Efficient Large-Scale Autoregressive Sequence Models

Anonymous EMNLP submission

Abstract

Large deep-learning based autoregressive models have shown state-of-the-art performance in many sequence-to-sequence tasks including neural machine translation. Deep Ensembles of these systems yield performance gains over individual models and enable uncertainty estimates, including knowledge uncertainty, on the predictions to be derived. The challenge with these ensembles is that training costs, memory requirements and inference costs all scale linearly with the number of members of the ensemble. In this work we explore how to train autoregressive models efficiently, while in a single forward pass, maintaining the ability to make robust uncertainty estimates. The approach combines efficient ensemble generation and distribution distillation techniques. This combination dramatically reduces the computational and memory costs compared to Deep Ensembles. Experiments on WMT16 and WMT20 show that single models trained using the proposed scheme can reach or outperform Deep Ensembles while being much cheaper at training and inference time. Additionally, by extending existing distribution distillation techniques, a single model can be trained to consistently outperform a Deep Ensemble on out-of-distribution detection.

1 Introduction

Large autoregressive neural networks based on attention have emerged in the past few years as the most competitive approach to many (structured) sequence tasks, especially in translation (Bahdanau et al., 2015; Vaswani et al., 2017; Ott et al., 2018), and are increasingly being used in practice. Additionally, one of the most straightforward ways to improve system performance, and allow for uncertainty estimation which is vital for safety and robustness of deep systems, is to train several models that ideally make independent errors in order to form an ensemble (Perrone and Cooper, 1993; Opitz and Maclin, 1999; Dietterich, 2000; Lakshminarayanan et al., 2017). However, for large-scale tasks such as automatic speech recognition and neural machine translation where single neural networks can have hundreds of millions, or billions, of parameters, it becomes increasingly infeasible to train and deploy the resulting ensembles.

Ensembles come with several benefits including better predictive power by exploiting the prediction diversity among the ensemble members (Krogh and Vedelsby, 1994). Furthermore, any committee of models (Krogh and Vedelsby, 1994) have the ability to decompose total uncertainty (Depeweg et al., 2018; Gal and Ghahramani, 2016) into a sum of data uncertainty, representing the intrinsic noise in the data being modelled, and knowledge uncertainty, which refers to the level of ignorance about the most optimal model parameters (Hullermeier and Waegeman, 2021). While this is a useful property, there has been limited use of such systems in practice due to high computational and memory requirements that scale linearly with the number of members in the ensemble.

Owing to this limitation, an ensemble compression technique called Knowledge Distillation[1] has become popular (Hinton et al., 2014a). It addresses the common scenario that while a practitioner has access to significant resources during training, efficiency during inference time is crucial. The important insight is that although the knowledge represented in the parameters of a neural network ensemble can be highly cryptic and unfathomable, neural networks can be abtractly viewed as functions, that map "input vectors to output vectors" (Hinton et al., 2014a; Bucilua et al., 2016). Therefore, instead of compressing a collection of parameters directly, the goal is to train a single efficient "student" model to predict the average output of the "teacher" ensemble. This approach is not only applicable to static tasks such as image classification but also

[1] This can also be used to distil large individual models.

to structured sequence tasks, where autoregressive models are often utilised.

Summary of contributions: In this paper, we address the computational demands of training and deployment of large autoregressive systems with a focus on yielding robust uncertainty estimates. While reducing training cost and increasing inference speed will be addressed, it will be assumed that disk space is abundant. These computational savings will be achieved by: (1) modification of efficient ensemble generation methods; and (2) extending recent distribution distillation approaches to generate well performing student models able to estimate and decompose uncertainty with a single forward pass. Additionally, (3) the diversity of teacher ensembles and performance of students is empirically investigated in order to aid the design process. Specifically, we will initially evaluate techniques on the En-De WMT16 dataset. We show that single student models have the ability to outperform transformer-based ensembles in both BLEU performance out-of-distribution detection. Finally, we investigate if similar conclusions can be drawn by experimenting on bigger transformers and the significantly larger En-Ru WMT20 dataset.

2 Background and Related Work

In this section we review ensemble-based uncertainty estimation for structured prediction tasks, such as those encountered in natural language and speech processing. We then discuss how the limitations of ensemble approaches can be addressed using recently developed distillation techniques. Finally, we describe how such ensembles can be generated in practice.

2.1 Uncertainty Estimation

We adopt a Bayesian perspective on ensembles as this offers a flexible framework within which uncertainties have an information theoretic justification. Here the posterior $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$ is derived given some observed data, $\mathcal{D}$. Unfortunately, this form Bayesian inference is often intractable and cannot be applied to large neural networks. Alternatively an approximation $\mathtt{q}(\boldsymbol{\theta})$ to the posterior $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$, is used. Samples from this distribution can then be drawn to generate an ensemble. Ensemble generation methods will be covered in more detail in a following subsection.

Take an ensemble $\{\mathtt{P}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$ sampled from an approximate posterior $\mathtt{q}(\boldsymbol{\theta})$ where each model maps a variable-length input of discrete elements $\boldsymbol{x} \in \mathcal{X}$ into a sequence of discrete elements $\boldsymbol{y} \in \mathcal{Y}$. The predictive distribution is obtained according to:

$$\mathtt{P}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}) = \mathbb{E}_{\mathtt{q}(\boldsymbol{\theta})}\left[\mathtt{P}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})\right] \tag{1}$$

From this predictive distribution, a measure of total uncertainty can be estimated using the entropy:

$$\mathcal{H}\left[\mathtt{P}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})\right] = \mathbb{E}_{\mathtt{P}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})}\left[\ln \frac{1}{\mathtt{P}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})}\right] \tag{2}$$

Furthermore, a measure of disagreement between models, also referred to as *knowledge* or *epistemic* uncertainty, can be estimated by using mutual information $\mathcal{I}$ between $\boldsymbol{y}$ and $\boldsymbol{\theta}$:

$$\mathcal{I}\left[\boldsymbol{y}, \boldsymbol{\theta}|\boldsymbol{x}, \mathcal{D}\right] = \mathbb{E}_{\mathtt{q}(\boldsymbol{\theta})}\left[\mathtt{KL}\big(\mathtt{P}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) \,\big\|\, \mathtt{P}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})\big)\right] \tag{3}$$

This estimate can also be decomposed into a measure of total and and data (*aleatoric*) uncertainty, as mentioned in Malinin and Gales (2021). There are also many other options for the measure of epistemic uncertainty such as expected pairwise KL-divergence or reverse mutual information, however, for the sake of simplicity we restrict our focus to the already mentioned eq. (2) and (3) since these represent uncertainties of differing nature.

2.1.1 Approximations

The measures of uncertainty provided above implicitly assume that one can enumerate all possible variable-length output sequences, and obtain model predictions for each and every one. This is evidently intractable; in practice, autoregressive models are used to factorize the distribution over $\boldsymbol{y}$ into a product of conditionals over a finite number of classes, referred to as vocabulary:

$$\mathtt{P}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) = \prod_{l=1}^{L} \mathtt{P}(y_l|\boldsymbol{y}_{<l}, \boldsymbol{x}, \boldsymbol{\theta}) \tag{4}$$

This equation illustrates the key mechanism behind autoregressive models, in which each token y_l is modelled based on conditioning on some back-history $\boldsymbol{y}_{<l} = \{y_1, y_2, ..., y_{l-1}\}$, and has been proven successful in many fields such as neural machine translation (Bahdanau et al., 2015; Vaswani et al., 2017) and end-to-end speech processing (Mohamed et al., 2019). However, autoregressive models come with high inference cost, and are often used in combination with n-best list decoding

schemes disallowing the direct estimation of uncertainties in eq. (2) and (3).

