

Compression of Generative Pre-trained Language Models via Quantization

Chaofan Tao¹, Lu Hou², Wei Zhang², Lifeng Shang²,
Xin Jiang², Qun Liu², Ping Luo¹, Ngai Wong¹

¹The University of Hong Kong ²Huawei Noah’s Ark Lab
cftao@connect.hku.hk, pluo@cs.hku.hk, nwong@eee.hku.hk
{houlu3, zhangwei379, shang.lifeng, jiang.xin, qun.liu}@huawei.com

Abstract

The increasing size of generative Pre-trained Language Models (PLMs) have greatly increased the demand for model compression. Despite various methods to compress BERT or its variants, there are few attempts to compress generative PLMs, and the underlying difficulty remains unclear. In this paper, we compress generative PLMs by quantization. We find that previous quantization methods fail on generative tasks due to the *homogeneous word embeddings* caused by reduced capacity, and *varied distribution of weights*. Correspondingly, we propose a token-level contrastive distillation to learn distinguishable word embeddings, and a module-wise dynamic scaling to make quantizers adaptive to different modules. Empirical results on various tasks show that our proposed method outperforms the state-of-the-art compression methods on generative PLMs by a clear margin. With comparable performance with the full-precision models, we achieve 14.4 \times and 13.4 \times compression rates on GPT-2 and BART, respectively.

1 Introduction

Transformer-based generative pre-trained language models (PLMs) show strong abilities of multi-task and few-shot learning, and achieve remarkable performances on various tasks (Radford and Narasimhan, 2018; Brown et al., 2020; Lewis et al., 2020; Raffel et al., 2020; Chen et al., 2021). However, they are usually expensive in terms of both computation and memory due to a large number of parameters, and the token-by-token generation process. Many methods have been proposed to compress PLMs, but mostly focus on understanding tasks like sentence classification with BERT (Lan et al., 2019; Sun et al., 2020b; Jiao et al., 2020; Shen et al., 2020; Hou et al., 2020). Recent works try to compress GPT-2 using tensor decomposition (Edalati et al., 2021), and knowledge distillation (Song et al., 2020), but the compression

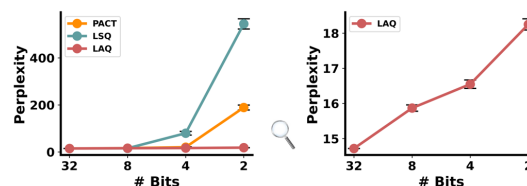


Figure 1: Performance of quantized GPT-2 with varying weight bit-widths and 8-bit activation, using different methods. The right figure takes a closer look at LAQ.

ratio achieved is much smaller than that of BERT. Yet the underlying difficulty remains unclear.

In this paper, we firstly explore compressing generative PLMs by quantizing the parameters from full-precision to lower bits. We find that directly applying previous quantization methods designed for BERT or computer vision tasks to generative PLMs lead to poor performance. Figure 1 shows that the performance drops sharply as the weight bit-width decreases. To investigate the difficulty of quantizing generative PLMs, we find that the learned embeddings tend to be homogeneous and hard to distinguish due to the reduced capacity caused by quantization, while the weight distributions also vary significantly across different modules and different Transformer layers. These problems are further magnified due to the nature of sequential left-to-right prediction of generative PLMs, as the quantization error will accumulate across time.

To alleviate the above problems, we propose a token-level contrastive distillation to contrast on tokens and make the word embedding distinguishable. Besides, we propose a module-wise dynamic scaling for the quantizer to better adapt to different modules. Empirical results on language modeling, next utterance prediction and summarization show that compared to the full-precision baseline, our quantized GPT and BART (abbreviated as QuantGPT and QuantBART) achieve comparable performance for 8/4-bit weight, and have only a slight drop for 2-bit weight, while being over 13 \times smaller.

QuantGPT also clearly outperforms previous GPT compression methods on language modeling.

To summarize, our main contributions are: 1) We find that generative PLMs are hard to quantize due to *homogeneous word embedding* and *varied weight distribution*. 2) We then propose the token-level contrastive distillation and module-wise dynamic scaling, to make the word embedding more distinguishable and make quantizers adapt to different modules, respectively. 3) Empirical results on various tasks show the efficacy of our method.

2 Difficulty of Quantizing Generative Pre-trained Language Models

In this section, we show that it is challenging to train a low-bit generative pre-trained model with conventional quantization approaches directly. Before diving into details, we first review the necessary backgrounds of quantization.

2.1 Network Quantization

In this paper, we apply the quantization-aware training (Courbariaux et al., 2015) to generative PLMs. Specifically, denote the vectorized full-precision weight as \mathbf{w} , each forward propagation first clips the weight by a positive clipping factor α , and then quantizes the clipped weight to b -bit as

$$\mathbf{w}_q = \alpha \cdot Q(\text{clip}(\mathbf{w}, -\alpha, \alpha)/\alpha), \quad (1)$$

where Q is the quantization function that maps each entry in $\text{clip}(\mathbf{w}, -\alpha, \alpha)/\alpha$ to its closest quantized value in the set of uniform discrete values $\{-1, -\frac{n-1}{n}, \dots, -\frac{1}{n}, 0, \frac{1}{n}, \dots, \frac{n-1}{n}, 1\}$ with $n = 2^{b-1} - 1$. Then we compute the loss $\ell(\mathbf{w}_q)$ with \mathbf{w}_q . During back propagation, we use the gradient with regard to the quantized weight $\nabla\ell(\mathbf{w}_q)$ as the Straight-Through-Estimator (Bengio et al., 2013) to update full-precision weights \mathbf{w} due to the non-differentiability of $Q(\cdot)$.

A good clipping factor is expected to take the majority of full-precision weight into account via clipping, *i.e.*, quantizing the range where data are densely distributed to reduce quantization error. To solve this problem, PACT (Choi et al., 2018) learns a parameterized clipping factor and achieves better results than setting a fixed clipping factor. Instead of learning the clipping factor, LSQ (Esser et al., 2020) learns the step size α/n , but requires a careful initialization and gradient update.

In practice, following previous works on BERT quantization (Zhang et al., 2020; Bai et al., 2021),

we use layer-wise quantization (*i.e.*, one clipping factor for elements in each weight matrix) for all weight matrices in the Transformer layers and row-wise quantization (*i.e.*, one clipping factor for each word embedding) for the embedding layer. We use asymmetric uniform quantization for activations after self-attention and GeLU function whose elements are mostly positive, and symmetric uniform quantization for other activations. We do not quantize layer-normalization layers, skip connections, biases due to small computational overhead.

2.2 Difficulty Analysis

We compare the following representative quantization methods including (i) LAQ (Zhang et al., 2020) for BERT; (ii) PACT (Choi et al., 2018) and LSQ (Esser et al., 2020) for computer vision tasks, to generative pre-trained model, GPT-2. Figure 1 shows the performance under different weight bit-widths, and the performance drops sharply as the bit-width decreases, especially for PACT and LSQ. In the following, we study the potential reasons behind the difficulty of quantizing generative PLMs, by empirically investigating the properties of the word embedding and model parameters.

Homogeneous Word Embedding. We first study the difficulty from the learned word embeddings of different models. In Figure 2, we visually compare the distributions of the word embeddings of the full-precision and quantized models under the same scale. As can be seen, the word embeddings of the full-precision model are scattered distinguishable, while those in previous quantization methods PACT, LSQ and LAQ learn homogeneous word embeddings which are clustered and less distinguishable, especially for PACT and LSQ. We speculate this is caused by the sequential computation nature of GPT. Specifically, unlike BERT which computes the representation of all tokens in parallel, GPT computes each token in left-to-right order, and the quantization error incurred in the previous tokens will pass on to future tokens, making the learning signal noisier over time, and finally less informative word embeddings.

A direct consequence of the homogeneous word embedding can be reflected in Figure 3. By comparing Figure 2 and Figure 3, we can find that the higher degree of homogeneity in the word embedding of a quantized model, the fewer dependencies among different tokens are kept.

