Knowledge Distillation for Efficient Training of Quantized LLMs

Abstract

Quantizing Large Language Models (LLMs) to low-bit formats is an effective strategy for reducing memory and computational cost. While 4 bit quantization has been widely studied and easily applied post-training, the behavior of LLMs under more extreme compression—particularly at 1–2 bits setting remains challenging without expensive training and large amounts of data.

In this paper, we investigate the feasibility of BitDistiller (Du et al., 2024), a novel compute efficient method for training quantized 2-bit and 3-bit LLMs with knowledge distillation, in extending to smaller models and the 1-bit regime. We also explore the effect of the teacher model on the quantized model performance, which had counterintuitive results in the original paper. Our findings clarify the limits of low-bit training and provide guidance for future work aiming to deploy highly compressed LLMs.

1 Introduction

As Large Language Models (LLMs) grow in scale, their deployment becomes increasingly constrained by memory, compute, and latency requirements. Quantization - a technique that downcasts model weights to lower precisions - has emerged as a promising direction for improving inference efficiency (Banner et al., 2019; Frantar et al., 2023; Lin et al., 2024). However, aggressive quantization, particularly below 2-bit precision, often results in performance degradation due to limited model expressivity (Li et al., 2025; Xu et al., 2024).

Post-training Quantization (PTQ) methods, which apply quantization after model pre-training, are effective above 4 bits (Banner et al., 2019) quickly degrade in the ultra low-bit regime (Zhao et al., 2025). Contrastingly, quantization-aware training (QAT) integrates quantization into the training process, which enables models to adapt to representational constraints, even for 1-bit quantization (Wang et al., 2023; Ma et al., 2024a; Xu et al., 2024). However, these methods are very compute intensive, often requiring training from scratch and/or large amounts of data similar to pretraining.

Knowledge Distillation (Hinton et al., 2015; Kim et al., 2019) presents a promising direction to address the issue of training compute for training ultra-low bit quantized models, allowing the superior embeddings of larger models to be efficiently transferred into smaller models. Du et al. (2024) introduced an efficient QAT method *BitDistiller* which relied purely on supervised finetuning for training 2-bit and 3-bit quantized LLMs with Knowledge Distillation. While their approach was successful and efficient, their approach was only tested on larger 7B and 13B models, which already have solid world knowledge, sufficient to reliably perform mathematical reasoning (Li et al., 2024). Their method was also not tested in the 1-bit regime, which was left as future work. Counterintuitively, using a larger teacher model with BitDistiller also degraded performance of the distilled quantized models, though there was insufficient data to make any conclusions.

In this report, we would like to address the previously mentioned unexplored directions in the following research questions:[1]

1. **Does BitDistiller remain effective at small scale?** Can the method preserve a similar level of performance of the full-precision model when applied to lightweight models (TinyLlama-1.1B, and LlaMA-3.2B) in the 2- and 3-bit quantization regime?

2. **Can BitDistiller be extended to 1-bit precision?** Existing methods at this level often rely on pretraining from scratch. We evaluate whether BitDistiller's QAT strategy can recover expressivity at ultra-low bit widths and offer a much more efficient training method.

3. **How does teacher model size influence distillation over different model sizes?** Can we reproduce BitDistiller's counterintuitive result for different model sizes where self-distillation performed better than using a superior teacher model?

2 Background

In this section, we review the key elements of *BitDistiller*, a novel method introduced by Du et al.

[1]Find our code using the following link https://github.com/BrownianNotion/BitDistiller

(2024) for efficient knowledge distillation training of quantized 2-bit and 3-bit LLMs.

2.0.1 Asymmetric Clipping and Quantization

Du et al. (2024) adopt the following quantization for 2-bit LLMs, where

$$\text{INT-Asym}: Q(w) = \text{Round}\left(\frac{w - z}{s}\right),$$

where s is a scaling factor based on the number of bits and the range of values in the weight group, and z is a learned zero point (Zhang et al., 2023).

The weights are quantized in buckets of 128, where each bucket is clipped according to the following process for outliers to improve the quantized distribution's expressive capacity and remove outliers:

$$w_c = \text{Clip}(w, \alpha, \beta), \text{ where } \begin{cases} \alpha \in [\min\_\text{val}, 0) \\ \beta \in (0, \max\_\text{val}] \end{cases},$$

$$\alpha^*, \beta^* = \arg\min_{\alpha,\beta} \|Q(w_c)X - wX\|.$$

2.1 Distillation: Confidence-Aware KL Divergence

In addition to their novel asymmetric clipping method, Du et al. (2024) introduce a novel loss function specifically for knowledge distillation training that adapts to the confidence level of the teacher. The loss function "Confidence-Aware KL Divergence" (CAKLD) calculates a weighted average between the Forward and Reverse KL divergence:

$$\mathcal{D}_{\text{CAKLD}}(P_T \parallel P_S) = \gamma \mathcal{D}_{\text{KL}}(P_S \parallel P_T) + (1-\gamma)\mathcal{D}_{\text{KL}}(P_T \parallel P_S), \quad (1)$$

where P_S, P_T are the quantized student and teacher distributions respectively and the weight γ represents the confidence-level of the teacher, calculated by averaging the token probabilities over all samples in the dataset.