Consider a beam of output sequences $\mathcal{B} = \{\boldsymbol{y}^{(b)}\}_{b=1}^{B}$ generated by a model $\mathtt{P}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$. The work of Malinin and Gales (2021) provides several approximations for estimating uncertainty based on such a beam. Since several approximations have been provided, we choose to focus on the conditional decomposition of uncertainties which have been hypothesised to be more stable:

$$\hat{\mathcal{H}}_{C}^{(B)}\left[\mathtt{P}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})\right] = \sum_{b,l} \frac{w_b}{L^{(b)}} \mathcal{H}\left[\mathtt{P}(y_l|\boldsymbol{y}_{<l}^{(b)}, \boldsymbol{x}, \mathcal{D})\right]$$

$$\hat{\mathcal{I}}_{C}^{(B)}\left[\boldsymbol{y}, \boldsymbol{\theta}|\boldsymbol{x}, \mathcal{D}\right] = \sum_{b,l} \frac{w_b}{L^{(b)}} \mathcal{I}\left[y_l, \boldsymbol{\theta}|\boldsymbol{y}_{<l}^{(b)}, \boldsymbol{x}, \mathcal{D}\right]$$

where the uncertainties have both been length-normalised and importance weighted according to:

$$w_b = \frac{\exp \frac{1}{T} \ln \mathtt{P}(\boldsymbol{y}^{(b)}|\boldsymbol{x}, \mathcal{D})}{\sum_{k=1}^{B} \exp \frac{1}{T} \ln \mathtt{P}(\boldsymbol{y}^{(k)}|\boldsymbol{x}, \mathcal{D})} \quad (5)$$

Length normalisation is vital in allowing comparisons of entropies of variable-length outputs while the importance weighting is needed to adjust the uncertainty associated with $\boldsymbol{y}^{(b)} \in \mathcal{B}$ according to its probability. For an in depth review of approximations and alternative uncertainties, see Malinin and Gales (2021).

2.2 Knowledge Distillation

Ensembles $\{\mathtt{P}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$ sampled from some posterior can be computationally demanding. One approach to efficiently exploit the information of the ensemble is to use knowledge distillation (KD) to yield a single student model (Hinton et al., 2014a; Kim and Rush, 2016).

Given an input and reference output pair (teacher-forcing) $(\boldsymbol{x}, \boldsymbol{y}) \sim \tilde{\mathtt{p}}(\boldsymbol{x}, \boldsymbol{y})$, a standard model might be trained using negative log-likelihood (NLL):

$$\mathcal{L}_{\mathtt{NLL}}(\boldsymbol{\theta}) = \frac{1}{L} \sum_{l=1}^{L} -\ln \mathtt{P}(y_l|\boldsymbol{y}_{<l}, \boldsymbol{x}, \boldsymbol{\theta}) \quad (6)$$

Similarly, a student model with parameters $\boldsymbol{\phi}$ can be trained to emulate a teacher ensemble by additionally using the average categorical prediction $\boldsymbol{\pi}_l, \pi_{l,k} = \mathtt{P}(y_l = \omega_k|\boldsymbol{y}_{<l}, \boldsymbol{x}, \mathcal{D})$ as 'soft' labels:

$$\mathcal{L}_{\mathtt{KL}}(\boldsymbol{\phi}) = \frac{1}{L} \sum_{l=1}^{L} \mathtt{KL}\big(\boldsymbol{\pi}_l \,\big\|\, \mathtt{P}(y_l|\boldsymbol{y}_{<l}, \boldsymbol{x}, \boldsymbol{\phi})\big) \quad (7)$$

However in practice, one normally optimises a convex combination of NLL and KL-divergence $\mathcal{L}_{\mathtt{KD}}(\boldsymbol{\phi}) = \lambda\mathcal{L}_{\mathtt{NLL}}(\boldsymbol{\phi}) + (1-\lambda)\mathcal{L}_{\mathtt{KL}}(\boldsymbol{\phi})$ for added supervision and stability. The probability mass functions in the KL-divergence can also be temperature scaled by some T to improve optimisation (Hinton et al., 2014a). Note that this criteria is only considered for the teacher-forcing case, more sophisticated distillation approaches exist, by sampling $(\boldsymbol{x}, \boldsymbol{y})$ from alternative distributions, but are out of scope for this work, see Kim and Rush (2016); Wong et al. (2016); Lee et al. (2018); Malinin et al. (2017) for details.

2.2.1 Ensemble Distribution Distillation

While knowledge distillation can be successful in many cases, the resulting student will not be able to estimate epistemic or knowledge uncertainty, since it only modelled the average ensemble prediction. To avoid this issue, Malinin et al. (2020) considers the task of distilling the distribution of ensemble predictions onto a single student. This allows the student to retain both predictive performance and information about diversity.

To explain the mechanics behind distribution distillation, consider modelling a distribution over ensemble predictions:

$$\{\boldsymbol{\pi}_l^{(m)}\}_{m=1}^{M},\ \pi_{l,k}^{(m)} = \mathtt{P}(y_l = \omega_k|\boldsymbol{y}_{<l}, \boldsymbol{x}, \boldsymbol{\theta}^{(m)})$$

An autoregressive student can be used for prediction based on the parameters $\boldsymbol{\alpha}_l$ of a Dirichlet distribution $\mathtt{Dir}(\boldsymbol{\pi}_l|\boldsymbol{\alpha}_l) = \mathtt{p}(\boldsymbol{\pi}_l|\boldsymbol{y}_{<l}, \boldsymbol{x}, \boldsymbol{\phi})$ (Fathullah et al., 2021). Since the Dirichlet is a prior for categorical distributions it is an ideal candidate for this task. The distribution distillation of such a model is achieved by optimising:

$$\mathcal{L}_{\mathtt{NLL}}^{\mathtt{DD}}(\boldsymbol{\phi}) = -\frac{1}{ML} \sum_{m,l} \ln \mathtt{Dir}(\boldsymbol{\pi}_l^{(m)}|\boldsymbol{\alpha}_l)$$

which is a straightforward application of negative log-likelihood. In this work we also consider generalisations to alternative distributions over categorical predictions.

2.3 Ensemble Generation

Up to this point, it has been assumed that an ensemble of models is available. This section reviews a number of methods for ensemble generation and how the weight-space can be explored, with reference to training and inference cost.

Two established and 'opposite' approaches for ensemble generation include Monte-Carlo Dropout (MCD) (Gal and Ghahramani, 2016) and Deep Ensembles (Lakshminarayanan et al., 2017). While both ensembles are equally expensive (in terms of inference) given the same number of draws, they are very different in terms training cost and diversity. MCD ensembles are cheap to train but have limited diversity over the ensemble members. Deep Ensembles have higher training costs, scaling with number of members, but yield higher ensemble diversity and expressiveness. Depending on computational and memory constraints one might favour one over the other. A third class of ensemble generation methods was proposed in Xie et al. (2013): storing checkpoint parameters throughout training results in a so called temporal ensemble. Furthermore, Huang et al. (2017) utilised cyclic learning rates to encourage the process to traverse and explore multiple minima in the weight-space achieving a more diverse temporal ensemble, referred to as a Snapshot Ensemble. The training cost of this form of ensemble varies depending on the cyclic learning rate schedule but can be set to operate at the equivalent of training a single model.

