Improving Minimum Bayes Risk Decoding with Weight Uncertainty

Anonymous ACL submission

Abstract

Minimum Bayes Risk (MBR) decoding has become a popular decoding strategy for different natural language generation tasks, especially machine translation. MBR relies on an estimator of an expected loss, where we use our learned model as a proxy for the target distribution that we wish to take this expectation with respect to. However, this reliance can be problematic if the model is a flawed proxy, for example, in light of a lack of training data in a specific domain. In this work, we show how using a posterior over model parameters, and decoding with a weighted-averaging over multiple models, can improve the performance of MBR by accounting for uncertainty over the learned model. We benchmark different methods for learning posteriors and show that performance correlates with the diversity of the combined set of models' predictions. Intriguingly, prediction diversity also determines whether risk can be successfully used for selective prediction.

1 Introduction

Natural language generation systems use decoding strategies to construct an output for a given input from a probabilistic model. Minimum Bayes Risk (MBR) decoding is one such strategy, which aims to find the string that minimizes an expected risk function, for example, the negative value of some standard quality metric. MBR was originally proposed in the era of statistical machine translation (Kumar and Byrne, 2002), with the motivation that while we might not be able to trust that our models accurately learn the mode of the target distribution, they are overall good representations of this distribution (Smith, 2011). More recent works have shown that these problems persist with modern models (Stahlberg and Byrne, 2019; Cohen and Beck, 2019), precipitating the resurgence of MBR.

Amidst this resurgence, there has been ample work on efficient variations of MBR (Eikema and Aziz, 2022; Fernandes et al., 2022; Cheng and Vlachos, 2023; Vamvas and Sennrich, 2024) and the effects of the chosen utility function (Freitag et al., 2022). On the other hand, little attention has been paid to a potentially large source of error: the model distribution. In MBR, the quality of the risk estimator—and consequently, the quality of the chosen string—relies on a good estimate of the target distribution. For over-parameterized models like large deep networks, parameters change drastically simply by using different random seeds (Fort et al., 2019, App. B). This suggests that there is large uncertainty in the learned model and this component of the MBR pipeline may be error-prone. Accordingly, increasing the robustness of MBR to such uncertainties is a clear path toward potential improvements.

In this work, we propose different methods for accounting for parameter uncertainty in MBR decoding. In short, we take an additional expectation over the posterior distribution of model parameters when computing expected risk. In practice, this boils down to combining the predictions of multiple models—sampled from an estimate of the Bayesian posterior—when generating the hypothesis set in MBR. Such model combination has been shown to provide better-calibrated distributions and can improve robustness and downstream performance, especially in low-resource settings (Blundell et al., 2015; Lakshminarayanan et al., 2017; Maddox et al., 2019; Shen et al., 2024). We explore both token- and sequence-level methods for combining model predictions.

Overall, we find strong evidence that accounting for weight uncertainty can improve MBR and make it more robust. We find that improvements trend with the expressiveness of the posterior distribution from which the combined models are obtained. Likely related to this observation, we see that the performance of uncertainty-aware MBR is highly correlated with the diversity of the hypothesis set generated from these models. We also find that weight uncertainty provides a useful signal for selective prediction, where we observe that uncertainty-aware expected risk can be used to decide when to predict vs. abstain from generation. Finally, we show that our methods scale well to a larger number of models and larger hypothesis set sizes.

2 Background

2.1 Probabilistic Language Generation

Modern models for language generation are predominantly locally-normalized, autoregressive models of the conditional distribution over next tokens. The probability of a sequence of tokens forming a string can be determined by the product of all next token probabilities in the sequence. Formally, given an input $\mathbf{x}$, the probability of an output sequence $\mathbf{y} = \langle y_1, y_2, \ldots \rangle$ can be computed as

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{t=1}^{|\mathbf{y}|} p(y_t \mid \mathbf{y}_{<t}, \mathbf{x}), \quad (1)$$

where each y_t is a token from some predetermined vocabulary $\mathcal{V}$.

Learning $p_{\boldsymbol{\theta}}$. We denote a single mode as $p_{\boldsymbol{\theta}}$, where $\boldsymbol{\theta}$ are its parameters (also called weights). These parameters are generally learned given paired examples $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$, a loss function and an optimization procedure. The loss function quantifies how far our model is from a chosen target distribution, for example the data-generating distribution $p(\cdot \mid \mathbf{x})$ that we assume $\mathcal{D}$ is sampled from.

Decoding from $p_{\boldsymbol{\theta}}$. At inference time, our goal is to generate a string from $p_{\boldsymbol{\theta}}(\cdot \mid \mathbf{x})$. The set of decision rules used in this process is often referred to as the decoding strategy. One such strategy is simply to sample tokens autoregressively until a stopping criterion (usually a fixed maximum length or a special end-of-sequence token) is met. Another strategy is to search for the maximum probability string according to $p_{\boldsymbol{\theta}}(\cdot \mid \mathbf{x})$. Both of these approaches have proved problematic empirically (Fan et al., 2018; Holtzman et al., 2020; Eikema and Aziz, 2020; Hewitt et al., 2022), prompting the exploration of alternative strategies. The shortcomings of these strategies have been (at least partially) attributed to the fact that they do not consider a string's utility, which may not perfectly align with its probability. Minimum Bayes Risk decoding aims to solve this issue.

2.2 Minimum Bayes Risk Decoding

Minimum Bayes Risk decoding is a strategy based on Bayesian Decision Theory which states that optimal decisions are those that minimize expected risk (or maximize expected utility), see DeGroot (2005, inter alia). Given a utility function $u : \mathcal{V}^* \times \mathcal{V}^* \to \mathbb{R}_{\geq 0}$ which assigns to each pair of strings a nonnegative utility value, we should aim to find the string that maximizes expected utility with respect to our target distribution. This principle is especially appealing when working with a possibly imperfect model of the target distribution, such as $p_{\boldsymbol{\theta}}$. Specifically, it allows us to make use of the full model distribution rather than relying on the adequacy of individual samples, which is argued to be the downfall of other decoding strategies (Eikema and Aziz, 2020). We thus choose the hypothesis that satisfies:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}' \in \mathcal{V}^*} \mathop{\mathbb{E}}_{\mathbf{y} \sim p_{\boldsymbol{\theta}}(\cdot \mid \mathbf{x})} \left[u(\mathbf{y}, \mathbf{y}') \right] \quad (2)$$

$$= \arg\max_{\mathbf{y}' \in \mathcal{V}^*} \sum_{\mathbf{y} \in \mathcal{V}^*} p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x}) u(\mathbf{y}, \mathbf{y}'). \quad (3)$$

There are several obstacles to the direct computation of Eq. (3). Namely, both summing over all possible strings in $\mathcal{V}^*$ to compute our expectation and searching over them to find the expectation-maximizing hypothesis are computationally infeasible. Thus, typically an approximation to the MBR problem Eq. (3) is used in practice.