As will be discussed in Section 3.1, we propose

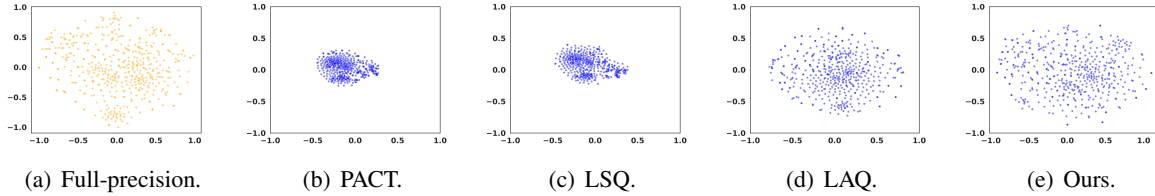


Figure 2: T-SNE visualization of the most frequent 500 word embeddings, of the full-precision and different 2-bit quantized models trained on PTB dataset. Embeddings of different methods show different degrees of homogeneity.

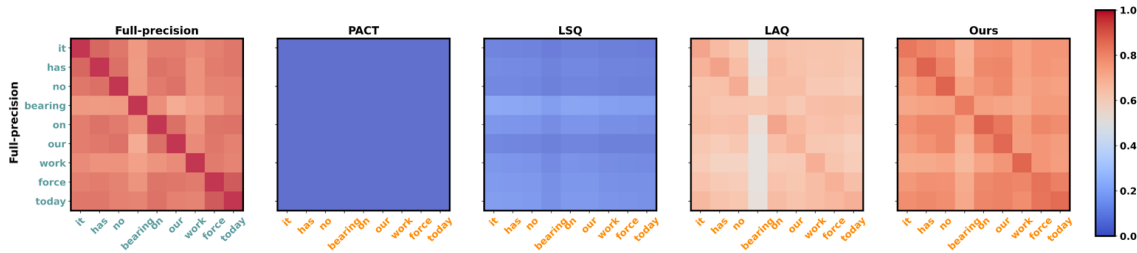


Figure 3: Matrices representing the cosine similarities between representations of all pairs of tokens in a sentence, between the full-precision model and 2-bit quantized models trained on PTB dataset. Token representations at the last decoder layer of GPT-2 are used. More visualizations are available in Appendix C.3.

a token-level contrastive learning to alleviate this problem. Compared with PACT, LSQ and LAQ, our method not only aligns the token representations between the quantized and full-precision networks (*i.e.*, diagonal boxes), but also captures the dependencies among different tokens (non-diagonal boxes). More visualizations are available in Appendix C.3. The non-distinguishable word embeddings and poor ability to capture contextualized dependencies also make methods like PACT and LSQ more likely to generate incorrect tokens, *e.g.* illogical and repeated text (Section 4.4).

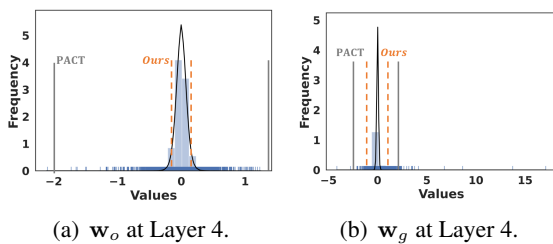


Figure 4: Distributions of output projection matrix w_o in the multi-head attention module and the second linear layer w_g in the feed-forward network of the 4-th layer from the 12-layer full-precision GPT-2. Other modules in other layers exhibit similar patterns. Vertical lines indicate the clipping factors learned by PACT and our method. Black curves show the estimated distribution by kernel density estimation.

Varied Distribution of Weights. Besides the learned word embeddings, we also investigate the

distribution of the weights in the full-precision model. Figure 4 shows that the weight distributions of a 12-layer full-precision GPT-2 are highly skewed with outliers. This causes difficulty in estimating the clipping factor α of the quantizer by heuristic methods, or even by PACT which learns the α through gradient descent. Specifically, in PACT, the approximated gradient of α only relies on the weights whose absolute values are larger than α . This solution ignores the effect of weights within $[-\alpha, \alpha]$ and depends heavily on the initialization of α . Figure 4 shows that an improper initialization together with the inaccurate gradient estimation of the clipping factor often make the learned α of PACT too large, and can not provide fine resolution to the majority of weights within the clipping range. The quantization error accumulated over time makes this problem more severe. In this work, we re-parameterize the clipping factor to make the quantizer adaptive to each module in the Transformer layers, and consider both weights outside and inside the clipping range when estimating the gradient of the clipping factor.

As will be discussed in Section 3.2, we propose a module-wise dynamic scaling to reduce the clipping factor’s sensitivity to initialization, and an improved gradient estimation that also considers the weights within $[-\alpha, \alpha]$. Figure 4 shows that the clipping factor learned by our method gives finer resolutions to the majority of the weights.

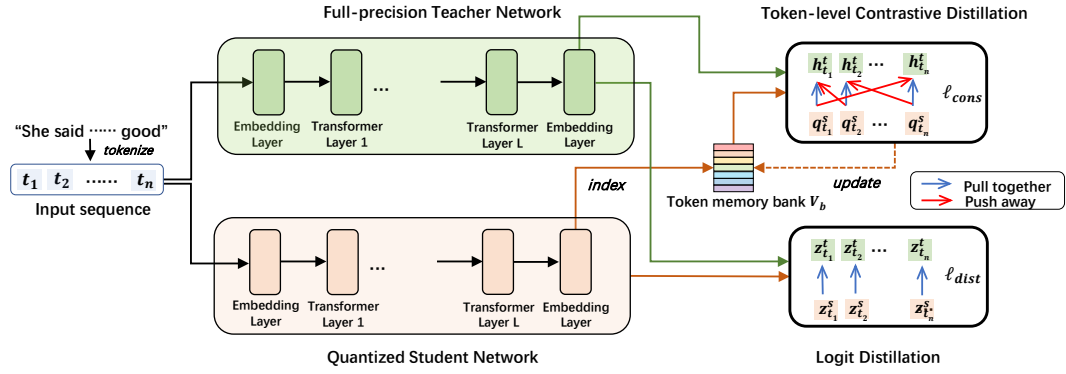


Figure 5: The training workflow of the proposed method. For each token in the quantized network, we compute both (i) the token-level contrastive distillation loss where the positive tokens and negative tokens are selected from the full-precision teacher network; and (ii) the distillation loss on the logits. The embedding layer and all weights in the Transformer layers are quantized with the proposed module-dependent dynamic scaling.

3 Proposed Method

Based on the observations in Section 2.2, we propose a quantization method which utilizes token-level contrastive distillation to make the word embedding distinguishable (Section 3.1) and a module-wise dynamic scaling adjustment to learn better clipping factors (Section 3.2).

3.1 Token-level Contrastive Distillation

The proposed token-level contrastive distillation contrast among tokens instead of sequences sequence, to learn distinguishable representations for each token. Inspired by Baevski et al. (2020), which uses in-utterance representation at different positions of the same utterance as negatives for speech feature learning, for each token of the quantized network, we use the representation of the same token from the full-precision teacher network as its positive, while representations of other tokens in the same sequence as negatives (Figure 5). Inspired by He et al. (2020) which uses a momentum encoder for more consistent representation, we build a memory bank to store momentum token representations from the quantized network. When computing the contrastive distillation loss, we load the representations of negative samples from the memory bank with cheap indexing operations.

Specifically, we use superscripts s and t to denote the quantized student network and full-precision teacher network, respectively. Denote the length- n input sequence of tokens as (t_1, t_2, \dots, t_n) . For the i -th token t_i , suppose its hidden states of the last Transformer layer from the quantized and full-precision network are linearly projected to $(\mathbf{h}_i^s, \mathbf{h}_i^t) \in \mathbb{R}^d$, and \mathbf{q}_i^s is the smoothed representation of \mathbf{h}_i^s in the memory bank. Denote

\mathcal{S}_i as the indices of the sampled negatives for token i , the token-level contrastive distillation loss for the length- n sequence can be formulated as

$$\mathcal{L}_{cont} = - \sum_{i=1}^n \log \frac{\exp(s(\mathbf{q}_{t_i}^s, \mathbf{h}_{t_i}^t)/\tau)}{\sum_{j \in \mathcal{S}_i} \exp(s(\mathbf{q}_{t_i}^s, \mathbf{h}_{t_j}^t)/\tau)}, \quad (2)$$

where $s(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$ computes the cosine similarity, and τ is a fixed temperature parameter.