Alternative generation methods exist such as MIMO (Havasi et al., 2021) based on implicit subnetwork generation within a neural network. These exploit the fact that many deep networks are overparametrised and have excessive capacity. However, this method not only suffers from high training cost, it has no trivial extension to handling variable-length data. SWAG (Maddox et al., 2019) is another method based on approximating the parameter checkpoints generated by SGD (with small learning rate) using a low-rank Gaussian distribution. This method, however, suffers from high memory cost, even with low-rank approximation, and low inference speed since it requires sampling large networks.

In this work we focus on two types of transformers based ensembles: Deep and Snapshot. Standard Deep Ensembles can be directly applied to NMT. For Snapshot Ensembles the learning rate schedule needs to be modified to be appropriate for transformers. Snapshot Ensembles are used in this work because they can be trained using similar resource requirements as training a single model while maintaining diversity and performance. Note, Snapshot Ensembles were critiqued in Wen et al. (2020) for their possible incompatibility with transformer-based models. Here we show that it is feasible to use cyclic learning rates with transformers.

3 Efficient Exploration/Exploitation

Before examining the proposed approach, we first touch on an interesting aspect of *stochastic weight averaging* and its Gaussian generalisation (Izmailov et al., 2018; Maddox et al., 2019). These approaches are based on modelling the parameter/weight-space by traversing the weight-space during training and generating a collection of checkpoints $\{\boldsymbol{\theta}^{(m)}\}_{m=1}^{M}$. In general neural network weights are not interpretable and cannot normally be compared directly to each other. Thus at the core of our approach will be to use checkpoints obtained by efficiently traversing the weight-space during training and storing the resulting predictions. These categorical predictions are directly related to each other. The two step approach that exploits this observation, *stochastic prediction averaging* (SPA), is described below.

Ensemble Generation: At the *exploration* stage we will make use of the ideas behind temporal ensembles. After training an initial model to convergence, we modify the learning rate (usually inverse square root decay for transformers) to a cyclic alternative and store the predictions $\{(\boldsymbol{x}, \boldsymbol{\pi}_{1:L}^{(m)})\}_{m=1}^{M}$ made by the checkpoints. The settings associated with this learning rate will determine the extent to which the ensemble members will have explored the weight space and associated local minima.

Ensemble Distillation: Once the predictions of the temporal ensemble have been saved, we switch to (distribution) distillation based training, the *exploitation* stage. Given a student able to model the space of categoricals $\mathrm{p}(\boldsymbol{\pi}_l|\boldsymbol{y}_{<l}, \boldsymbol{x}, \boldsymbol{\phi})$ it can be trained on the stored ensemble predictions:

$$\mathcal{L}_{\texttt{NLL}}^{\texttt{DD}}(\boldsymbol{\phi}) = -\frac{1}{ML}\sum_{m,l} \ln \mathrm{p}\left(\boldsymbol{\pi}_l^{(m)}|\boldsymbol{y}_{<l}, \boldsymbol{x}, \boldsymbol{\phi}\right)$$

As previously mentioned in section 2.2, a combination of losses is often used in practice:

$$\lambda\mathcal{L}_{\texttt{NLL}}(\boldsymbol{\phi}) + (1-\lambda)(\mathcal{L}_{\texttt{KL}}(\boldsymbol{\phi}) + \mu\mathcal{L}_{\texttt{NLL}}^{\texttt{DD}}(\boldsymbol{\phi})) \quad (8)$$

One option for the student is to use Dirichlet distributions. Additionally we will examine students with diagonal Gaussian and Laplace distributions in the (pre-softmax) logit space. These distributions require an additional output head, in order to predict both the mean and diagonal covariance

of the logits. The rationale for these more complex models is that additional flexibility in the student distribution enables more knowledge to be exploited from the teacher for both decoding and accurate uncertainty modelling. Note, the softmax of the mean is used as an approximation to the predictive distribution as analytic expectations over logits are intractable, in depth discussion of these distributions and implementation will be covered in Appendix B.

3.1 Computational Cost

As described above, there are multiple costs associated with stochastic prediction averaging. **Firstly**, one needs to explore the weight-space by training a converged model using cyclic learning rates (LRs) to generate high quality temporal checkpoints. In this paper, we focus on a case where the cyclic regime is run for 30-50% of the cost (measured in terms of GPU-hours) of training the original model and we always choose to store $M = 5$ checkpoints (5 cycles and checkpoint whenever the learning rate hits its minimum value). **Secondly**, we distribution distil the system utilising an additional 50-70% of the original cost, resulting in a student that requires similar compute to training two models. We also initialise the parameters of the student with the final checkpoint in order to save additional compute. **Thirdly**, there is some memory cost associated with storing the predictions, since one often has to work with large vocabularies. In this work we will neglect the cost of disk space, but will investigate the case where only top-k elements of temporal ensemble are saved, similar to Tan et al. (2019).

The resulting model should, with a single forward pass, be able to estimate and decompose uncertainty whilst maintaining good decoding performance. Compared to an ensemble, this will be a significant reduction of the number of forward passes required.

3.2 Drawbacks and Advantages

In stochastic prediction averaging it is necessary to determine the learning rate (LR) schedule for both the exploration and exploitation stage once an initial converged model has been obtained. In our work we found that the cyclic learning schedule is flexible (for transformers) and the hyperparameters can be selected depending on the computational restrictions of the user. For example, we restricted experiments to triangular cyclic schedules with 2-3 epoch cycle length. The peak cyclic LR was set to a value larger than the original model's peak LR. The minimum value was set to a value smaller than the original model's final LR. For the exploitation stage, we chose a schedule that mimics the original schedule closely.

There are many options at the exploration stage of the proposed scheme, this is one of the strengths of this approach. If a practitioner has no limitations during training time, one could simply replace the temporal generation method with the predictions made by a, for example, Hyper Deep Ensemble (Wenzel et al., 2020). Alternatively, one could easily modify the number cycles and learning rate to encourage exploration of the weight space and generate diverse but high quality predictions. We will also show empirically that although temporal ensembles are less diverse than Deep Ensembles, this drawback doesn't have a significant impact and in some cases is actually beneficial to the SPA student performance.

4 Experimental Evaluation

This section reports on performance of base transformers trained on the smaller En-De WMT16 dataset consisting of 4.5 million sentence pairs. We use newstest13/14 as development and evaluation sets. Additionally, we investigate out-of-domain detection on the publically available Khresmoi-Summary (Khresmoi) (Dušek et al., 2017), MTNT (Michel and Neubig, 2018) and Kyoto Free Translation Task (KFTT) (Neubig, 2011) datasets. These datasets relate to medical articles, Reddit based noisy text and specialised Wikipedia articles, respectively. Furthermore, we apply insights from training these systems to "big" transformers trained on the larger En-Ru WMT20 dataset made of about 58 million pairs post processing. In this case we use newstest19 as development data and evaluate on newstest20; out-of-domain detection used the same out-of-domain datasets as above.