Both the forward and backward KL divergence are standard loss functions in Knowledge Distillation (see section 3) and Du et al. (2024) found that CAKLD was able to balance the advantages of both loss functions - the Backward KL divergence $\mathcal{D}_{\text{KL}}(P_S \parallel P_T)$ promotes mode-seeking behavior and performs best on instruction tuning (Agarwal et al., 2024) while the Forward KL divergence $\mathcal{D}_{\text{KL}}(P_T \parallel P_S)$ promotes which is mode covering behavior and performs better on general text generation (Narayan et al., 2018).

2.2 Quantization Aware Training

Du et al. (2024) then combine their novel clipping method to perform quantization aware training (see Figure 1). The clipped teacher model is used to initialize the weights of the student, the quantized weights are used to calculate the loss, and gradient updates are performed on the student's full precision weights, which are updated alongside the quantized weights.

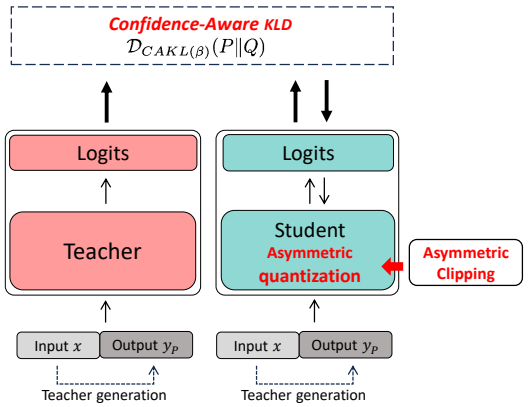


Figure 1: Overview of the BitDistiller Training method. Figure from Du et al. (2024).

3 Related Work

3.1 Weight Quantization and Clipping

Reducing the precision of LLM weights is a well-established strategy for improving inference efficiency. While post-training quantization techniques have achieved strong performance at 8-bit and 4-bit precision (Frantar et al., 2023), extending these methods to sub 4-bit regimes introduces significant challenges. Chief among these is the loss of representational capacity and the presence of heavy-tailed weight distributions, where a small number of high-magnitude weights dominate quantization error (Dettmers et al., 2023). Standard quantization methods struggle in these settings due to rounding error and insufficient granularity at low bit-widths.

To address this, quantization-aware training (QAT) methods introduce clipping strategies to mitigate the influence of outliers during training. BitDistiller (Du et al., 2024) proposed a learned asymmetric clipping strategy that limits extreme values while preserving high-weight fidelity. Such clipping approaches have proven especially important in sub-4-bit quantization, where small changes

in scale can cause disproportionately large errors. However, these techniques are inherently lossy and must be carefully tuned, particularly in small models where capacity is already constrained.

For 3-bit quantization, (Du et al., 2024) used the NF-Asym quantization which performed slightly better than Int-Asym. For ultra-low bit representations, 2-bit, BitDistiller worked with INT-Asym as it performed empirically better than NF-Asym.

Several recent efforts have explored 1-bit and 2-bit quantization. While early approaches adopted hybrid precision or residual representations to offset degradation (Ma et al., 2024a), more recent work has targeted full quantization. Notably, Wang et al. (2023) demonstrated that ternary weight sets could reach near-parity with full-precision baselines. However, BitNet required training from scratch and did not fully eliminate higher-precision components. In contrast, our work focuses on fine-tuning pretrained models to 1–2 bit precision using efficient, low-resource QAT strategies, and directly investigates the limits of quantization when model size and bitwidth are both constrained.

3.2 Knowledge Distillation

Knowledge distillation (KD) is a widely used technique in model compression, transferring the predictive behavior of a larger teacher model to a smaller student (Hinton et al., 2015). In quantized LLMs, KD has proven particularly useful for stabilizing training and compensating for the reduced representational capacity of low-bit models. Instead of learning from one-hot labels, the student is trained to match the teacher's output distribution, providing a richer and smoother learning signal.

BitDistiller (Du et al., 2024) integrates this principle into its QAT pipeline through a self-distillation mechanism, where the full-precision model teaches its own quantized counterpart. A novel confidence-aware KL divergence (CAKLD) objective interpolates between forward and reverse KL based on the teacher's confidence, balancing generalization and specificity. This approach yields strong performance in the 2–3 bit regime with limited compute and minimal data.

Despite these advances, the role of teacher size remains underexplored. BitDistiller found that a 13B teacher model did not outperform a 7B self-distilled teacher for a 7B student, suggesting diminishing returns or potential misalignment. Our findings extend this line of inquiry, showing that overly large teachers can degrade performance in small, low-bit students—likely due to mismatched logit distributions and the student's limited capacity to absorb high-entropy outputs. This underscores the need for more nuanced teacher-student design in extreme quantization regimes.