Then we update the representation of token t_i in the memory bank with the moving-average of token representations from the quantized network:

$$\mathbf{q}_{t_i}^s \leftarrow m \mathbf{q}_{t_i}^s + (1 - m) \mathbf{h}_{t_i}^s, \quad (3)$$

where $m \in [0, 1)$ is the momentum coefficient that controls the smoothness of the token representation.

Besides, we use an additional distillation loss \mathcal{L}_{dist} over the logits. For the i -th token t_i , suppose the logits of the quantized and full-precision network are $\mathbf{z}_{t_i}^s, \mathbf{z}_{t_i}^t \in \mathbb{R}^{|V|}$, where $|V|$ is the vocabulary size. \mathcal{L}_{dist} is computed with the soft cross-entropy loss:

$$\mathcal{L}_{dist} = - \sum_{i=1}^n \mathbf{z}_{t_i}^t \log(\mathbf{z}_{t_i}^s). \quad (4)$$

Thus the total training loss is

$$\mathcal{L} = \lambda \mathcal{L}_{cont} + \mathcal{L}_{dist}, \quad (5)$$

where λ is a trade-off factor set as 0.1 by default.

Intuitively, for each token in the quantized network, \mathcal{L}_{dist} only encourages it to mimic its corresponding token of the teacher network, while \mathcal{L}_{cont} not only pulls it close to its positive, but also pushes it away from its negatives. In this way, \mathcal{L}_{cont} helps the student to capture more information from the

teacher’s representation, as is also theoretically discussed in Tian et al. (2019).

The proposed token-level contrastive distillation is crucial to the performance, and outperforms the sequence-level counterpart (as will be shown empirically in Section 5.1.1). We conjecture this is because (i) token-level contrast alleviates the problem of homogeneous word embedding (Figure 2) in the low-bit quantization; and (ii) similar to speech, the order of natural language is also sequential instead of spatial like images; and (iii) the self-attention mechanism allows other tokens to learn representations contextualized on the studied token, and these in-sequence negatives are harder than those from in-batch sequences, allowing more efficient representation learning.

3.2 Module-dependent Dynamic Scaling

Based on the observation of varied weight distribution in Section 2.2, we propose a simple-yet-effective dynamic scaling according to the statistics of each module weight. Specifically, instead of directly learning the original clipping factor α as PACT, we turn to learn a new scaling factor γ , which is multiplied with the average weight magnitude $\frac{\|\mathbf{w}\|_1}{n}$ to get clipping factor α :

$$\alpha = \gamma \cdot \frac{\|\mathbf{w}\|_1}{n}, \quad (6)$$

where $\|\cdot\|_1$ denotes ℓ_1 norm. The scaling γ is initialized as 1, which not only eases the initialization but also ensures the initial clipping factor α does not deviate far from the full-precision weights, regardless of the diversity of weight distribution.

Besides, we also design a more accurate gradient estimation of the scaling factor than PACT (Choi et al., 2018). Previous PACT only back propagates through weights whose absolute values are larger the clipping factor (*i.e.* $|w| \geq \alpha$). Instead, we also consider the weights inside the clipping range (*i.e.* $|w| < \alpha$) as:

$$\frac{\partial \ell}{\partial \gamma} = \begin{cases} \frac{\partial \ell}{\partial \mathbf{w}_q} Q(\mathbf{u}) \frac{\|\mathbf{w}\|_1}{n}, \mathbf{w} < -\alpha \\ \frac{\partial \ell}{\partial \mathbf{w}_q} \left[-\frac{\mathbf{w}}{\alpha} + Q(\mathbf{u})\right] \frac{\|\mathbf{w}\|_1}{n}, -\alpha \leq \mathbf{w} \leq \alpha, \\ \frac{\partial \ell}{\partial \mathbf{w}_q} Q(\mathbf{u}) \frac{\|\mathbf{w}\|_1}{n}, \mathbf{w} > \alpha \end{cases} \quad (7)$$

where ℓ is the total training loss and $\mathbf{u} = \text{clip}(\mathbf{w}, -\alpha, \alpha)/\alpha$ in Eq. (1). The detailed derivation can be found in Appendix A.

Intuitively, the update of clipping factor should be influenced by both weights outside and inside

$[-\alpha, \alpha]$, since α controls the quantization error of both, *i.e.*, a large clipping factor results in small quantization error for weights outside $[-\alpha, \alpha]$, while large error for weights inside. Our new estimation of the gradient of γ in Eq. (7) considers weights both outside and inside $[-\alpha, \alpha]$. Additionally, the proposed scaling is less sensitive to the varied distribution of weight than PACT, since the gradient of scaling $\frac{\partial \ell}{\partial \gamma}$ is proportional to the average weight magnitude $\frac{\|\mathbf{w}\|_1}{n}$.

4 Experiments

4.1 Setup

Tasks and Models. In this section, we evaluate the efficacy of our proposed quantization method on three kinds of generative tasks on two kinds of generative pre-training models. Specifically, we perform the proposed quantization approach on language modeling and next utterance prediction tasks on GPT-2 (Radford and Narasimhan, 2018), and abstractive summarization using BART (Lewis et al., 2020), and call the resultant models QuantGPT and QuantBART. The token-level contrastive distillation is performed on the hidden states of the last layer of GPT-2 or the BART decoder. More details about the datasets and model architectures can be found in Appendix B.1 and B.2.

Implementation Details. For each downstream task with our proposed method, we first fine-tune a full-precision network using the pre-trained checkpoint from huggingface¹ for both GPT-2 and BART. Then we use this fine-tuned network as the full-precision teacher network and to initialize the quantized student network. We train each task with 8 V100 GPUs based on the Pytorch framework. The detailed hyper-parameters for each task are available in Appendix B.3.

Compared Methods. Since there are very few attempts to compress generative PLMs, we self-implement three baseline quantization methods PACT (Choi et al., 2018), LSQ (Esser et al., 2020) and LAQ (Hou and Kwok, 2018) for comparison. Details about these methods are in Appendix B.4.

4.2 Language Modeling

The task of language modeling is to predict the probability distribution over a sequence of words.

¹<http://huggingface.co/models>

Method	#Bits (W-E-A)	Size (MB) (↓)	WikiText2 PPL (↓)	PTB PPL (↓)	WikiText103 PPL (↓)	Persona-Chat Acc(%) (↑)
-	<i>full-prec.</i>	474.9	14.48	14.72	14.19	77.01
<i>PACT</i>	8-8-8	121.4	17.49	16.11	16.76	74.73
<i>LSQ</i>	8-8-8	121.4	16.75	15.43	15.24	75.28
<i>LAQ</i>	8-8-8	121.4	16.91	15.87	15.88	76.02
<i>QuantGPT</i>	8-8-8	121.4	15.31	14.90	14.58	76.12
<i>PACT</i>	4-4-8	62.4	19.23	20.17	20.15	25.13
<i>LSQ</i>	4-4-8	62.4	78.99	79.76	75.12	45.10
<i>LAQ</i>	4-4-8	62.4	17.12	16.55	16.91	71.71
<i>QuantGPT</i>	4-4-8	62.4	15.55	14.95	15.31	76.57
<i>PACT</i>	2-2-8	33.0	173.02	189.13	171.03	5.52
<i>LSQ</i>	2-2-8	33.0	847.54	544.98	1470.86	5.54
<i>LAQ</i>	2-2-8	33.0	19.15	18.25	18.97	71.36
<i>QuantGPT</i>	2-2-8	33.0	17.30	16.12	16.98	74.78

Table 1: Results of language modeling on the test set of WikiText2, PTB and WikiText103 datasets, and next utterance prediction on the validation set of Persona-Chat dataset, with quantized GPT-2. “#Bits (W-E-A)” represents the bit-width for weights of Transformer layers, word embedding, and activations.

For language modeling, we experiment on WikiText2 (Merity et al., 2016), Penn Treebank (PTB) (Mikolov and Zweig, 2012) and WikiText103 (Merity et al., 2016). We use perplexity (PPL) to evaluate the performance for language modeling.

Comparison with the Full-precision Model.