Text Processing: Data is cleaned and tokenized using Moses[2]. For WMT16, a shared dictionary is trained using Byte Pair Encoding (BPE) with 32,000 merge operations (Sennrich et al., 2016). Similarly, for WMT20 we learn disjoint dictionaries using BPE with 40,000 merge operations.

Metrics: System performance will be evaluated using corpus-level BLEU (Post, 2018). Furthermore, to measure diversity of models, indepen-

[2] https://github.com/moses-smt/mosesdecoder

Table 1: BLEU and CrossBLEU on newstest14 (± 2 std). The **top row of each block** represents the **teacher** model/ensemble that remaining models were distilled from. ***Standard** can be seen as a Deep Ensemble with identical members. Table also includes number of parameters and relative training time. Inference speed for all single models are similar.

Model	**Params** ↓	**Training Time** ↓	**BLEU** ↑	**CrossBLEU** ↓
Standard*	60.9M	1.0	25.85 ± 0.17	100.00 ± 0.00
Self distillation	60.9M	1.8	26.48 ± 0.15	
Self-distribution dist.	60.9M	1.7	25.89 ± 0.20	
Deep Ensemble	304.5M	5.0	26.72	62.96 ± 0.62
Distillation	60.9M	5.9	26.70 ± 0.26	
Snapshot Ensemble	304.5M	1.5	26.54 ± 0.16	73.06 ± 3.35
SPA (Categorical)	60.9M	1.9	**27.02** ± 0.19	
SPA (Dirichlet)	60.9M	2.0	**26.96** ± 0.06	
SPA (Gaussian)	77.8M	1.9	**26.90** ± 0.28	
SPA (Laplace)	77.8M	1.9	**27.08** ± 0.20	

dent of level of confidence (temperature), we use corpus-level BLEU between outputs (referred to as CrossBLEU). For detection, we use the ubiquitous threshold independent AUROC metric (Manning and Schütze, 1999), with baseline random detection corresponding to an AUROC of 50%.

Models: The individual transformer models are trained using an inverse square root with warmup. A Deep Ensemble is formed by taking $M = 5$ such models. The last checkpoint of standard trained model is used in conjunction with a cyclic learning rate to generate a temporal ensemble of $M = 5$ members, the Snapshot Ensemble. The level of exploration in Snapshot Ensembles is generally lower than its Deep equivalent due to being constrained by the same initial checkpoint. These models are then (distribution) distilled using students with either categorical predictions (standard ensemble distillation), or distribution distillation with Dirichlet, (logit) Gaussian or (logit) Laplace outputs. The hyperparameters (λ, μ) in eq. (8) and level of weight averaging of last few checkpoints, are tuned on the development set. In addition, we evaluate three baseline approaches: knowledge distillation (Hinton et al., 2014a); self distillation[3] (SD) (Zhang et al., 2019; Allen-Zhu and Li, 2021); and self-distribution distilled systems (S2D) (Fathullah and Gales, 2022). The latter is included because it is cheaper to train as it avoids the need for ensembles while able to estimate knowledge uncertainty in a single pass. All single model experiments were run 5 times. Details of learning rates, regularisation and various other hyperparameters are provided in Appendix A.

4.1 En-De WMT16 Results

Table 1 shows both efficiency and performance of a wide range of systems on newtest14 evaluation data. As expected the diversity of the Snapshot Ensemble is less than the Deep Ensemble equivalent, and as a consequence, the ensemble performance gain is less. Snapshot Ensembles can be used as

[3]This refers to distilling a model onto a student with identical architecture.

Table 2: CrossBLEU (lower score means higher diversity) and BLEU newstest14 performance (± 2 std).

Model	**Teacher BLEU** ↑	**Student BLEU** ↑	**Difference BLEU** ↑	**Diversity CrossBLEU** ↓
Standard	25.85 ± 0.17	26.48 ± 0.15	0.63 ± 0.14	100.00 ± 0.00
Limited Ensemble	26.07 ± 0.11	26.59 ± 0.11	0.52 ± 0.04	83.09 ± 0.61
Snapshot Ensemble	26.54 ± 0.16	27.02 ± 0.19	0.48 ± 0.25	73.06 ± 3.35
Deep Ensemble	26.72	26.70 ± 0.26	-0.02 ± 0.26	62.96 ± 0.62

Table 3: Out-of-distribution detection using the %AUROC ↑ metric (± 2 std).

Model	Khresmoi		MTNT		KFTT	
	TU	KU	TU	KU	TU	KU
Standard	47.5 ± 0.8	—	63.5 ± 1.3	—	30.6 ± 1.2	—
Self-distribution dist.	48.7 ± 2.8	54.4 ± 3.3	63.8 ± 1.9	58.9 ± 2.0	31.3 ± 2.3	31.4 ± 3.1
Deep Ensemble	48.0	61.9	64.5	63.7	30.1	44.0
Snapshot Ensemble	49.0 ± 0.6	62.6 ± 1.1	63.8 ± 1.2	63.1 ± 0.7	31.7 ± 0.9	47.4 ± 2.5
SPA (Categorical)	48.0 ± 1.4	—	64.6 ± 0.9	—	31.3 ± 0.5	—
SPA (Dirichlet)	49.6 ± 1.3	57.1 ± 1.4	65.1 ± 1.7	65.6 ± 2.0	31.0 ± 0.9	36.2 ± 1.4
SPA (Gaussian)	59.5 ± 1.1	71.7 ± 1.9	66.3 ± 1.6	64.0 ± 2.1	35.8 ± 1.2	44.0 ± 0.2
SPA (Laplace)	65.1 ± 1.8	**73.1** ± 1.7	65.1 ± 1.5	**66.8** ± 1.8	37.8 ± 0.2	**48.8** ± 1.4

the teacher ensemble to train SPA students, the complete training cost is similar to training two standard models (significantly cheaper than distillation and similar to self distillation). Notably, both self distillation and stochastic prediction averaging significantly outperform their teachers. However, unlike self distillation, all SPA student models are able to outperform the Deep Ensemble, the standard baseline, and requires both less training time and only a single forward pass during inference.

To investigate why certain students are able outperform their teachers, we train an additional ensemble on the WMT16 data specifically designed to have restricted diversity (higher CrossBLEU), see Table 2. There is a negative correlation between corpus-level diversity ("Diversity CrossBLEU") and the relative performance of the student to outperform its teacher ("Difference BLEU"). Higher level of exploration of the weight space for the ensemble, resulting in more diverse models, seems to make the exploitation of the resulting ensemble more challenging. Hence, if the aim is to generate a single, distilled, model for efficient inference limiting training ensemble diversity may be beneficial.

Next, we compare threshold independent out-of-domain detection performance of the baseline systems with SPA models. The comparison will be made using newstest14 as in-domain data and one of {Khresmoi, MTNT, KFTT} test datasets as out-of-domain. From Table 3, while self-distribution distillation can in one case (Khresmoi) produce useful knowledge uncertainties (KU), improving upon a standard model, it does not manage to do so consistently. Secondly, the Snapshot Ensemble is able to compete with the Deep equivalent while being more than 3 times cheaper to train. Thirdly, the (logit) Laplace SPA model is able to outperform the Deep Ensemble in all three detection splits, producing either similar or significantly better total (TU) and knowledge uncertainty estimates. Similarly, the (logit) Gaussian model trailing the Laplace equivalent in performance. However, the Dirichlet version is unable to match the quality of total or knowledge uncertainties, possibly because the logit based distributions have an additional head giving them more flexibility and expressiveness.

