Post-Training Quantization of CLIP Vision-Language Model

Anirudh Kanneganti Sri Sai Sundareswar Pullela Satvarsh Gondala Juily Y. Deshpande
Oregon State University
{kannegaa, pullelas, gondalas, deshpanj}@oregonstate.edu

Abstract

Quantization is an effective technique for compressing neural networks. However, applying quantization on vision-language models is under-explored. This is because the vision-language models are more sensitive to post-training quantization due to uneven activation distributions. Our work helps with the analysis and effective implementation of post-training quantization on the Vision-Language model (CLIP). Our post-training quantization analysis is carried out by various methods ranging from basic quantization on CLIP to applying twin uniform quantization and using Hessian guided metric to find the scaling factors for activations and weights for every layer respectively. We will also use five vision-language tasks as benchmarks to further analyze and evaluate the post-training quantization of the CLIP model. Our experiments show the quantized CLIP achieves a near loss-less prediction accuracy (for 8-bit quantization) on the ImageNet Classification task. Our open-source code is present at https://github.com/VLQuant/PTQ4CLIP

1. Introduction

Post-Training Quantization (PTQ) is a technique used in deep learning to compress neural network models with quantized weights and activations after an initial training phase with 32-bit precision. This allows for the deployment of neural network models on resource-constrained hardware platforms or on systems that operate on low energy and memory. Not only does it make the said models more accessible and compatible, but it also does this with a minimal or no hit to its accuracy. PTQ is very successful for CNNs, but directly bringing it to a vision transformer results in more than 10% accuracy drop even with 8-bit quantization [7]. It is worth noting that most of the future applications in AI are multi-modal and real-time inference of these models on edge devices is an interesting area to explore, which made us start the analysis of post-training quantization of vision-language models. There have been previous works on applying PTQ across a Vision model and a Language model separately, but there was never any study on the implementation of PTQ for a vision-language model.

Our paper focuses on the implementation of PTQ across various components of a Vision-Language model. The vision-language model we are using is CLIP (Contrastive Language - Image Pre-training). We are using CLIP as it has a lot of applications and is trained on over 400 million image-text pairs. This model uses a Modified ResNet/Vision Transformer as its image encoder and uses a transformer as its Text Encoder. CLIP trains these encoders simultaneously to learn a contrastive representation [1]. CLIP uses a text encoder to convert prompts into embeddings and then calculates the encoding for each prompt during inference. All of this happens simultaneously as the image encoder generates the encoding for the image. CLIP then chooses the image-text embedding pair with the highest cosine similarity as its prediction. CLIP has shown impressive performance in zero-shot transfer learning, achieving a remarkable top-1 accuracy of 76.2% on the ImageNet dataset without being exposed to any of its labels during training [1].