3.3 BitNet and FBI-LLM

Alternative approaches such as BitNet (Wang et al., 2023) and FBI-LLM (Ma et al., 2024a) explore training binarized LLMs from scratch. BitNet b1.58 constrains all weights to a ternary set {-1, 0, +1} during training, with group-wise scaling to preserve information across layers. This allowed a 1.3B model to reach full-precision performance when trained on large-scale corpora. However, BitNet's reliance on ternary weights (rather than true binary), lack of knowledge distillation, and full pretraining requirements limit its flexibility and applicability to existing models.

FBI-LLM adopts a similar from-scratch philosophy but introduces KD from a pretrained teacher throughout training. This enables fully binarized transformer blocks at up to 7B scale, albeit with several unquantized components (e.g., embeddings, layer norms). While effective, FBI-LLM requires massive compute and still shows a gap to full-precision performance.

In contrast, our work focuses on binarizing and fine-tuning pretrained models—providing a more practical path for model compression. We extend BitDistiller's QAT framework into the 1–2 bit regime, systematically explore the impact of teacher size, and demonstrate results on a variety of benchmarks, in low-capacity settings. Our ablations offer insight into failure modes and distillation dynamics at the limit of model compression.

4 Methodology

4.1 1-bit Quantization

For 1-bit quantization, we adopt the same `INT-Asym` quantizer from Du et al. (2024) which performed better on very low precision formats. The only change we introduce is to lower the quantization precision to 1 bit, effectively turning this rounded-to-nearest (RTN) (Xu et al., 2024) method into a threshold on each weight group.

4.2 Larger Teacher Models

To use larger teacher models for Knowledge Distillation training, we follow the setup of Du et al.

(2024) which aims to align the logit distribution of the teacher model with the smaller student model on the training dataset pre-generated by the larger teacher model.

5 Experiments

5.1 Experimental Setup

5.1.1 Metrics

We use the same range of QA benchmarks in Du et al. (2024) to measure the Natural Language Understanding (NLU) of our models as well as perplexity to gauge the next token prediction ability. The QA benchmarks include ARC-Easy/ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2019) and PIQA (Bisk et al., 2019). We elect to focus on NLU over maths or code given our exploration of small models, which would likely struggle in these domains.

5.1.2 Models

We use TinyLlama 1.1B (Zhang et al., 2024), Llama 3.2 3B (Touvron et al., 2023a) and Llama 2 7B (Touvron et al., 2023b) for our experiments. Llama 2 7B was used by Du et al. (2024) in their original experiments, and we have chosen our smaller models in the same family to keep the architecture and tokenizer fairly consistent.

5.1.3 Baselines

For our baselines, we use a full-precision model to measure the degradation induced by quantization and a model with randomly initialized weights since all of our QA benchmarks are multiple choice questions.

5.1.4 Data

All models were trained using a combination of language modelling and instruction tuning datasets. WikiText-2 (Merity et al., 2016) was used as the primary dataset for language modeling evaluation and training with subsets of Alpaca (Taori et al., 2023) also used.

5.1.5 Training Implementation

Following Du et al. (2024), we train each model for 4 epochs using the AdamW Optimizer (Loshchilov and Hutter, 2019) with zero weight decay, using a learning rate of 8e-6 for the 3B and 7B Llama models. We use a larger learning rate of 2e-5 for faster convergence on the smaller TinyLlama model. We keep the same batch size of 16 and hyper-parameter values of $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon =$ 1e-8 values for AdamW.

6 Analysis and Discussion

6.1 RQ1 - Does BitDistiller work on Smaller Models?

From Figure 2, it is clear that scaling up the number of bits used in quantization for BitDistiller significantly increases model performance on QA benchmarks. The perplexities between 2-bit and 3-bit models are quite similar, suggesting the next token prediction ability of models in these regimes may be similar, whereas there is a noticeable deterioration at 1-bit. From Figure 3, we see that 2-bit and 3-bit models retain similarly high fractions of the QA average performance compared to its full-precision counterpart, with 2-bit TinyLlama and 2-bit Llama-2-7B performing 75% and 81% as well as their full-precision models respectively. Across the different QA benchmarks, we generally observe clear improvements as the number of bits increases. However the quantizations of TinyLlama perform particularly poorly on WinoGrande and ARC-Challenge - the accuracies are statistically indistinguishable from random Table 1 and on Winogrande, there is no consistent improvement as the number of bits increases. This suggests the models may be deficient in benchmarks that emphasize linguistic nuance and multi-step reasoning, and the poor performance is consistent broader observations in the quantization literature, where smaller models with low bit-widths often degrade significantly on these two benchmarks (Frantar et al., 2023; Liu et al., 2023). We do not observe poor performance on these two benchmarks on the larger models, suggesting that it is more likely an issue of model size compounding the issue of bit-width in the case of TinyLlama.

Overall, given the consistent improvements across most QA benchmarks as we increase the number of quantization bits and the high proportion of average QA performance retained compared to their full-precision counterparts, we deduce that BitDistiller remains a feasible efficient approach for training quantized low-bit LLMs for model sizes smaller than 7B.