Table 4: BLEU and CrossBLEU on newstest20 (± 2 std). The **top row of each block** represents the **teacher** model/ensemble that remaining models were distilled from. Table also includes number of parameters and relative training time. Inference speed for all single models are similar.

Model	Params ↓	Training Time ↓	BLEU ↑	CrossBLEU ↓
Standard	271M	1.0	26.28 ± 0.34	100.00 ± 0.00
Self distillation	271M	1.8	26.56 ± 0.23	
Deep Ensemble	1.35B	5.0	26.81	68.67 ± 0.76
Snapshot Ensemble	1.35B	1.5	26.42 ± 0.23	76.62 ± 2.56
SPA (Categorical)	271M	2.1	26.73 ± 0.16	
SPA (Laplace)	320M	2.2	26.71 ± 0.18	

Table 5: Out-of-distribution detection using the %AUROC ↑ metric (± 2 std).

Model	Khresmoi		MTNT		KFTT	
	TU	KU	TU	KU	TU	KU
Standard	39.0 ± 0.7	—	69.6 ± 0.9	—	50.8 ± 1.1	—
Deep Ensemble	39.3	53.2	70.8	69.0	51.0	60.3
Snapshot Ensemble	40.8 ± 0.5	55.0 ± 0.8	70.1 ± 0.5	69.3 ± 0.9	51.1 ± 0.6	60.9 ± 1.4
SPA (Categorical)	40.4 ± 0.8	—	70.9 ± 1.0	—	50.9 ± 0.6	—
SPA (Laplace)	51.0 ± 0.9	**63.4** ± 1.2	**72.6** ± 0.8	70.2 ± 0.6	63.2 ± 1.0	**70.2** ± 1.1

4.2 En-Ru WMT20 Results

Next we train the best performing models on the En-Ru WMT20 dataset. Due to the much larger dataset and architecture (big transformer), we do not perform extensive hyperparameter optimisation and use instead the same training script as in Malinin et al. (2021) and the best found hyperparameters (λ, μ) from the En-De WMT16 experiments. The averaging of the last few checkpoints will be determined based on performance on development set (newstest19). Again we will mainly investigate models that can be trained in significantly less time than a Deep Ensemble (with the ensemble as the baseline to beat). The performance of the systems are shown in Table 4.

Interestingly, the SPA student models were all able to outperform their teacher Snapshot Ensemble, and reach close to Deep Ensemble performance, within approximately a standard deviation. This is despite the fact that they have far fewer parameters and training cost. Similarly, self distilled models were able to outperform their identically designed teacher but show an insignificantly smaller gain in performance than SPA students. Additionally, when it comes to uncertainty estimation, self distillation is inadequate due to the lacking ability to estimate knowledge uncertainty. These results follow very similar patterns to what was observed on the WMT16 dataset.

Additionally, we perform out-of-distribution detection but with newstest20 as the in-domain dataset, see Table 5. Results using newstest14 as the in-domain dataset is given in Appendix C.2. Again we observe the Laplace student model is able to outperform the Deep Ensemble in all cases, providing both better total and knowledge uncertainties. While the added flexibility of the Laplace model (having two heads) does explain why it outperforms the categorical equivalent, it does not explain its ability to outperform its teacher or the Deep Ensemble in detection. We explore this question in Appendix C.3 by scaling both Snapshot and Deep Ensembles to include many more members and investigate if detection ability increases dramatically. Alternatively, an explanation could be that logit based distributions are able to extract better 'dark knowledge' (Hinton et al., 2014b) specifically useful for estimating robust uncertainties. Furthermore, unlike in Table 3 where no model was able to beat a random detector on the KFTT detection, the larger WMT20 based models are able to differentiate between newstest20 and KFTT (using newstest14 as in-domain does not deflate results). While uncertainties from a standard individual model are not much different from simply outputting random values on KFTT, knowledge uncertainties, and especially those from Laplace SPA, are significantly better.

5 Conclusion

We have empirically investigated the balance between exploration of the weight space in the form of ensemble generation, and exploitation of such ensembles in the form of (distribution) distillation. In this process its been demonstrated how one can train single models, requiring only a single forward pass, to outperform ensembles in established machine translation tasks. Furthermore, it has also been shown that such distribution distilled models are able to consistently outperform their teachers and Deep Ensembles in detecting out-of-distribution inputs, vital for ensuring these models are used in a safe way in practice.

Another interesting observation is that when ensemble distillation approaches are used, the best performing ensemble does not necessarily yield the best performing distilled model. Limiting ensemble diversity can be beneficial. Additional experiments are required to confirm this phenomenon.

6 Limitations

In section 3.2 it was discussed how stochastic prediction averaging has a large hyperparameter space associated with it, but that the choice of parameters are flexible in many cases. In addition, an experimental limitation of this work is the evaluation of uncertainty quality purely based on out-of-distribution detection. Sequence based uncertainties can be used for a wide range of tasks such as Active Learning and Adversarial Detection, and there is no guarantee that SPA shows promise in those other tasks (Shen et al., 2017; Radmard et al., 2021; Ebrahimi et al., 2018). Finally, there is still a lack of understanding in what makes a well performing autoregressive out-of-distribution detector, in understanding the effect of architecture, vocabulary, beam size and calibration, discussed in slightly more detail in section C.2.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2021. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *arXiv*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. 2016. Model compression. In *International Conference on Knowledge Discovery and Data Mining*.

Stefan Depeweg, Jose Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. 2018. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Learning Representations*.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15.

Ondřej Dušek, Jan Hajič, Jaroslava Hlaváčová, Jindřich Libovický, Pavel Pecina, Aleš Tamchyna, and Zdeňka Urešová. 2017. Khresmoi summary translation test data 2.0. http://hdl.handle.net/11234/1-2122. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. *arXiv*.

Yassir Fathullah and Mark J. F. Gales. 2022. Self-distribution distillation: Efficient uncertainty estimation. In *arXiv*.

Yassir Fathullah, Mark J. F. Gales, and Andrey Malinin. 2021. Ensemble distillation approaches for grammatical error correction. In *International Conference on Acoustics, Speech, and Signal Processing*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*.

Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew M. Dai, and Dustin Tran. 2021. Training independent subnetworks for robust prediction. In *International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014a. Distilling the knowledge in a neural network. In *Conference on Neural Information Processing Systems Deep Learning Workshop*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014b. Dark knowledge. Invited talk at the BayLearn Bay Area Machine Learning Symposium.

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*.

Eyke Hullermeier and Willem Waegeman. 2021. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. In *Machine Learning*.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Conference on Empirical Methods in Natural Language Processing*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*.

Anders Krogh and Jesper Vedelsby. 1994. Neural network ensembles, cross validation, and active learning. In *Conference on Neural Information Processing Systems*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Conference on Neural Information Processing Systems*.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*.

Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson. 2019. A simple baseline for bayesian uncertainty in deep learning. In *Conference on Neural Information Processing Systems*.

Andrey Malinin, Neil Band, German Chesnokov, Yarin Gal, Mark J. F. Gales, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, Vyas Raina, Mariya Shmatova, Panos Tigas, and Boris Yangel. 2021. Shifts: A dataset of real distributional shift across multiple large-scale tasks. In *Conference on Neural Information Processing Systems Track on Datasets and Benchmarks*.