From Table 1, the performance of the proposed method with 8-bit weight is comparable to the full-precision counterpart on PTB and WikiText103, while drops slightly on WikiText2. A slightly more severe performance drop is observed as the bit-width decreases from 8 to 4, with a drop of around 1 PPL point on WikiText2 and WikiText103, and less than 0.1 PPL point on PTB. When the bit-width of weight further goes down to 2, our method has an average of 2 PPL points drop, but achieves 14.4× model size reduction.

Comparison with Other Quantization Methods.

From Table 1, our method outperforms PACT, LSQ and LAQ for all bit-widths and tasks. As the bit-width decreases from 8 to 4, the PPL of LSQ greatly increases, with the average PPL of LSQ increasing by over 5 times. As the bit-width further decreases to 2, both LSQ and PACT fail on all datasets, despite their good performance on understanding tasks on BERT (Bai et al., 2021). We conjecture it is because though both PACT and LSQ have learnable parameters, the accumulated quantization error of generative PLMs makes the updates of these parameters by gradient descent less stable. On the other hand, the proposed module-wise dynamic scaling alleviates the problem.

Comparison with Other Compression Methods.

In Table 2, we compare our quantization method

Method	Size (MB)(↓)	WikiText2 PPL(↓)	PTB PPL(↓)	WikiText103 PPL(↓)
<i>full-prec.</i>	474.9 (1.0x)	14.4	14.6	13.9
KnGPT2	332.0 (1.4x)	-	-	20.5
DistilGPT2	329.6 (1.4x)	-	-	21.1
LightPAFF	268.0 (1.8x)	18.8	22.8	16.4
<i>Ours(8-8-8)</i>	121.4 (3.9x)	15.3	14.9	14.6
<i>Ours(4-4-8)</i>	62.4 (7.6x)	15.6	15.0	15.3
<i>Ours(2-2-8)</i>	33.0 (14.4x)	17.3	16.1	17.0

Table 2: Comparison between our proposed quantization method and other compression methods on GPT-2.

against recent GPT-2 compression methods, including tensor decomposition method KnGPT2 (Edalati et al., 2021), as well as distillation methods DistilGPT2 and LightPAFF (Song et al., 2020). From the comparison, our method outperforms the others in terms of model size and performance, even when weights are compressed to only 2 bits.

4.3 Next Utterance Prediction

The task of next utterance prediction predicts the next utterance given the dialogue context. It tests the language understanding ability of generative models. For this task, we use a large-scale dialogue dataset, Persona-Chat (Zhang et al., 2018).

From Table 1, all quantization methods incur a clear performance drop compared to the full-precision baseline, even in the 8-bit setting. As the quantization becomes more aggressive, *i.e.*, the bit-width gets smaller, the performance of PACT and LAQ decrease more significantly than ours. In particular, LSQ diverges for 2-bit weight and its accuracy is only 5%, which is no better than a random guess as there are 20 classes.

4.4 Abstractive Summarization

Abstractive summarization aims at generating a terse summary that captures the main ideas of the source article. We experiment on XSum (Narayan et al., 2018), whose ground-truth summarizations are highly abstractive and are challenging for many extractive strategies. ROUGE 1, 2, L are used to evaluate the performance of this task.

Method	#Bits (W-E-A)	Size (MB)(↓)	XSum		
Metric			R1 (↑)	R2 (↑)	RL (↑)
-	<i>full-prec.</i>	532.0	40.75	18.10	33.05
<i>PACT</i>	8-8-8	138.1	39.16	16.60	31.60
<i>LSQ</i>	8-8-8	138.1	39.09	16.72	31.56
<i>LAQ</i>	8-8-8	138.1	39.10	16.74	31.65
<i>QuantBART</i>	8-8-8	138.1	40.25	17.78	32.70
<i>PACT</i>	4-4-8	72.4	32.68	11.52	26.03
<i>LSQ</i>	4-4-8	72.4	38.94	16.48	31.46
<i>LAQ</i>	4-4-8	72.4	39.03	16.68	31.63
<i>QuantBART</i>	4-4-8	72.4	40.24	17.71	32.69
<i>PACT</i>	2-2-8	39.6	7.76	1.30	6.96
<i>LSQ</i>	2-2-8	39.6	37.09	14.88	29.76
<i>LAQ</i>	2-2-8	39.6	37.48	15.27	30.13
<i>QuantBART</i>	2-2-8	39.6	39.15	16.72	31.72

Table 3: Results of abstractive summarization on the test set of the XSum dataset, with quantized BART.

Table 3 shows the results of the abstractive summarization. As can be seen, our method constantly outperforms other methods again with a clear margin. Example generated summarizations of different methods in Appendix C.2 show that the summaries generated by QuantBART are logical and terse, while those from PACT have repeated texts.

5 Discussion

5.1 Ablation on Contrastive Learning

5.1.1 Choices of Negative Sampling

As shown in Figure 6, we ablate on how to choose negative samples in contrastive learning. Specifically, we compare our method with variants of token-level contrastive learning, which select negative samples of each token from (a) representations of other tokens in both the full-precision and quantized networks (*fp+quan.*); (b) representations of other tokens in the quantized network (*quan. only.*); and (c) the whole vocabulary randomly for each training iteration (*global.*). Besides, we compare with (d) sequence-level contrastive learning by pulling together representations of the same sequence, and pushing away representations of differ-

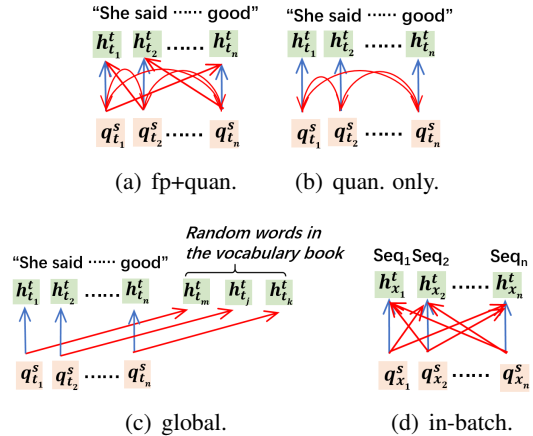


Figure 6: Four variants of negative sampling.

	Sampling method	WikiText2	PTB	WikiText103
-	<i>QuantGPT</i>	17.30	16.12	16.98
Tok-level	fp+quan.	17.38	16.51	17.13
	quan. only	17.35	16.54	17.15
	global	17.71	16.63	17.55
Seq-level	in-batch (bz=32)	17.62	19.23	18.97
	in-batch (bz=16)	17.48	17.11	18.16

Table 4: Ablation study on negative sampling for 2-bit weight, “bz” denotes for the batch size. “Tok” and “Seq” are abbreviation for token and sequence, respectively.

ent ones from the teacher network (*in-batch*). Representation of a sequence is defined as the mean of representations of all tokens in the sequence.

From Table 4, “*fp+quan.*” and “*quan. only.*” performs worse than QuantGPT, which uses full-precision representations of other tokens as negative samples. This indicates that noisy representations of tokens from the not-fully-trained quantized network may not be sufficient. “*global.*” performs even worse, which we conjecture is because, for one token, negative tokens chosen from the same sequence are contextually related to it and more informative than random tokens. “*in-batch.*” performs worse than all token-level variants, which may be because generative tasks make predictions in a token-wise manner and rely heavily in finer-grained token-wise representations. Interestingly, contrary to in-batch negative sampling in computer vision (Chen et al., 2020), we find that reducing the number of negative samples by reducing the batch size from 32 to 16 slightly improves performance.

5.1.2 Number of Negative Samples

In Figure 7, we plot the PPL of 2-bit QuantGPT on the PTB dataset, with varying number of negative samples. We plot the mean results with standard

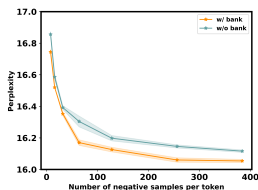


Figure 7: Effect of the number of negative samples.

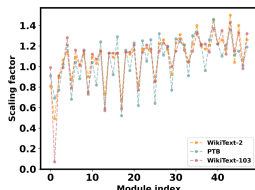


Figure 8: Scaling factors in the 2-bit QuantGPT.

