



Quantization Methods for Transformer-Based Models on Edge Devices

Zhixiang Zeng

School of Advanced Manufacturing, Guangdong University of Technology, Guangzhou, 510006, China

zengzhixiang1@mails.gdut.edu.cn

Abstract. As the rapid development and the broad usage of Transformer-based models like ChatGPT and Gemini, and the popularity of edge devices like mobile phones, the request of deployment of Transformer-based models on edge devices is growing urgently. However, the original Transformer-based models have numerous parameters and large expense of computation and storage, which made it almost impossible for deployment on edge devices. To tackle this problem, quantization, which is an efficient and energy-saving model compression approach, eventually emerged and gradually developed into the mainstream method of compressing models on edge devices. This article reviews various state-of-the-art quantization methods, focusing on three key areas: image recognition, large language models, and image generation. We analyze their applicability, application requirements, and performance across different tasks, and compare them with typical experiments and case studies. We further identify shortcomings in existing methods in terms of accuracy preservation, model robustness, and hardware adaptability, and propose potential areas for improvement. These findings aim to contribute to a deeper understanding of on-device model quantization and provide a reference for more efficient and universal large-scale model deployment.

Keywords: Quantization, Large Language Models, Edge Devices.

1 Introduction

Since the launch of ChatGPT in November 2022, artificial intelligence technology has entered a new stage of development [1]. In the following 18 months, large language models (LLMs), including Gemini, Claude, and the open-source model DeepSeek, have experienced explosive growth. AI applications have gone beyond text generation, and the launch of Sora in December 2024 has made the leap from text to video. At the same time, with the rapid popularization of mobile Internet and smartphones, the demand for AI functions in terminal devices has become increasingly prominent. In this context, how to achieve the optimized deployment of large models on mobile terminals has become a key technical challenge that needs to be solved urgently.

Thanks to the breakthrough of the Transformer architecture, the above technical vision is gradually being realized. The self-attention mechanism adopted by the

Transformer gives it excellent parallel computing characteristics, which can make full use of modern hardware accelerators to significantly improve training efficiency [2]. However, large-scale models based on the Transformer architecture are still difficult to run efficiently on consumer-grade computing devices. For example, the GPT-3 model contains 175 billion parameters. A single token generation requires 350GFLOPs operations and requires at least 350GB (FP16 format) of memory space [3]. This far exceeds the processing power of existing civilian computing devices. Through this background, quantization technology has emerged as an important model compression method. This technology converts high-precision parameters in the original model (32-bit floating point FP32) into low-bit representation (such as 8-bit integer INT8 or 4-bit integer INT4), effectively reducing the computational intensity of multiplication and accumulation (MAC) operations, thereby reducing the overall computational overhead at the system level [4] [5]. Compared with other model compression methods, quantization technology has become the mainstream technical solution for mobile Transformer model compression due to its advantages such as high computational efficiency, convenient deployment and low memory usage, and has achieved rapid development in recent years [6] [7].

This study systematically explores quantization techniques for Transformer-based large models in application scenarios across multiple types of edge devices. The overall structure of the paper is arranged as follows: first, the theoretical foundation section defines the core concepts of quantization and provides an in-depth analysis of its two methodological systems—post-training quantization (PTQ) and quantization-aware training (QAT) and their technical characteristics; subsequently, for different edge application scenarios such as mobile terminals and embedded systems, the study offers a comprehensive review of the innovation mechanisms, practical effectiveness, and technical challenges of existing mainstream quantization methods, and proposes targeted optimization schemes; finally, the research findings are systematically summarized, and possible directions for future technological breakthroughs in this field are outlined. The academic value of this study lies in establishing a classification framework for large-model quantization techniques oriented toward multi-scenario edge devices, which not only provides a clear research framework for the academic community but also offers a reference pathway for engineering practice in industry.

2 Quantitative Methods and Domain Analysis

2.1 Image Classification

Image recognition tasks on edge devices require small models and high performance. Hybrid architectures (CNN + Transformer) and hardware-aware PTQ are currently hot research topics. This article introduces two cutting-edge image recognition quantization technologies: the PTQ-based HyQ quantization method and the QAT-based GradQ-ViT method.

Method Introduction. The HyQ quantization algorithm is optimized for emerging hybrid Transformer-based models. Based on the hardware design of AI accelerators such as GPUs, the method uses linear functions to approximate the softmax function using only integers in the Transformer module, reducing the amount of computation and improving efficiency. At the same time, the method introduces quantization-aware distribution scaling to address the large outliers caused by inter-channel variance in the convolutional layer, alleviating the problem of insufficient accuracy after quantization to a certain extent [8].