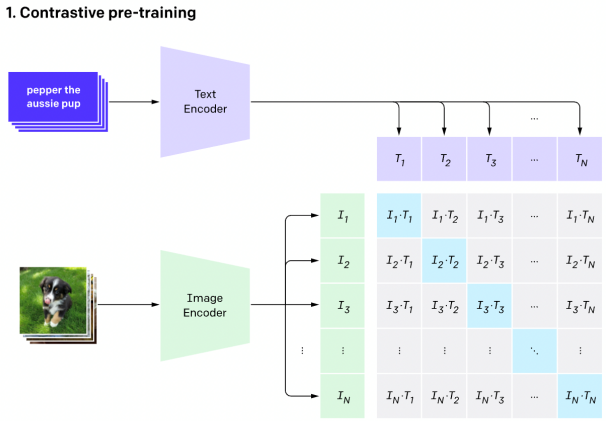

Figure 1. Contrastive Pre-Training of CLIP

One of the biggest challenges of applying PTQ across a

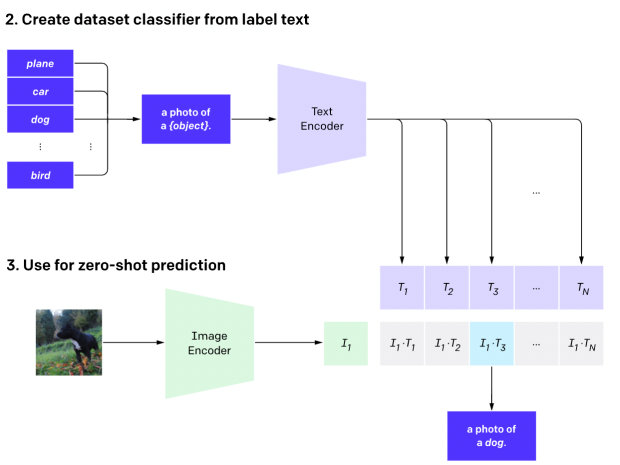

Figure 2. Next steps of CLIP

vision-language model is the fact that vision-language models are more sensitive to quantization due to uneven activation distributions. We are overcoming this problem by applying twin uniform quantization to find the scaling factor for activations and a Hessian-guided metric to find the scaling factor of weights of each layer [7].

2. Related Works

Our project deals with the compression of a vision-language model. Pruning, quantization, and knowledge distillation are some of the most used compression techniques that are related to our project.

Unified and Progressive pruning (UPop) is a pruning approach applied to vision-language models that enables the unified search of multimodal subnets within the original model by exploring a continuous optimization space. Then, the UPop incorporates progressive search and retains the identified subjects, ensuring convergence between the search and retaining process [4].

However, quantization is proven to be the most efficient compression method when compared to pruning and knowledge distillation. While none of the works focus on applying quantization to a vision-language model, some works below focus on applying quantization to vision models and language models separately.

PTQ for Language models was done using a vector-wise quantization with separate normalization constraints for each inner product in matrix multiplication, which then includes a mixed-precision decomposition scheme for the emergent outliers; this isolates the outlier feature dimensions into a 16-bit matrix multiplication [2]. Some other quantization models applied across Language Models are : SmoothQuant [5] and ZeroQuant [6]

PTQ for Vision models was done using Twin-Uniform Quantization and hessian-guided metric to find the scaling factors for activations and weights respectively [7]. This shows the quantized vision transformers achieve near-lossless prediction accuracy (less than 0.5% drop at 8-bit quantization) on the ImageNet classification task [7]. Our current work, PTQ4CLIP, utilizes a similar approach that is applied across all the components of the vision-language model (CLIP) uniformly. We can not use separate quantization schemes as the CLIP components are trained on the common loss function.

Paper	Similarity with our project
UPop [4]	Application (Vision-Language Model)
LLM.Int8() [2]	Technique (Quantization)
PTQ4VIT [7]	Technique (Quantization)

Table 1. Table highlighting the similarity between related works and our project

3. Methodology

In this section, we will discuss two quantization methods for the CLIP model. First, we will talk about the vanilla quantization strategy for the image and language encoders of the CLIP model. Then, we will discuss the PTQ4CLIP approach that uses twin uniform quantization strategy applied to both the encoders, where the Hessian guided metric is used to achieve better performance.

3.1. Vanilla Quantization

This is the first approach we used to quantize the CLIP model. In this, we applied quantization to the weights and activations of the convolutional and linear layers for the image encoder, and the linear layers for the language encoder. For the image encoder, i.e., ResNet-50, we quantized all the linear and convolutional layers, and for the language encoder, i.e., the transformer, linear layers inside multi-head attention, where the q-k-v inputs are fed and the final linear after the scaled dot product operation, and the linear layers pre and post-GELU activation are quantized.