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

Andrey Malinin, Bruno Mlodozeniec, and Mark J. F. Gales. 2020. Ensemble distribution distillation. In *International Conference on Learning Representations*.

Andrey Malinin, Anton Ragni, Kate Knill, and Mark J. F. Gales. 2017. Incorporating uncertainty into deep learning for spoken language assessment. In *Association for Computational Linguistics*.

Chris Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. In *Conference on Empirical Methods in Natural Language Processing*.

Abdelrahman Mohamed, Dmytro Okhonko, and Luke Zettlemoyer. 2019. Transformers with convolutional context for asr. In *arXiv*.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

David Opitz and Richard Maclin. 1999. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198.

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 NAACL-HLT 2019: Demonstrations*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Michael P. Perrone and Leon N. Cooper. 1993. When networks disagree: Ensemble methods for hybrid neural networks. *Neural networks for speech and image processing*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Association for Computational Linguistics*.

Puria Radmard, Yassir Fathullah, and Aldo Lipani. 2021. Subsequence based deep active learning for named entity recognition. In *Association for Computational Linguistics*.

Max Ryabinin, Andrey Malinin, and Mark J. F. Gales. 2021. Scaling ensemble distribution distillation to many classes with proxy targets. In *Conference on Neural Information Processing Systems*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Association for Computational Linguistics*.

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*.

Leslie N. Smith. 2017. Cyclical learning rates for training neural networks. In *Winter Conference on Applications of Computer Vision*.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Conference on Neural Information Processing Systems*.

Yeming Wen, Dustin Tran, and Jimmy Ba. 2020. Batchensemble: An alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*.

Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. 2020. Hyperparameter ensembles for robustness and uncertainty quantification. In *Conference on Neural Information Processing Systems*.

Jeremy H. M. Wong, Mark J. F. Gales, and Yu Wang. 2016. Sequence-level knowledge distillation. In *IEEE/ACM Transactions on Audio Speech and Language Processing*.

Jingjing Xie, Bing Xu, and Zhang Chuang. 2013. Horizontal and vertical ensemble with deep representation for classification. In *International Conference on Machine Learning*.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *International Conference on Computer Vision*.

A Experimental Configuration

This section will provide detailed information about the datasets used for training, development, evaluation and detection. It will also give the exact training and various hyperparameters used for all models.

A.1 Datasets

We utilise two training sets WMT16/20, each with a pair of development and evaluation datasets based on newstest13/14 and newstest19/20. Additionally, we utilise three out-of-domain datasets for evaluating detection performance of a wide range of transformer models, see Table 6. As stated previously, all data is cleaned and tokenized using Moses. For WMT16, a shared dictionary is learned using BPE with 32,000 merge operations. On WMT20 we learn disjoint dictionaries using BPE with 40,000 merge operations. A consequence of the larger disjoint dictionary on WMT20 is the significantly lower number of unknown tokens in the OOD datasets.

A.2 En-De WMT16 Training

We use the base transformer from Vaswani et al. (2017) implemented in `fairseq` (Ott et al., 2019) and train it using 4 NVIDIA© A100 with an update frequency of 32. This is virtually equivalent to training on $4 \times 32 = 128$ GPUs. A per-gpu batch has a maximum of 3584 tokens. Models are optimized with Adam (Kingma and Ba, 2015) using β_1 = 0.9, β_2 = 0.98, and ϵ= 1e-8. We use a similar learning rate schedule to Vaswani et al. (2017), i.e., the learning rate increases linearly for 4000 warmup steps to a learning rate dependent on $d_{\texttt{model}}$ after which it is decayed proportionally to the inverse square root of the number of steps:

$$\eta = (\texttt{step} \cdot d_{\texttt{model}})^{-0.5} \min\left(1, \frac{\texttt{step}}{\texttt{warmup}}\right)^{1.5}$$

We use label smoothing with 0.1 weight for the uniform prior distribution over the vocabulary. The last 10 weight checkpoints were averaged. Training was stopped after 31 epochs corresponding to approximately a total of 18 GPU-hours. At inference, a beam of 4 with a length-penalty of 0.6 is used for all models.

SD/KD: Self and knowledge distilled models are first initialised by one of teacher members and then trained using the knowledge distillation loss $\mathcal{L}_{\texttt{KD}}$ provided in section 2.2 with $\lambda = 0.50$. The student was trained with a warmup of 1026 steps (3 epochs), from $\eta = 4.0 \times 10^{-4}$ to $\eta = 7.0 \times 10^{-4}$ after which it decays for a total of 24 epochs. A temperature of $T = 0.8$ was used in the KL-divergence loss as this was found to be mildly beneficial. All other hyperparameters match the standard case above.

S2D: The self-distribution distillation models are trained using the proxy Dirichlet approach (Fathullah and Gales, 2022; Ryabinin et al., 2021) together with multiplicative Gaussian noise (with

Table 6: Dataset information together with average source and target sentence sizes post tokenization and processing. The OOD testsets Khresmoi, MTNT and KFTT have two quoted numbers for each field as they were processed using either the En-De WMT16 or En-Ru WMT20 BPE based dictionaries. Additionally, only source side information is provided for OOD sets as these are only used for unsupervised uncertainty estimation.

Dataset	Type	Number of Sentences	Tokens per Sentence		Fraction of Unknown Tokens in Source
			Source	Target	
En-De WMT16	policy, news, web	4.5M	29.5	30.6	0.01%
En-De newstest13	news	3.0K	26.0	28.0	0.00%
En-De newstest14		3.0K	27.6	29.1	0.00%
En-Ru WMT20	policy, news, web	58.4M	27.8	27.5	0.00%
En-Ru newstest19	news	2.0K	29.9	33.4	0.00%
En-Ru newstest20		2.0K	30.9	32.5	0.00%
Khresmoi	medical	1.0K	30.9/30.3	—	0.78%/0.00%
MTNT	noisy reddit	1.4K	21.1/21.3	—	0.45%/0.06%
KFTT	encyclopedia	1.2K	35.4/35.2	—	1.46%/0.01%

standard deviation of $\sigma = 0.1$). We use a weight of $\mu = 2 \times 10^{-6}$ for the proxy Dirichlet KL-divergence loss. Similarly, all other hyperparameters match the standard case above.

Snapshot: The Snapshot Ensemble was generated by first starting from the last checkpoint of a standard trained transformer. At this point, a cyclic triangular learning rate schedule (Smith, 2017) was employed oscillating between the values of $\eta_{\texttt{min}} = 1.0 \times 10^{-4}$ and $\eta_{\texttt{max}} = 1.0 \times 10^{-3}$ with a period of 3 epochs. Note that the maximum learning rate in this cyclic phase is notably larger than the peak learning rate (7.0×10^{-4}) during standard training This setting was run for 15 epochs generating an ensemble with 5 members.

SPA: All stochastic prediction averaging models were trained using the same parameters as the distilled students but where however, trained for only 12 epochs. For each distribution, Dirichlet, Gaussian or Laplace, $\mu \in \{1.0, 2.0, 3.0\} \times 10^{-6}$ was tried and the best performing value on the development newstest13 was chosen, see eq. (8).

A.3 En-Ru WMT20 Training

We use the big transformer from Vaswani et al. (2017) again implemented in `fairseq` and trained using 4 NVIDIA© A100 with an update frequency of 32. A per-gpu batch has a maximum of 5120 tokens. Dropout was set to a value of 0.10 and weight decay to 0.0001. In this case we train the model for 20 epochs, corresponding to 53960 update steps and approximately 230 GPU-hours. The last 5 checkpoints were averaged leading to improved performance. At inference, a beam of 5 with a length-penalty of 1.0 is used for all models.