The standard approach to circumvent the aforementioned obstacles is to employ an estimator—often specifically a Monte Carlo estimator—of our expected utility and limit the search space to a subset of $\mathcal{V}^*$. Since the estimator requires a sample of strings from the distribution of interest, the same strings are often used in both the approximate search and utility estimation.[1] We refer to this subset as the hypothesis set and denote the sample used in our estimator as $\mathcal{H} = \{\mathbf{y}^{(i)}\}_{i=1}^{N}$. In the case of a Monte Carlo estimator where $\mathbf{y}^{(i)} \sim p_{\boldsymbol{\theta}}$, we denote this set as $\mathcal{H}_{\boldsymbol{\theta}}$. This leads to the following

[1]Some works have explored using different subsets for these two steps (Eikema and Aziz, 2022; Fernandes et al., 2022); we leave the exploration of the interaction of this design choice with our methods to future work.

approximation to Eq. (3):[2]

$$\widehat{\mathbf{y}}^* = \underset{\mathbf{y}' \in \mathcal{H}_{\boldsymbol{\theta}}}{\arg\max} \sum_{\mathbf{y} \in \mathcal{H}_{\boldsymbol{\theta}}} u(\mathbf{y}, \mathbf{y}'). \tag{4}$$

Most prior work has focused on making the approximation in Eq. (4) more efficient (Eikema and Aziz, 2022; Fernandes et al., 2022; Cheng and Vlachos, 2023; Vamvas and Sennrich, 2024) or on better choices for utility functions (Freitag et al., 2022). Yet, few have considered the important underlying assumption of MBR: that $p_{\boldsymbol{\theta}}$ is a good substitute for p. In the face of high **weight uncertainty**, that is, uncertainty about suitable values of model parameters $\boldsymbol{\theta}$, this assumption may not hold. Rather, a single choice of model parameters may not provide a robust substitute for the target distribution. Given that weight uncertainty occurs often in overparameterized deep learning models, when the model has not seen sufficient data in a specific domain, this should be an important consideration when using MBR. In this work, we show how weight uncertainty can be incorporated into MBR to create a more robust decoding method.

3 Minimum Bayes' Risk Decoding with Weight-Uncertainty

Our goal is to show how weight uncertainty can be used to improve MBR decoding. We first introduce weight uncertainty, and then present two decoding methods based on it.

3.1 Weight Uncertainty

Placing a probability distribution over model parameters is an oft-employed method for modeling weight uncertainty (Graves, 2011; Blundell et al., 2015; Maddox et al., 2019; Osawa et al., 2019; Möllenhoff and Khan, 2023; Yang et al., 2024). The distribution $q(\cdot)$, which in our case will be an approximation to the Bayesian posterior distribution, attaches a probability to each parameterization. There are numerous methods one can use for obtaining $q(\cdot)$; we discuss the ones that we employ in §4.1 and §4.2.

Using the posterior $q(\cdot)$, we can create a more robust version of our model (Maddox et al., 2019) by combining the predictions of multiple $p_{\boldsymbol{\theta}}$, weighted by the probability attached to each parameterization $\boldsymbol{\theta}$. The resulting distribution is often referred to as the **predictive posterior** distribution. Our proposed method to account for weight uncertainty is then simply to replace the definition of $p_{\boldsymbol{\theta}}$ in Eq. (3) with the predictive posterior p_Θ, leading to the following variant of the MBR problem:

$$\mathbf{y}^\Theta = \underset{\mathbf{y}' \in \mathcal{V}^*}{\arg\max} \sum_{\mathbf{y} \in \mathcal{V}^*} p_\Theta(\mathbf{y} \mid \mathbf{x}) u(\mathbf{y}, \mathbf{y}') \tag{5}$$

For autoregressive sequence generation, there are two logical definitions of this predictive posterior. The first uses the product of the expectations of token-level probabilities:

$$p_\Theta^{(\text{tok})}(\mathbf{y} \mid \mathbf{x}) \coloneqq \prod_{t=1}^{T} \underset{\boldsymbol{\theta} \sim q}{\mathbb{E}} \left[p_{\boldsymbol{\theta}}(y_t \mid \mathbf{y}_{<t}, \mathbf{x}) \right]. \tag{6}$$

The second uses the expectation of the probability of full sequences under each parameterization:

$$p_\Theta^{(\text{seq})}(\mathbf{y} \mid \mathbf{x}) \coloneqq \underset{\boldsymbol{\theta} \sim q}{\mathbb{E}} \left[p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x}) \right]. \tag{7}$$

Under mild assumptions, these two quantities should be identical. However, their approximations using a finite sample size—which are necessary since the exact computation of these quantities is infeasible—are different.[3] We thus explore the use of Monte Carlo estimates of Eqs. (6) and (7) in place of p_Θ in Eq. (5). Note that regardless of which estimator is used, the ensemble is usually sampled i.i.d. from q (Maddox et al., 2019, inter alia). We denote such ensembles as $\mathcal{M} = \{\boldsymbol{\theta}^{(i)} \sim q(\boldsymbol{\theta})\}_{i=1}^{M}$.

3.2 Token-Level Averaging

We first explore the use of token-level predictive posteriors, i.e., Eq. (6), for incorporating weight uncertainty into MBR. We can approximate $p_\Theta^{(\text{tok})}$ with a Monte-Carlo estimator, which uses our ensemble of sampled models $\mathcal{M}$:

$$\widehat{p}_\Theta^{(\text{tok})}(y_t \mid \mathbf{y}_{<t}, \mathbf{x}) = \frac{1}{|\mathcal{M}|} \sum_{\boldsymbol{\theta} \in \mathcal{M}} p_{\boldsymbol{\theta}}(y_t \mid \mathbf{y}_{<t}, \mathbf{x}) \tag{8}$$

We then sample our set of hypotheses $\mathcal{H}_\Theta$ from this approximation of the predictive posterior. With this, we get the following estimator for Eq. (5):[2]

$$\widehat{\mathbf{y}}^\Theta = \underset{\mathbf{y}' \in \mathcal{H}_\Theta}{\arg\max} \sum_{\mathbf{y} \in \mathcal{H}_\Theta} u(\mathbf{y}, \mathbf{y}'). \tag{9}$$

There are several intuitive reasons why averaging the outputs of multiple models should help in

[2]We drop the normalizing term in our Monte Carlo estimator for succinctness as it does not affect the arg max operation.

[3]These definitions are also discussed in Malinin and Gales (2021, Sec. 3); the theoretical properties of their estimators are derived in Appendix A of the same work.