First, we derive the min-max value of the tensor using a range tracker. Then, we calculate the scaling factor, which divides the given range of real values r into a number of partitions. This scaling factor is calculated using the

formula:

$$S = \frac{\beta - \alpha}{'2^b - 1} \quad (1)$$

where $[\alpha, \beta]$ denote the clipping range and b is the quantization bit width [3]. The clipping range is determined with the help of the min/max values of the signal using two approaches:-

Symmetric approach : $-\alpha = \beta = max(|r_{max}|, |r_{min}|)$

Asymmetric approach :$\alpha = r_{min}$, and $\beta = r_{max}$

For the weights, we used per-channel symmetric quantization. To ensure that each filter has its own unique scaling range, the per-channel approach was used, since the clipping range will be calculated multiple times, we applied symmetric quantization to reduce the computational overhead. Whereas for the activation, we used per-layer asymmetric quantization. Symmetric quantization partitions the clipping using a symmetric range. This has the advantage of easier implementation, as it leads to Z = 0. However, it is suboptimal for cases where the range could be skewed and not symmetric. Hence, we applied asymmetric quantization to the activations. Next, this scaling factor is applied to the input. Then we clipped the result and finally de-quantized it.

3.2. PTQ4CLIP

This method uses Twin Uniform Quantization and Hessian guided metric on both image and text encoders. Twin uniform quantization uses two quantization ranges, R1 and R2, controlled by two scaling factors, ΔR1 and ΔR2, respectively, to ease the difficulty of accurately quantifying both positive and negative values achieved after the softmax and GELU functions, with symmetric uniform quantization [7].

In this approach, we used a ViT-B/32 vision transformer instead of a ResNet-50 image encoder. Here, we applied the Twin Uniform Quantization strategy on the linear layers of the vision transformer after the Multi-Head Self Attention (MHSA) block. In this, the MHSA inputs are not quantized.

In the Twin Uniform approach, we computed two ranges: R1 for negative values and R2 for positive values. Each part is further divided into smaller intervals or steps. The quantization aims to accurately represent both negative and positive values. R1 covers the range of negative numbers with smaller steps, while R2 covers the range of positive numbers with larger steps. This allows for a precise representation of negative values while still capturing a wide range of positive values. To find the optimal quantization for a transformer model, we performed a calibration process. During calibration, suitable step sizes (ΔR1 and ΔR2) were determined for R1 and R2 respectively, ensuring accurate quantization.

To integrate Twin Uniform Quantization with the CLIP model, the original Vision Transformer module from the Timm library was substituted with the CLIP model. The CLIP layers were then replaced with quantized layer classes of PTQ. Next, we modified the hessian quant calibration class to accommodate multimodal inputs of the CLIP. Following calibration, the resulting model was utilized for performing inference tasks.

4. Experimental Setup

In this section, we will explain experiments that we performed using Vanilla Quantization and discuss the results obtained. For PTQ4CLIP we used the same setup as described for Vanilla Quantization.

We performed five tasks using the vanilla quantized CLIP model and evaluated the performance in comparison to non-quantized CLIP execution.

i) **Task 1 : Image Classification**
Setup: For this task we used the ImageNetV2 dataset with 10,000 samples in the image classification experiment. The assignment centered on zero-shot classification utilizing a contrastive technique, which allows classification without requiring individual training for each class. Our assessment was based on Top-1 Accuracy metrics, which assesses the accuracy of the top projected class. In this, the CLIP original paper claimed a zero-shot performance of 47% without a prompt and 53% with the prompt.

Each image in the dataset is matched against the target classes embedded within the prompts. A similarity score is computed between the image and each target class, resulting in a set of similarity scores. The target class with the highest similarity score is designated as the generated target for the given image. The generated target is then compared against the ground truth target, allowing for the evaluation of accuracy. This approach enables the assessment of the model's performance in accurately predicting the target class based on the image input, leveraging prompt-based comparisons and similarity scoring.

ii) **Task 2 : Image Captioning**
Setup: In the image captioning experiment, we used

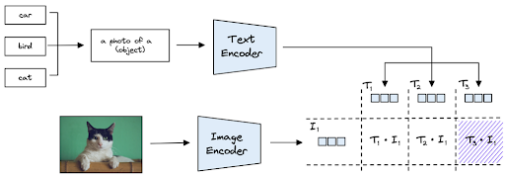

Figure 3. Image Classification

the MS-COCO dataset with a selection of 100 samples out of a total of 5000. The captioning strategy used was unsupervised, with a CLIP-based loss guiding and improving the captions of the GPT2 model for better alignment with the given images. The assessment metric utilized was CIDEr, which evaluates the consensus-based evaluation of picture descriptions. As a reference standard, the ZeroCap approach received a score of 34.5.