Training loss	Training time (sec/iter) (\downarrow)	Memory (MB) (\downarrow)	PPL (\downarrow)
\mathcal{L}_{dist}	0.61	14700	16.93
$\mathcal{L}_{dist} + \lambda\mathcal{L}_{cont}$	0.67	14839	16.12

Table 5: Efficiency study of the token-level contrastive learning. The results are reported on the PTB dataset on 2-bit GPT-2. “sec/iter” means the needed time in seconds per iteration. Memory denotes the GPU consumption per device.

deviations from 5 independent runs. As can be seen, the performance improves and converges gradually as the number of negative samples increases.

Figure 7 also shows that using the moving-average representations ($\mathbf{q}_{t_i}^s$ in Eq. (3)) of negative samples in the memory bank has better performance than using the immediate representations ($\mathbf{h}_{t_i}^s$ in Eq. (3)), because of a smoother and more consistent representation of tokens.

5.1.3 Training Cost of the Contrastive Loss

In Table 5, we report the training speed and memory consumption of training the GPT-2 model on the PTB dataset with and without the proposed token-level contrastive loss. Batch size is set as 4 per device, which can be increased by using GPUs with larger memory or reducing the sequence length of samples. As can be seen, with the proposed token-level contrastive loss, the performance clearly improves with only slightly slower training speed and more memory consumption.

5.1.4 Representations for the Contrastive Loss

In Table 6, we compare the different representations to perform the contrastive loss. The “decoder-last” (resp. “decoder-first”) denotes performing the proposed token-level contrastive loss on the hidden states from the last decoder layer (resp. first decoder layer) followed by a linear transformation.

From Table 6, “decoder-last” performs better than “decoder-first”. A possible reason is that the hidden states of the last decoder blocks contain rich information from all previous layers (Xiong

et al., 2020). Since the experiments of abstractive summarization are conducted on BART, which has both encoder and decoder layers, we also study the contrastive loss on the “encoder-last” and “encoder-first”. In the ablation on the encoder, the contrastive loss \mathcal{L}_{cont} are computed on the source input (articles), instead of target input (summaries). From Table 6, “decoder-last” also has better ROUGE 1, 2, L values than other counterparts.

5.2 Ablation on Dynamic Scaling

Figure 8 shows the learned scaling γ of different modules in the 2-bit GPT-2 model. As can be seen, the scalings of different modules vary a lot, verifying the need for module-wise dynamic scaling.

In addition, we investigate the effect of the proposed dynamic scaling and the new estimation of the gradient in Eq. (7) with two variants: 1) \mathcal{L}_{dist} only which removes the token-level contrastive learning; and 2) *Ours with PACT* which removes the contrastive learning, and estimates the gradient with PACT which only considers the weights whose absolute values are larger than the clipping factor α . As shown in Table 7, the performance gets worse without contrastive learning to learn the distinguishable representations of tokens. The performance drops significantly when using PACT to estimate the gradient of the proposed scaling, especially for the WikiText103 dataset, verifying the efficacy of the new gradient estimation.

6 Related Work

Compression of Generative Pre-trained Language Models. Some early explorations compress the generative pre-trained language models. KnGPT2 (Edalati et al., 2021) applies the Kronecker decomposition to compress the GPT. DistilGPT2² distills a 12-layer GPT-2 to a 6-layer one, which is twice as fast during inference. LightPAFF (Song et al., 2020) proposes a distillation approach that the training loss is a combination of a maximum likelihood loss of the student model, and the KL divergence between the output of teacher and student models. SpAtten (Wang et al., 2021) proposes a sparse model with algorithm and architecture co-design, which removes uninformative tokens and attention heads. Compared with these methods, we not only study the difficulties of compression from the properties of generative tasks,

²<https://transformer.huggingface.co/model/distil-gpt2>

-	WikiText2	PTB	WikiText103	Persona-Chat	XSum		
Metric	PPL (\downarrow)	PPL (\downarrow)	PPL (\downarrow)	Acc(%) (\uparrow)	R1 (\uparrow)	R2 (\uparrow)	RL (\uparrow)
decoder-last	17.30	16.12	16.98	74.78	39.15	16.72	31.72
decoder-first	18.02	16.61	17.25	74.75	39.11	16.70	31.62
encoder-last	-	-	-	-	38.91	16.72	31.67
encoder-first	-	-	-	-	38.87	16.70	31.56

Table 6: Representations for the contrastive loss \mathcal{L}_{cont} in 2-bit setting. The “decoder-last” means the contrastive loss is computed on the hidden states from the last Transformer layer of the decoder after a linear transform. The naming format works for other variants.

Method	WikiText2	PTB	WikiText103
<i>QuantGPT</i>	17.30	16.12	16.98
\mathcal{L}_{dist} only	17.85	16.93	17.78
Ours with PACT	20.03	17.78	25.54

Table 7: Ablation study on the learning of the clipping factor with 2-bit GPT-2 on the language modeling task.

but also study both decoder and encoder-decoder generative models.

Quantization of Pre-trained Language Models.

Quantization compresses a model by representing the 32-bit floating-point parameter with a low-bit representation, and has been widely used in various domains as it does not require designing a new model architecture. There have been many attempts to quantize task-specific BERT models (Zafir et al., 2019; Shen et al., 2020; Zadeh et al., 2020) with only negligible performance drop on natural language understanding tasks. Recent works (Zhang et al., 2020; Bai et al., 2021) even push the weight bit-width down to as low as 1-bit. Despite the success of these approaches for BERT models, attempts to quantize generative PLMs are scarce, and the underlying difficulty remains unclear.

Contrastive Learning. Contrastive learning aims at pushing the representations of similar samples together while pulling those of dissimilar ones apart. and is widely used for large-scale self-supervised learning in various domains (Chen et al., 2020; Sun et al., 2020a; Baevski et al., 2020; Huang et al., 2022), and multi-modal learning (Radford et al., 2021; Jia et al., 2021). SimCLR (Chen et al., 2020) directly uses other in-batch samples as negatives, and sufficient large batch size is required to work well. MoCo (He et al., 2020) maintains a large number of negative samples in a queue and uses a moving average key encoder to improve consistency. Contrastive learning without negative samples is also proposed in BYOL (Grill et al.,

2020) and SimSiam (Chen and He, 2021). Contrastive representation distillation (Tian et al., 2019) distills the knowledge from the teacher network to the student network by maximizing the mutual information between them.

The closest work with our token-level contrastive distillation is Wav2vec 2.0 (Baevski et al., 2020), which use in-utterance representations at different positions as negatives in speech learning. Besides the difference in the modality and tasks, our method also differs from theirs in (1) *Model*: We quantize the model parameters and activations while they do not; (2) *Representation*: For each sample, we use the output of the full-precision and the quantized networks as its two views, while they use the quantized and the contextualized representation. (3) *Loss*: We calculate loss over all tokens in an auto-regressive manner, while they only calculate over the masked tokens non-autoregressively.

7 Conclusion

This paper studies low-bit quantization of generative PLMs. We find that the difficulty of quantizing generative PLMs lies in homogeneous word embedding and varied distribution of weights. To alleviate the two problems, we propose token-level contrastive learning to learn more distinguishable token embeddings, as well as a module-dependent dynamic scaling for more accurate quantization. Extensive experiments on language modeling, next utterance prediction and abstractive summarization demonstrate the efficacy of our proposed method. We hope our work sheds a light on the compression of generative PLMs in future exploration.

Acknowledgements

This work is supported in part by the General Research Fund (GRF) project 17206020, and in part by ACCESS, AI Chip Center for Emerging Smart Systems, Hong Kong SAR.