MBR. Perhaps the foremost is that the probabilities obtained from this style of model averaging are usually better-calibrated than those of a single model and better reflect predictive uncertainty, for instance, in out-of-domain settings (Shen et al., 2024, inter alia). Since predictive uncertainty has been shown to correlate with hallucinations (Xiao and Wang, 2021), one hope would be that incorporating weight uncertainty would downweigh potentially hallucinated outputs, such as mistranslations.

Even though the method introduces an additional overhead during inference because $|\mathcal{M}|$ models have to be evaluated for approximating the expectation, the number of comparisons required for the Bayes risk estimator stays the same. To be precise, $|\mathcal{H}|^2$ evaluations are required for a given hypothesis set. We next explore sequence-level methods for approximating Eq. (5).

3.3 Sequence-Level Averaging

Our second approach uses estimators for Eq. (7)—which requires sequence-level averaging—to find an approximate solution to Eq. (5). We make use of an important result: when our utility function u is bounded[4] or nonnegative we can apply Fubini's theorem to switch the order of the two expectations in Eq. (5) (one of which is implicit in the definition of p_Θ) (DeGroot, 2005, Sec. 8.9):

$$\mathbf{y}^\Theta = \operatorname*{arg\,max}_{\mathbf{y}' \in \mathcal{V}^*} \sum_{\mathbf{y} \in \mathcal{V}^*} \mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim q} [p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x})] \, u(\mathbf{y}, \mathbf{y}') \tag{10}$$

$$= \operatorname*{arg\,max}_{\mathbf{y}' \in \mathcal{V}^*} \mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim q} \Big[\sum_{\mathbf{y} \in \mathcal{V}^*} p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x}) u(\mathbf{y}, \mathbf{y}') \Big]. \tag{11}$$

Eq. (10) follows by the definition of $p_\Theta^{(\text{seq})}$. This suggests that another valid estimator of Eq. (5)—and therefore another method for incorporating weight uncertainty in MBR—is to average per-sequence utilities (rather than token probabilities) across the posterior. One interpretation of this approach is as a consensus decoding that prefers outputs with high utility under many models.

In practice, approximating Eq. (11) can be done simply by using a specific $\mathcal{H}$ in Eq. (4). Given an ensemble of models $\mathcal{M}$, let $\mathcal{H}_\mathcal{M} = \cup_{\boldsymbol{\theta} \in \mathcal{M}} \mathcal{H}_{\boldsymbol{\theta}}$. Our approximate solution then becomes:[2]

$$\widehat{\mathbf{y}}^\Theta = \operatorname*{arg\,max}_{\mathbf{y}' \in \mathcal{H}_\mathcal{M}} \sum_{\boldsymbol{\theta} \in \mathcal{M}} \sum_{\mathbf{y} \in \mathcal{H}_{\boldsymbol{\theta}}} u(\mathbf{y}, \mathbf{y}'). \tag{12}$$

Note that the same hypothesis can be contained in multiple $\mathcal{H}_{\boldsymbol{\theta}}$, and this will potentially have a large effect on that hypothesis's utility. This differentiates our approach from other works that have used multiple models in MBR, where summation occurs over the union of individual hypothesis sets (Kobayashi, 2018, Alg. 1). These prior approaches therefore do not provide an unbiased estimate of the expected risk in Eq. (11).

3.4 Selective Prediction with Bayes' Risk

For some inputs, the quality of model predictions might be poor and even using MBR cannot lead to outputs of sufficient quality. For example, an input may be out of domain or contain errors, making it unlikely that the model can provide a good output. In such situations, the best action is arguably to abstain from answering and, e.g., defer to a human expert instead. This is the approach taken in selective prediction: answers are only given for queries in which inputs (or outputs) score highly according to some criterion (Geifman and El-Yaniv, 2017; Ren et al., 2023; Kuhn et al., 2023). Formally, selective prediction defines a criterion $s : \mathcal{V}^* \to \mathbb{R}$ that assigns a score for a given input $\mathbf{x}$; this score may depend solely on the input or involve an assessment of model outputs, for example by transforming the predictive distribution (Ren et al., 2023). Given a factor α and a test-dataset $\mathcal{D}_{\text{test}}$, we consider the model's answers for the top-$\lceil \alpha \cdot |\mathcal{D}_{\text{test}}| \rceil$ examples according to s. Only this subset is evaluated; we "abstain" from providing model answers to the remaining examples. If s is reliable, performance should improve as α decreases and we evaluate a smaller and smaller subset of outputs.

Expected utility is a logical candidate for such a criterion. If it is low for a particular input, we should abstain from answering; if it is high, we can place more trust in the model's answer. Here we compare different methods for using expected utility as the selective prediction criterion. We first consider the utility of the maximum-utility output in $\mathcal{H}_\Theta$ or $\mathcal{H}_\mathcal{M}$, i.e:[2]

$$s^*_{\text{tok}}(\mathbf{x}) = \max_{\mathbf{y}' \in \mathcal{H}_\Theta} \sum_{\mathbf{y} \in \mathcal{H}_\Theta} u(\mathbf{y}, \mathbf{y}') \tag{13}$$

$$s^*_{\text{seq}}(\mathbf{x}) = \max_{\mathbf{y}' \in \mathcal{H}_\mathcal{M}} \sum_{\boldsymbol{\theta} \in \mathcal{M}} \sum_{\mathbf{y} \in \mathcal{H}_{\boldsymbol{\theta}}} u(\mathbf{y}, \mathbf{y}') \tag{14}$$

[4]Many commonly used utility functions for MBR are bounded and non-negative. For example, BLEU (Papineni et al., 2002) and BERTScore (Zhang et al., 2020) return scores from 0 to 100 or 0 to 1, respectively.

Another strategy is to use the expected utility *across* outputs for the given input. We can do this by averaging the utility of all outputs in the hypothesis set $\mathcal{H}_\Theta$ or $\mathcal{H}_\mathcal{M}$.[2]

$$\bar{s}_{\text{tok}}(\mathbf{x}) = \sum_{\mathbf{y}'\in\mathcal{H}_\Theta} \sum_{\mathbf{y}\in\mathcal{H}_\Theta} u(\mathbf{y}, \mathbf{y}') \tag{15}$$

$$\bar{s}_{\text{seq}}(\mathbf{x}) = \sum_{\boldsymbol{\theta}\in\mathcal{M}} \sum_{\mathbf{y}'\in\mathcal{H}_{\boldsymbol{\theta}}} \sum_{\mathbf{y}\in\mathcal{H}_{\boldsymbol{\theta}}} u(\mathbf{y}, \mathbf{y}') \tag{16}$$

3.5 Discussion

In the incorporation of weight uncertainty in MBR, is not clear a priori whether token- or sequence-level estimators should lead to a better-performing decoding strategy. While Malinin and Gales (2021) found that token-level methods performed best for predictive uncertainty estimation with entropy-based measures, model architectures have changed considerably since this study; we thus explore both methods. We note, however, that there are clear advantages in terms of computational complexity between the two approaches. While both require evaluating $|\mathcal{M}|$ models, token-level aggregation only requires $|\mathcal{H}|^2$-many MBR evaluations, whereas sequence-level aggregation needs $|\mathcal{M}| \cdot |\mathcal{H}|^2$ calculations. On the other hand, Eq. (11) is easier to parallelize if models are kept on separate devices as aggregation does not need to happen at every time step.

