IMPROVING THE CALIBRATION OF FINE-TUNED LANGUAGE MODELS VIA DENOISING VARIATIONAL AUTO-ENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large pre-trained language models have demonstrated strong performance on natural language understanding (NLU) tasks through fine-tuning. However, fine-tuned models still suffer from overconfident predictions, especially in out-of-domain settings. In this paper, we tackle the problem of calibrating fine-tuned language models. Based on the observation that pre-trained language models are well-calibrated yet fine-tuned models distort pre-trained features, we propose a simple fine-tuning method that enforces the consistency between the pre-trained and fine-tuned features. Specifically, we improve calibration on downstream tasks by introducing an auxiliary generative language modeling objective in the fine-tuning phase. The auxiliary objective is defined with a denoising variational auto-encoding framework, which encourages the model to learn generative representations. Empirical results show that our method achieves competitive accuracy and the lowest expected calibration error compared to several strong baselines under both in-domain and out-of-domain settings on three downstream NLU tasks. Furthermore, our fine-tuned models preserve the ability of language modeling. They can generate text given labels on downstream datasets and thus have better interpretability compared to vanilla fine-tuning.

1 INTRODUCTION

Fine-tuning pre-trained language models (PLMs) is a dominating paradigm for natural language understanding (NLU) with state-of-the-art results for a variety of NLU tasks (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019; He et al., 2021a). However, as with other modern neural networks, the fine-tuned language models are shown to suffer from overconfidence (Desai & Durrett, 2020; Jiang et al., 2021), i.e., their posterior probabilities (or confidence) do not align well with the true correctness likelihood, which can lead to disastrous consequences in safety-critical applications (Johnson et al., 2020; Sandagiri et al., 2020).

Uncertainty estimation of fine-tuned models is challenging due to the small amount of available data for fine-tuning. Several approaches have been developed to calibrate the fine-tuned model on NLU tasks, including temperature scaling and label smoothing (Desai & Durrett, 2020), stochastic network architectures (Fan et al., 2020; Zhang et al., 2021), and Mixup (Park & Caragea, 2022). However, these existing works mostly utilize general calibration methods for supervised learning, while specific properties of the pre-training & fine-tuning paradigm are still largely neglected.

In this work, we tackle the calibration of fine-tuned models from a different angle. We observe that PLMs themselves are actually *well-calibrated* for the masked language modeling (MLM) task. We hypothesize that the pre-trained features could model the predictive uncertainty well across different tasks, so preserving them is helpful for the calibration on downstream tasks. We examine this hypothesis through a carefully designed empirical study. However, the pre-trained features are only used as initialization and are distorted by the fully discriminative fine-tuning.

To improve the calibration on downstream tasks, we propose a simple fine-tuning method that utilizes the MLM objective to maintain the consistency between the pre-trained and fine-tuned models to improve the calibration of the fine-tuned models. Our objective is defined with a denoising vari-

ational auto-encoder (Im et al., 2017), which encourages the model to learn both generative and discriminative representations while fine-tuning on downstream NLU tasks.

Empirical results demonstrate that our proposed method achieves the lowest expected calibration error and competitive accuracy compared to existing methods in both in-domain (ID) and out-of-domain (OD) settings on three NLU tasks, including natural language inference, paraphrase detection, and commonsense reasoning. In addition to being well-calibrated, our fine-tuned models preserve the generative power. They can generate text with given labels on downstream datasets with simple sampling methods, which exhibits better interpretability than vanilla fine-tuning.

2 Related Work

Previous works have demonstrated that PLMs can be beneficial for improving the robustness and calibration on downstream tasks compared to non-pre-trained models (Hendrycks et al., 2020; Desai & Durrett, 2020). However, PLMs can still fail to model their predictive uncertainty on downstream tasks. For example, Desai & Durrett (2020); Kong et al. (2020) have shown that fine-tuned masked language models (e.g., BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019)) are overconfident on text classification tasks, while Jiang et al. (2021) has shown that powerful generative PLMs (e.g., T5 (Raffel et al., 2020), BART (Lewis et al., 2020), and GPT-2 (Radford et al., 2019)) are poorly calibrated on QA tasks.

Several approaches have been developed to calibrate the fine-tuned language model on NLU tasks. For instance, Desai & Durrett (2020) demonstrate that temperature scaling and label smoothing can improve the calibration of the models in ID and OD settings, respectively. He et al. (2021b) introduce a new discriminative objective under the noise contrastive estimation (NCE) framework to jointly train an energy-based model defined on the classifier, which leads to better ID calibration. Fan et al. (2020); Zhang et al. (2021) model the attention weights as random variables and design a series of methods to optimize the stochastic attention layer with variational inference, which yields better performance in accuracy and calibration compared to vanilla deterministic attention layers. There are also several works adopting Mixup (Zhang et al., 2018) for calibrating fine-tuned language models. For example, Kong et al. (2020) use Mixup as a regularization to mitigate overconfidence for OD data, and Park & Caragea (2022) mixup two categorized training sets guided by Area Under the Margin and Saliency. The mixup-based methods can effectively improve calibration in both ID and OD settings.

The uncertainty estimation problem of fine-tuned language models is also studied on other tasks besides NLU. Si et al. (2022) observe overconfident tendencies under ID and OD settings for neural text retrievers on open domain QA (OpenQA) tasks and calibrate the model by enforcing the prediction consistency along the training trajectory. Tam et al. (2022) show that parameter-efficient prompt tuning methods can improve the generalization and calibration of neural text retrievers on OpenQA tasks. Xie et al. (2021) validate that standard fine-tuning can destroy the output structure of pre-trained generative denoiser such as BART and show that preserving pre-trained features can improve out-of-distribution generalization.

3 Preliminaries

3.1 Masked Language Models

Masked language models generally consist of a transformer-based text encoder f_ϕ parameterized by ϕ and a linear language modeling head g_θ parameterized by θ. In the pre-training phase, the model handles the masked language modeling task (Devlin et al., 2019). Assume we have unsupervised sequence inputs $\boldsymbol{x}$ sampled from large-scale corpora $p_u(\boldsymbol{x})$. A subset of $\boldsymbol{x}$ is first masked by a corruption function or distribution. Denote the indices of the masked tokens as $\mathbb{M}$, the set of masked tokens as $\boldsymbol{x}_{\mathbb{M}}$, and the observed unmasked input as $\boldsymbol{x}_{\setminus\mathbb{M}}$. The model is trained to recover the masked tokens $\boldsymbol{x}_{\mathbb{M}}$. In particular, the masked language model first uses the text encoder to get a hidden representation of the input, denoted as $f_\phi(\boldsymbol{x}_{\setminus\mathbb{M}})$. Then the language modeling head g_θ with softmax function is applied to $f_\phi(\boldsymbol{x}_{\setminus\mathbb{M}})$ to obtain a conditional categorical distribution $p_{\text{mlm}}(x_i|\boldsymbol{x}_{\setminus\mathbb{M}})$ over

the vocabulary $\mathbb{V}$ for each masked position $i \in \mathbb{M}$. The masked language modeling objective is:

$$\mathcal{L}_{\text{mlm}} = -\mathbb{E}_{p_u(\boldsymbol{x})} \left[\sum_{i \in \mathbb{M}} \log\, p_{\text{mlm}}(x_i | \boldsymbol{x}_{\setminus \mathbb{M}}; \phi, \theta) \right] \tag{1}$$

In the fine-tuning phase, assume we have samples in the form of $(\boldsymbol{x}, y)$ sampled from data distribution p_d, where $\boldsymbol{x}$ corresponds to the text input, and y corresponds to the label. For classification tasks, a task-specific head h_φ that parameterized by φ is applied on the hidden representation of input to obtain the logit for each class. The predictive posterior distribution $q(y|\boldsymbol{x})$ is given by the logits after softmax operation. In standard fine-tuning, the pre-trained encoder f_ϕ and the task-specific head h_φ are jointly optimized using the cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = -\mathbb{E}_{p_d(\boldsymbol{x}, y)} \left[\log\, q(y|\boldsymbol{x}; \phi, \varphi) \right] \tag{2}$$

which is also known as full fine-tuning (Full-FT).