The conventional approach for image captioning involves utilizing the latent representation of images as input to a language model for caption generation, which falls under the category of supervised image captioning. In our experiment, we adopted an unsupervised methodology that incorporates CLIP (Contrastive Language-Image Pretraining) and GPT2 (Generative Pretrained Transformer 2). In this approach, a loss function based on CLIP is utilized to guide and influence the caption generation process of GPT2. By leveraging the capabilities of CLIP and GPT2 in an unsupervised manner, we aim to enhance the quality and coherence of the generated captions.

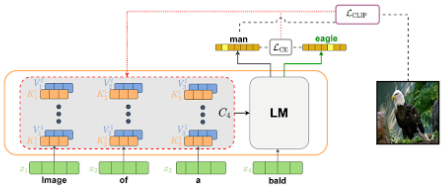

Figure 4. Image Captioning

iii) **Task 3 : Image-Text Retrieval**
Setup: We analyzed the MS-COCO dataset, which contained 5000 samples, in the image-text retrieval experiment. The primary metric used to evaluate retrieval performance was Recall@K, with a focus on the top-ranked result (R@1). The established benchmark in the CLIP paper, using the VIT-Large model, achieved a recall rate of 58.4%.

In this experimental setup, we employ a pairwise cosine similarity computation to create a matrix that captures the similarity between each item in the dataset. When the task involves generating captions for images, the similarity scores between the image and all the text samples are computed and stored in a row of the matrix. The text sample with the highest similarity score in that row is selected as the result, indicating the most relevant caption for the given image based on similarity metrics. By utilizing this approach, we aim to associate images with their most semantically similar captions, facilitating accurate image captioning.

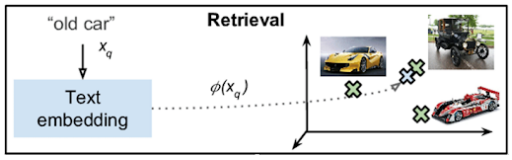

Figure 5. Image-Text Retrieval

iv) **Task 4 : Text-Image Retrieval**
Setup: We used the MS-COCO dataset with 5000 samples in the text-image retrieval experiment. The metric used to assess retrieval performance was Recall@K, with a focus on the top-ranked result (R@1). The standard performance reported in the CLIP paper using the VIT-Large model achieved a recall rate of 37.8%.

In this particular experimental setup, we utilize the previously computed similarity matrix and perform a transpose operation on it. By transposing the matrix, we switch the rows and columns, effectively swapping the text samples with the corresponding image samples. When the objective is to retrieve an image given a specific text, we select the row of similarity scores corresponding to the text sample of interest. From this row, we identify the image sample with the highest similarity score, which is then presented as the result, indicating the most relevant image associated with the given text based on similarity metrics. This approach allows for effective image retrieval based on textual queries.

v) **Task 5 : Visual Question Answering**
Setup: For this task, we used the VQA2 dataset, specifically the validation set. The dataset consists of 40,504 images and 214,354 associated questions. The performance of the models is evaluated using the VQA Accuracy metric, which measures the accuracy of answering questions based on the provided images. The standard set by the MCAN paper suggests achiev-

ing an accuracy range of 65% to 70% for this task, with the specific range depending on the depth of the Co-Attention mechanism used.

In this experimental setup, we adopted the MCAN paper's configuration for the Visual Question Answering (VQA) task. However, we made modifications to the original MCAN architecture by substituting the text encoder (consisting of GLoVe embeddings and LSTM) and the image encoder (utilizing Faster R-CNN) with the encoders from the CLIP model. Subsequently, we retrained the model for 13 epochs using the VQA2 training set. By incorporating CLIP's encoders, we aimed to leverage the benefits of the CLIP model's visual and textual understanding capabilities and explore their impact on VQA performance.