GradQ-ViT, which builds on QAT, achieve higher accuracy for demanding image recognition applications. This method focuses on solving the gradient quantization problem of ViT in the QAT process. By proposing a gradient quantization framework and using an outlier-resistant loss function to solve the training stability problem, it also applies gradient scaling and quantization error analysis to adaptively allocate learning rates to ensure the performance of model training [9].

Performance analysis. In terms of accuracy, HyQ, under INT8 full quantization (weights and activations), only reduces MobileViT-S's accuracy by 0.39% (Top-1) compared to FP32 on ImageNet-1k, according to the data in Table 1. Fig. 1 also shows that GradQ-ViT's accuracy on MobileViT-S decreases by approximately 0.09%, making it nearly lossless. Furthermore, GradQ-ViT maintains good accuracy across different models. Overall, the difference between the two is not significant, which means both can be considered high accuracy.

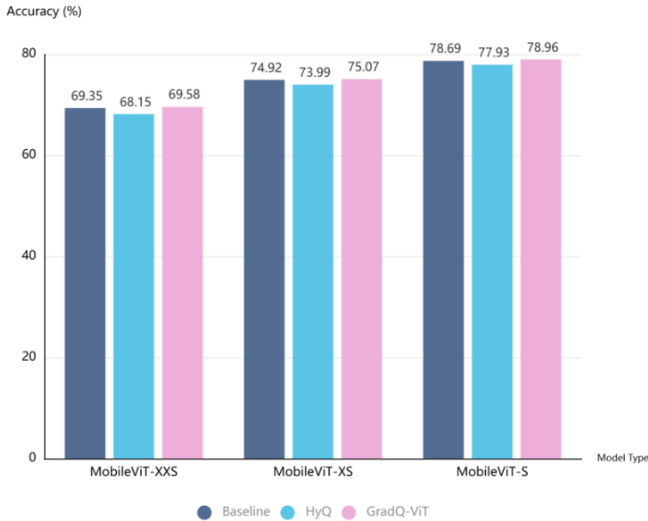


Fig. 1. Quantization accuracy of different quantization methods on the MobileViT models (%)

In terms of efficiency, HyQ reduces model static storage to approximately 1/4 of the original size. In FPGA implementations, its resource usage for lookup tables and flip-flops can be reduced to 1/1.8 to 1/2.1 of the original. GradQ-ViT significantly reduces

storage consumption during training. For example, in experimental data from MobileNet, the intermediate storage usage significantly reduced by enabling gradient quantization in a CUDA 11.8 environment, ultimately achieving a 2.06x speedup.

In terms of time consumption, HyQ as a post-training quantization (PTQ) method only requires a small number of calibration samples to complete training, while GradQ-ViT achieves measured improvements in training throughput and memory efficiency by reducing the bit width and data transfer requirements of the gradient optimizer state.

Method classification Through the above analysis, we classify the applicable scenarios of the two methods and summarize them in Table 1 below.

Table 1. Summary of applicable scenarios for quantization method on image classification

Application Scenarios	Recommend Approach	Typical devices	Reasons
Static memory usage sensitive scenarios	HyQ	Embedded GPUs, FPGA	Int8 quantization can significantly reduce model size.
Specific low-power hardware requirement	HyQ	Mobile SoC GPUs	HyQ is explicitly designed for integer-only implementations of the softmax approximation
Training memory & bandwidth sensitive scenarios	GradQ-ViT	32-bit SoCs,	Quantized gradients reduce bandwidth and memory requirements

Current Challenges and Solution. Admitting the limitations of current methods above, Table 2 gives their statements and tries to give possible solutions for quantization in the field of image classification:

Table 2. Current challenges and solutions for quantization on image classification

Challenges	Challenges Details	Solutions
Inconsistent hardware and software stacks	Most evaluations are performed on GPUs/FPGAs; replicating the results on common mobile SoCs (ARM CPUs, Adreno/Mali, NPUs) requires additional work and specialized kernel adaptation.	Improve the unified testing platform. At this stage, test data is still limited to high-end hardware, and there is only a theoretical description of the feasibility of mobile devices. In the future, benchmark data on mobile SoC chips can be added.
Difficulty on trade-off	Reducing the bit width does not necessarily reduce energy	Lightweight Quantization with calibration. Minimal adjustments on key

between energy consumption and latency on edge devices	consumption. The energy consumption during actual inference or training needs to be measured on the target hardware.	modules or using knowledge distillation can reduce training resources occupation, which is beneficial for reducing latency and energy consumption.
--	--	--

2.2 Large Language Models (LLM)

Large language models (LLMs) are widely used on edge devices. In this field, the key metrics for LLM model quantization are task accuracy (MMLU), inference latency and speed (TTFT/TGS), and deployment energy consumption [10]. This section compares and analyzes several LLM model quantization methods that excel in these metrics, categorizes them, and provides recommendations for specific edge device scenarios.