3.2 Paramater-Efficient Tuning for Pre-Trained Transformers

Parameter-efficient tuning is a series of fine-tuning methods which keep the pre-trained parameters of the text encoder ϕ frozen and update only a small number of extra parameters ϕ_Δ while preserving the competitive performance with Full-FT. For example, Adapter (Houlsby et al., 2019) adds a light-weight bottleneck module after the output of each sub-layer in the transformer block; LoRA (Hu et al., 2021) updates the attention weight matrix using low-rank reparameterization; Prefix Tuning (Li & Liang, 2021) prepends tunable prefix vectors to keys and values of the multi-head attention layers. The objective of parameter-efficient tuning methods is identical to standard fine-tuning, except only the extra parameters ϕ_Δ and the task-specific head h_φ are optimized.

3.3 Confidence Calibration

The framework of confidence calibration under the supervised classification setting can be expressed as a joint distribution $P(\hat{y}, \hat{p})$ over the label prediction $\hat{y} \in |\mathcal{Y}|$ and the corresponding confidence $\hat{p} \in [0, 1]$. A perfectly calibrated model holds $P(\hat{y} = y | \hat{p} = p) = p$. One way to evaluate calibration through finite samples is expected calibration error, i.e., ECE (Naeini et al., 2015; Guo et al., 2017). The model's predictive confidences are first grouped into M equal-sized bins. Denote B_m as the indices of samples whose confidences are in the interval $(\frac{m-1}{M}, \frac{m}{M}]$. Suppose we have N samples, the ECE is calculated by the weighted average of the difference between confidence and accuracy in each bin:

$$\begin{gathered} \text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}\left(\hat{y}_i = y_i\right), \;\; \text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i \\ \text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)| \end{gathered} \tag{3}$$

In this work, we set $M = 10$ following Desai & Durrett (2020).

4 Are Pre-trained Language Models Themselves Calibrated?

In this section, we attempt to explore the connection between the pre-trained and fine-tuned models in terms of calibration by examining: (1) The calibration of the pre-trained language models themselves. (2) The effect of fine-tuning on the pre-trained features in calibration. First, we evaluate the calibration of PLMs on the MLM task and investigate the pre-trained features' role in the calibration of downstream tasks with parameter-efficient tuning methods. Such methods keep the pre-trained text encoder frozen when adapting to downstream tasks, which makes them likely to retain more pre-trained features. Then we illustrate the effect of fine-tuning on the pre-trained features by the calibration performance of full fine-tuning and parameter-efficient tuning on both MLM and NLU tasks, along with the feature consistency between the two fine-tuning methods and the PLMs.

4.1 SETUP

We use pre-trained $\texttt{RoBERTa}_{\text{BASE}}$ (Liu et al., 2019) as our backbone model and perform full fine-tuning and parameter-efficient tuning on downstream NLU tasks. We report the ECE of both pre-trained and fine-tuned models on the MLM task and report the accuracy and ECE of two fine-tuning methods on the NLU tasks. Since fine-tuning methods do not update the language modeling head, we simply use the pre-trained one when evaluating the calibration of fine-tuning methods on the MLM task. We use the WikiText-103 (Merity et al., 2016) test set with the identical masking approach (i.e., 80-10-10 strategy with 15% masking rate) as in the pre-training phase (Devlin et al., 2019) for evaluating calibration on the MLM task and follow the same setup described in §6.1 for evaluating full fine-tuning on the NLU tasks. For parameter-efficient tuning methods, we choose three representative approaches: Adapter (Houlsby et al., 2019), LoRA (Hu et al., 2021) and Prefix Tuning (Li & Liang, 2021). We use the default hyperparameter configuration provided by OpenDelta (Ding et al., 2022) library for all three methods and conduct a grid search for learning rates on different tasks. The detailed setup for each method can be found in Appendix A.2.1.

Table 1: Samples generated by pre-trained and different fine-tuned models using the Mask-Predict algorithm.

Text Prefix:	*a little girl sitting in a seat.*
Input:	*She is* `[MASK] [MASK] [MASK]`
Pre-trained:	*She is a little girl.*
Adapter:	*She is not very tall.*
LoRA:	*She is 5 years old.*
Prefix Tuning:	*She is a a grade size*
Full-FT:	*She is I, I,*
Text Prefix:	*A man holding his two adorable babies.*
Input:	*A man* `[MASK] [MASK] [MASK]`
Pre-trained:	*A man with a big heart*
Adapter:	*A man being a man.*
LoRA:	*A man in his office.*
Predix Tuning:	*A man's a senior too*
Full-FT:	*A man said he said,*

Table 2: Calibration of different fine-tuning methods on in-domain (SNLI, QQP, SWAG) and out-of-domain (MNLI, TwitterPPDB, HellaSWAG) datasets of three NLU tasks.

RoBERTa-base	ID		OD	
	Acc	ECE	Acc	ECE
Task: SNLI/MNLI				
Full-FT	$\mathbf{91.89}_{0.18}$	$2.21_{0.12}$	$\mathbf{79.87}_{0.32}$	$4.06_{0.20}$
Adapter	$91.30_{0.09}$	$1.32_{0.20}$	$79.17_{0.30}$	$1.98_{0.47}$
LoRA	$89.86_{0.23}$	$\mathbf{1.28}_{0.12}$	$77.52_{0.23}$	$1.70_{0.33}$
Prefix Tuning	$89.08_{0.19}$	$1.68_{0.08}$	$76.09_{0.28}$	$\mathbf{1.53}_{0.39}$
Task: QQP/TwitterPPDB				
Full-FT	$\mathbf{91.26}_{0.13}$	$2.44_{0.10}$	$\mathbf{86.15}_{0.35}$	$10.09_{0.49}$
Adapter	$90.18_{0.09}$	$1.37_{0.08}$	$85.63_{0.30}$	$9.84_{0.47}$
LoRA	$88.54_{0.10}$	$\mathbf{1.05}_{0.09}$	$85.51_{0.46}$	$8.55_{0.51}$
Prefix Tuning	$87.67_{0.12}$	$1.46_{0.16}$	$85.26_{0.22}$	$\mathbf{8.02}_{0.21}$
Task: SWAG/HellaSWAG				
Full-FT	$\mathbf{83.98}_{0.27}$	$\mathbf{1.38}_{0.25}$	$42.99_{0.97}$	$8.80_{0.65}$
Adapter	$82.37_{0.12}$	$3.25_{0.19}$	$\mathbf{43.94}_{1.08}$	$11.13_{0.76}$
LoRA	$80.01_{0.18}$	$5.12_{0.15}$	$43.17_{0.42}$	$\mathbf{4.38}_{0.99}$
Prefix Tuning	$80.66_{0.20}$	$4.20_{0.56}$	$42.86_{0.63}$	$5.76_{1.09}$