The method presented in Eq. (11) draws parallels between MBR and PAC-Bayes bounds (Alquier, 2021) which study the risk (or negative utility) of predictive posteriors and can be used for further theoretical insights. Token- and sequence-level aggregation methods can also be combined to obtain a similar hierarchical method to Manakul et al. (2023). Finally, our work provides a framework that encompasses earlier system aggregation methods that have weighted model predictions to arrive at similar decision rules, for example by optimizing scalar model weights using a minimum-risk objective (González-Rubio et al., 2011, Eq. 8). We believe this work can therefore aid in the understanding of these prior methods.

4 Experiments & Results

In this section, we demonstrate empirically that incorporating model weight uncertainty into the MBR framework can improve decoding strategy performance. We first provide common experimental details in §4.1. Then, we compare token- and sequence-level ensembling and show results with different estimates of the posterior distributions in §4.2. §4.3 explores a trade-off between performance and ensemble diversity and §4.4 shows results when using Bayes' risk for selective prediction. Finally, we provide intuitions into the scaling behavior of various methods in §4.5.

4.1 Experimental Details

We focus our experiments on machine translation using neural language generation models.[5] All of our models follow the Transformer$_{\text{base}}$ architecture from Vaswani et al. (2017) and are encoder-decoder models with 6 layers for each component. We use two datasets: the WMT14 English to German (En-De) translation task (Bojar et al., 2014), where we evaluate on newstest2014, and the IWSLT14 German to English (De-En) translation task (Cettolo et al., 2014). We evaluate all models using the SacreBLEU implementation (Post, 2018) of BLEU (Papineni et al., 2002) and the quality estimator COMET$_{22}$ (Rei et al., 2022). We train all models using the IVON optimizer (Shen et al., 2024), as described in the next paragraph. We use BLEU for u. Further details are given in App. B.

Learning weight uncertainty We use the variational learning algorithm IVON to estimate a posterior distribution over model weights. We learn a unimodal Gaussian posterior with diagonal covariance, i.e., $q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid \mathbf{m}, \Sigma)$ for mean $\mathbf{m}$ and covariance matrix Σ. Setting model parameters equal to the mode of this distribution ($\mathbf{m}$) is similar to standard neural network training but Σ also provides an estimate of its stability. To be precise, for each parameter m_i the variance Σ_{ii} indicates how much this parameter can be changed without significant performance degradation. Each training run has only negligible overhead compared to AdamW (Loshchilov and Hutter, 2019) and gives comparable performance.

4.2 Weight Uncertainty & Model Combination

We show that different posterior forms can be used to improve token- and sequence-level model combination in MBR. Results on WMT14 and IWSLT14 are shown in Tab. 1. All models are evaluated using a hypothesis set size of 20 obtained using both

[5] For other tasks like dialog or summarization, no improvements over beam search or ancestral sampling were observed when using MBR. We leave the exploration of these results for future work.

Method	WMT14 En-De Sampling BLEU	WMT14 En-De Sampling COMET	WMT14 En-De Beam Search BLEU	WMT14 En-De Beam Search COMET	IWSLT14 De-En Sampling BLEU	IWSLT14 De-En Sampling COMET	IWSLT14 De-En Beam Search BLEU	IWSLT14 De-En Beam Search COMET	MBR comparisons	Effective beam size
Best Mean	23.25	67.74	26.55	73.25	33.69	74.71	35.90	76.65	400	20
	24.07	68.47	26.55	73.33	34.53	75.18	36.07	76.76	1600	40
Sequence-level										
Unimodal	24.08	68.90	26.60	73.30	34.65	75.20	35.99	76.67	1600	80
Mixture	**24.10**	**69.20**	26.81	73.70	**35.42**	**75.84**	37.42	77.69	1600	80
Token-level										
Unimodal	23.38	67.77	26.57	73.26	33.62	74.68	35.94	76.66	400	80
Mixture	23.37	67.74	**27.19**	**74.04**	34.61	75.06	**38.56**	**78.31**	400	80

Table 1: Using four model samples to incorporate weight-uncertainty can improve the performance of MBR decoding on WMT14 and IWSLT14. More complex mixture posteriors offer further improvements over simpler unimodal posterior, which can not always improve over MBR for the equivalent number of comparisons. Interestingly, sequence-level aggregation provides stronger improvements when hypothesis sets are obtained via sampling, whereas token-level aggregation is better mainly when beam search is used.

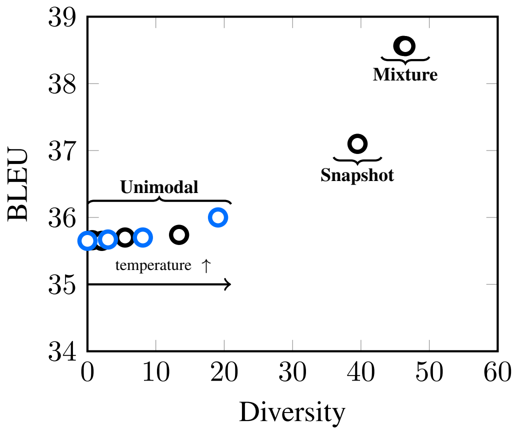


Figure 1: Weight-uncertainty is more successful when the ensembled models are diverse. We compare a diagonal Gaussian posterior (unimodal) to one mixture-of-Gaussian posterior that mixes models from one training run (snapshot) and one that uses models from multiple runs (mixture). Sampling from a unimodal posterior with larger temperature can increase diversity and improve performance (in blue). Results with token-level combination on IWSLT14 using beam search.