References

- Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33.
- Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jing Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. 2021. Binarybert: Pushing the limit of bert quantization. In *Annual Meeting of the Association for Computational Linguistics*.
- Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. Technical Report arXiv:1308.3432.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*.
- Cong Chen, Chaofan Tao, and Ngai Wong. 2021. Litegt: Efficient and lightweight graph transformers. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 161–170.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607.
- Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758.
- Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. 2018. Pact: Parameterized clipping activation for quantized neural networks. Preprint arXiv:1805.06085.
- Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2015. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, pages 3123–3131.
- Ali Edalati, Marzieh Tahaei, Ahmad Rashid, Vahid Partovi Nia, James J Clark, and Mehdi Rezagholizadeh. 2021. Kronecker decomposition for gpt compression. In *Advances in Neural Information Processing Systems*.
- Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. 2020. Learned step size quantization. In *International Conference on Learning Representations*.
- Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. In *Neural Information Processing Systems*.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.
- Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). Technical Report arXiv:1606.08415.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. Technical Report arXiv:1503.02531.
- Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic bert with adaptive width and depth. In *Advances in Neural Information Processing Systems*, volume 33.
- Lu Hou and James T. Kwok. 2018. Loss-aware weight quantization of deep networks. In *International Conference on Learning Representations*.
- Lu Hou, Yao Quanming, and James T. Kwok. 2017. Loss-aware binarization of deep networks. In *International Conference on Learning Representations*.
- Wenyong Huang, Zhenhe Zhang, Yu Ting Yeung, Xin Jiang, and Qun Liu. 2022. Spiral: Self-supervised perturbation-invariant representation learning for speech pre-training. In *International Conference on Learning Representations*.
- Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.
- Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *IEEE Spoken Language Technology Workshop*, pages 234–239. IEEE.
- Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.
- Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.
- Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Q-bert: Hessian based ultra low precision quantization of bert. In *AAAI Conference on Artificial Intelligence*.
- Kaitao Song, Hao Sun, Xu Tan, Tao Qin, Jianfeng Lu, Hongzhi Liu, and Tie-Yan Liu. 2020. Lightpaff: A two-stage distillation framework for pre-training and fine-tuning. Preprint arXiv:2004.12817.
- Siqi Sun, Zhe Gan, Yuwei Fang, Yu Cheng, Shuohang Wang, and Jingjing Liu. 2020a. Contrastive distillation on intermediate representations for language model compression. In *Conference on Empirical Methods in Natural Language Processing*, pages 498–508.
- Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020b. Mobilebert: a compact task-agnostic bert for resource-limited devices. In *Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170.
- Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. Contrastive representation distillation. In *International Conference on Learning Representations*.
- Hanrui Wang, Zhekai Zhang, and Song Han. 2021. Spatten: Efficient sparse attention architecture with cascade token and head pruning. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 97–110. IEEE.
- Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tiejun Liu. 2020. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR.
- Ali Hadi Zadeh, Isak Edo, Omar Mohamed Awad, and Andreas Moshovos. 2020. Gobo: Quantizing attention-based nlp models for low latency and energy efficient inference. In *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 811–824. IEEE.
- Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. Preprint arXiv:1910.06188.
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213.
- Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. 2020. Ternarybert: Distillation-aware ultra-low bit bert. In *Conference on Empirical Methods in Natural Language Processing*.

A Derivation of Gradient of Dynamic Scaling

In this section, we provide the derivation of the gradient of the proposed dynamic scaling γ . The quantization in the forward process can be written as

$$\begin{aligned}\alpha &= \frac{\|\mathbf{w}\|_1}{n}\gamma, \\ \mathbf{u} &= \text{clip}(\mathbf{w}, -\alpha, +\alpha)/\alpha, \\ \mathbf{w}_q &= Q(\mathbf{u})\alpha,\end{aligned}$$

where $Q(\cdot)$ is an uniform quantization function as described in Section 2.2. Based on the chain rule, the gradient of scaling γ w.r.t. the training loss function ℓ is:

$$\begin{aligned}\frac{\partial \ell}{\partial \gamma} &= \frac{\partial \ell}{\partial \mathbf{w}_q} \left[\frac{\partial \mathbf{w}_q}{\partial Q(\mathbf{u})} \frac{\partial Q(\mathbf{u})}{\partial \alpha} \frac{\partial \alpha}{\partial \gamma} + \frac{\partial \mathbf{w}_q}{\partial \alpha} \frac{\partial \alpha}{\partial \gamma} \right] \\ &= \frac{\partial \ell}{\partial \mathbf{w}_q} \left[\alpha \frac{\partial Q(\mathbf{u})}{\partial \alpha} \frac{\|\mathbf{w}\|_1}{n} + Q(\mathbf{u}) \frac{\|\mathbf{w}\|_1}{n} \right] \quad (8) \\ &= \frac{\partial \ell}{\partial \mathbf{w}_q} \left[\alpha \frac{\partial Q(\mathbf{u})}{\partial \alpha} + Q(\mathbf{u}) \right] \frac{\|\mathbf{w}\|_1}{n}.\end{aligned}$$

We use straight through estimator (STE) to estimate the gradient of uniform quantizer $Q(\cdot)$, *i.e.*, $\forall i, \frac{\partial Q(\mathbf{u}_i)}{\partial \mathbf{u}_i} = 1$. Thus the gradient $\frac{\partial Q(\mathbf{u})}{\partial \alpha}$ can be written as:

$$\frac{\partial Q(\mathbf{u})}{\partial \alpha} = \frac{\partial Q(\mathbf{u})}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \alpha} = \begin{cases} 0, & \mathbf{w} \leq -\alpha \\ -\frac{\mathbf{w}}{\alpha^2}, & -\alpha < \mathbf{w} < \alpha \\ 0, & \mathbf{w} \geq \alpha \end{cases} \quad (9)$$

By combining Eq. (8) and Eq. (9), we get

$$\frac{\partial \ell}{\partial \gamma} = \begin{cases} \frac{\partial \ell}{\partial \mathbf{w}_q} Q(\mathbf{u}) \frac{\|\mathbf{w}\|_1}{n}, & \mathbf{w} \leq -\alpha \\ \frac{\partial \ell}{\partial \mathbf{w}_q} \left[-\frac{\mathbf{w}}{\alpha} + Q(\mathbf{u}) \right] \frac{\|\mathbf{w}\|_1}{n}, & -\alpha < \mathbf{w} < \alpha \\ \frac{\partial \ell}{\partial \mathbf{w}_q} Q(\mathbf{u}) \frac{\|\mathbf{w}\|_1}{n}, & \mathbf{w} \geq \alpha \end{cases}$$

where $\frac{\partial \ell}{\partial \gamma}$ considers both the weight inside and outside the clipping value, and proportional to the weight magnitude $\frac{\|\mathbf{w}\|_1}{n}$.

B More Experimental Settings

B.1 Datasets

The train/val/test splits for different datasets are shown on Table 8.

B.2 Model Architectures

GPT-2. The vocabulary size of GPT-2 is 50527. We use GPT-2-small with 12 decoder layers and hidden state dimension as 768, for experiments

Dataset	Training	Validation	Test
WikiText2	36,717	3,760	4,358
PTB	42,068	3,370	3,761
WikiText103	1,801,350	3,760	4,358
Persona-Chat	8,939	1,000	968
XSum	204,045	11,332	11,334

Table 8: Data splits of different datasets.

in Sections 2.2, 4 and 5. GeLU (Hendrycks and Gimpel, 2016) is used as the activation function. In the experiments of Appendix C.1, we adopt GPT-2-base with 24 decoder layers and hidden state dimension as 1024, to evaluate the quantization ability on larger models.

BART. The vocabulary size of BART is 50265. We use BART-base with 6 encoder layers, 6 decoder layers and hidden state dimension as 768 for experiments in Section 4. In the experiments of Appendix C.1, we adopt BART-large with 12 encoder layers, 12 decoder layers and hidden state dimension 1024, to evaluate the quantization ability on larger models.

B.3 Hyperparameters

Language Modeling. The sequence length is 512. The learning rate is initialized to 0.0005 (resp. 0.001) for the GPT-2 backbone parameters (resp. clipping factor γ) and then linearly decays to 0. The number of negative samples in each sequence is 64 for the PTB dataset, and 32 for the WikiText2 and WikiText103. The temperature τ and momentum coefficient m is 0.1 and 0.5 respectively. We train with the AdamW optimizer (Loshchilov and Hutter, 2017) with batch size 32. The training epochs for WikiText2, PTB and WikiText103 are set as 80, 120, 8, respectively.

Next Utterance Prediction. The sequence length is 512. The learning rate is initialized to 0.0005 (resp. 0.001) for the GPT-2 backbone parameters (resp. clipping factor γ) and then linearly decays to 0. The number of negative samples in each sequence is 32. The temperature τ and momentum coefficient m is 0.1 and 0.5, respectively. We train with the AdamW optimizer with batch size 16, for a total of 2 epochs.