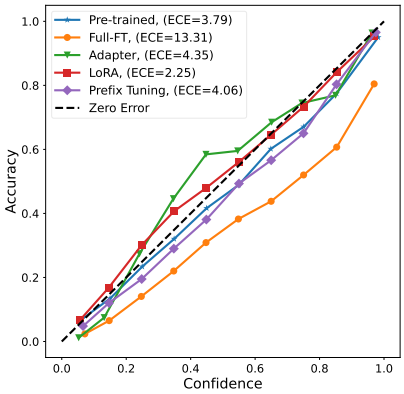


Figure 1: Reliability diagram (DeGroot & Fienberg, 1983; Niculescu-Mizil & Caruana, 2005) for masked language modeling task on WikiText-103 test set.

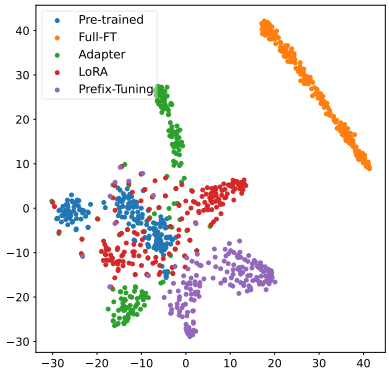


Figure 2: t-SNE (Van der Maaten & Hinton, 2008) visualization for hidden representations of sampled inputs from the QQP test set given by the pre-trained and different fine-tuned models.

4.2 RESULTS AND ANALYSIS

As shown in Figure 1, the PLMs and parameter-efficient tuning methods are relatively well-calibrated on the MLM task, while the full fine-tuned model exhibits overconfidence on this task, which suggests that the pre-trained feature could model the predictive confidence of the MLM task

accurately. For the downstream NLU tasks, Table 2 shows that parameter-efficient tuning methods are generally more calibrated than full fine-tuning under both ID and OD settings. We also examine how the parameter-efficient tuning methods related to the pre-trained models on downstream tasks. As shown in Figure 2, parameter-efficient tuning methods are more consistent with the pre-trained model in the feature space compared to full fine-tuning. From these empirical results, we hypothesize that *the pre-trained features could model the predictive uncertainty well, and preserving them are helpful for the calibration on downstream tasks.*

Secondly, we adopt the Mask-Predict algorithm (Ghazvininejad et al., 2019) on all methods, letting them predict several masked positions given text prefixes to study the effects of fine-tuning on the pre-trained features. As shown in Table 1, parameter-efficient tuning methods can generate readable sentences given text prefixes as the pre-trained model does. This makes them reasonable surrogates for the pre-trained models when evaluating the calibration on downstream tasks. On the other hand, the full fine-tuned models lose the ability of language modeling. Combining this with previous observations that full fine-tuning degrades the calibration of the language models on both MLM and NLU tasks, we posit that *fine-tuning on downstream tasks distorts the pre-trained features of the language models in terms of calibration.*

5 Our Approach

5.1 A new fine-tuning criterion using MLM Objective

Inspired by the observations in §4, besides fixing the parameters of the text encoder like parameter-efficient tuning methods, we can also enhance the consistency between pre-training and fine-tuning by jointly optimizing the MLM objective while fine-tuning:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{dvae}} + \mathcal{L}_{\text{cls}} \tag{4}$$

Note that the MLM objective introduces the corruption distribution $p(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ in the fine-tuning phase, which allows us to re-define the generative objective with denoising variational auto-encoding (DVAE) framework (Im et al., 2017). Specifically, given the corrupted input $\tilde{\boldsymbol{x}} \sim p(\tilde{\boldsymbol{x}}|\boldsymbol{x})$, the text encoder f_ϕ can be viewed as an amortized inference network $q_\phi(\boldsymbol{z}|\tilde{\boldsymbol{x}})$, where the latent variable $\boldsymbol{z}$ corresponds to the hidden representation of the input $f_\phi(\tilde{\boldsymbol{x}})$. On the other hand, the pre-trained language modeling head g_θ can be treated as the generative network $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ defined with the categorical distribution. Then we could obtain the following denoising variational lower bound (ELBO):

$$\begin{aligned}\log p_\theta(\boldsymbol{x}) &\geq \mathbb{E}_{p(\tilde{\boldsymbol{x}}|\boldsymbol{x})}\mathbb{E}_{q(\boldsymbol{z}|\tilde{\boldsymbol{x}})}\left[\frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\tilde{\boldsymbol{x}})}\right] \\ &= \mathbb{E}_{p(\tilde{\boldsymbol{x}}|\boldsymbol{x})}\left[\mathbb{E}_{q(\boldsymbol{z}|\tilde{\boldsymbol{x}})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{\text{KL}}\left(q_\phi(\boldsymbol{z}|\tilde{\boldsymbol{x}})\|p(\boldsymbol{z})\right)\right] = -\mathcal{L}_{\text{dvae}}\end{aligned} \tag{5}$$

In this work, we do not change how we calculate the hidden representation of inputs. Hence, the variational distribution $q_\phi(\boldsymbol{z}|\tilde{\boldsymbol{x}})$ is defined on the Dirac delta function $\delta_{f(\tilde{\boldsymbol{x}})}(\boldsymbol{z})$ and we choose fixed gaussian as the prior $p(\boldsymbol{z})$. Putting the ELBO in Eq. (5) and the predictive distribution $q_\varphi(y|\boldsymbol{x})$ on the classification task together, we get our final training objective under the framework of DVAE:

$$\begin{aligned}\mathcal{L}_{\text{joint}} = &-\mathbb{E}_{p_u(\boldsymbol{x})}\mathbb{E}_{p(\tilde{\boldsymbol{x}}|\boldsymbol{x})}\left[\alpha\mathbb{E}_{q(\boldsymbol{z}|\tilde{\boldsymbol{x}})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \beta D_{\text{KL}}\left(q_\phi(\boldsymbol{z}|\tilde{\boldsymbol{x}})\|p(\boldsymbol{z})\right)\right] \\ &-\mathbb{E}_{p_d(\boldsymbol{x},y)}\left[q_\varphi(y|\boldsymbol{x})\right]\end{aligned} \tag{6}$$

where α, β are scaling factors corresponding to the reconstruction loss and the KL term. In practice, we simply take $\mathbb{E}_{q(\boldsymbol{z}|\tilde{\boldsymbol{x}})}[\sum_{i\in\mathbb{M}}\log p_\theta(x_i|\boldsymbol{z})]$ from original MLM objective as the reconstruction term, which is a lower bound of $\mathbb{E}_{q(\boldsymbol{z}|\tilde{\boldsymbol{x}})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]$.