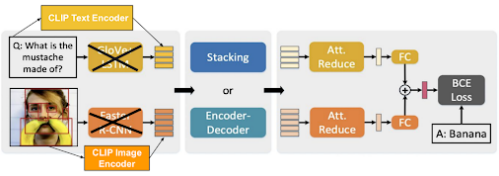

Figure 6. Visual Question Answering

5. Results

The evaluation of Vanilla Quantization on the given tasks resulted in the following performance metrics:

Across various computer vision tasks, the impact of reducing the bit-width on accuracy varies. In image classification, there is a significant drop of 11% in accuracy when the bit-width is reduced from 16 to 8, and it further deteriorates to almost 0 when the bit-width reaches 4. This sharp decline suggests a substantial effect of reducing the bit precision on classification performance. (*Figure 7*)

In contrast, image captioning exhibits a relatively smaller drop of 2% in CIDEr score between bit-width 32 and bit-width 8. However, as the bit-width decreases to 4, the accuracy drop increases to 4%. Despite this drop, it is still significantly less severe compared to image classification. (*Figure 8*)

For image-text retrieval, the accuracy drop is approximately 3% when transitioning from bit-width 32 to bit-width 8, which is slightly higher than that observed in image captioning. However, the accuracy drops to 0 when the bit-width is reduced to 4, indicating a complete loss of performance. (*Figure 9*)

Text-image retrieval demonstrates the least sensitivity to bit-width reduction, with only a 1% drop in accuracy from bit-width 32 to bit-width 8. The accuracy drop approaches 0 as the bit-width decreases to 4, indicating minimal impact on retrieval performance. (*Figure 10*)

In visual question answering (VQA), there is a 4% drop in accuracy from bit-width 32 to bit-width 8. However, the accuracy does not reach 0 even at bit-width 4, suggesting that VQA is more resilient to the decrease in bit precision compared to other tasks. (*Figure 11*)

In summary, reducing the bit-width has varying effects on accuracy across different computer vision tasks. Image classification and image-text retrieval experience signifi-