Abstractive Summarization. We set the length of the source sequence (articles) as 512, and pad the target sequence (summaries) to maximum length. We use beam search to generate summaries, with beam size 6 and length penalty 1. The learning rate

Method	#Bits (W-E-A)	Size (MB)(↓)	WikiText2	PTB	WikiText103	Size (MB)(↓)	XSum		
Metric			PPL (↓)	PPL (↓)	PPL (↓)		R1 (↑)	R2 (↑)	RL (↑)
-	<i>full-prec.</i>	1353.7	12.46	12.35	12.37	1550.0	45.25	22.11	37.07
PACT	8-8-8	342.5	12.86	13.95	13.90	394.8	43.55	20.57	35.55
<i>Ours</i>	8-8-8	342.5	12.53	12.40	12.68	394.8	44.34	21.41	36.32
PACT	4-4-8	174.0	16.10	14.19	18.07	202.2	19.45	3.53	15.58
<i>Ours</i>	4-4-8	174.0	13.34	12.41	14.12	202.2	44.18	21.31	36.25
PACT	2-2-8	89.7	98.74	68.55	86.60	106.0	8.53	0.93	7.25
<i>Ours</i>	2-2-8	89.7	14.53	13.22	14.52	106.0	42.38	19.75	34.57

Table 9: Ablation study on larger models. We report the results on 24-layer GPT-2 and 24-layer BART.

is initialized to 0.0002 (resp. 0.001) for the BART backbone parameters (resp. clipping factor γ) and then linearly decays to 0. The number of negative samples is 32. The temperature τ and momentum coefficient m is 0.1 and 0.5, respectively. We train with the AdamW optimizer with batch size 128, for a total of 8 epochs.

B.4 Description of the Compared Methods

PACT. PACT (Choi et al., 2018) learns a learnable clipping factor for each module by gradient descent. To make the quantization more accurate, we adopt a flexible variant of the original PACT, with different positive and negative clipping factors $[-\alpha_{neg}, \alpha_{pos}]$, where both α_{neg} and α_{pos} are initialized as 2.5.

LSQ. LSQ (Esser et al., 2020) learns the step-size of quantizer for each module by gradient descent. We use the recommended initialization strategy of the step size as (Esser et al., 2020).

LAQ. LAQ (Hou et al., 2017; Hou and Kwok, 2018) is a loss-aware quantization method that views quantization as an optimization problem and solve it via proximal Newton algorithm. We use the approximate solver in (Hou and Kwok, 2018) to compute the quantized weights before each forward propagation.

For the self-implemented methods PACT, LSQ and LAQ, we adopt the commonly-used distillation loss adopted in (Hinton et al., 2015; Jiao et al., 2020). Note that these methods are only used for weights and embeddings, while the activations of these methods follow the same setting as our proposed method in Section 2.1. We also tried using the original language modeling loss w.r.t. the ground-truth labels, and distillation loss over the the attention as (Jiao et al., 2020). However, these two losses worsens the performance on all three

methods.

B.5 Frameworks of Double-head GPT-2 and BART

Since we adopt double-head GPT-2 and BART for next utterance prediction and abstractive summarization, the frameworks for these tasks are slightly modified from that on language modeling due to the difference of tasks. In Figure 9 and 10, we illustrate the framework for double-head GPT-2 and BART, respectively. In the double-head GPT-2, we also quantize the final linear layer in the output head.

C More Experimental Results

C.1 Performance of Larger Models

In Table 9, we experiment with GPT-base and BART-large, which both have 24 Transformer layers. For all bit-widths, the training of our method converges successfully without gradient exploding/vanishing problems. QuantGPT outperforms PACT by a large margin in all tasks, especially for 2-bit weight. Our quantization method on larger models also has better performance than that on 12-layer GPT-2 and 12-layer BART in Section 4.

C.2 Examples of Summarizations

In Table 10, we provide the example summarizations on the XSum dataset. By comparing the articles, references and generations, the generated summaries by our quantized model are more logical and terse than PACT, LSQ and LAQ, which face problems of homogeneous word embeddings to some extent as discussed in Section 2.2.

C.3 More Visualizations for the Token Representations

In Figure 11, we provide the visualizations of token representations on more samples. The observations

Article: On Tuesday, a BBC Spotlight programme revealed that eight children had gone missing in Northern Ireland. Two of the girls were Somali teenagers who disappeared in 2005 and 2012. The Health and Social Care Board has said new guidelines are in place and add that no children have gone missing since 2014. Separated children are children outside their country of origin and separated from their parents or legal guardian. The term can also include unaccompanied asylum-seeking children and trafficked children. When they arrive in Northern Ireland they are taken into the care of the local health trust. Eight children have gone missing since 2005 and they remain missing. The SDLP's health spokesman Mark H Durkan said he would be raising the issue at the Northern Ireland Assembly's health committee and his party colleague Alex Attwood would be raising it at the justice committee. "The number of children who cannot be accounted for is something that needs urgent inquiry and investigation," he said. "There is a lot of very good work being done to look after the welfare of unaccompanied young people, but clearly we now have some very big questions that need to be answered." Ulster Unionist MLA Jo-Anne Dobson said it was "frankly appalling" to hear that eight children had gone missing. "I have written to Health Minister Michelle O'Neill on this issue to seek further clarification and to demand details of how the department, health trusts and the Health and Social Care Board have sought to address each of the cases involved in the investigation," she added. The Green Party leader Steven Agnew also said it was extremely worrying that children can disappear without a trace. Paula Bradshaw from the Alliance Party added that the health trusts and police "need to work closer over the handling of these cases". In a statement, the Police Ombudsman for Northern Ireland said: "Our director of investigations will be reviewing the contents of the programme to ascertain if there are any issues of police conduct which may need further investigation." The Police Service of Northern Ireland has said that in the two cases identified in the programme, investigations were robust and all information available at the time was followed. The Health and Social Care Board has said that new guidelines are in place and stress that no children have gone missing since 2014. BBC Spotlight's investigation is now available on BBC iPlayer.

Reference: An urgent inquiry is needed into separated children who have gone missing from care, the Social Democratic and Labour Party has said.

PACT: TheTheAAATheTheTheAnAnAnTheThe an an an been been been been jailed.

LSQ: The SDLP has called for an urgent inquiry into the welfare of unaccompanied children in Northern Ireland.

LAQ: The SDLP has called for "urgent inquiry and investigation" into the handling of unaccompanied children in Northern Ireland.

Ours: The SDLP is calling for an urgent inquiry and investigation into the disappearance of unaccompanied young people.

Article: The dairies operation, which processes and distributes milk, is being sold to Germany's Mueller for Â£80m. It comes as profits at the UK's largest dairy food company fell 95% to Â£900,000 in the six months to September. Dairy Crest processes and delivers around 1.3 billion litres of milk a year for retailers and homes. Dairy Crest said in a statement that the deal was in the best interests of consumers, customers and dairy farmers. The dairies business accounts for about 70% of the company's revenues, which rose 1% to Â£682.1m during the six months. After the sale, which still needs shareholder approval and could take several months, Dairy Crest will focus on its profitable cheese and spreads operations. There are about 14,000 dairy farmers in the UK, producing 3.3 million litres a day. However, with milk prices having fallen, there has been much debate about whether the economics of the industry are sustainable. Investors approved of the Dairy Crest's decision to get out of a loss-making sector, sending its shares 10% higher in morning trading on Thursday. Muller said the deal would lead to lower costs and larger exports of dairy products made in the UK. Ronald Kers, chief executive of Muller UK & Ireland, said: "We are concerned that the dynamics of the UK fresh milk market are unsustainable for dairy processors in the mid to long term and this acquisition will allow us to reduce our costs, increase our efficiencies and invest in the future." Under the deal, Mueller's UK division - Muller Wiseman Dairies - will take over factories at Foston, in Derbyshire, Chadwell Heath, in Essex, and Severnside, near Gloucester. The deal also includes the Hanworth glass bottling site in Middlesex, where Dairy Crest is consulting with employees on the site's future, and 72 depots. Muller bought Robert Wiseman in 2012.

Reference: Dairy Crest, maker of Cathedral City cheese and Country Life butter, has announced a big slump in profits and the sale of its milk business.

PACT: More than than more more more than more than than to be be be will will will be be are are are be be to the.

LSQ: Dairy Crest is to sell its Dairies business to a German company for an undisclosed sum.

LAQ: Dairy giant Dairy Crest is to sell its UK business to a German company for an undisclosed sum.