5.2 Improving DVAEs with Pre-training Corpus and Knowledge Distillation

The objective in Eq. (6) is designed to preserve the pre-trained features while fine-tuning on the downstream tasks. We introduce two simple techniques to further strengthen the connection between the pre-trained model and our fine-tuned model:

Utilizing corpus of the pre-training phase: The dataset for the MLM task is not required to be labeled as for the downstream task, so we can choose any text dataset. In this work, we present two

criteria for selecting $p_u(\boldsymbol{x})$ of the MLM task. The first one uses the dataset of downstream tasks, referred to as DVAE-D. The second one introduces the corpus of the pre-training phase, referred to as DVAE-P. We suppose that using the corpus in the pre-training phase is helpful for preserving the pre-trained features.

Distillation from the pre-trained model: We aim to enforce the consistency between the pre-trained and fine-tuned models through the MLM task, which reconstructs the masked positions to the original texts. However, this could be an insufficient regularization when fine-tuning on the NLU task, as the two distinct objectives may exhibit competitive behavior (Standley et al., 2020). To address this problem, we perform knowledge distillation (Hinton et al., 2015) from the pre-trained language model. Specifically, instead of calculating the reconstruction term using the original text as hard labels, we use the KL-divergence between the predictive distribution of the pre-trained language model and our model $\mathbb{E}_{q(\boldsymbol{z}|\tilde{\boldsymbol{x}})}[\sum_{i\in\mathbb{M}} D_{\mathrm{KL}}(p_{\theta_0}(x_i|\tilde{\boldsymbol{x}})\|p_\theta(x_i|\boldsymbol{z}))]$ as the reconstruction term, where $p_{\theta_0}(\boldsymbol{x}|\tilde{\boldsymbol{x}})$ is the fixed predictive distribution of the pre-trained language model.

Following previous works (Desai & Durrett, 2020; Park & Caragea, 2022), we also apply label smoothing (Szegedy et al., 2016) on the classification task, which can mitigate overconfident predictions by distributing a σ fraction of probability mass of the ground-truth label equally to other non-ground-truth classes.

5.3 Mask-Predict with Rejection Sampling

Since the hidden variable $\boldsymbol{z}$ of our fine-tuned model is encouraged to learn both discriminative and generative representations, we can perform conditional generation with the MLM head g_ϕ and classifier h_φ. In this work, we choose Mask-Predict (Ghazvininejad et al., 2019) as our basic sampling algorithm, which can incorporate the rejection sampling framework to generate samples from the conditional distribution $p(\boldsymbol{x}|y^*)$ given the desired label y^*. We show the detailed sampling process in Alg 1.

Algorithm 1: Mask-Predict with Rejection Sampling

Input: Number of iterations T, Inference network $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, Generative network $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, Classifier $q_\varphi(y|\boldsymbol{x})$, Desired label y^*, Number of initial masked tokens N.
Initialize: $\boldsymbol{x}^{(0)} \leftarrow$ `[MASK]`^{N}
for $t \leftarrow 0$ **to** $T-1$ **do** // Mask-Predict iteration
  $n \leftarrow f(N,t)$ // Determine number of tokens to mask at iteration t
  repeat // Rejection sampling loop
    $\boldsymbol{z}^{(t)} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(t)})$
    $\hat{\boldsymbol{x}}^{(t)} \sim p_\theta(\boldsymbol{x}|\boldsymbol{z}^{(t)})$ // Select prediction with highest probability for every `[MASK]` token.
    $\mathbb{M}_t \leftarrow \arg\min_i(p_\theta(x_i = \hat{x}_i^{(t)}|\boldsymbol{z}^{(t)}), n)$
    $\hat{\boldsymbol{x}}^{(t)}_{\mathbb{M}_t} \leftarrow$ `[MASK]` // Mask n tokens with the lowest probability scores
  until $u \sim \mathcal{U}(0,1)$, $u \leq q_\varphi(y = y^*|\hat{\boldsymbol{x}}^{(t)})/\tau$
  $\boldsymbol{x}^{(t+1)} \leftarrow \hat{\boldsymbol{x}}^{(t)}$
end
Output: $\boldsymbol{x}^{(T)} \sim p(\boldsymbol{x}|y^*)$

6 Experiments

6.1 General Setup

Datasets: We conduct experiments on three natural language understanding tasks: natural language inference (NLI), paraphrase detection (PD), and commonsense reasoning (CR). Each task consists of a pair of in-domain (ID) and out-of-domain (OD) datasets. Specifically, SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) are ID and OD datasets for NLI; QQP (Shankar et al., 2017) and TwitterPPDB (Lan et al., 2017) are ID and OD datasets for PD; SWAG (Zellers et al., 2018) and HellaSWAG (Zellers et al., 2019) are ID and OD datasets for CR. We use the same train/validation/test split for those six datasets published by Desai & Durrett (2020). For each task, we fine-tune the model using the ID training set and evaluate the model's performance with both ID

and OD test sets. We use WikiText-103 (Merity et al., 2016) as the corpus of the pre-training phase for DVAE-P. The detailed statistics for each dataset can be found in Appendix A.1.

Setup: We set a learning rate of 1e-5 and batch size of 32 across all three tasks as Desai & Durrett (2020) does [1]. For DVAEs, we use a Bernoulli corruption distribution [2]. We conduct hyperparameter search for the mask probability p_{mask}, the scaling factors α_{mlm}, β_{kl} of the MLM loss, and the KL term on the validation set of each task independently. We also search the hyperparameter σ_{ls} for label smoothing. Completed setup details for each task can be found in Appendix A.2.2. All experiments are run for 3 training epochs and are deployed on a single NVIDIA A40 48G GPU within 2 hours to fine-tune a single model.

Evaluation: In this section, we use pre-trained $\texttt{RoBERTa}_{\text{BASE}}$ (Liu et al., 2019) for all experiments. For each NLU task, we report accuracy and expected calibration error (ECE) on both ID and OD test sets. We evaluate "out-of-the-box" calibration of our method, which does not apply any post-hoc calibration methods such as temperature scaling (Guo et al., 2017).

Table 3: Out-of-the-box calibration results of different fine-tuning methods on in-domain (SNLI, QQP, SWAG) and out-of-domain (MNLI, TwitterPPDB, HellaSWAG) datasets. We report the averaged accuracy and ECE across five random fine-tuning runs. We also report the corresponding standard deviation in subscript.

