

Stable Quantization-Aware Training for Vision-Language Models

Brian Ebiyau*, Ifeanyi Igwegbe George*, Martins Awojide*, Mboh Bless*,

* Carnegie Mellon University Africa, Kigali, Rwanda

Email: {bebiyau, ici, mawojide, mpearlnc}@andrew.cmu.edu

I. INTRODUCTION

Vision Language Models (VLMs) have emerged as a powerful tool in artificial intelligence, enabling machines to understand and generate language in conjunction with visual data. VLMs are designed to bridge the gap between visual and textual information, allowing for a wide range of applications, such as image captioning [1], visual question answering (VQA) [2] and more.

The training of VLMs requires substantial computational resources due to the complexity of the models and the extensive datasets involved. To address this challenge, researchers have been exploring low-precision training techniques, which aim to reduce computational costs while maintaining convergence rates [3]. These techniques involve quantizing the model parameters and activations from high-precision floating-point representations to lower-precision formats, such as int8 [4] or bfloat16 [5]. By doing so, the memory footprint and computational demands of VLMs can be significantly reduced, making them more suitable for deployment on resource-constrained devices like mobile phones and edge computing platforms.

Recently, quantization techniques for VLMs have garnered significant attention, with various approaches being proposed to strike a balance between model performance and efficiency. These approaches range from post-training quantization (PTQ), which quantizes a pre-trained model without requiring access to the original training data, to quantization-aware training (QAT), which integrates quantization into the training process itself

In this report, we focus on advancing the state-of-the-art in quantization techniques for VLMs. We select the work by Wortsman et al. [6], titled "Stable and low-precision training for large-scale vision-language models," due to its significant contributions and outstanding performance in VLM quantization. Building upon this, we propose extensions by experimenting with various backbones of the CLIP model and exploring different precision values to investigate their impact on performance.

The rest of this report is organized as follows: Section II provides an overview of the related work in VLM quantization. Section III describes the baseline selection and justifies its choice. Section IV presents a detailed description of the baseline models, followed by implementation in Section V. Section VI introduces our proposed model extensions and improvements

II. RELATED WORK

Quantization in the context of large-scale vision-language models refers to the process of converting high-precision floating-point parameters (weights and activations) into lower-precision representations. This conversion is essential for reducing the memory footprint and computational demands of these models, making them more suitable for deployment on resource-constrained devices such as mobile phones and edge computing platforms. There have been various research and studies that have been done in quantizing VLM's and in this section we shall review these methods.

Post-Training Quantization (PTQ) quantizes a pre-trained model without requiring access to the original training data. This approach typically involves quantizing the weights and activations directly after training, which can lead to a reduction in model size and an increase in inference speed [7]. However, PTQ often results in accuracy degradation, particularly when quantizing to lower bit-widths [8]. While traditional methods employ uniform quantization and tensor clipping to minimize quantization noise [9], recent work has focused on cross-layer dependency optimization. Wang et al. [10] proposed a framework that leverages activation entropy to guide block partitioning and optimizes the visual encoder for more efficient search space decomposition, achieving 2.78x memory compression and 1.44x speed improvement on LLaVA models without performance degradation.

Quantization Aware Training (QAT) integrates quantization into the training process itself, allowing the model to learn how to mitigate the effects of quantization during training. This approach typically involves using a straight-through estimator (STE) to approximate gradients through quantized layers, enabling effective backpropagation [11]. While QAT has been shown to produce models with significantly lower accuracy loss compared to PTQ, especially when targeting low-bit quantization, it often requires substantial training resources. To address this limitation, Sun et al. [12] proposed "Prompt for Quantization" (P4Q), a hybrid approach that bridges QAT and PTQ by using learnable prompts and contrastive learning. Their method [12] achieves 4x compression on CLIP-ViT/B-32 while maintaining comparable or better accuracy than full-precision models, demonstrating that lightweight architectures can effectively balance the trade-offs between quantization performance and training costs. Wortsman et al. [6] made significant progress in low-precision training by introducing

SwitchBack, achieving INT8 quantized training of CLIP ViT-Huge with only 0.1% accuracy drop while providing 13-25% speedup compared to bfloat16 training. Their work also demonstrated that proper initialization techniques, particularly zero-initialized layer scaling, can enable effective lower precision training even with standard approaches.

Hybrid approaches combine quantization with other model compression techniques, such as pruning or knowledge distillation, which has shown promise in enhancing model efficiency while maintaining accuracy. For instance, knowledge distillation can be used alongside quantization to train a smaller model (the student) to replicate the behavior of a larger model (the teacher), which can be particularly effective in natural language processing tasks [13]

III. BASELINE SELECTION

For this project, we have chose to replicate the work done by Wortsman et al. [6] on QAT of VLMs. Wortsman et al. demonstrated the gains in training VLMs using Contrastive Language-Image Pre-training (CLIP) models. CLIP models are important in the area of computer vision and have been used in the area of classification and also in the underlying techniques for images generation models like DALL-E-2 and Stable Diffusion [6] For their experiment, they trained ViT CLIP models on LAION-2B [14] using the OpenCLIP Library [15] They are evaluated using zero-shot classification on benchmark datasets such as the ImageNet1k and CIFAR100. For their experiments, they did not focus on performance but rather on the instability of the training task at different hyperparameter tunes.

