanford Question Answering (SQuAD) Datasets. There are 8 tasks in GLUE, including Microsoft Research Paraphrase Corpus (MRPC; (Dolan & Brockett, 2005)), Corpus of Linguistic Acceptability (CoLA; (Warstadt et al., 2019)), Recognizing Textual Entailment (RTE; (Dagan et al., 2005)), Multi-Genre NLI (MNLI; (Williams et al., 2017)), Question NLI (QNLI; (Rajpurkar et al., 2016)), Quora Question Pairs (QQP; (Iyer et al., 2017)), Stanford Sentiment Treebank (SST; (Socher et al., 2013)), and Semantic Textual Similarity Benchmark (STS;(Cer et al., 2017)). For SQuAD, we include both SQuAD v1.1 and SQuAD v2.0. We use the codebase[2] from Huggingface Transformers (Wolf et al., 2019). For the detailed experimental settings, we summarize in Table 6.

Table 6. Experimental settings of each task for in-domain pretrained language model (LR: learning rate, BSZ: batch size, DR: dropout rate, TS: training steps, WS: warmping steps, MSL: maximum sentence length).

	LR	BSZ	ALBERT DR	Classifier DR	TS	WS	MSL
CoLA	$1.00e^{-5}$	16	0	0.1	5336	320	512
STS	$2.00e^{-5}$	16	0	0.1	3598	214	512
SST2	$1.00\,e^{-5}$	32	0	0.1	20935	1256	512
MNLI	$3.00\,e^{-5}$	128	0	0.1	10000	1000	512
QNLI	$1.00\,e^{-5}$	32	0	0.1	33112	1986	512
QQP	$5.00\,e^{-5}$	128	0.1	0.1	14000	1000	512
RTE	$3.00\,e^{-5}$	32	0.1	0.1	800	200	512
MRPC	$2.00\,e^{-5}$	32	0	0.1	800	200	512
SQuAD v1.1	$5.00\,e^{-5}$	48	0	0.1	3649	365	384
SQuAD v2.0	$3.00\,e^{-5}$	48	0	0.1	8144	814	512

Table 7. Efficiency on ALBERT-base models.

Attention	Params ↓	s/step ↓
Base	11.7M	0.26
BAM	11.7M	0.35
BABN	12.4M	0.41

[2] https://github.com/huggingface/transformers

A.1.3. Model Specifications for Domain Generalizations

We follow Desai & Durrett (2020) to use bert-base-uncased (Devlin et al., 2018) and roberta-base (Liu et al., 2019) as the baseline models. We also include the results of two non-pretrained models DA (Parikh et al., 2016) and ESIM (Chen et al., 2017) from Desai & Durrett (2020), which are obtained with the open-source implementation in AllenNLP (Gardner et al., 2017). The pretrained models are provided by HuggingFace Transformers (Wolf et al., 2019). Largely following the settings from Desai & Durrett (2020). we finetune BERT with a maximum of 3 epochs, batch size of 16, learning rate of $2e^{-5}$, gradient clip of 1.0, and no weight decay. For RoBERTa, we finetune with a maximum of 3 epochs, batch size of 32, learning rate of $1e^{-5}$, gradient clip of 1.0, and weight decay of 0.1. AdamW (Loshchilov & Hutter, 2018) is used as the optimizer in experiments.

A.1.4. Experimental Settings for Domain Generalizations

For all datasets, we follow the settings from Desai & Durrett (2020) and split the development set in half to obtain a held-out, non-blind test set.

We conduct experiments on three tasks: (1) *Natural Language Inference.* The Stanford Natural Language Inference (SNLI) corpus is a large-scale entailment dataset (Bowman et al., 2015). The similar entailment data across domains is also included in Multi-Genre Natural Language Inference (MNLI) (Williams et al., 2018). Thus the MNLI can be used as an unseen out-of-domain test dataset. (2) *Paraphrase Detection.* Quora Question Pairs (QQP) contains sentence pairs from Quora that are semantically equivalent (Iyer et al., 2017). TwitterPPDB (TPPDB), considered as out-of-domain data, contains the sentence pairs from the paraphrased tweets (Lan et al., 2017). (3) *Commonsense Reasoning.* Situations With Adversarial Generations (SWAG) is a grounded commonsense reasoning task (Zellers et al., 2018). The out-of-domain data is HellaSWAG (HSWAG), which is a more challenging benchmark (Zellers et al., 2018).