RoBERTa-base	ID		OD	
	Acc	ECE	Acc	ECE
Task: SNLI/MNLI				
baseline (Desai & Durrett, 2020)	91.23	1.93	78.79	3.62
baseline (Our run)	$91.89_{0.18}$	$2.21_{0.12}$	$79.87_{0.32}$	$4.06_{0.20}$
BABN (Zhang et al., 2021)	91.70	2.62	79.86	2.67
Mixup (Park & Caragea, 2022)	$91.61_{0.5}$	$1.34_{0.7}$	$78.01_{0.6}$	$1.41_{1.9}$
Mixup + LS (Park & Caragea, 2022)	$91.24_{0.3}$	$1.28_{0.6}$	$78.86_{0.5}$	$1.37_{1.7}$
DVAE-D	$\mathbf{91.95}_{0.12}$	$\mathbf{0.67}_{0.05}$	$79.85_{0.29}$	$1.26_{0.50}$
DVAE-P	$91.74_{0.15}$	$1.09_{0.15}$	$79.36_{0.27}$	$\mathbf{1.20}_{0.43}$
DVAE-P + LS	$91.78_{0.09}$	$1.48_{0.17}$	$\mathbf{80.00}_{0.15}$	$1.90_{0.07}$
Task: QQP/TwitterPPDB				
baseline (Desai & Durrett, 2020)	91.11	2.33	86.72	9.55
baseline (Our run)	$91.26_{0.13}$	$2.44_{0.10}$	$86.15_{0.35}$	$10.09_{0.49}$
BABN (Zhang et al., 2021)	**91.72**	1.74	87.31	9.42
Mixup (Park & Caragea, 2022)	$89.19_{0.4}$	$2.47_{0.6}$	$87.13_{0.8}$	$3.94_{0.9}$
Mixup + LS (Park & Caragea, 2022)	$89.75_{0.6}$	$2.18_{0.7}$	$\mathbf{87.63}_{1.0}$	$3.96_{1.6}$
DVAE-D	$90.42_{0.08}$	$\mathbf{0.57}_{0.12}$	$86.67_{0.22}$	$6.49_{0.44}$
DVAE-P	$90.50_{0.09}$	$0.91_{0.50}$	$86.50_{0.90}$	$\mathbf{1.27}_{0.39}$
DVAE-P + LS	$90.42_{0.08}$	$0.77_{0.28}$	$86.38_{0.93}$	$2.21_{1.06}$
Task: SWAG/HellaSWAG				
baseline (Desai & Durrett, 2020)	82.45	1.76	41.68	11.93
baseline (Our run)	$83.98_{0.27}$	$1.38_{0.25}$	$42.99_{0.97}$	$8.80_{0.65}$
BABN (Zhang et al., 2021)	83.12	1.32	43.11	9.72
Mixup (Park & Caragea, 2022)	$81.47_{0.8}$	$1.24_{0.1}$	$40.95_{1.4}$	$2.40_{1.8}$
Mixup + LS (Park & Caragea, 2022)	$82.69_{0.7}$	$1.12_{0.4}$	$41.37_{1.1}$	$1.86_{0.9}$
DVAE-D	$\mathbf{84.49}_{0.24}$	$\mathbf{0.91}_{0.25}$	$43.75_{0.92}$	$7.64_{0.43}$
DVAE-P	$83.51_{0.05}$	$1.94_{0.12}$	$\mathbf{44.78}_{0.22}$	$5.23_{0.43}$
DVAE-P + LS	$83.61_{0.27}$	$1.20_{0.29}$	$44.08_{0.61}$	$\mathbf{1.62}_{0.29}$

6.2 RESULTS

We summarize our results in Table 3. Our baselines include full fine-tuning and two advanced methods based on the pre-trained language models: Bayesian Attention Belief Networks (BABN) (Zhang et al., 2021) and Mixup (Park & Caragea, 2022). We copied the results of the same setting from the original paper. We also report our full fine-tuning runs following the default behavior of the training scripts provided by the Huggingface Transformers library (Wolf et al., 2020) We present the results

[1]The only exception is training DVAE-P on SWAG with label smoothing, where we use a larger learning rate of 5e-5.

[2]In our fine-tuning method, we only perform the mask operation instead of the 80-10-10 strategy as in the pre-training phase.

following the objective in Eq. (6) for DAVE-D and the results using knowledge distillation with and without label smoothing for DVAE-P.

As shown in Table 3, full fine-tuned models generally have low in-domain ECE compared to out-of-domain, which illustrates that the fine-tuned language models tend to be overconfident under OD settings. BABN exhibits better generalization performance than deterministic methods, but it has a limited effect on the OD calibration. Mixup significantly lowers the ECE under both ID and OD settings while preserving comparable accuracy to vanilla fine-tuning.

Meanwhile, our proposed DVAE-D and DVAE-P achieve the lowest ECE across all three NLU tasks in ID and OD settings, respectively. In particular, DVAE-D best calibrates the model under ID settings and improves the OD calibration compared to full fine-tuning, while DVAE-P better calibrates the model under OD settings. Combining DVAE-P with label smoothing can lead to further gains in the OD settings, such as the calibration on HellaSWAG. In addition to being well-calibrated, our methods are competitive in accuracy. The proposed method achieves the best accuracy in both ID and OD settings on NLI and CR tasks and maintains the accuracy within $< 1\%$ drop compared to vanilla fine-tuning on the PD task. We also highlight that the DVAEs have relatively low standard deviations for both accuracy and ECE compared to other methods.

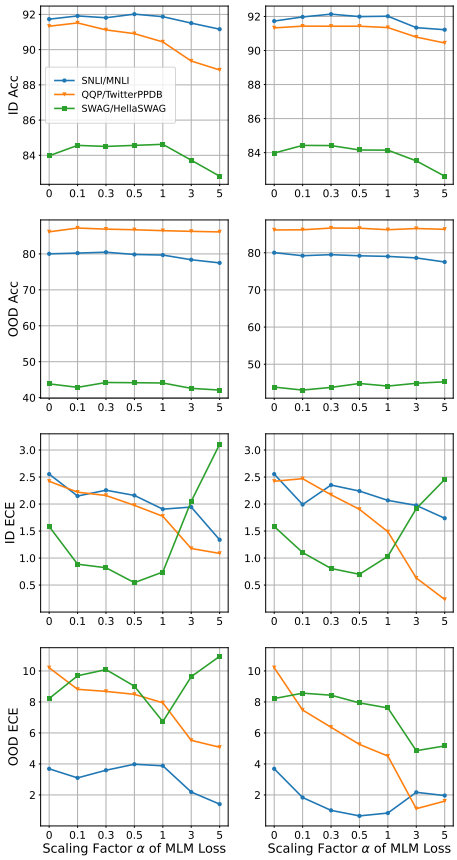


Figure 3: Accuracy and ECE of DVAE-D (left) and DVAE-P (right) over different scaling factors α of the MLM Loss in ID and OD settings.

Table 4: DVAE results with/without knowledge distillation.

RoBERTa-base	ID		OD	
	Acc	ECE	Acc	ECE
Task: SNLI/MNLI				
DVAE-D (no distill)	$91.95_{0.12}$	$0.67_{0.05}$	$79.85_{0.29}$	$1.26_{0.50}$
DVAE-D (distill)	$91.64_{0.19}$	$0.75_{0.21}$	$79.99_{0.30}$	$1.31_{0.28}$
DVAE-P (no distill)	$91.72_{0.14}$	$1.12_{0.17}$	$79.45_{0.25}$	$1.19_{0.17}$
DVAE-P (distill)	$91.74_{0.15}$	$1.09_{0.15}$	$79.36_{0.27}$	$1.20_{0.43}$
Task: QQP/TwitterPPDB				
DVAE-D (no distill)	$90.42_{0.08}$	$0.57_{0.12}$	$86.67_{0.22}$	$6.49_{0.44}$
DVAE-D (distill)	$90.86_{0.04}$	$0.50_{0.08}$	$86.26_{0.35}$	$6.06_{0.52}$
DVAE-P (no distill)	$89.75_{0.19}$	$0.78_{0.47}$	$86.77_{0.61}$	$2.86_{1.01}$
DVAE-P (distill)	$90.50_{0.09}$	$0.91_{0.50}$	$86.50_{0.90}$	$1.27_{0.39}$
Task: SWAG/HellaSWAG				
DVAE-D (no distill)	$84.49_{0.24}$	$0.91_{0.25}$	$43.75_{0.92}$	$7.64_{0.43}$
DVAE-D (distill)	$84.03_{0.14}$	$0.58_{0.08}$	$44.68_{0.49}$	$10.08_{0.36}$
DVAE-P (no distill)	$82.39_{0.03}$	$2.85_{0.11}$	$42.68_{0.24}$	$7.64_{0.37}$
DVAE-P (distill)	$83.51_{0.05}$	$1.94_{0.12}$	$44.78_{0.22}$	$5.23_{0.43}$

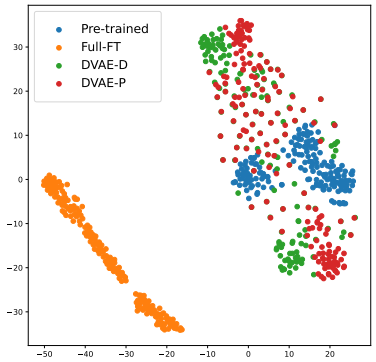


Figure 4: t-SNE visualization for hidden representations of sampled inputs from QQP test set given by DVAEs fine-tuned on QQP.