Method Introduction. AWQ discovered that not all weights are equally important, while quantizing LLM weights. Protecting just 1% of significant weights can significantly reduce quantization error. Therefore, AWQ uses activation statistics to identify salient weight channels, quantizing only the 99% of weights and scaling them to reduce error [11]. This approach eliminates the need for retraining, resulting in significant results and hardware-friendly performance.

SpinQuant optimizes the problem of large errors caused by outliers in the quantization process [12]. The algorithm identifies a series of suitable rotation parameterization methods and achieves the same output in the full-precision Transformer architecture, thereby improving quantization accuracy.

LLM-QAT is a quantization method based on QAT and optimized for large language models. To address the efficiency and memory usage limitations of QAT, LLM-QAT quantizes the KV cache, improving throughput. It also enables data-free distillation during training, preserving the original output distribution [13]. This method achieves a comprehensive balance between task accuracy and training efficiency.

Performance analysis. In terms of task performance, Fig. 2 shows that SpinQuant achieves task accuracy close to BF16 at W4A8 precision, significantly improving over baseline methods. LLM-QAT demonstrates its MMLU advantage in larger-scale llama models. AWQ achieves a significant advantage in complex inference on llama models of similar size to those used in the LLM-QAT test, as evidenced by its leading GSM8K performance.

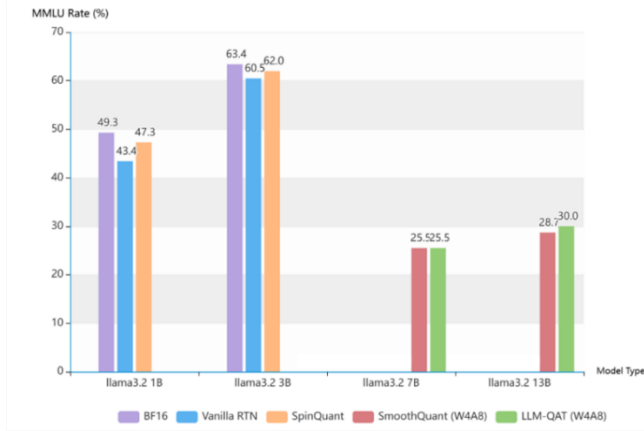


Fig. 2. MMLU performance of LLM-QAT, SpinQuant, and their baseline methods under different llama models.

In terms of operational efficiency, SpinQuant's TTFT value decreases with increasing batch size (BS) when deploying the LLaMA-3 70B model on an NVIDIA H100, indirectly demonstrating its latency advantage as batch size increases. AWQ's additional TinyChat inference stack achieves approximately 3x speedup compared to HuggingFace FP16 on desktop and mobile GPUs. In contrast, LLM-QAT does not demonstrate a significant efficiency advantage, as its efficiency depends on the actual model size.

In terms of memory usage, SpinQuant uses 4-bit full quantization, theoretically reducing memory usage to 1/4 (compared to FP16 or FP32). AWQ's support for INT3 quantization significantly reduces weight storage and may even enable deployment of large 70B Llama-2 models on certain mobile GPUs. LLM-QAT focuses on quantizing the KV cache, significantly reducing peak memory usage and allowing this memory to be used to accommodate longer batches or context data.

Method classification Through the above analysis, we classify the applicable scenarios of the two methods and summarize them in Table 3 below:

Table 3. Summary of applicable scenarios for quantization method on LLM

Application Scenarios	Recommend Approach	Typical devices	Reasons
Real-time inference scenarios, resource-constrained devices	AWQ	Mobile GPU, NPU, FPGA devices	Low-bit quantization significantly reduces the size, the model does not require training, and has strong generalization capabilities.

High-precision demanding scenarios, multi-modality LLMs	SpinQuant	Edge devices with mildly strong GPUs	The highest accuracy which is closest to full precision among three methods brings it a significant advantage.
Training memory & bandwidth sensitive scenarios	LLM-QAT	NPU with sufficient memory, low-memory Graphic Cards	Performance is steadily improved at extremely low bit rates, and KV Cache quantization effectively reduces peak memory usage.