6.2 RQ2 - Does BitDistiller work with 1-bit Quantization?

Based on our experiments, it is clear that the BitDistiller method is ill-suited for 1-bit quantiza-

Model: TinyLlama 1.1B	PPL	PIQA	HellaSwag	WinoGrande	ARC-c	ARC-Easy	QA Avg.
Full-precision baseline	7.71	72.63 ± 1.04	46.70 ± 0.50	59.59 ± 1.38	30.89 ± 1.35	61.45 ± 1.00	54.25
3-bit	11.99	66.21 ± 1.10	39.00 ± 0.49	50.28 ± 1.41	24.32 ± 1.25	41.96 ± 1.01	44.35
2-bit	23.76	60.45 ± 1.14	32.35 ± 0.48	52.88 ± 1.40	22.10 ± 1.21	35.90 ± 0.98	40.71
1-bit	2297.22	52.83 ± 1.16	25.68 ± 0.44	51.22 ± 1.40	21.50 ± 1.20	26.64 ± 0.91	35.58
Random baseline	47917.22	53.43 ± 1.16	25.62 ± 0.44	50.20 ± 1.41	23.89 ± 1.25	25.00 ± 0.89	35.63

Table 1: Performance comparison across different bit levels for TinyLlama.

Model: Llama-3 3B	PPL	PIQA	HellaSwag	WinoGrande	ARC-c	ARC-Easy	QA Avg.
Full-precision baseline	7.36	76.49 ± 1.08	54.53 ± 0.49	65.50 ± 1.40	43.96 ± 1.30	64.21 ± 1.02	60.94
3-bit	11.59	74.92 ± 1.01	49.45 ± 0.50	65.11 ± 1.34	36.69 ± 1.41	69.23 ± 0.95	59.08
2-bit	16.90	68.82 ± 1.08	39.76 ± 0.49	54.30 ± 1.40	27.39 ± 1.30	56.44 ± 1.02	47.57
1-bit	7062.74	53.37 ± 1.16	26.04 ± 0.44	49.17 ± 1.41	20.05 ± 1.17	25.76 ± 0.90	34.88
Random baseline	238567.86	54.41 ± 1.16	25.42 ± 0.43	51.70 ± 1.40	21.93 ± 1.21	25.55 ± 0.89	35.80

Table 2: Performance comparison across different bit levels for Llama-3 3B.

tion, with all the QA benchmarks being statistically indifferent from random. For example, TinyLlama with 1-bit quantization achieves 26.64% on ARC-Easy, which is within two standard errors of random choice (25%), compared to 35.90% and 41.96% on the 2-bit and 3-bit quantizations. Additionally, the 1-bit quantized TinyLlama model has a perplexity which is two order of magnitudes higher than either the 2-bit or 3-bit counterparts, suggesting the model is significantly worse at the more basic task of next token prediction. The larger 3B and 7B models achieve a similar level of performance on 1-bit quantization suggesting that BitDistiller, at least without significant adapatation, is ill-suited for such a low-bit regime. Indeed, we see similar levels of performance on the same benchmarks when applying LLM-QAT (Liu et al., 2023) or GPTQ (Frantar et al., 2023) for 1-bit quantization. We reiterate that performance on these QA benchmarks with 1-bit quantization on models of similar sizes is possible, but these require expensive compute in either a large amount of data (Xu et al., 2024), or pretraining from scratch (Wang et al., 2023; Ma et al., 2024a). A potential explanation for this gap in performance between 1-bit and 2/3-bit quantizations could be that with the extreme lack of expressivity in the 1-bit regime, the model must learn a very different distribution from the pretrained model to succeed. Indeed, we observe from Figures 5, 6 and 7 that while the post-fine-tuning weight distributions from BitDistiller are relatively similar to the pre-trained full precision 7B model, the distributions for FBI-LLM (Ma et al., 2024a) has a much larger spread and is not Gaussian-like for the attention projection layer. This would be consistent with findings from previously successful methods for sub 2-bit quantization (Wang et al., 2023; Ma et al., 2024b,a), which found no difference between training from scratch and on the pretrained weights of an LLM. Thus, while the BitDistiller method is efficient for training 2-bit and 3-bit LLMs using only fine-tuning, there is no evidence of its feasibility for 1-bit quantization.

6.3 RQ3 - What is the Effect of Teacher Size on Performance?

Turning to the final question, we examine how the size of the teacher model affects the performance of a quantized student. Du et al. (2024) observed that a 7B model distilled from a larger 13B teacher underperformed compared to self-distillation; they hypothesized that the mismatch in teacher–student size architecture may have resulted in difficulty in the alignment of weight distributions. However, this result was based on a single comparison and lacked systematic evaluation.