6.3 Ablation Study

Effect of the MLM objective: As shown in Figure 3, compared to vanilla fine-tuning ($\alpha = 0$), introducing the MLM objective lowers the ECE effectively in both ID and OD settings with a relatively small effect on accuracy. However, as the weight of the MLM loss becomes too large compared to the classification loss, the performance of the NLU task will be apparently damaged. We also exam-

ine how our method affects the features of the fine-tuned models. As shown in Figure 4, the DVAEs are far closer to the pre-trained model compared to the full fine-tuned model, which indicates that the DVAEs are likely to preserve the pre-trained features.

Effect of introducing corpus of pre-training phase: From Table 3 and Figure 3, we observe that performing the MLM task with the corpus of the pre-training phase has lower OD ECE on all three tasks. This confirms our belief that using the corpus of the pre-training phase can preserve more helpful features from the PLMs. We also notice that sometimes DVAE-P degrades the ID calibration (e.g., calibration on SWAG in Table 3), however, applying label smoothing on downstream tasks can relieve this negative effect.

Effect of knowledge distillation from the pre-trained model: Table 4 shows that applying knowledge distillation has little effect on DVAE-D but enhances the accuracy and ECE of DVAE-P in most cases. We hypothesize that performing the MLM task with the corpus that is distinct from the downstream datasets can hurt relatively more performance on the downstream tasks, and distilling from the pre-trained model's predictive distribution might be a smoother and more effective regularization.

6.4 Sampling from DVAEs

Table 5 shows the samples generated by DVAE-D with the sampling algorithm described in §5.3. The fine-tuned DVAEs are able to generate text with the given labels, which can be seen as a diagnostic for the model. For example, the models prefer to generate contradictory hypotheses by changing the objective's entity or adjectives with high confidence and tend to copy the prefix to generate positive textual entailment, which may expose some spurious correlations that the models rely on when performing the NLU tasks (Tu et al., 2020). We present more generated samples in Appendix A.3.1.

Table 5: Samples generated by DVAE-D with SNLI test set using the Mask-Predict algorithm with rejection sampling. The model generates hypotheses given class labels {none (-), entailment, contradiction, natural} and premise prefixes. We also report the corresponding confidence of the class of the generated text after the label.

Text Prefix:	*A mountain biker rides up a hill on a red bicycle.*
[-]	*A mountain biker rides a bike on a hill.*
[Entailment, 99%]	*A mountain biker rides a bike up a hill.*
[Contradiction, 99%]	*A mountain biker rides **downhill** on a **blue** bicycle.*
[Natural, 65%]	*A mountain biker is trying to climb a hill.*
Text Prefix:	*A man plays the french horn as his pianist plays the supporting melody on stage.*
[-]	*A man is playing a french horn for a concert.*
[Entailment, 97%]	*A man is playing a french horn on a stage.*
[Contradiction, 99%]	*A man is playing a **flute** for a **crowd**.*
[Natural, 91%]	*A man is playing a song on a concert stage.*

7 Conclusions

We improve the calibration of the fine-tuned language models on NLU tasks by introducing the generative MLM objective in the fine-tuning phase. From the empirical results, we conclude two main factors contributing to the calibration on downstream NLU tasks: (1) the MLM objective, (2) the consistency between pre-trained and fine-tuned features. First, as discussed in §5.1, the MLM objective can be re-defined with an ELBO of $p(\boldsymbol{x})$. Intuitively, encouraging the models to be aware of $p(\boldsymbol{x})$ while fine-tuning on the discriminative tasks helps the model better calibrate their predictive confidence. This is demonstrated by the well-calibrated predictions in the ID settings given by DVAE-D, which is also exhibited by joint energy based models (Grathwohl et al., 2020; He et al., 2021b). Secondly, utilizing the corpus from the pre-training phase and performing knowledge from the PLMs further improve the OD calibration of DVAEs, which corroborates our assumption that preserving the pre-trained features is helpful to the calibration on downstream tasks. We hope our work can draw more attention to deeper exploitation of the pre-trained features learned by PLMs with large corpora.

REPRODUCIBILITY STATEMENT

We have provided detailed setup for all experiments in §4.1, §6.1, A.1, and A.2, we submit our codes as the supplementary material. The information we provided is sufficient to reproduce our results.

REFERENCES

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 1983.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models, 2022.

Xinjie Fan, Shujian Zhang, Bo Chen, and Mingyuan Zhou. Bayesian attention modules. *Advances in Neural Information Processing Systems*, 2020.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. 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)*, 2019.

Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021a.

Tianxing He, Bryan McCann, Caiming Xiong, and Ehsan Hosseini-Asl. Joint energy-based model training for better calibrated natural language understanding models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021b.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

Daniel Jiwoong Im, Sungjin Ahn, Roland Memisevic, and Yoshua Bengio. Denoising criterion for variational auto-encoding framework. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How Can We Know When Language Models Know? On the Calibration of Language Models for Question Answering. *Transactions of the Association for Computational Linguistics*, 2021.

Alistair E. W. Johnson, Lucas Bulgarelli, and Tom J. Pollard. Deidentification of free-text medical records using pre-trained bidirectional transformers. CHIL '20, pp. 214–221. Association for Computing Machinery, 2020.

Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. Calibrated language model fine-tuning for in- and out-of-distribution data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

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

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, 2005.

Seo Yeon Park and Cornelia Caragea. On the calibration of pre-trained language models using mixup guided by area under the margin and saliency. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

S.P.C.W Sandagiri, B.T.G.S Kumara, and Banujan Kuhaneswaran. Deep neural network-based approach to identify the crime related twitter posts. In *2020 International Conference on Decision Aid Sciences and Application (DASA)*, pp. 1000–1004, 2020.

Iyer Shankar, Dandekar Nikhil, and Csernai Kornél. Quora question pairs. 2017.

Chenglei Si, Chen Zhao, Sewon Min, and Jordan Boyd-Graber. Revisiting calibration for question answering, 2022.

Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Weng Lam Tam, Xiao Liu, Kaixuan Ji, Lilong Xue, Xingjian Zhang, Yuxiao Dong, Jiahua Liu, Maodi Hu, and Jie Tang. Parameter-efficient prompt tuning makes generalized and calibrated neural text retrievers, 2022.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. An Empirical Study on Robustness to Spurious Correlations using Pre-trained Language Models. *Transactions of the Association for Computational Linguistics*, 2020.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020.