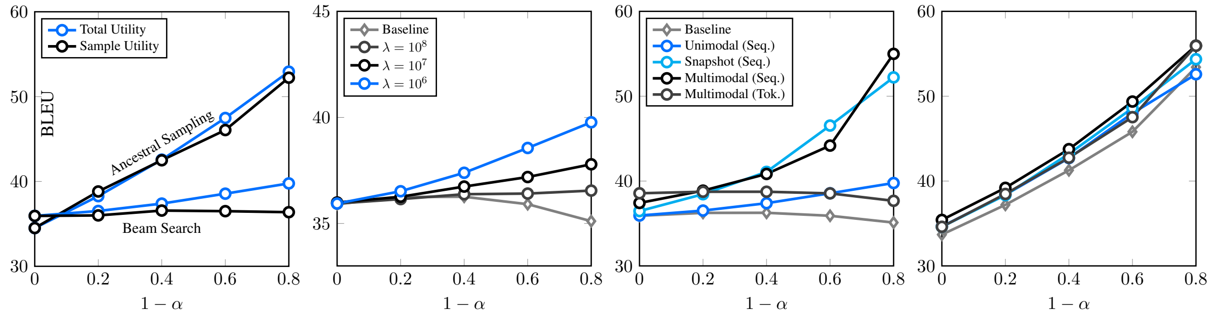
ancestral sampling and beam search. Further details are in App. B. We first discuss the changes in performance that we observe as a function of the estimation of q—the posterior distribution over model parameters—before discussing the performance of token- and sequence-level combinations for creating the predictive posterior.

Comparison of parameter posteriors Here we compare using a uni- and a multimodal Gaussians for q. While the unimodal posterior is faster to train, because it only requires one training run, the multimodal posterior can capture a more complex distribution which can be beneficial. We train four unimodal posteriors using IVON with different seeds. We either use the best-performing (according to the performance of $\mathbf{m}$) posterior (unimodal) or compose a mixture-of-Gaussian (mixture) of all independently-trained posteriors. We obtain this mixture by setting the weight of each mixture component to be equal. This resembles common ensembling techniques in deep learning like deep ensembles (Lakshminarayanan et al., 2017) but has shown superior performance (Shen et al., 2024).

Tab. 1 shows results with both posteriors. We use either 4 samples from the unimodal posterior or the mean of each mixture component. We find that both can give improvements over using just one model when the beam size of this one model matches the number of beams per model when using a posterior. When controlling for the number of MBR comparisons, the improvements of uncertainty-aware MBR with only a unimodal posterior relative to standard MBR can vanish. Mixture-based posteriors, though, give much stronger improvements and even outperform the best mean when MBR comparisons are matched but require more training effort. We take this as indication that incorporating knowledge of weight uncertainty is helpful and that more complex posteriors provide further improvements, potentially due to incorporating knowledge from various loss basins (Lion et al., 2023).

Token- vs. sequence-level combination Here, we compare the use of token- and sequence-level posteriors (Eqs. (9) and (11)) in MBR. Since Tab. 1 shows similar trends for unimodal and mixture-based posteriors, we mainly discuss the latter.

We find that, in comparison to ancestral sam-



(a) Total vs. sample utility (b) Unimodal, Beam Search (c) Beam Search (d) Sampling

Figure 2: Weight-untertainty and MBR can be combined for selective prediction. Both the total risk and best-output-risk can be used effectively for selective prediction (a) but creating the hypothesis set with ancestral sampling performs better than beam search. Increasing temperature when sampling from unimodal posteriors also improves selective prediction (b). Using ancestral sampling, selective prediction generally works well (d) but when using beam search the performance depends on increased diversity and more expressive posteriors. Results on IWSLT14.

pling, the improvements over the baseline from using beam search with token-level combination to form the hypothesis set are much stronger. Still, when using a mixture-based posterior, performance is improved in both settings but larger for IWSLT14 than for WMT14. Sequence-level combination, on the other hand, provides similar improvements for both settings. As discussed, these improvements also hold when matching the number of MBR comparisons. Hence, the preferred method may depend on the decoding algorithm used to create the hypothesis set. Overall, modeling weight uncertainty reduces hallucinations, as shown in App. C.1 and the qualitative examples in App. C.2, which show fewer translation errors.

4.3 Correlation of Quality and Diversity

Next, we show that the performance of MBR with weight-uncertainty is strongly correlated with the prediction diversity of the models that are ensembled. This is in line with prior works on ensembling for classification tasks which have found that diversity is often important for good performance (Fort et al., 2019; Masegosa, 2020) but can also form a trade-off with, for example, individual model performance (Abe et al., 2022; Wood et al., 2023).

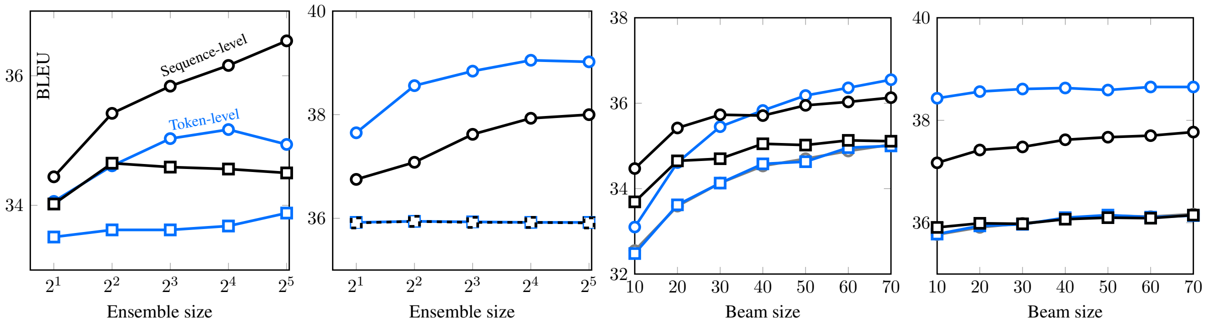
We hypothesize that prediction diversity, and incorporating knowledge from multiple loss basins–regions with low loss–due to a more complex posterior, is the main reason why multimodal posteriors outperform unimodal posteriors. We empirically validate the former claim in Fig. 1, where we plot BLEU on IWSLT14 with token-level averaging against the prediction diversity, which we measure as 100 minus average self-BLEU; self-BLEU scores are measured on the set of greedy decoding outputs of each ensemble member, similar to Shen et al. (2019). The plot shows a clear correlation between both metrics. Hence, we ask two questions: 1) can diversity be promoted in unimodal posteriors to improve performance and 2) can we find a method with the same training overhead as a unimodal posterior but more expressiveness?

For the first, we note that the variance of the IVON posterior is $\boldsymbol{\sigma}^2 = 1/\lambda(\mathbf{h} + \delta)$, where $\mathbf{h}$ is the expected Hessian of the loss, δ is weight-decay and λ the effective sample size which can be seen as an (inverse) temperature parameter. We decrease λ gradually, which samples models from the posterior with higher temperature. This improves diversity and can improve performance. For the latter, we use a mixture-of-Gaussian consisting of checkpoints from one training run, denoted by "snapshot" due to its similarity to snapshot ensembles (Huang et al., 2017). This comes at no training time increase but can improve performance by incorporating knowledge from different regions along the optimization trajectory. Our results show that good posterior approximation is important and we expect further improvements from better approximations.