SD: Similar to the previous section, the self distillation student is initialised from its teacher but is trained using a learning rate warmup of 2698 steps (one epoch) from $\eta = 2.0 \times 10^{-4}$ to $\eta = 4.0 \times 10^{-4}$ after which it decays for a total of 16 epochs. The last 3 or 5 epochs are averaged, based on development newstest19 performance.

Snapshot: Based on the last checkpoint of a standard trained big transformer, a triangular cyclic learning rate is utilised, oscillating between $\eta = 5.0 \times 10^{-5}$ and $\eta = 5.0 \times 10^{-4}$ every 2 epochs for 10 epochs. This results in an ensemble with 5 members.

SPA: Following self distillation, SPA models are trained using the same parameters, but only for 12 epochs. The best found parameter μ (different for each student distribution) in the WMT16 experiments is to be used here. No hyperparameter search is performed at this stage.

B Novel Distribution Distillation

In section 3 it was described in general terms how maximum-likelihood training can be used to train students able to predict a distribution over categoricals. This allows the student to estimate both total and knowledge uncertainty in a single forward pass. In this section we describe in detail how this is achieved for both the novel Gaussian and Laplace student distributions in the logit space.

Given a collection of logits provdided by an ensemble $\{\boldsymbol{z}_{1:L}^{(m)}\}_{m=1}^{M}$ one first has to normalise these logits due to an invariance in the shift. While one could subtract the minimum or maximum logit, we choose the following:

$$\hat{\boldsymbol{z}}_l^{(m)} = \boldsymbol{z}_l^{(m)} - \mathbf{1}\left(\log \sum_k \exp z_{l,k}^{(m)}\right)$$

using the logsumexp trick. This choice is mainly based on its close relationship to the softmax function. Say a student then predicts a mean $\boldsymbol{\mu}_l$ and scale $\boldsymbol{\sigma}_l$ which describes a diagonal Gaussian in the logit space (given some back-history $\boldsymbol{y}_{<l}$ and source $\boldsymbol{x}$) the distribution distillation loss becomes:

$$\mathcal{L}_{\texttt{NLL}}^{\texttt{DD}}(\boldsymbol{\phi}) = -\frac{1}{ML}\sum_{m,l,k} \ln \mathcal{N}\left(\hat{\boldsymbol{z}}_{l,k}^{(m)} | \mu_{l,k}, \sigma_{l,k}^2\right)$$

where $\mathcal{N}$ is simply the univariate Gaussian. The Laplace equivalent follows the same sequence of steps. Note that at inference time, we simple use the softmax of the mean as an approximation to the predictive distribution. For uncertainty estimation however, we sample in parallel 20 logits from this distribution and use them for quantifying total and knowledge uncertainty. This represents an inexpensive operation since we restrict ourselves to diagonal multivariate distributions.

C Ablation Studies

This section will explore a wide range of experiments briefly mentioned in the main paper:

1. Distribution distillation when only the top-k probabilities are saved reflecting the case when limited disk space is available,

2. the impact of detection when changing the in-domain dataset,

Table 7: BLEU newstest14 performance (± 2 std). The vocabulary size is $|\mathcal{V}| = 32768$.

Model	$k = \|\mathcal{V}\|$	$k = 128$	$k = 32$
Snapshot Ensemble	26.54 ± 0.16	—	—
SPA (Categorical)	27.02 ± 0.19	27.09 ± 0.14	27.00 ± 0.11
SPA (Dirichlet)	26.96 ± 0.06	27.02 ± 0.23	26.99 ± 0.02
SPA (Gaussian)	26.90 ± 0.28	26.99 ± 0.32	27.02 ± 0.26
SPA (Laplace)	27.08 ± 0.20	26.99 ± 0.30	26.97 ± 0.19

3. and BLEU and detection performance of ensembles with increasing number of members.

C.1 Saving the Top-K Predictions

While the focus of this paper has been computational efficiency in regards to training and deploying robust single models we have neglected the memory cost of storing a large number of ensemble predictions. This section therefore, investigates the impact on performance when the practitioner only saves the top-k probabilities similar to Tan et al. (2019). However, a key difference is that at distillation time, we distribute the missing probability to remaining classes instead of renormalising the top-k probabilities. The experiments were only run on the smaller WMT16 dataset, see Table 7.

This shows a promising pattern, that storing predictions over the whole vocabulary is not necessary. In many cases, one can even achieve small performance boosts when only storing top-k predictions. The resulting conclusion is that even when there is not an abundant level of disk space available, SPA style approaches can still be used by simply discarding low probability classes.

C.2 Varying In-Domain Datasets for Detection

The detection results provided in sections 4.1 and 4.2 are not directly comparable due to a difference in in-domain dataset. This section simply provides the detection numbers for a WMT20 trained system using newstest14 as in-domain, see Table 8. While there still are significant differences in architecture and dictionaries used between WMT16 and WMT20 trained systems, these results at least provide the difference in performance when the same original data is used.

Significantly more experiments, out of the scope of this paper, need to be run to isolate the impact of dictionary size, joint or disjoint dictionaries, transformer architecture size and beam search parameters. Additionally, one might need to analyse the impact of dataset size, target language complexity on the ability to perform unsupervised out-of-distribution detection.

C.3 Scaling Ensembles

One intriguing aspect of the detection results shown in Table 8 is the ability of Laplace based SPA to outperform the Deep Ensemble in all experiments.

Table 8: Out-of-distribution detection using the %AUROC ↑ metric (± 2 std). The last three block columns represent the system trained, which in-domain dataset was used and the uncertainty utilised.

Dataset	**Model**	**WMT16 newstest14**		**WMT20 newstest14**		**WMT20 newstest20**	
		TU	**KU**	**TU**	**KU**	**TU**	**KU**
khresmoi	Deep Ensemble	48.0	61.9	42.5	55.5	39.3	53.2
	SPA (Laplace)	65.1 ± 1.8	**73.1** ± 1.7	52.6 ± 0.7	**63.8** ± 1.1	51.0 ± 0.9	**63.4** ± 1.2
mtnt	Deep Ensemble	64.5	63.7	73.0	71.1	70.8	69.0
	SPA (Laplace)	65.1 ± 1.5	**66.8** ± 1.8	**73.8** ± 0.7	70.6 ± 0.3	**72.6** ± 0.8	70.2 ± 0.6
kftt	Deep Ensemble	30.1	44.0	53.8	62.7	51.0	60.3
	SPA (Laplace)	37.8 ± 0.2	**48.8** ± 1.4	64.6 ± 0.8	**70.7** ± 1.2	63.2 ± 1.0	**70.2** ± 1.1