We expand this analysis across multiple settings. In the case of TinyLlama (2-bit), distillation from a 3B teacher yields a QA average of 43.23, outperforming the 7B teacher variant at 40.86, despite the latter achieving slightly better perplexity (16.94 vs 17.17). Furthermore, on benchmarks, such as HellaSwag and PIQA, there is no clear pattern between teacher size and TinyLlama performance, though

Model: Llama-2 7B	PPL	PIQA	HellaSwag	WinoGrande	ARC-c	ARC-Easy	QA Avg.
Full-precision baseline	5.47	77.97 ± 0.97	57.11 ± 0.49	69.22 ± 1.30	43.34 ± 1.45	76.30 ± 0.87	64.79
3-bit	5.99	74.66 ± 0.89	39.76 ± 1.43	76.88 ± 0.98	67.88 ± 1.31	55.16 ± 0.50	62.87
2-bit	7.87	74.05 ± 1.02	48.79 ± 0.50	61.64 ± 1.37	33.02 ± 1.37	67.09 ± 0.96	56.92
1-bit	4082.93	52.45 ± 1.17	25.88 ± 0.44	51.78 ± 1.40	22.44 ± 1.22	26.26 ± 0.90	35.76
Random baseline	74461.16	52.45 ± 1.17	25.54 ± 0.44	50.99 ± 1.40	23.38 ± 1.24	25.29 ± 0.89	35.53

Table 3: Performance comparison across different bit levels for Llama-2 7B

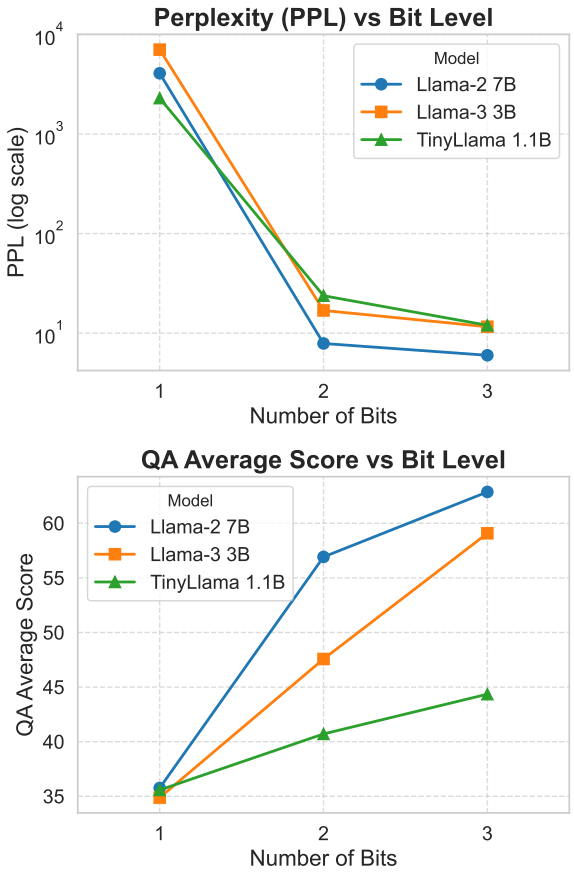


Figure 2: Model performance on QA Average and perplexity versus the number of bits for quantization.

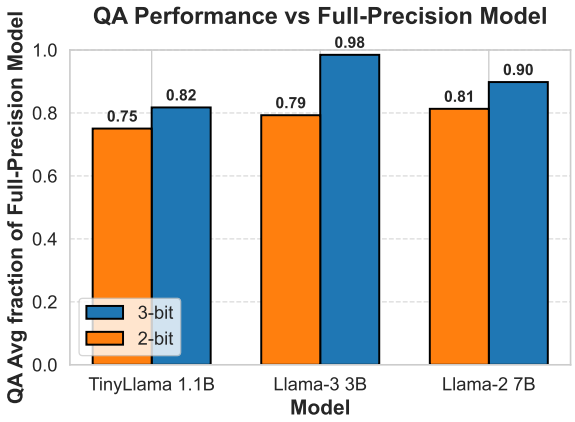


Figure 3: Fraction of QA Average performance compared to full-precision model.

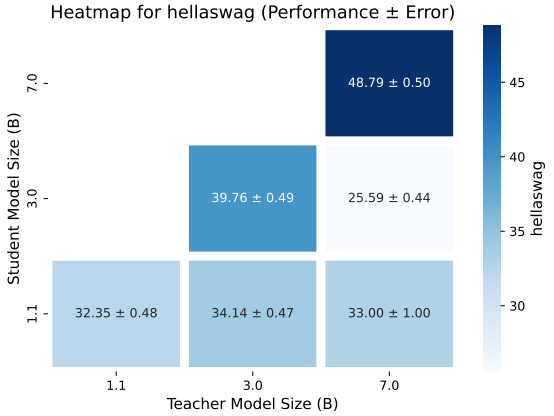
performance degrades significantly when distilling from a Llama 2 7B teacher to a Llama 3.2 3B student (Figure 4). This observation is likely due to the difference in vocabulary between the teacher and the student ($\sim$ 32,000 tokens for Llama 2 vs $\sim$ 128,000 tokens for Llama 3.2), which highlights the importance of tokenizer alignment in the distillation process.

These findings support the hypothesis that a larger teacher does not necessarily lead to better student performance. One possible explanation lies in the characteristics of the logit distributions. Larger models tend to produce sharper, high-confidence outputs that may be difficult for smaller, quantized students to match, particularly under logit-based objectives like the Confidence-Aware KL Divergence (CAKLD) used in BitDistiller. In such settings, CAKLD shifts toward Reverse KL, encouraging mode-seeking behavior (Agarwal et al., 2024) that can limit generalization in low-capacity students.