4.4 Selective Prediction with Bayes' Risk

Here, we explore the use of expected Bayes' risk for selective prediction. We observe that both the maximum output utility and the expected output utility (i.e., average expected utility across outputs) can be used effectively for selective prediction. Our results are summarized in Fig. 2.

First, we find in Fig. 2 (a) that using the average expected utility across outputs as our selective pre-



(a) Ancestral Sampling (b) Beam Search (c) Ancestral Sampling (d) Beam Search

Figure 3: Scaling behavior of weight-uncertainty in MBR on IWSLT14 in terms of ensemble (a, b) and hypothesis set size (c, d). (a, b) For a unimodal posterior (□), larger ensembles improve token-level combination using sampling but not beam search. For mulitmodal posteriors (○), larger ensembles generally improve performance. (c, d) Sequence-level combination performs better for smaller beam sizes but is outperformed by token-level combination at larger ones. Scaling the hypothesis set produces stronger improvements for ancestral sampling than beam search.

diction criterion performs slightly better than just using the best-expected-output utility. This seems especially true when creating hypothesis sets with beam search, which performs much worse than ancestral sampling in general in this setting. Next, we again sample from the unimodal posterior with different temperatures (via decreasing λ). We find that this improves selective prediction with MBR when using beam search to create the hypothesis set, and likewise corresponds to an increase in prediction diversity, as shown in Fig. 2 (b).

Finally, we evaluate the influence of the posterior approximation. First, we find that a hypothesis set built with ancestral sampling is reliable independent of the used posterior. Even the single model baseline works well but is outperformed by using an ensemble and more expressive posteriors give bigger improvements. For beam search, the baseline completely fails and token-level can be unreliable. Sequence-level combination performs much better, especially with more expressive multimodal posteriors. These results are shown in Fig. 2 (c, d).

In short, ancestral sampling provides better selective prediction but worse downstream performance than beam search, where multimodal posteriors and sequence-level combination are preferable.

4.5 Scaling Behavior

Finally, we examine the scaling behavior of token- and sequence-level combination with different posteriors. Results are summarized in Fig. 3 and show scaling both ensemble and hypothesis set size.

First, we show scaling the ensemble size in Fig. 3 (a) for ancestral sampling and beam search (b). Using beam search, both token- (in blue) and sequence-level (in black) combination using unimodal posteriors provide no improvements. For ancestral sampling, we find improvements with a unimodal posterior, especially at larger ensemble sizes of 32 models, but sequence-level combination of a unimodal posterior only improves until 4 models. In all other settings, scaling the ensemble size is usually beneficial. This again shows that if the number of models is scaled, it is helpful if they are diverse and the posterior more expressive.

When scaling hypothesis sets with beam search, the improvements are small, likely because the hypothesis sets lack diversity. Note that the per-model hypothesis set size is shown. Ancestral sampling shows a different picture and we obtain strong improvements when scaling hypothesis sets. Intriguingly, for small hypothesis sets it is better to use sequence-level ensembling but for larger sizes token-level combination is better.

5 Discussion & Conclusion

In this work, we explore the effects of accounting for weight uncertainty in MBR. We investigate different methods within this realm, combining predictions from multiple models during generation or afterwards, ensembling their individual hypothesis sets. We benchmark these methods on different machine translation tasks and show that modeling weight uncertainty can effectively improve MBR. We evaluate the effects of using different posterior distributions. More complex distributions provide stronger performance improvements. Perhaps related, prediction diversity is important for both standard MBR and when using the expected utility of MBR for selective prediction.

6 Limitations

One key limitation of our work is that the methods we use introduce an overhead either at inference time, training time, or both. For example, learning multiple distributions for a mixture-of-Gaussian posterior results in the training time being increased by a factor proportional to the number of distributions that are being learned. In a similar vein, using multiple models during decoding requires as many additional forward passes as there are models. If their predictions are kept for sequence-level averaging, the inference cost of MBR is also increased.

Another limitation is the scale of our models. While we experiment mostly with smaller transformers, current LLMs are often many magnitudes larger and it would be interesting to see how our approaches can improve such large models. In a similar vein, we only evaluate on the task of machine translation, because initial experiments suggested that other tasks did not benefit from MBR at all, either with or without the incorporation of weight uncertainty.

Finally, our evaluation also only covers translation between German and English language and therefore has limited coverage of language families.

7 Ethics and Broader Impact Statement

Our work uses probabilistic language models to generate machine translations. Such models can produce outputs that are, among others, harmful, toxic, and hallucinated and our methods can not guarantee that such outputs are not generated. However, we aim to improve the robustness of language generation methods and, therefore, aim to alleviate these issues. Therefore, we believe there to be no direct ethical concern in our work.

References

Taiga Abe, E. Kelly Buchanan, Geoff Pleiss, and John Patrick Cunningham. 2022. The best deep ensembles sacrifice predictive diversity. In *I Can't Believe It's Not Better Workshop: Understanding Deep Learning Through Empirical Falsification*.

Pierre Alquier. 2021. User-friendly introduction to pac-bayes bounds. *arXiv preprint arXiv:2110.11216*.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ale s Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 2–17, Lake Tahoe, California.

Julius Cheng and Andreas Vlachos. 2023. Faster minimum Bayes risk decoding with confidence-based pruning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12473–12480, Singapore. Association for Computational Linguistics.

Eldan Cohen and Christopher Beck. 2019. Empirical analysis of beam search performance degradation in neural sequence models. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1290–1299. PMLR.

Morris H DeGroot. 2005. *Optimal statistical decisions*. John Wiley & Sons.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2022. Sampling-based approximations to minimum Bayes risk decoding for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10978–10993, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. 2019. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30.

Jesús González-Rubio, Alfons Juan, and Francisco Casacuberta. 2011. Minimum Bayes-risk system combination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1268–1277, Portland, Oregon, USA. Association for Computational Linguistics.

Alex Graves. 2011. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24.

John Hewitt, Christopher Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Anas Himmi, Guillaume Staerman, Marine Picot, Pierre Colombo, and Nuno M. Guerreiro. 2024. Enhanced hallucination detection in neural machine translation through simple detector aggregation. *arXiv preprint arXiv:2402.13331*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. 2017. Snapshot ensembles: Train 1, get m for free. In *International Conference on Learning Representations*.

Hayato Kobayashi. 2018. Frustratingly easy model ensemble for abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4165–4176, Brussels, Belgium. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Shankar Kumar and William Byrne. 2002. Minimum Bayes-risk word alignments of bilingual texts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 140–147. Association for Computational Linguistics.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Kai Lion, Gregor Bachmann, Lorenzo Noci, and Thomas Hofmann. 2023. How good is a single basin? In *UniReps: the First Workshop on Unifying Representations in Neural Models*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. 2019. A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, 32.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Potsawee Manakul, Yassir Fathullah, Adian Liusie, Vyas Raina, Vatsal Raina, and Mark Gales. 2023. CUED at ProbSum 2023: Hierarchical ensemble of summarization models. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 516–523, Toronto, Canada. Association for Computational Linguistics.