Sang Michael Xie, Tengyu Ma, and Percy Liang. Composed fine-tuning: Freezing pre-trained denoising autoencoders for improved generalization. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Shujian Zhang, Xinjie Fan, Bo Chen, and Mingyuan Zhou. Bayesian attention belief networks. In *ICML*, pp. 12413–12426, 2021.

A APPENDIX

A.1 DATASET DETAILS

The general information of in-domain and out-of-domain datasets of the three NLU tasks are shown below:

Natural Language Inference: Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) requires the model to learn the textual entailment by predicting the relationship between given premise and hypothesis among *entailment*, *contradiction*, or *neutral*. Multi-Genre Natural Language Inference (MNLI) (Williams et al., 2018) shares the same task form as SNLI, but the samples are from more diverse domains than SNLI.

Paraphrase Detection: Quora Question Pairs (QQP) (Shankar et al., 2017) contains question pairs from Quora. The model needs to discriminate whether the given pairs are semantically equivalent. TwitterPPDB (Lan et al., 2017) is the out-of-domain dataset that collects sentence pairs shared with the same URLs.

Commonsense Reasoning: Situations With Adversarial Generations (SWAG) (Zellers et al., 2018) is a common sense reasoning task that requires the model to choose the most plausible continuation of a sentence given four possible candidates. HellaSWAG (Zellers et al., 2019) is designed as a more challenging commonsense reasoning task for the pre-trained language models built with Adversarial Filtering.

The statistic of the datasets for both MLM and NLU tasks are shown in Table 6. For the datasets of NLU tasks (SNLI/MNLI, QQP/TwitterPPDB, SWAG/HellaSWAG), we use the published version by Desai & Durrett (2020). For WikiText-103, we use the version provided by Huggingface Datasets (Lhoest et al., 2021) library. Note that the training splits of the OD datasets (MNLI, TwitterPPDB, HellaSWAG) are not used.

Table 6: The size of the training, validation, test splits and number of labels for all datasets.

Dataset	#Train	#Validation	#Test	#Labels
SNLI (Bowman et al., 2015)	549,368	4,922	4,923	3
MNLI (Williams et al., 2018)	392,702	4,908	4,907	3
QQP (Shankar et al., 2017)	363,871	20,216	20,217	2
TwitterPPDB (Lan et al., 2017)	46,667	5,060	5,060	2
SWAG (Zellers et al., 2018)	73,547	10,004	10,004	4
HellaSWAG (Zellers et al., 2019)	39,905	5,021	5,021	4
WikiText-103 (Merity et al., 2016)	1,801,350	3,760	4,358	—

A.2 SETUPS

A.2.1 SETUP FOR FINE-TUNING METHODS

We conduct the experiments with the Huggingface Transformers library (Wolf et al., 2020). For parameter-efficient tuning methods, we use the implementations of OpenDelta (Ding et al., 2022) library for the three parameter-efficient tuning methods (Adapter, LoRA, Prefix Tuning). We use the same default hyperparameters provided by the OpenDelta library for each method across all three tasks [3].

All fine-tuning methods are trained with the AdamW optimizer (Loshchilov & Hutter, 2019). For full fine-tuning, we use a learning rate of 1e-5 across all tasks. For parameter-efficient tuning methods, we search the learning rate among {1e-5, 2e-5, 5e-5, 1e-4, 2e-4, 5e-4}. We use the one that has the best ID accuracy on the validation set, which is the widespread application scenario for the parameter-efficient tuning methods. Table 7 shows the learning rate used to fine-tune all the methods. For other training hyperparameters, we follow the default setup of the Huggingface Trans-

[3]The default hyperparameters can be found in the definition of each class on the document: `https://opendelta.readthedocs.io/en/latest/modules/deltas.html`.

formers library [4]. In specific, we set a batch size of 32, a maximum sequence length of 256, and a weight decay of 0.1. We use a linear decay learning rate scheduler without warmup and do not apply gradient clipping. Note that the reported baselines for full fine-tuning and Mixup (Desai & Durrett, 2020; Park & Caragea, 2022) in Table 3 do not use learning rate scheduler and use a gradient clip of 1.0 compared to our runs. There are also some minor differences, such as the padding strategy for input texts between our fine-tuning setup and theirs.

Table 7: Learning rate of fine-tuning methods on each task. The models are fine-tuned on the training split of SNLI for natural language inference (NLI), QQP for paraphrase detection (PD), SWAG for commonsense reasoning (CR).

Task	Full-FT	Adapter	LoRA	Prefix Tuning
NLI	1e-5	2e-4	2e-4	1e-4
PD	1e-5	2e-4	2e-4	1e-4
CR	1e-5	1e-4	1e-4	2e-4

A.2.2 Setup for Denoising VAEs

We conduct hyperparameter search with the ID/OD validation set for the DVAEs. Specifically, we set a learning rate of 1e-5 and a batch size of 32 across all three tasks as Desai & Durrett (2020) does, except for fine-tuning DVAE-P with label smoothing on SWAG, where we use a larger learning rate of 5e-5. For other training parameters, we adopt the same setup described in A.2.1 except for fine-tuning DVAE-P on QQP, where we do not use a learning rate scheduler. For the hyperparameter of DVAEs, we search the scaling factor of the MLM loss $\alpha_{\text{mlm}} \in \{0.1, 0.3, 0.5, 1, 2, 3, 4, 5\}$, the scaling factor of the KL term $\beta_{\text{kl}} \in \{$1e-9, 1e-8, 1e-7, 1e-6, 1e-5, 1e-4$\}$, the masking probability for a sentence $p_{\text{mask}} \in \{0.05, 0.15, 0.3, 0.4, 0.5, 0.6\}$, and the hyperparameter of label smoothing $\sigma_{\text{ls}} \in \{0.01, 0.03, 0.05\}$ for each task. We also search the maximum sequence length of the MLM task for DVAE-P and the batch size of MLM tasks for both DVAE-D and DVAE-P. In detail, we use a batch size for the MLM task of 32 on NLI and PD tasks and a batch size for the MLM task of 8 on the CR task. The maximum sequence lengths of the MLM task for DVAE-P are set to 32/32/64 for NLI, PD, and CR tasks, respectively. Training a single model for 3 epochs can be done in two hours with a single NVIDIA A40 48G GPU. We present the detailed hyperparameter setup of each method on each task in Table 8.