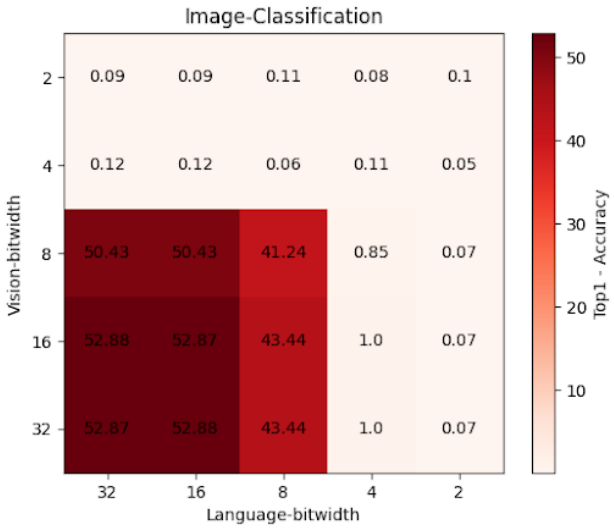

Figure 7. Result of Image-Classification

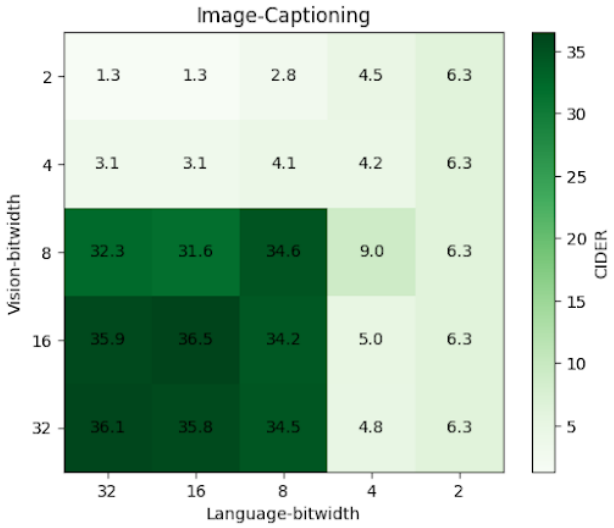

Figure 8. Result of Image-Captioning

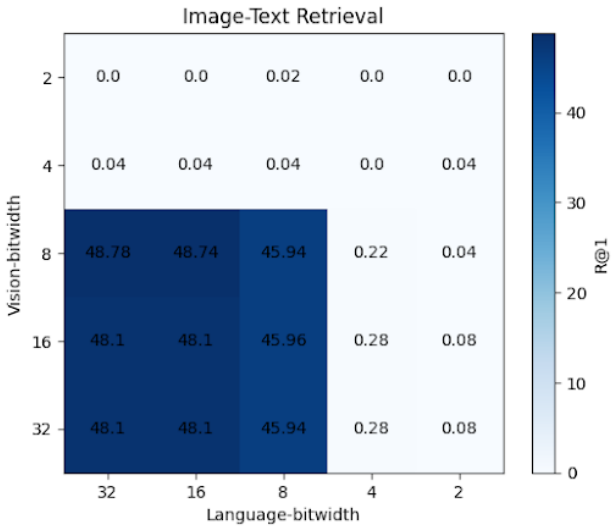

Figure 9. Result of Text-Image Retrieval

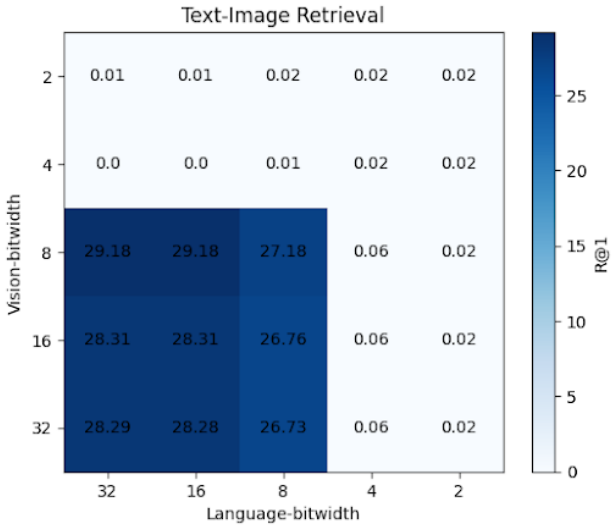

Figure 10. Result of Text-Image Retrieval

cant drops, while image captioning and VQA exhibit more resilience. Text-image retrieval demonstrates the least sensitivity to bit-width reduction, with minimal accuracy degradation.

i) **Cosine Similarity Graph :**
The cosine similarity graphs illustrate the pairwise distribution of different bit-widths across the three datasets employed in the experiments. As the bit-widths approach 32, the graphs exhibit less deviation from the non-quantized cosine similarities distribution. Notably, the lines representing graphs with bit-widths closer to 2 display the highest degree of deviation, indicating a significant departure from the expected pattern. (*Figure 12, 13, 14*)