Andres Masegosa. 2020. Learning under model misspecification: Applications to variational and ensemble methods. *Advances in Neural Information Processing Systems*, 33:5479–5491.

Thomas Möllenhoff and Mohammad Emtiyaz Khan. 2023. SAM as an optimal relaxation of bayes. In *The Eleventh International Conference on Learning Representations*.

Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. 2019. Practical deep learning with bayesian principles. *Advances in neural information processing systems*, 32.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. 2023. Out-of-distribution detection and selective generation for conditional language models. In *International Conference on Learning Representations*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. Mixture models for diverse machine translation: Tricks of the trade. In *International conference on machine learning*.

Yuesong Shen, Nico Daheim, Bai Cong, Peter Nickl, Gian Maria Marconi, Clement Bazan, Rio Yokota, Iryna Gurevych, Daniel Cremers, Mohammad Emtiyaz Khan, and Thomas Moellenhoff. 2024. Variational learning is effective for large deep networks. *ICML*.

Noah A. Smith. 2011. *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3356–3362, Hong Kong, China. Association for Computational Linguistics.

Jannis Vamvas and Rico Sennrich. 2024. Linear-time minimum bayes risk decoding with reference aggregation. *arXiv preprint arXiv:2402.04251*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Danny Wood, Tingting Mu, Andrew M Webb, Henry WJ Reeve, Mikel Lujan, and Gavin Brown. 2023. A unified theory of diversity in ensemble learning. *Journal of Machine Learning Research*, 24(359):1–49.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Adam X. Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. 2024. Bayesian low-rank adaptation for large language models. In *The Twelfth International Conference on Learning Representations*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Relationship of Token- and Sequence-level Averaging

$$\log \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim q} p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x}) = \log \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim q} \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(y_t \mid \mathbf{y}_{<t}, \mathbf{x}) \tag{17}$$

$$\geq \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim q} \log \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(y_t \mid \mathbf{y}_{<t}, \mathbf{x}) \tag{18}$$

$$= \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim q} \sum_{t=1}^{T} \log p_{\boldsymbol{\theta}}(y_t \mid \mathbf{y}_{<t}, \mathbf{x}) \tag{19}$$

$$= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim q} \log p_{\boldsymbol{\theta}}(y_t \mid \mathbf{y}_{<t}, \mathbf{x}) \tag{20}$$

The second step follows from Jensen's inequality (log is strictly concave), then we use linearity of expectation.

B Experimental Details

Datasets Our usage of the WMT14 English-to-German translation tasks (Bojar et al., 2014) follows the set-up from (Vaswani et al., 2017) but augments the training data by the *news-commentary-v12* data from WMT17 (Bojar et al., 2017). In total, we train on ca. 3.9M paired examples. We also use a validation set during training in order to pick checkpoints which consists of ca 39.4K examples. We use the original *newstest2014* data which consists of 3,003 examples.

We also use the IWSLT14 German-to-English translation task (Cettolo et al., 2014) which consists of ca 160K training examples. The validation set consists of ca. 7.3K examples. The test set consists of 6,750K examples.

All data usages can be reproduced by following the instructions from the Fairseq repository under `https://github.com/facebookresearch/fairseq/tree/main/examples/translation` and will be published along our code.

Models All models follow the Transformer$_{\text{base}}$ architecture from Vaswani et al. (2017) and consist of an encoder-decoder Transformer with 6 encoder and 6 decoder layers. The models use a vocabulary of Byte-Pair-Encoding tokens (Sennrich et al., 2016). The WMT model has an input vocabulary size of 40480 and an output vocabulary size of 42720. Altogether, the model has $86,736,896$ parameters. The IWSLT model has an input vocabulary size of 8848 and an output vocabulary size of 6632 for in total $39,469,056$ parameters. The input and output embedding parameters of the decoder are shared.

Training We train all models from scratch using the fairseq library (Ott et al., 2019) which we extend for variational learning and a Bayesian interpretation of neural networks. Fairseq is licensed under MIT license[6] which permits our form of usage. We will release our code publicly in the future for further research in a software repository under Apache License 2.0[7]. We train all models with the IVON optimizer (Shen et al., 2024) and place a diagonal Gaussian posterior over neural networks. We use IVON with a isotropic Gaussian prior and initialize all entries of the Hessian with 0.1. We use an effective sample size of $1 \cdot 10^{-8}$, a small weight-decay of 0.0001, and a learning rate of 0.1. We set $\beta_1 = 0.9$ and $\beta_2 = 0.9999$. All models are trained with a batch size of 32 and we use 2 MC samples from the posterior during training. While this roughly doubles training time when compared to AdamW, it is also possible to use just 1 MC sample and arrive at similar results. We train the models until performance in terms of BLEU has not improved for at least 3 epochs and then stop. Afterwards, we use the distribution that has best validation performance.

For the snapshot-like approach, we add 3 randomly-sampled distributions that were trained with at least 10 epochs to the best-performing one. For the multi-IVON mixture-of-Gaussian approach we always use the best performing distribution for runs with different random seeds. In all experiments we sample from the posterior "as-is" and only vary the temperature by reducing the effective sample size when explicitly mentioned.

All models are trained on a single GPU which is an NVIDIA GPU with either 40GB, 32GB or 24GB GPU memory. Training takes around 1-2 hours for the IWSLT14 models and 1-2 days for the WMT models.

Method	Sampling BLEU	COMET	LaBSE	Beam Search BLEU	COMET	LaBSE
Best Mean	33.69	74.71	85.33	35.90	76.65	86.44
Sequence-level						
Unimodal	34.65	75.20	85.68	35.99	76.67	86.45
Mixture	**35.42**	**75.84**	**86.07**	37.42	77.69	86.97
Token-level						
Unimodal	33.62	74.68	85.39	35.94	76.66	86.45
Mixture	34.61	75.06	85.88	**38.56**	**78.31**	**87.34**

Table 2: Measuring hallucinations with LaBSE (higher is better) on IWSLT14 with hypothesis set of size 20 shows similar trends as quality estimation metrics: incorporating weight-uncertainty can reduce hallucinations, especially when a complex posterior is used.