The baseline for our experiments is a number of CLIP models with select backbones trained with bfloat16 precision. In place of the default precision of float32, the quantized precision of bfloat16 which uses 8 bits for the exponent and 7 bits for the mantissa, with a range of $(10^{-38}, 3 \times 10^{38})$ was chosen instead of float16 uses 5 bits for the exponent and 10 bits for the mantissa, giving a range of $(5.96 \times 10^{-8}, 65504)$ since it has been proven to produce training runs with convergence compared to float16, which has been documented to lead to divergence for QAT for VLMs [16].

CLIPs model with different backbones ranging from basic CNN architecture models to ViTs and they perform differently [17]. The availability and cost of computational resources for bfloat16 training constrained the choice of backbones. The paper used multiple nodes of A100 GPUs for the training which was not readily available for our work. Since we opted for A100 GPUs on TensorDock (in place of AWS due to vCPUs request constraints for P4d.24xlarge instances) we had to consider the cost of running the baseline.

Since the cost of training is dependent on the cost of hyperparameter tuning, the size of the dataset and the cost of training one epoch [18], we attempted to limit the cost of training each epoch for each backbone since the dataset has been set for our work and the hyperparameters from the main paper. To restrict the training cost, we set an initial number of training epochs (or steps based on the batch size) and chose

backbones that are likely to perform better with a small dataset in this case MSCOCO 2017. Large ViT models used in the paper are likely to underperform when trained with a relatively smaller dataset due to inductive bias and ViT models' focus on global learning compared to local learning.

The list of CLIP-based models selected for the baseline considering the project constraint includes:

- Resnet 50 and 101,
- Convnext small and Convnext Large
- Vision Tranformer
- MobileCLIP

IV. BASELINE MODEL DESCRIPTION

The baseline models are CLIP models trained with **bfloat16 precision**.

$$Q_{\text{row}} \left(\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \right) = \text{round} \left(\begin{bmatrix} \frac{127}{\max(|x_1|)} \cdot x_1 \\ \vdots \\ \frac{127}{\max(|x_n|)} \cdot x_n \end{bmatrix} \right)$$

$$Q_{\text{tensor}}(X) = \text{round} \left(\frac{127}{\max(|X|)} \cdot X \right)$$

These models were trained on the MSCOCO 2017 dataset in place of the LAION-2B dataset and their performance was assessed using zero-classification on ImageNet1k based on 80 prompt templates from CLIP. At the core, the distances between embeddings are compared using cosine similarity.

$$\text{cosine_similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

Where A and B are two vectors being compared to see how closely related they are to each other.

V. BASELINE IMPLEMENTATION

Training is performed with a linear warmup for the first 5,000 iterations and cosine decay for the remaining 15,000. Other hyperparameters include a **learning rate of 2e-3** and a **weight decay of 0.2**.

The code used for the baseline implementation was the OpenCLIP library. We create new scripts to run models with our select backbones.

Our project ablations are available at this wandb project. It is important to note that the goal of the experiments in the sources is **not to achieve high final accuracy but rather to compare different methods for low-precision training and stability**. Therefore, opting for shorter training runs is still acceptable.

Fig 1. and Fig. 2 are wandb plots showing the Top 1 and Top 5 zero-shot performance on ImageNet1k. Table 1 gives further information on the runs for each backbone.

We intend to adapt the OpenCLIP library code to run two new optimizers, StableAdamW and AEOS. The library currently provides support for int8 training and the Switch-Back but would be extended to accommodate float8 training and different initialization choices. Also, we are considering including CoCa models in the list which would then be evaluated using the BLEU score and ARO benchmarks.



Fig. 1. Top 1 Zero-shot plot

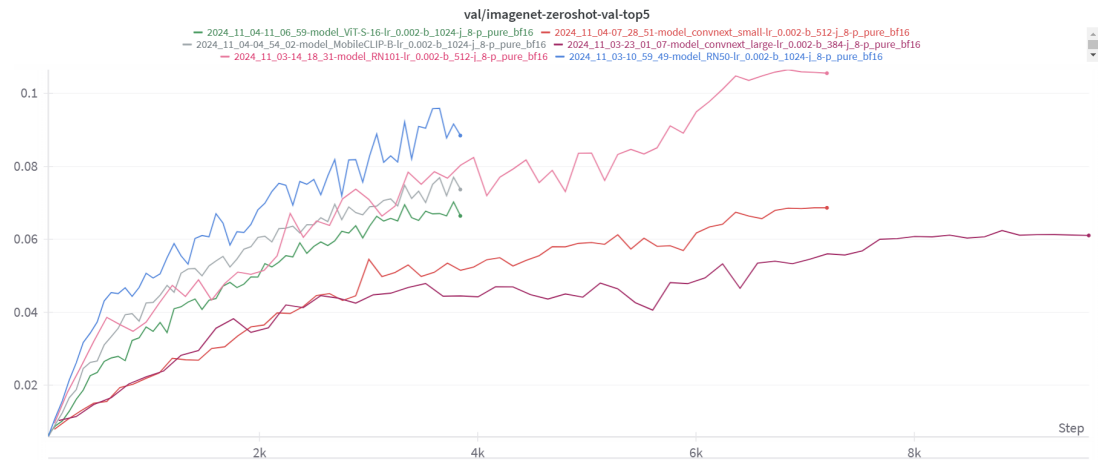


Fig. 2. Top 5 Zero-shot plot

TABLE I
ZERO-SHOT PERFORMANCE METRICS FOR VARIOUS MODEL BACKBONES.