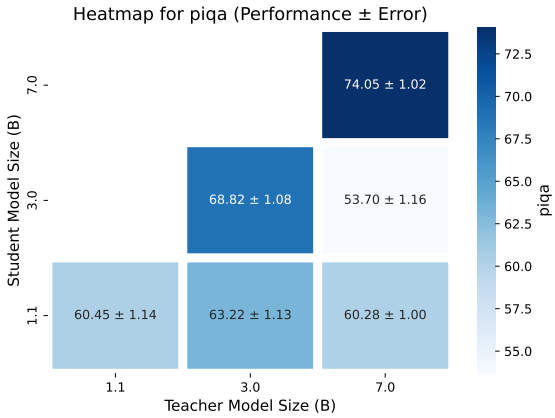
Moreover, larger teachers may introduce a representational mismatch, where the student struggles to approximate the richer predictive space of the teacher. This is especially problematic in extreme low-bit regimes, where the student's expressivity is already constrained. In contrast, teachers with similar architectures or capacities (e.g., 3B teacher for TinyLlama) produce more tractable distributions and lead to more stable and effective distillation. These findings reinforce the BitDistiller paper's own results (Table 4) and highlight the importance of teacher–student compatibility, particularly when operating under quantization constraints.

Student Model	Teacher Model	PPL	PIQA	HellaSwag	WinoGrande	ARC-c	ARC-Easy	QA Avg.
TinyLlama 1.1B	LLaMA 7B	16.94	60.28 ± 1.00	33.00 ± 1.00	53.99 ± 1.00	20.14 ± 1.00	36.91 ± 1.00	40.86
TinyLlama 1.1B	LLaMA 3B	17.17	63.22 ± 1.13	34.14 ± 0.47	51.78 ± 1.40	21.84 ± 1.21	45.16 ± 1.02	43.23
TinyLlama 1.1B	Self-Distilled	23.76	60.45 ± 1.14	32.35 ± 0.48	52.88 ± 1.40	22.10 ± 1.21	35.90 ± 0.98	40.71
LLaMA-3.2 3B	LLaMA 7B	914841.62	53.70 ± 1.16	25.59 ± 0.44	48.70 ± 1.40	20.14 ± 1.17	25.25 ± 0.89	34.68
LLaMA-3.2 3B	Self-Distilled	16.90	68.82 ± 1.08	39.76 ± 0.49	54.30 ± 1.40	27.39 ± 1.30	56.44 ± 1.02	47.57
LLaMA-2 7B	Self-Distilled	7.87	74.05 ± 1.02	48.79 ± 0.50	61.64 ± 1.37	33.02 ± 1.37	67.09 ± 0.96	56.92

Table 4: Effect of teacher model size on 2-bit quantized student models.



(a) HellaSwag Benchmark



(b) PIQA Benchmark

Figure 4: Performance of teacher size as a function of student size on the benchmarks HellaSwag and PIQA.

6.4 Ablation Studies

6.4.1 Autodistillation on Larger Teacher Data

The dataset generated by TinyLLama used for fine-tuning contained examples with incoherent passages and snowballing repetition, a common observation among the smaller Llama models. We observed an increase in the data quality from larger models, hence, we also investigated the performance of training with the teacher model data while maintaining self-distillation. Although, there were small improvements in perplexity, we found the difference in performance on QA Average to be marginal on TinyLlama for 2-bit quantization (see Table 5), suggesting that the student model's performance, at least for smaller models, is more bottlenecked by its size and quantization level than the data quality for distillation.

Teacher Data Model	PPL	QA Avg
TinyLlama 1.1B	23.76	40.71
Llama-3 3B	17.11	41.54
Llama-2 7B	16.73	40.83

Table 5: Performance of TinyLlama with 2-bit quantization using different models to generate data.

6.4.2 Gamma Masking

When calculating γ in the CAKLD loss function (Equation 1), Du et al. (2024) include mask token probabilities, biasing the value low due to averaging over a larger number of uninformative probabilities. We exclude these probabilities from our calculation of γ, resulting in a 5.54% increase over four randomly selected training splits of our dataset.

6.4.3 Smaller Group Size for 1-bit Quantization

Du et al. (2024) quantize the model weights in buckets of 128, however, given the large loss in expressivity in 1-bit quantization, we studied if larger bucket sizes could increase resolution and increase performance. Despite a large drop in perplexity from 128 to 64, we did not observe any noticeable improvements on the QA benchmarks (see Table 6) and therefore elected keep the default value.