Ours: Dairy Crest, the world's largest dairy producer, is to sell its UK operations to a German firm.

Table 10: Example summaries generated by 2-bit BART quantized with different methods.

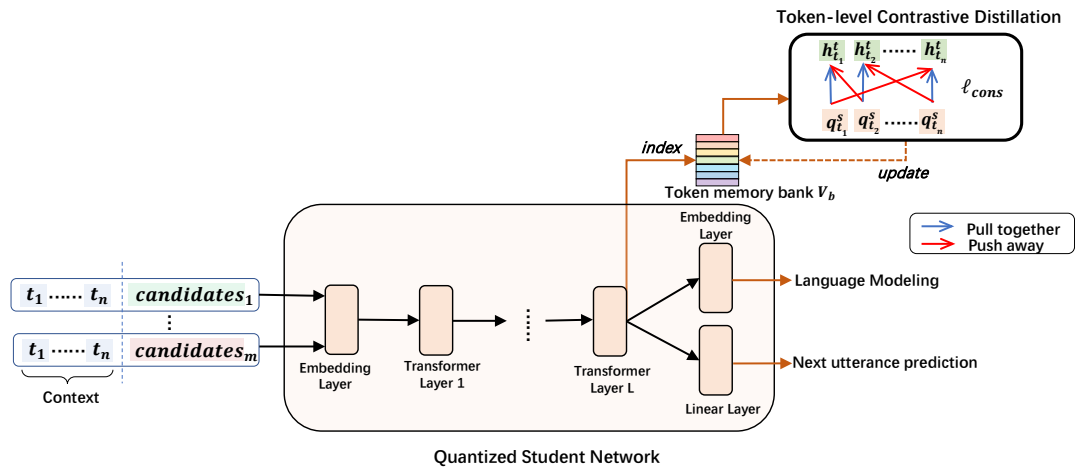


Figure 9: The training workflow of the proposed method for double-head GPT-2 quantization in the task of next utterance prediction. The model is trained to find the correct candidate. The full-precision teacher network and distillation loss \mathcal{L}_{dist} are omitted for simplicity.

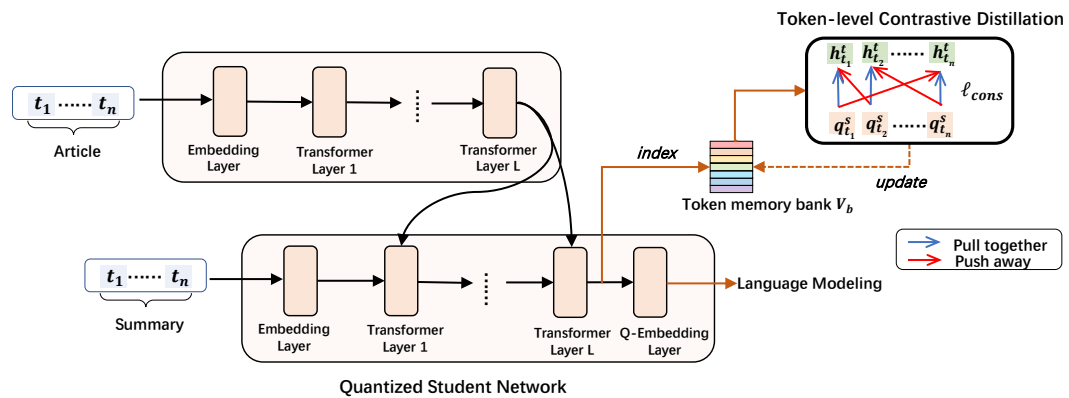
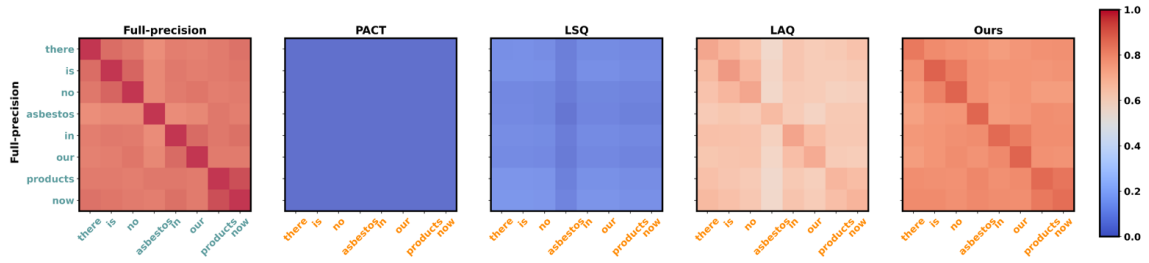
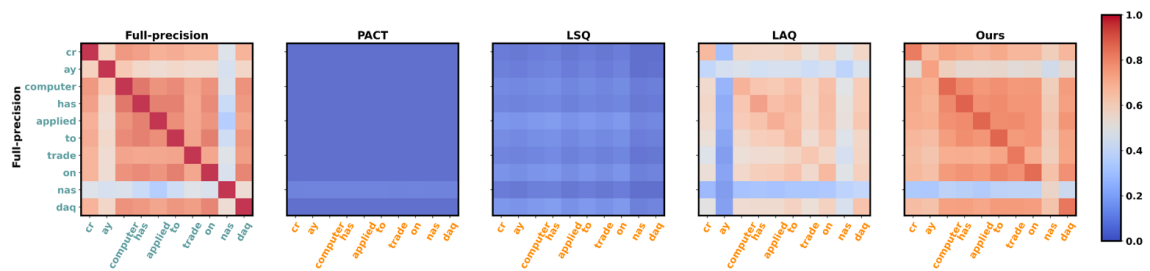


Figure 10: The training workflow of the proposed method for BART quantization in the task of abstractive summarization. The full-precision teacher network and distillation loss \mathcal{L}_{dist} are omitted for simplicity.

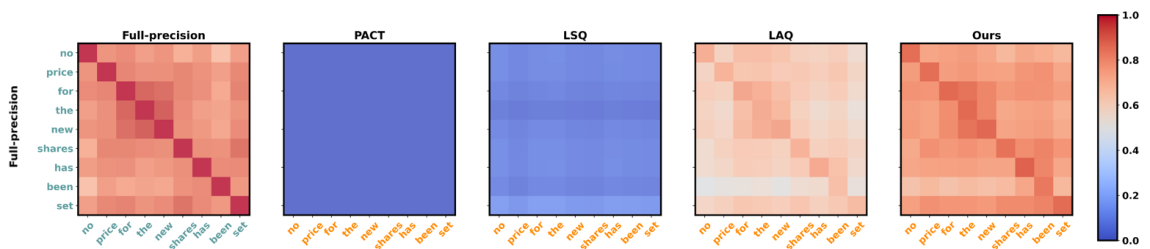
are similar to those in Section 2.2.



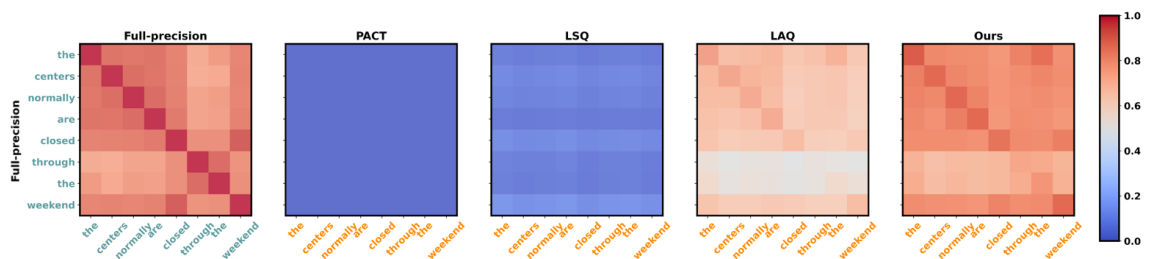
(a) "there is no asbestos in our products now"



(b) "cray computer has applied to trade on nasdaq"



(c) "no price for the new shares has been set"



(d) "the centers normally are closed through the weekend"

Figure 11: More Visualizations: matrices representing the cosine similarities between representations of all pairs of tokens in a sentence, between the full-precision model and 2-bit quantized models trained on PTB dataset. Token representations at the last decoder layer of GPT-2 are used.