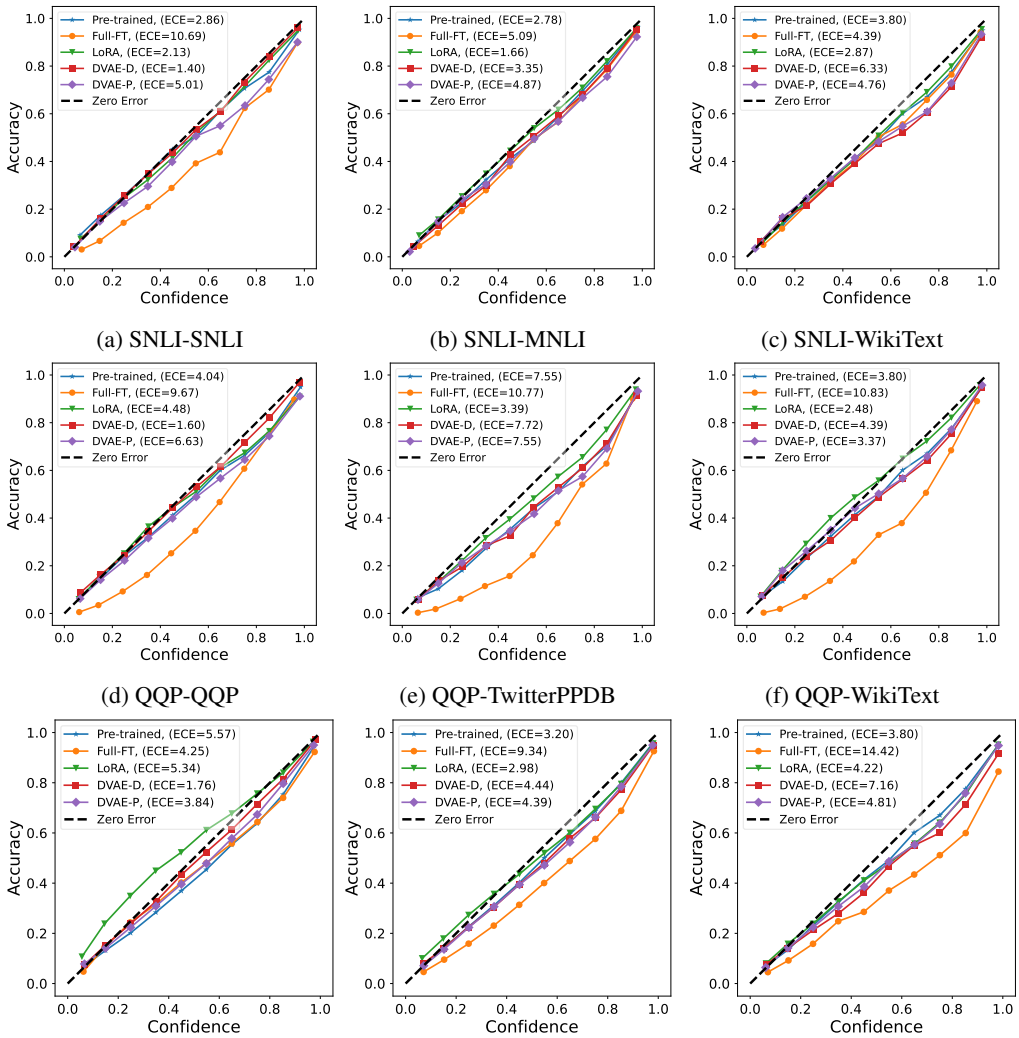
Table 8: Hyperparameters of DVAEs for each NLU task.

	α_{mlm}	p_{mask}	β_{kl}	σ_{ls}
Task: SNLI/MNLI				
DVAE-D	0.3	0.4	1e-5	—
DVAE-P	0.3	0.4	1e-5	—
DVAE-P + LS	0.5	0.6	1e-8	0.03
Task: QQP/TwitterPPDB				
DVAE-D	1	0.15	1e-5	—
DVAE-P	4	0.15	1e-7	—
DVAE-P + LS	4	0.15	1e-9	0.01
Task: SWAG/HellaSWAG				
DVAE-D	1	0.3	1e-9	—
DVAE-P	3	0.3	1e-9	—
DVAE-P + LS	3	0.05	1e-4	0.05

[4]The NLI and PD tasks correspond to the text classification setting, while the CR task corresponds to the multiple choice criterion of huggingface transformers library.

A.3 Additional Experimental Results

A.3.1 Calibration of pre-trained and fine-tuned models on MLM tasks



(a) SNLI-SNLI (b) SNLI-MNLI (c) SNLI-WikiText

(d) QQP-QQP (e) QQP-TwitterPPDB (f) QQP-WikiText

(g) SWAG-SWAG (h) SWAG-HellaSWAG (i) SWAG-WikiText

Figure 5: Reliability diagram and ECE of pre-trained and fine-tuned models on the MLM task with the identical masking strategy as in the pre-training phase. For the models fine-tuned on each task (Full-FT, LoRA, DVAEs), we evaluate their calibration of the MLM task on the ID and OD test sets and the WikiText corpus. The caption of each sub-figure represents the training dataset for fine-tuning and the test dataset for evaluating the calibration on the MLM task. For example, "SNLI-MNLI" represents the models that are fine-tuned on SNLI perform the MLM task on the MNLI test set.

A.3.2 SAMPLING FROM DVAES

Table 9: Samples generated by DVAE-D with SNLI and QQP test set using the Mask-Predict algorithm with rejection sampling. For SNLI, the model generates hypotheses given class labels {none (-), entailment, contradiction, natural} and premise prefixes. For QQP, the model generates sentences given class labels {none (-), non-paraphrase (non-para), paraphrase (para)} and question prefixes. We mark the generated samples whose assigned label is consistent with the given label using [ ] and mark the failure cases where the assigned label is not consistent with the given label using [ ]. We also report the corresponding confidence of the class of the generated text after the label.

Text Prefix:	*Two young men in unusual clothing are jumping in a gym.*
[-]	*Two men are playing basketball.*
[Entailment, 96%]	*Two men are jumping around.*
[Contradiction, 90%]	*Two men are jumping outside.*
[Natural, 44%]	*Two men are playing basketball.*
Text Prefix:	*a blue and gray race car driving on a dirt track.*
[-]	*A race car is driving on a dirt track.*
[Entailment, 98%]	*A race car is driving on a dirt track.*
[Contradiction, 99%]	*A race car is parked on a dirt track.*
[Natural, 66%]	*A race car is racing on a dirt track.*
Text Prefix:	*A boy poses in karate form and uniform.*
[-]	*A boy is practicing karate.*
[Entailment, 75%]	*A boy is practicing karate.*
[Contradiction, 2%]	*A boy is in a costume.*
[Natural, 25%]	*A boy is practicing karate.*
Text Prefix:	*Four females wearing helments are riding on an ATV.*
[-]	*Four females are wearing helments on an ATV.*
[Entailment, 98%]	*Four females are wearing helments on an ATV.*
[Contradiction, 99%]	*Four females are riding on a horse in the park.*
[Natural, 99%]	*Four females are riding a ATV in the desert.*
Text Prefix:	*How does cloud computing work?*
[-]	*How does cloud computing in India work?*
[Non-Paraphrase, 95%]	*How does Google think cloud computing work?*
[Paraphrase, 74%]	*How does I understand cloud computing work?*
Text Prefix:	*Do you think time travel is possible?*
[-]	*Do you think space and time travel is possible?*
[Non-Paraphrase, 73%]	*Do you think gravity can make time travel possible?*
[Paraphrase, 98%]	*Do you think it is possible to time travel?*
Text Prefix:	*What is a good data analysis book?*
[-]	*What is a good free data analysis book?*
[Non-Paraphrase, 78%]	*What is a good book for analysis books?*
[Paraphrase, 67%]	*What is the best free data analysis book?*
Text Prefix:	*Feeling bored. What do I do?*
[-]	*What to do to get bored?*
[Non-Paraphrase, 97%]	*What should I do to myself?*
[Paraphrase, 28%]	*What should I start to do?*