6.4.4 Cross Entropy Term in Loss

In knowledge distillation training, the KL divergence term is often combined with a cross-entropy (CE) term (Equation 2) to balance betwen learning the distribution from the teacher model and

Group Size	PPL	QA Avg
TinyLlama 1.1B	2720.33	37.79
Llama-3 3B	2707.56	37.23
Llama-2 7B	3488.38	37.63

Table 6: Performance of 1-bit quantized TinyLlama on different group sizes

achieving accurate next token prediction (Hinton et al., 2015; Ma et al., 2024a). We test the effect of this additional term which is excluded from the BitDistiller's CAKLD loss, following the scaling and weight factors of Kim et al. (2019).

$$D_{\text{CAKLD+CE}} = \Big[(1-\gamma)D_{\text{KL}}\left(\frac{P_T}{T} \parallel \frac{P_S}{T}\right) + \gamma D_{\text{KL}}\left(\frac{P_S}{T} \parallel \frac{P_T}{T}\right)\Big]T^2 + L_{CE} \quad (2)$$

The new temperature hyper-parameter T (similar in concept but distinct from the LLM sampling temperature) determines the shape of the distributions - a higher parameter results in flatter distributions. The gradients are scaled down by T^{-2} in the division, so a T^2 factor is multiplied outside the CAKLD term to keep scaling consistent.

Training runs were performed on 2 epochs for temperature values of 2, 5 and 20 (Table 7). Since $T = 20$ performed the best, we then continued to train for a total of 4 epochs, however, the results did not show significant differences from CAKLD across the QA benchmarks while the perplexity slightly increased.

KD Temperature	QA Avg
20	40.36
20 (2 epochs)	40.92
5 (2 epochs)	40.53
2 (2 epochs)	39.84

Table 7: Performance for CAKLD + CE loss function with varying temperatures for 2-bit TinyLlama 1.1B. We focus on QA average as the most well-rounded metric to compare these methods. The full table can be found in Appendix Table 8. The increased QA score for the 2 epoch runs is due to a higher ARC-easy, however other metrics are slightly outperformed in the 4 epoch run.

7 Conclusion

In this work, we investigate the applicability of BitDistiller in the ultra-low-bit regime, specifically extending it to 1-bit quantization for models smaller than LLaMA 7B. Our findings demonstrate that BitDistiller remains effective at 2-bit and 3-bit precision with smaller models, demonstrating large consistent improvements above the random baselines on QA benchmarks. However, extending the framework to 1-bit quantization presents significant challenges, indicating that additional methodological innovations may be required to preserve performance at this ultra low bit-width. In our exploration of the impact of teacher complexity, we did not observe any consistent statistical trends across the hard targets generated by different teacher model sizes or any significant impact to model performance when varying the soft targets used in the distillation process. We therefore believe that the reasons for degradation of student performance when using more complex teacher models likely lie elsewhere.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes.

Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. 2019. Post-training 4-bit quantization of convolution networks for rapid-deployment.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Dayou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. 2024. Bitdistiller: Unleashing the potential of sub-4-bit llms via self-distillation.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers.

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

Jangho Kim, Yash Bhalgat, Jinwon Lee, Chirag Patel, and Nojun Kwak. 2019. Qkd: Quantization-aware knowledge distillation.

Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. 2024. Common 7b language models already possess strong math capabilities.

Zhen Li, Yupeng Su, Runming Yang, Congkai Xie, Zheng Wang, Zhongwei Xie, Ngai Wong, and Hongxia Yang. 2025. Quantization meets reasoning: Exploring llm low-bit quantization degradation for mathematical reasoning.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for llm compression and acceleration.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2023. Llm-qat: Data-free quantization aware training for large language models.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Liqun Ma, Mingjie Sun, and Zhiqiang Shen. 2024a. Fbi-llm: Scaling up fully binarized llms from scratch via autoregressive distillation.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. 2024b. The era of 1-bit llms: All large language models are in 1.58 bits.

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

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpaca: A Strong, Replicable Instruction-Following Model.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. 2023. Bitnet: Scaling 1-bit transformers for large language models.

Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and Wanxiang Che. 2024. Onebit: Towards extremely low-bit large language models.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence?

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model.

Yijia Zhang, Sicheng Zhang, Shijie Cao, Dayou Du, Jianyu Wei, Ting Cao, and Ningyi Xu. 2023. Afpq: Asymmetric floating point quantization for llms.

Jiaqi Zhao, Miao Zhang, Ming Wang, Yuzhang Shang, Kaihao Zhang, Weili Guan, Yaowei Wang, and Min Zhang. 2025. Ptq1.61: Push the real limit of extremely low-bit post-training quantization methods for large language models.

A Appendix

A.1 Weight Distributions

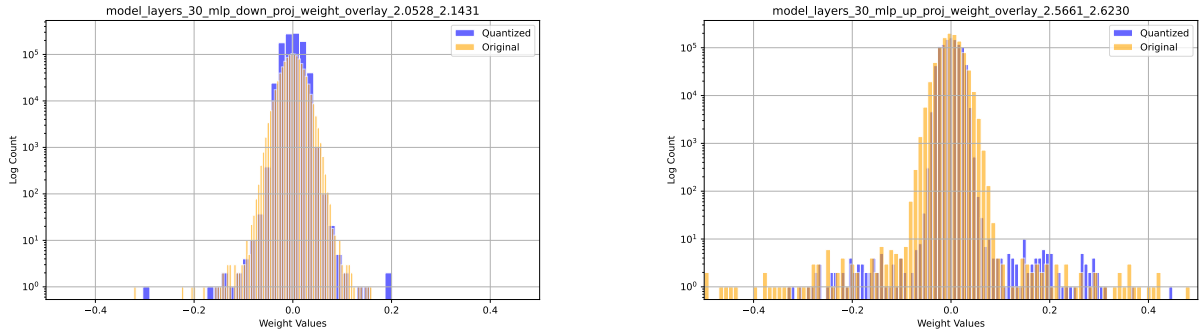


Figure 5: Weight distribution of Llama 2 7B 1-bit.