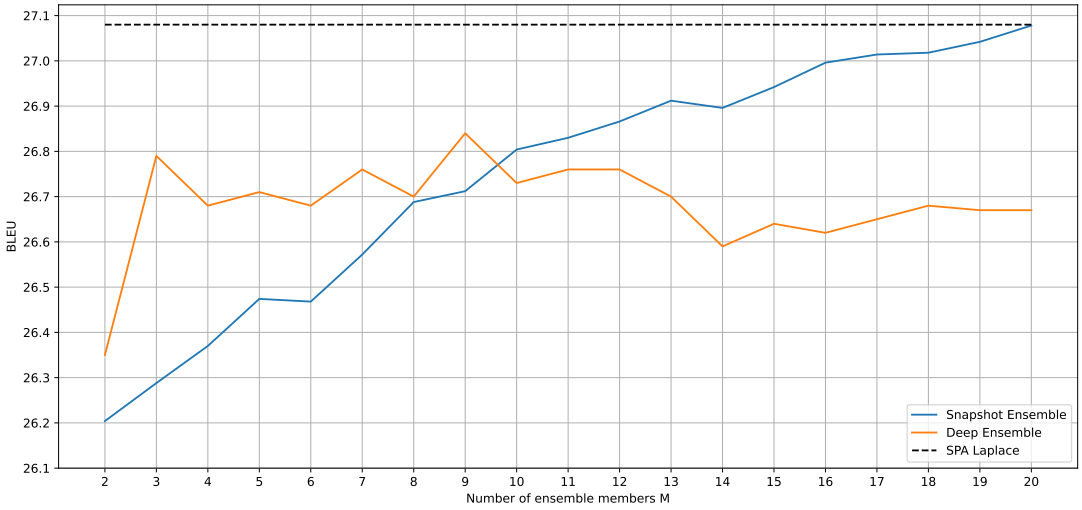
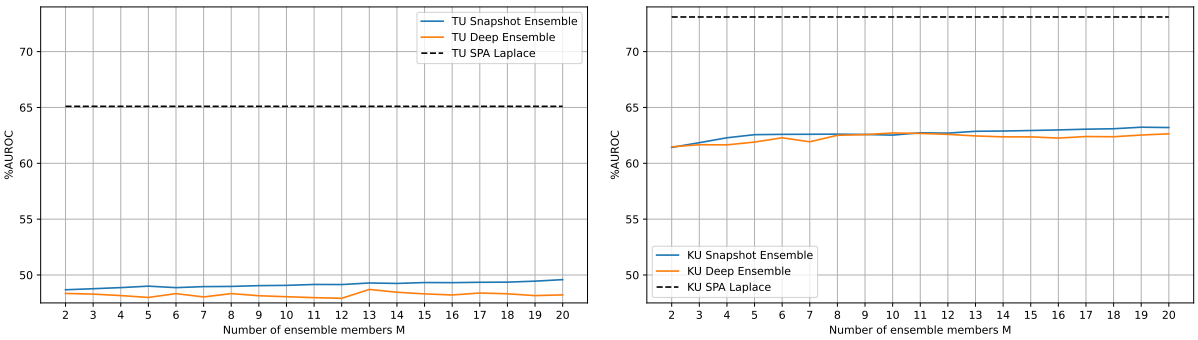


Figure 1: BLEU performance vs number of ensemble members on newstest14.

Therefore, an interesting element would be to investigate if this pattern still holds when scaling both Deep and Snapshot Ensembles to much larger sizes and empirically verifying if the SPA model is carrying out some type of interpolation of ensemble predictions. Note, the cost of training a Deep Ensemble scales with M (where M is the number of members) while a Snapshot Ensemble only requires approximately $1 + 0.097M$ in our WMT16 setup.

Before investigating detection, BLEU performance on newstest14 is reported using the the WMT16 setup, see Figure 1. While the Deep Ensemble plateaus quickly, the Snapshot Ensemble is able to have consistent gains in performance as it grows. This has to do with each successive Snapshot Ensemble member having increasingly better performance coupled with higher diversity from its predecessors, leading to larger ensemble gains.

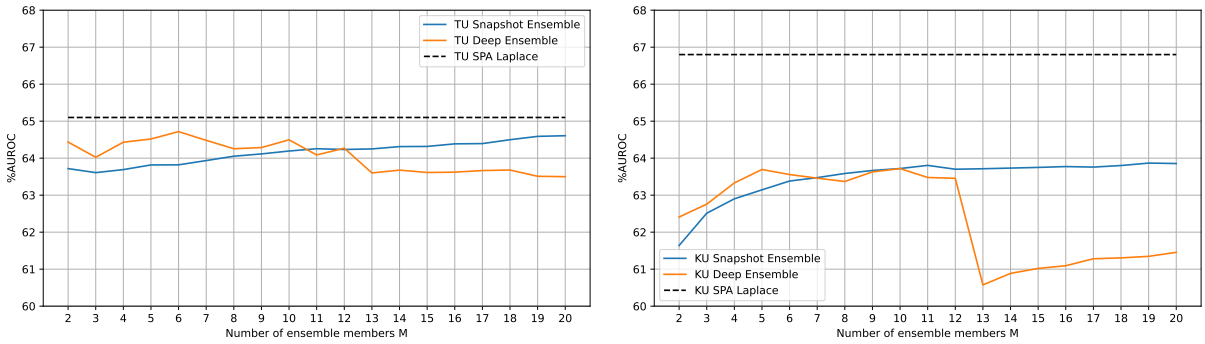
Regarding detection, there seems to be no clear increasing pattern when scaling ensembles, see Figures 2, 3 and 4. While the Snapshot Ensemble does show small gains in detection performance using both total and knowledge uncertainty, it is not able to reach the Laplace model in any case. On top of that, the Deep Ensemble shows even more detrimental results, in many cases displaying worse performance as the ensemble grows past a certain size. Therefore, this points to logit based distribution distillation possibly extracting and overvaluing more dark knowledge from low probability classes and not just simply interpolating ensemble predictions.



(a) Total Uncertainty

(b) Knowledge Uncertainty

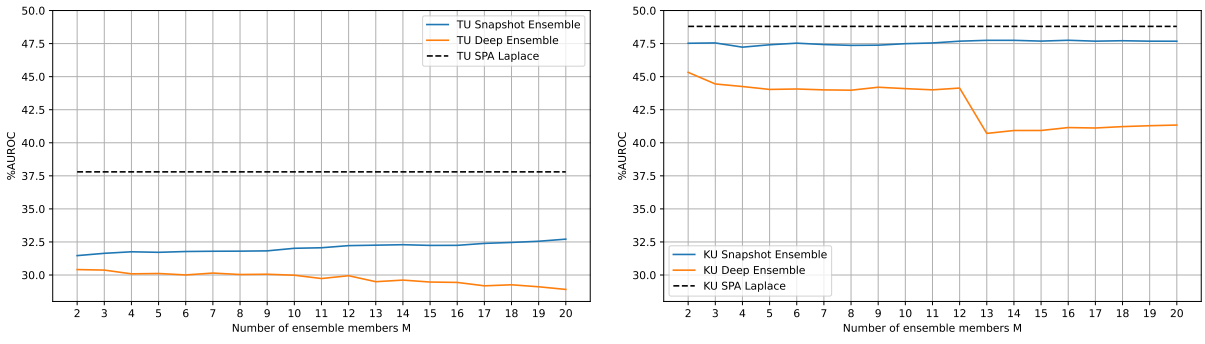
Figure 2: %AUROC detection performance on Khresmoi with increasing ensemble size.



(a) Total Uncertainty

(b) Knowledge Uncertainty

Figure 3: %AUROC detection performance on MTNT with increasing ensemble size.



(a) Total Uncertainty

(b) Knowledge Uncertainty

Figure 4: %AUROC detection performance on KFTT with increasing ensemble size.