Method	Sampling BLEU	COMET	LaBSE	Beam Search BLEU	COMET	LaBSE
Best Mean	23.25	67.74	86.74	26.55	73.25	88.60
Sequence-level						
Unimodal	24.08	68.90	87.09	26.60	73.30	88.62
Mixture	**24.10**	**69.20**	**87.28**	26.81	73.70	88.86
Token-level						
Unimodal	23.38	67.77	86.70	26.57	73.26	88.60
Mixture	23.37	67.74	86.62	**27.19**	**74.04**	**88.87**

Table 3: Measuring hallucinations with LaBSE (higher is better) on WMT14 with hypothesis set of size 20 shows similar trends as quality estimation metrics: incorporating weight-uncertainty can reduce hallucinations, especially when a complex posterior is used.

C Additional Results

C.1 Incorporating Weight Uncertainty Reduces Hallucinations

In this section, we additionally measure the amount of hallucinations generated by various strategies. We use LaBSE (Feng et al., 2022) to evaluate hallucinations which has shown strong correlation with human judgements (Himmi et al., 2024). The hallucination score is calculated by the cosine similarity of the LaBSE embedding of input and output, respectively. Note that a higher score means less hallucinations. We use the checkpoint from Sentence Transformers (Reimers and Gurevych, 2019) which is available on the huggingface (Wolf et al., 2020) hub[8]. Results are shown in Tab. 2 for IWSLT14 and in Tab. 3 for WMT14. We find that the overall trends follow the same pattern as observed in terms of quality estimation metrics. Using weight uncertainty improves over a single model, especially when a multimodal posterior is used. Similarly, for a hypothesis set size of 20, we find that sequence-level combination outperforms token-level combination when using ancestral sampling but not when using beam search.

C.2 Qualitative Examples

Tab. 4 shows qualitative examples that were generated with MBR and a hypothesis set of size 20 created with beam search for IWSLT14de-en and WMT14en-de. We compare the source and target translations to a translation produced by the best mean, as well as sequence- and token-level model combination of models from a multimodal posterior. We find in general that the amount of hallucinations is reduced when comparing the single model baseline to a model combination, for example the additional "i'll tell you" in the first example of IWSLT14. Furthermore, grammar and translation mistakes are reduced. This highlights how modeling weight uncertainty can effectively improve MBR.

[6] https://github.com/facebookresearch/fairseq/blob/main/LICENSE

[7] https://www.apache.org/licenses/LICENSE-2.0

[8] https://huggingface.co/sentence-transformers/LaBSE

IWLST14de-en

Source	wenn sie jetzt mal ein stückchen weiter denken .
Target	now let's take that a step further .
Best Mean	now , if you think a little farther , i'll tell you
Sequence-level	now , if you think a little bit further
Token-level	now , if you keep thinking a little bit further
Source	passiert ist das maschinenzeitalter und passiert ist das buch und passiert ist die ist die fotolitografie
Target	then the machine age happened and the book and then we had photolithography
Best Mean	and the machine age has happened , and what's happened is the book , and that's what's happened
Sequence-level	what happened is the machine age , and what happened is the book , and what happened is the photolitography
Token-level	what's happened is the machine age , and what's happened is the book , and what's happened is the photolitography
Source	und das ist halt nicht im hobbybereich , aber es ist eine wirklich coole technologie , die auch sehr viel kann
Target	and that's not within the range of amateurs , but it's a really cool technology , which is capable of a lot
Best Mean	and that's not in the hobbyist , but it's a really cool technology , it can do a lot
Sequence-level	and that's not in the hobbysphere , but it's a really cool technology that can do a lot
Token-level	and that's not in the hobbyist , but it's a really cool technology that can do a lot
Source	also es war eine neugier und ich wollte irgendwie näher ran an die materie , so
Target	so it was curiosity and i somehow wanted to get closer to the material , that was it
Best Mean	so it was a curiosity , and i wanted to get closer to the subject , like that
Sequence-level	so it was a curiosity , and i wanted to sort of get closer to the material , like that
Token-level	so it was a curiosity , and i wanted to sort of get closer to the matter , like that
Source	also für ihn waren sie nur bauklötze
Target	so to him , they were just blocks
Best Mean	so for him , they were just timbers
Sequence-level	so for him , they were just building blocks
Token-level	so for him , they were just building blocks

WMT14en-de

Source	Together with an accomplice , who procured the women , he brought them to various brothels in the south-west
Target	Zusammen mit einem Kollegen , der die Frauen vermittelte , brachte er sie in verschiedene Bordelle im Südwesten
Best Mean	Zusammen mit einem Komplizen , der die Frauen beschaffte , brachte er sie zu verschiedenen Brüdern im Südwesten
Sequence-level	Zusammen mit einem Komplizen , der die Frauen beschaffte , brachte er sie in verschiedene Bordelle im Südwesten
Token-level	Zusammen mit einem Komplizen , der die Frauen beschaffte , brachte er sie in verschiedene Bordelle im Südwesten
Source	Founding Chairman Henne reported in depth on the work of the organisation
Target	Der erste Vorsitzende Henne berichtete ausführlich von der Arbeit des Vereins
Best Mean	Gründervorsitzender Henne hat ausführlich über die Arbeit der Organisation berichtet
Sequence-level	Gründervorsitzender Henne berichtete ausführlich über die Arbeit der Organisation
Token-level	Der Gründungsvorsitzende Henne berichtete ausführlich über die Arbeit der Organisation
Source	They then drove away
Target	Anschließend fuhren sie davon
Best Mean	Sie verjagten sich
Sequence-level	Dann fuhren sie weg
Token-level	Dann sind sie weg
Source	They earn as much as teachers and can make a good living
Target	Sie verdienten so viel wie Lehrer und könnten gut davon leben
Best Mean	Sie verdienen so viel wie Lehrer und können ein gutes Leben
Sequence-level	Sie verdienen so viel wie Lehrer und können gut leben
Token-level	Sie verdienen so viel wie Lehrer und können ein gutes Leben führen
Source	The fact that the dog was spotted is unbelievable
Target	Die Tatsache , dass der Hund entdeckt wurde , ist unglaublich
Best Mean	Dass der Hund aufgedeckt wurde , ist unglaublich
Sequence-level	Die Tatsache , dass der Hund entdeckt wurde , ist unglaublich
Token-level	Die Tatsache , dass der Hund entdeckt wurde , ist unglaublich
Source	There are lots of well-publicized theories about the causes of precocious puberty
Target	Es gibt viele umfassend publizierte Theorien über die Ursachen frühzeitiger Pubertät
Best Mean	Es gibt viele gut publizisierte Theorien über die Ursachen von bösartiger Pubertät
Sequence-level	Es gibt viele gut publizierte Theorien über die Ursachen von bösartiger Pubertät
Token-level	Es gibt eine Menge gut publizierter Theorien über die Ursachen der bösartigen Pubertät

Table 4: Qualitative examples of outputs generated by the best learned mean, as well as sequence- and token-level model combination using beam search and a multimodal posterior.