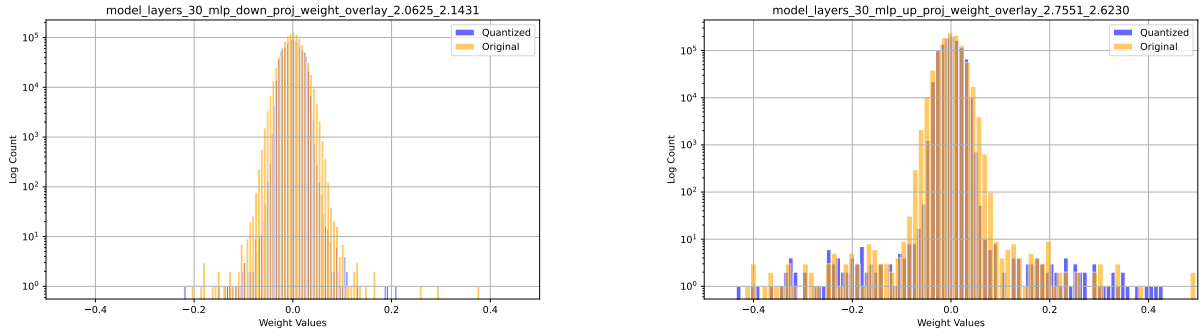


Figure 6: Weight distribution of Llama 2 7B 2-bit.

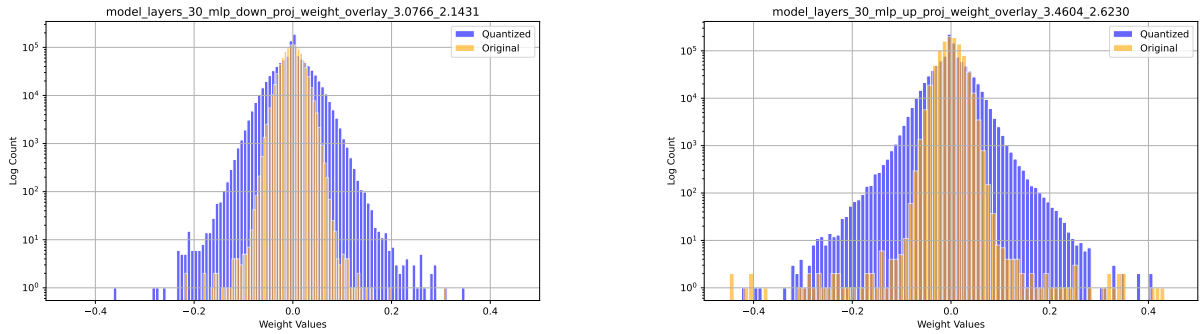


Figure 7: Weight distribution of FBI-LLM 1-bit. Plot title shows the layer name and entropy of the quantized and full-precision models

To understand the degradation of performance from our 2-bit quantized model to our 1-bit quantized model, we compare the layer-wise weight distribution of these models against that of the full precision pre-trained model. We selected 2 of the layers for illustration, but the general pattern can be seen across multiple layers. Although we can see from the histograms that our 2-bit model resembles more closely the distribution of the full precision model when compared to our 1-bit model, their difference is not sufficient to explain the gap in performance. This potentially indicates that in the 1-bit regime, a completely different distribution may be required. In the bottom figure we perform the same analysis but on the 1-bit FBI-LLM

model trained from scratch. We indeed see a significant shift in weight distribution. Although distributions of weights are only highly indicative and their actual impact on model capabilities is more nuanced, this appears to support the idea that a quantized 1-bit model may require a different weight distribution to remain performant.

A.2 Loss Function Table in Full

KD Temperature	PPL	PIQA	HellaSwag	WinoGrande	ARC-c	ARC-Easy	QA Avg.
20	22.43	60.55 ± 1.14	32.22 ± 0.47	51.93 ± 1.40	21.67 ± 1.20	35.40 ± 0.98	40.36
20 (2 epochs)	—	60.12 ± 1.14	31.21 ± 0.46	51.78 ± 1.40	21.50 ± 1.20	39.98 ± 1.01	40.92
5 (2 epochs)	—	60.29 ± 1.15	31.09 ± 0.46	52.80 ± 1.40	21.84 ± 1.21	36.16 ± 0.99	40.53
2 (2 epochs)	—	61.37 ± 1.14	31.37 ± 0.46	51.38 ± 1.41	20.56 ± 1.18	34.51 ± 0.98	39.84

Table 8: Full table for performance for CAKLD + CE loss function with varying temperatures for 2-bit TinyLlama 1.1B