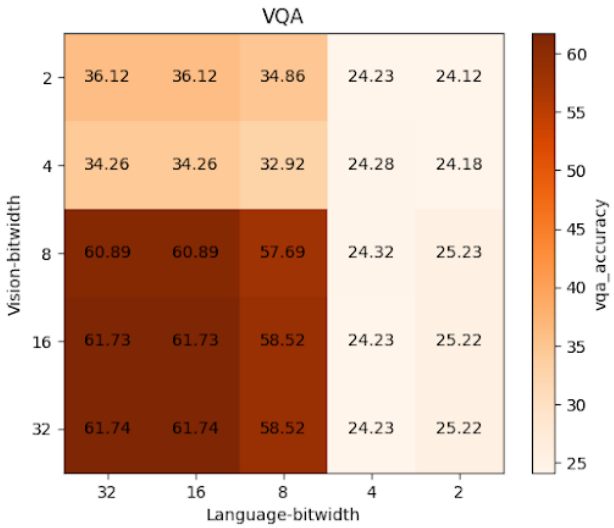

Figure 11. Result of VQA

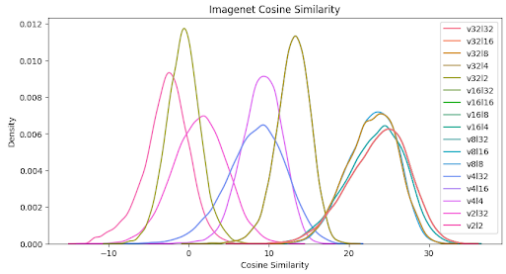

Figure 12. Imagenet Cosine Similarity

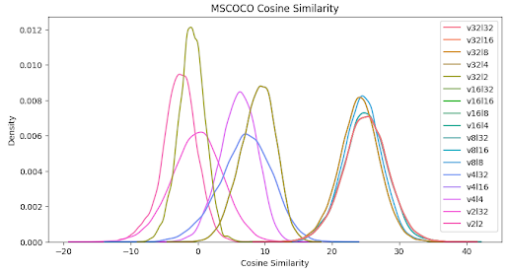

Figure 13. MSCOCO Cosine Similarity

The evaluation of PTQ4CLIP on the specific tasks resulted in the following performance metrics:

- Text-to-image retrieval: R@1 (Recall at 1) achieved a score of 26.74%.

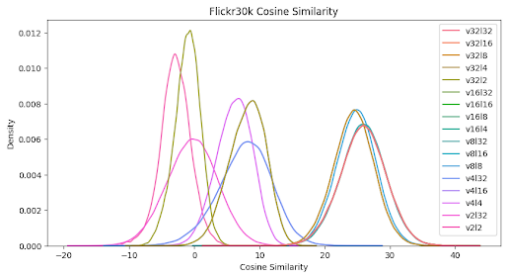

Figure 14. Flickr30k Cosine Similarity

- Image-to-text retrieval: R@1 (Recall at 1) attained a score of 45.72%.

The following table shows a comparative analysis of the approaches used in the experiments:

Image-Classification Task:

Technique	Bit-width	Score (%)
Original CLIP	32	55.92
Vanilla Quantization	8	41.24
Vanilla Quantization	4	0.11
PTQ4CLIP	8	51.13
PTQ4CLIP	6	42.23
PTQ4CLIP	4	5.39

Table 2. Table showing the accuracy of each technique

Image-Captioning Task:

Technique	Bit-width	Score (%)
Original CLIP	32	36.1
PTQ4CLIP	8	32.5

Table 3. Table showing the accuracy of each technique

6. Conclusion

We quantized the CLIP model and performed basic multimodal tasks to evaluate the performance. The empirical results suggest that a vanilla PTQ applied on a vision-language model yields a considerable decrease in the accuracy of the model for a vision and language with lower bit-widths completely destroys the model. Whereas, applying twin uniform quantization along with hessian guided metric increases the accuracy while also giving an advantage in performance. From the data, it can be concluded that PTQ applied on CLIP results in better accuracy than Vanilla quantization where accuracy from PTQ is around 51% compared to the accuracy for vanilla quantization is around 41% for 8-bit quantization. Our approach can be considered as a baseline in the area of vision-language model quantization and can be scaled across various State-Of-the-Art models and requires more thorough testing. Also, there is a large scope for improvising the quantization strategy by applying mixed precision and smoothing techniques.

References

[1] Feiyang Chen, Yadi Cao, and Zhaoqian Wang. Dq-clip: Post-training quantization for dynamic quantized clip. 2022. 1

[2] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*, 2022. 2

[3] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*, 2021. 3

[4] Dachuan Shi, Chaofan Tao, Ying Jin, Zhendong Yang, Chun Yuan, and Jiaqi Wang. Upop: Unified and progressive pruning for compressing vision-language transformers. *arXiv preprint arXiv:2301.13741*, 2023. 2

[5] Guangxuan Xiao, Ji Lin, Mickael Seznec, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. *arXiv preprint arXiv:2211.10438*, 2022. 2

[6] Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183, 2022. 2

[7] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, pages 191–207. Springer, 2022. 1, 2, 3