DI725
Quantized Vision-Language Adapter (QVLA) for Efficient Fine-Tuning and Inference
Final Report

Nisan Yıldız
Dept. of Medical Informatics
Informatics Institute, METU
Ankara, Turkey
nisany@metu.edu.tr

***Abstract*—Recent advances in vision language models (VLMs) favor large base models followed by task-specific fine-tuning. With such a framework, full fine-tuning has become another associated cost for deploying VLMs. To address this cost, parameter efficient methods like Adapters and LoRA have been developed, with QLoRA introducing quantization for higher efficiency. Here, we propose the Quantized Vision Language Adapter (QVLA), based on the Bottleneck Adapter and combining quantization with adapters to enable efficient, scalable fine-tuning. Combined with the PaliGemma base-model, a VLM emphasizing a compact and efficient architecture; QVLA aims to offer practical deployment in resource-constrained settings.**

I. Introduction

Large scale neural networks trained on massive datasets have led to significant breakthroughs in unimodal domains such as natural language processing (NLP) and computer vision (CV). However, many real world problems require inferences across multiple modalities. Vision Language Models (VLMs) have emerged as a key era of research, incorporating visual and textual modalities in a unified framework.

A recent trend in VLM development mirrors those in NLP [1] [2], where pre-trained (large) base models are later fine-tuned for various applications. This paradigm has incentivized ever increasing numbers of parameters, reaching billions in the base models. As a result, it has also become increasingly cumbersome to fine-tune these models with the traditional way of updating the entire set of billions of parameters. In addition to the computational cost of fine tuning, fine-tuning can also make large models susceptible to catastrophic forgetting [3]. This prompted research into parameter efficient fine-tuning (PEFT) methods, where the aim is to limit the number of parameters updated during fine-tuning.

Research on parameter efficient fine-tuning has mainly focused on prompt-tuning or methods that fine-tune over a significantly decreased size of parameters. We will not be going into the details about prompt-tuning here. Examples of the latter approach include Adapter based methods where small adapter heads are inserted into the pre-trained model and the resulting model is fine-tuned with (usually) the pre-trained weights frozen; it also includes methods like LoRA where a low rank representation of the model is trained with the frozen weight of the pre-trained model and concatenated with the original model. Both of the latter approaches perform similarly in terms of training cost and accuracies [4] [5] [6], however, Adapter based methods incur a small inference cost in contrast to LoRA.

For tasks that require a good degree of interaction between different modalities, such as image captioning or visual question answering, Adapter architectures with cross-modal interactions, such as interactions between image encoders and text decoders, have been shown to perform the best, in contrast to architectures that incorporate non-interacting Adapters [5]. One such example is the VL-Adapter [4], which has been shown to offer better accuracy than LoRA in image captioning tasks with similar training cost. Quantization is a method to decrease memory usage and computational costs of neural networks in both pre- and post-training by lowering the bit lengths of parameters saved [7]. Recently, QLoRA introduced a method to quantize the base model to a much lower memory footprint, with higher precision operations for Low Rank Matrix updating; achieving high performance with substantial memory and computational savings [8]. Although QLoRA focuses on LoRA based fine-tuning, its general findings are applicable to other Adapter based fine-tuning methods. However, effects of quantization with adapters have not yet been thoroughly investigated in the VLM field.

II. Dataset

We utilize a remote sensing imagery dataset that consists of 4521 sattelite images with 5 captions per image that briefly describe it, for a total of 222605 captions. See figure 1 for examples. A more detailed summary of the dataset is available in the notebook titled ”dataset_visualizations.ipynb”.

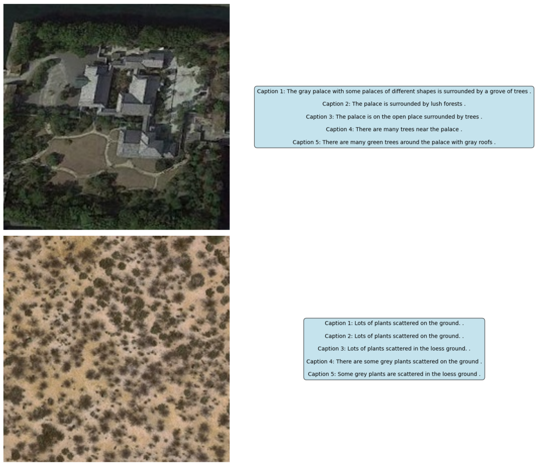

Fig. 1. Two examplary sattelite images from the dataset with associated captions.

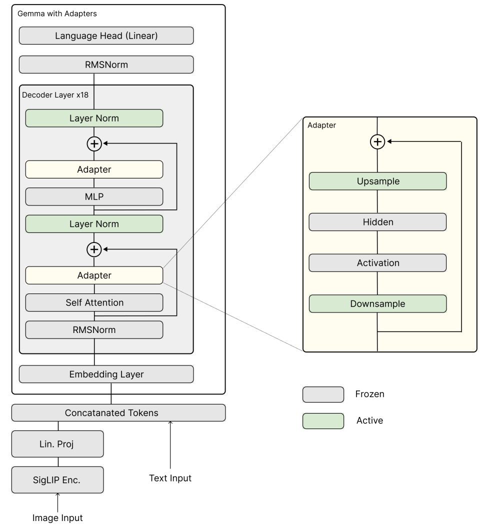

Fig. 2. Illustration of QVLA framework. Each Decoder Layer within the Language Model (Gemma) is supplemented with two adapter modules. Majority of the model weights are frozen during training. Layer widths are not in scale.