Backbone	Epochs	Train Steps	Top-1 (%)	Top-5 (%)
ResNet-50	60	3,839	3.16	8.85
ResNet-101	60	7,199	3.75	10.56
ConvNeXt small B	60	7,199	2.15	6.86
ConvNeXt large	60	9,599	1.92	6.11
ViT S-16	60	3,839	2.20	6.64
MobileCLIP	60	3,839	2.54	7.37

REFERENCES

- [1] S. Sharifzadeh, C. Kaplanis, S. Pathak, D. Kumaran, A. Ilic, J. Mitrovic, C. Blundell, and A. Banino, "Synth²: Boosting visual-language models with synthetic captions and image embeddings," 2024. [Online]. Available: <https://arxiv.org/abs/2403.07750>
- [2] A. C. A. M. de Faria, F. de Castro Bastos, J. V. N. A. da Silva, V. L. Fabris, V. de Sousa Uchoa, D. G. de Aguiar Neto, and C. F. G. dos Santos, "Visual question answering: A survey on techniques and common trends in recent literature," 2023. [Online]. Available: <https://arxiv.org/abs/2305.11033>
- [3] M. Cacciola, A. Frangioni, M. Asgharian, A. Ghaffari, and V. P. Nia, "On the convergence of stochastic gradient descent in low-precision number formats," 2023.
- [4] H. Wu, P. Judd, X. Zhang, M. Isaev, and P. Micikevicius, "Integer quantization for deep learning inference: Principles and empirical evaluation," 2020. [Online]. Available: <https://arxiv.org/abs/2004.09602>
- [5] D. Kalamkar, D. Mudigere, N. Mellempudi, D. Das, K. Banerjee, S. Avancha, D. T. Vooturi, N. Jammalamadaka, J. Huang, H. Yuen, J. Yang, J. Park, A. Heinecke, E. Georganas, S. Srinivasan, A. Kundu, M. Smelyanskiy, B. Kaul, and P. Dubey, "A study of bfloat16 for deep learning training," 2019. [Online]. Available: <https://arxiv.org/abs/1905.12322>
- [6] M. Wortsman, T. Dettmers, L. Zettlemoyer, A. Morcos, A. Farhadi, and L. Schmidt, "Stable and low-precision training for large-scale vision-language models," 2023. [Online]. Available: <https://arxiv.org/abs/2304.13013>

- [7] Z. Liu, Y. Wang, K. Han, W. Zhang, S. Ma, and W. Gao, "Post-training quantization for vision transformer," 2021.
- [8] Z. Yuan, Y. Chen, Q. Wu, and G. Sun, "post-training quantization framework for vision transformers," 2021.
- [9] G. Shomron, F. Gabbay, S. Kurzum, and U. Weiser, "Post-training sparsity-aware quantization," 2021.
- [10] C. Wang, Z. Wang, X. Xu, Y. Tang, J. Zhou, and J. Lu, "Q-vlm: Post-training quantization for large vision-language models," 2024. [Online]. Available: <https://arxiv.org/abs/2410.08119>
- [11] T. Allenet, O. Bichler, and O. Sentieys, "Disentangled loss for low-bit quantization-aware training," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2787–2791, 2022.
- [12] H. Sun, R. Wang, Y. Li, X. Cao, X. Jiang, Y. Hu, and B. Zhang, "P4q: Learning to prompt for quantization in visual-language models," 2024. [Online]. Available: <https://arxiv.org/abs/2409.17634>
- [13] J. Jin, C. Liang, T. Wu, L. Zou, and Z. Gan, "Kdlsq-bert: a quantized bert combining knowledge distillation with learned step size quantization," 2021.
- [14] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, K. Aarush, C. H. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "Laion-5b: An open large-scale dataset for training next generation image-text models," 2022.
- [15] G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, and A. Farhadi, "Editing models with task arithmetic," 2022.
- [16] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, "Reproducible scaling laws for contrastive language-image learning," 2022.
- [17] A. Bartsch-Jimenez, M. Błażkiewicz, H. Azadjou, R. Novotny, and F. J. Valero-Cuevas, "“fine synergies” describe motor adaptation in people with drop foot in a way that supplements traditional “coarse synergies”," 2023.
- [18] R. Verdecchia, J. Sallou, and L. J. Cruz, "A systematic review of green computing," 2023.