A.1.5. Adversarial Robustness

We utilized the same models and training procedures as the in-domain evaluation. The settings for adversarial attack follow those from Morris et al. (2020) with maximum sentence length 512.

A.2. Neural Machine Translation

A.2.1. Model Specifications

Following the Neural Machine Translation (NMT) setting from Deng et al. (2018), we utilize the bidirectional LSTM to embed each source sentence to source representations.

Attention is utilized, during the decoding stage, to identify which source positions should be used to predict the target using a function of previous generated tokens as the query. The aggregated features are passed to an MLP to produce the distribution over the next target word (see details in Deng et al. (2018)).

A.2.2. Experimental Settings

For NMT we use the IWSLT dataset (Cettolo et al., 2014). We follow the same preprocessing as in Edunov et al. (2017) which uses Byte Pair Encoding vocabulary over the combined source/target training set to obtain a vocabulary size of 14k tokens (Sennrich et al., 2015) with sequences of length up to 125. A two-layer bi-directional LSTM with 512 units is used as the encoder and another two-layer LSTM with 768 units is used as the decoder. Other training details include: the batch size 6, dropout rate 0.3, and learning rate $3e^{-4}$ with Adam optimizer (Kingma & Ba, 2014). During testing, we use beam search with beam size 10 and length penalty as 1 (Wu et al., 2016).

A.3. Visual Question Answering

A.3.1. Model Specifications

The state-of-the-art VQA model, MCAN (Yu et al., 2019), is used in the experiments. The MCAN consists of MCA layers. Each MCA layer consists of self-attention (SA) over question and image features, and guided-attention (GA) between question and image features. Multi-head structure as in Vaswani et al. (2017), including the residual and layer normalization components, is incorporated in the MCA layer. MCAN represents the deep co-attention model which consists of multiple MCA layers cascaded in depth to gradually refine the attended image and question features. We adopt the encoder-decoder structure in MCAN (Yu et al., 2019) with four co-attention layers.

A.3.2. Experimental Settings

We conduct experiments on the commonly used benchmark, VQA-v2 (Goyal et al., 2017), containing human-annotated question-answer (QA) pairs. There are three types of questions: Yes/No, Number, and Other. The dataset is split into the training (80k images and 444k QA pairs), validation (40k images and 214k QA pairs), and testing (80k images and 448k QA pairs) sets. We perform evaluation on the validation set as the true labels for the test set are not publicly available (Deng et al., 2018). To construct the noisy dataset, we incorporate the Gaussian noise (mean 0, variance 5) to image features. We use the same model hyperparameters and training settings in Yu et al. (2019) as follows: the dimensionality of input image features, input question features, and fused multi-modal features are set to be 2048, 512, and 1024, respectively. The latent dimensionality in the multi-head attention is 512, the number of heads is set to 8, and the latent dimensionality for each head is 64. The size of the answer vocabulary is set to $N = 3129$ using the strategy in Teney et al. (2018). To train the MCAN model, we use the Adam optimizer (Kingma & Ba, 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The base learning rate is set to $\min(2.5te^{-5}, 1e^{-4})$, where t is the current epoch number starting from 1. After 10 epochs, the learning rate is decayed by $1/5$ every 2 epochs. All the models are trained up to 13 epochs with the same batch size of 64.

A.3.3. Ablation Study

Table 8. Ablation study of the upward path in BABN on VQA.

	Accuracy ↑		PAvPU ↑	
	Original	Noisy	Original	Noisy
$\rho = 0$, σ=$1.00e^{-6}$	44.62	32.16	50.93	53.22
$\rho = 1.5$, σ=0	66.78	64.04	69.99	69.02
$\rho = 1.5$, σ=$1.00e^{-6}$	**66.92**	**64.40**	**72.21**	**70.43**

We conduct ablation study to exam the role of the upward-downward structure by turning the weight parameters ρ and σ to zeros. Table 8 shows that tuning either parameter to zero would lead to performance drop, especially the parameter ρ, which demonstrates the necessity and effectiveness of the upward-downward structure. We also found that the experimental results are not sensitive to the choice of the value of the ρ. Any number from 1 to 4 would give similar results. The other is the scaling factor σ that controls the importance of the $\boldsymbol{h}^l$ in $\boldsymbol{\lambda}^l$. We found that the performance is not that sensitive to its value and it is often beneficial to make it smaller. In all experiments considered in the paper, which cover various noise levels and model sizes, we have simply fixed it at $1.00e^{-6}$.