III. Modeling

Here, we propose the Quantized Vision Language Adapter (QVLA) that incorporates quantization with the Adapter framework to achieve efficient fine-tuning of Vision Language Models [9]. We use PaliGemma [10], a pre-trained VLM model intended to be a versatile base for various fine-tuning applications, as our base model. The PaliGemma family of models propose compact architectures compared to similar performing contemporary models, that seek to maintain performance while improving efficiency.

We utilize a quantization method as outlined in [8]. QLoRA quantization have been shown to significantly decrease the memory cost while having a performance similar to that of a full 16-bit model. We use the Adapter architecture as proposed by [9].

We use the following configurations on the RISC image captioning dataset, to compare their accuracy, training and inference times:

- Traditional fine-tuning of the PaliGemma model
- VL-Adapter fine-tuning without quantization
- QVLA (quantization + Adapters)

A. Bottleneck Adapters

We introduce Bottleneck Adapters into the decoder layers of the language model (Gemma) of the PaliGemma model. Previous work VL-Adapters had shown that adding Adapters into the Language encoding layer of a VL model didn't significantly increase the final accuracies after training [4]. Following these results, we do not insert adapters into the vision encoder part of the architecture (SigLIP). The model specifics are illustrated in 2. Briefly, we insert two adapter modules within each decoder block inside Gemma, of which there are 18; making the total adapter modules introduced 36.

During fine-tuning, we train the adapters, as well as the normalization layers after each adapter. This method reduces the number of parameters updated from approximately 3 billion to approximately 19 million, just 0.65% of the original parameter count I.

B. Quantization

In addition to the introduction of adapters, we quantize our model as outlined [8]. Briefly, we use 4-bit quantization of the base model weights with the NF4 datatype; that achieves better results compared to a standard 4-bit numeric type for data that is approximately normal, which neural network weights have shown to adhere.

We use double quantization to achieve faster look-ups during conversion to NF4, followed by a 16-bit data type used during weight update computations.

All steps of quantization are outlined in the original QLoRA paper, here we adapt it to non-LoRA adapter types.

IV. Evaluation

As mentioned above, we perform an ablation study to compare inference & training times, and accuries of the base PaliGemma model (paligemma-3b-pt-224), PaliGemma with Adapters, and QVLA (PaliGemma with Adapters and Quantization). We report training and inference times as achieved on a single A100 GPU with 40GB of Memory. We use CIDEr metric to evaluate the accuricies of the models after training, except in the case of base model; which we were not able to train due to memory constraints. The training time given for the Base-Model is therefore an estimate, given by the Transformers Trainer method.

During training, we use batches of 4 image-text inputs. We use ”caption en” as the prefix to the model, as suggested by the authors [10].

We use the RISCM dataset for training and inference, with a total of 44k images. We utilize a 80-10-10 split for traing-evalutation-test sets. Inference are done on the test set of approximately 4k images. During training and inference, we randomly pick a caption out of 5 total captions for each image.

V. Results

Introduction of adapters into the model decreases the number of parameters updated to mere 0.65% of the base model I. This significantly decreases the memory requirements during training. Both QVLA and PaliGemma with Adapters were able to be fine-tuned in a single A100 GPU, whereas it was not possible to fine-tune the base model.

TABLE I
Adapter and Model Parameter Summary

Name	Architecture	#Param	%Param	Active
adapter_lm	bottleneck	18,952,704	0.648	1
Full model	–	2,923,466,480	100.000	

Introducing the quantization further leads to decrease the memory requirements of the model, where the base-model, as well as model with adapters require approximately 11.5 GBs of memory to just load the weights, the QVLA model requires only 2.8 GBs. of memory; a 4-fold decrease II. We can further see memory requirements during training in 3.

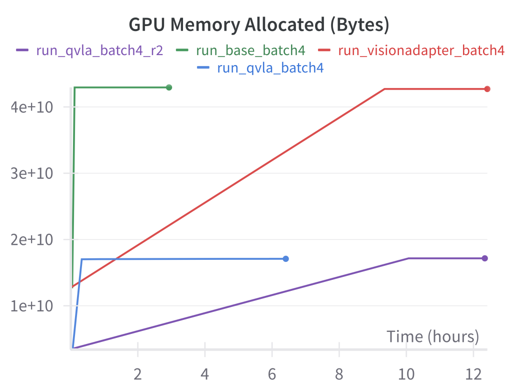

Fig. 3. Memory requirements of base-model, model with Adapters and QVLA during training.

Both QVLA and Adapters decrease the inference time required, from about 2.5 hours for 4k images to about 1.5 hours in both cases. We would expect QVLA to further decrease the inference times, but we did not observe such a trend.

TABLE II
Comparison of Model Architectures: Training/Inference Time and Memory Usage

Architecture	Training Time (h)	Inference Time (h)	Memory (min / mean)(GB)
Base Model	12.00*	2.51	11.5 / –
Vision Adapter	9.18	1.39	11.5 / 39
QVLA	9.42	1.54	2.8 / 15.5

We also see a dramatic increase in caption accuracies when using Adapters and QVLA compared to pre-trained base model III, from a CIDEr score of 0.006 in base model to 1.4336 and 1.5533 in Adapter model and QVLA respectively. Here, QVLA seems to have achieved better accuracy despite having lower bit resolution of weights, however, since no repeats were done, this probably isn't significant. We can at least say we do not see a decrease in accuracy with quantization.

See figure 4 for example captions generated in each step and associated notebooks for a larger set of example captions.

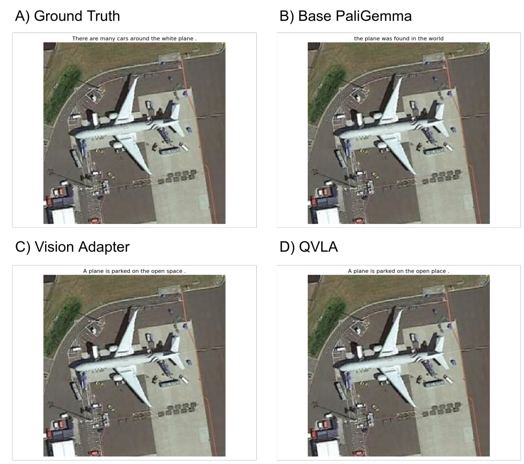

Fig. 4. Example captions generated by (B) base PaliGemma model without fine-tuning, (C) PaliGemma fine-tuned with vision adapters, and (D) PaliGemma fine-tuned with QVLA, along with the ground truth caption (A)

TABLE III
CIDEr Scores for Model Architectures

Architecture	CIDEr Score
Base Model	0.0006
Vision Adapter	1.4336
QVLA	1.5533

VI. Discussion

An inspection of the captioning accuracies of QVLA and Vision adapters tells us that both of the methods achieve a satisfactory fine-tuning performance, in contrast to low score achieved by the base PaliGemma model. Qualitative inspections of captions generated by QVLA and Vision Adapters tells us that they produce very similar results, while QVLA offers the advantage of significant memory reduction during training. One of the drawbacks of the current study is the lack of full fine-tuning of the PaliGemma model for comparisons with QVLA and Vision Adapters; although the inability to fine-tune the full model in classical ways does tell us something about the prohibitive cost of such methods when fine-tuning large models.

VII. Conclusion

Although we cannot compare our results to a full fine-tuning without adapters due to memory constraints, our results suggest that QVLA is a good and viable option for fine-tuning vision-language models with low computational over-heads. It offers high-quality fine-tuning of large models, as examplified by PaliGemma in this study, in situation where computational costs are prohibitive.

VIII. Code and Data

All of the code associated with the project is available at: https://github.com/NisanYildiz/DI725-project

Wandb report available at: https://api.wandb.ai/links/yildizz-nisan-middle-east-technical-university/9op1wp8f

References

[1] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[2] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph, "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[3] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. Zhang, "An empirical study of catastrophic forgetting in large language models during continual fine-tuning," 2025. [Online]. Available: https://arxiv.org/abs/2308.08747

[4] Y. Sung, J. Cho, and M. Bansal, "Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks," *CoRR*, vol. abs/2112.06825, 2021. [Online]. Available: https://arxiv.org/abs/2112.06825

[5] J. Xing, J. Liu, J. Wang, L. Sun, X. Chen, X. Gu, and Y. Wang, "A survey of efficient fine-tuning methods for vision-language models — prompt and adapter," *Computers Graphics*, vol. 119, p. 103885, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0097849324000128

[6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *CoRR*, vol. abs/2106.09685, 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[7] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, "A white paper on neural network quantization," *CoRR*, vol. abs/2106.08295, 2021. [Online]. Available: https://arxiv.org/abs/2106.08295

[8] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," 2023. [Online]. Available: https://arxiv.org/abs/2305.14314

[9] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 2790–2799. [Online]. Available: https://proceedings.mlr.press/v97/houlsby19a.html

[10] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, T. Unterthiner, D. Keysers, S. Koppula, F. Liu, A. Grycner, A. Gritsenko, N. Houlsby, M. Kumar, K. Rong, J. Eisenschlos, R. Kabra, M. Bauer, M. Bošnjak, X. Chen, M. Minderer, P. Voigtlaender, I. Bica, I. Balazevic, J. Puigcerver, P. Papalampidi, O. Henaff, X. Xiong, R. Soricut, J. Harmsen, and X. Zhai, "Paligemma: A versatile 3b vlm for transfer," 2024. [Online]. Available: https://arxiv.org/abs/2407.07726