Current Challenges and Solution. Admitting the limitations of current methods above, Table 4 gives their statements and tries to give possible solutions for quantization in the field of LLM.

Table 4. Current challenges and solutions for quantization on LLM.

Challenges	Challenges Details	Solutions
Limitations in extreme low-bit or complex scenarios	AWQ mainly focuses on weight quantization and does not cover activations. Its performance may degrade when applying lower bit-widths or activation quantization. SpinQuant relies on rotation matrices to remove outliers without weight fine-tuning, which can restrict its flexibility. Although LLM-QAT shows excellent performance under 4-bit settings, its activation quantization (especially at 4-bit) remains unsatisfactory.	Explore stability mechanisms under ultra-low precision. For extreme quantization scenarios ($W \leq 4, A \leq 4$), investigate mixed-precision protection strategies (retaining higher precision in critical layers or channels) or residual correction mechanisms (introducing lightweight correction networks) to mitigate catastrophic quantization collapse.
Lack of a unified evaluation framework	Current methods often adopt different evaluation standards, resulting in the absence of a consistent, end-to-end benchmarking system across models, tasks, and bit-widths. This makes direct comparisons between methods less intuitive and less reliable.	Establish a unified benchmarking platform. Design a standardized evaluation framework that covers multiple model sizes and quantization levels (e.g., W4/A8/KV4, W4/A4, or even W2). This would allow comprehensive comparisons of methods across various models (e.g., LLaMA-2/3 of different scales) and downstream tasks (zero-shot reasoning, generation, etc.).

Contradiction between generalization ability and hardware efficiency	The three methods above all need extra resource expense for generalization and specific hardware architecture for efficiency, which makes it difficult for resource-limited scenarios and hardware variety.	Providing better support for hardware-friendly environment and generalization. Optimize the computational flow and data layout of hybrid methods based on NPU or mobile embedded hardware characteristics such as INT8/INT4 instruction support and GEMM unit limitations, and testing new methods via multiple modalities or through different specific tasks.
--	---	---

2.3 Image Generation

Image generation tasks demand higher accuracy and real-time responsiveness, such as intelligent image inpainting or one-click enhancement in photo applications. While diffusion models remain the dominant paradigm, recent studies have shown that replacing the U-Net backbone with a Transformer can accelerate generation and improve output quality. This section analyzes quantization methods for Diffusion Transformers (DiT), with a focus on enabling efficient deployment on mobile devices.

Method Introduction PTQ4DiT addresses extreme channel outliers and timestep dependencies in DiT models. It applies Channel-wise Saliency Balancing (CSB) and Spearman’s ρ guided Saliency Calibration (SSC) to reduce activation variance and quantization error [14].

ViDiT-Q combines PTQ with a metric-decoupled mixed-precision strategy, aiming to balance accuracy and memory efficiency [15]. For layers or timesteps that are highly sensitive to bit-width reduction, it introduces ViDiT-Q-MP, a decoupled mixed-precision rule that improves stability under quantization.

Performance analysis. As shown in Fig.3, in terms of image quality, evaluation on Fréchet Inception Distance (FID) under W8A8 and W4A4 configurations shows that PTQ4DiT and ViDiT-Q achieve nearly comparable results at W8A8. At W4A8, however, PTQ4DiT suffers from a one-order-of-magnitude increase in FID degradation compared to full precision, while ViDiT-Q maintains a much smaller gap (~0.88), demonstrating stronger robustness at lower precision.

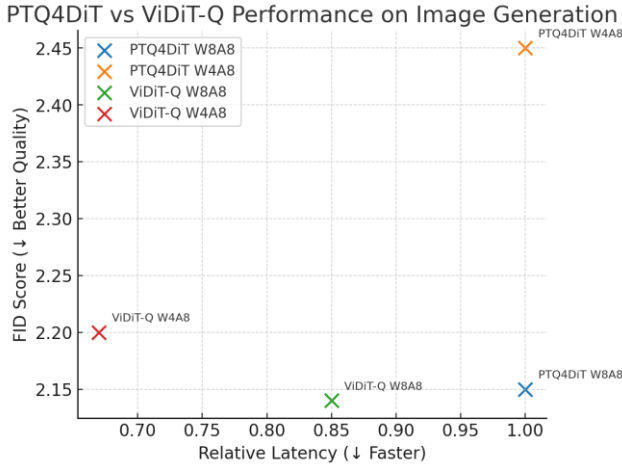


Fig. 3. FID Score and relative latency of two image generation quantization approach under different quantizing modes

Regarding memory, ViDiT-Q reports 2–2.5× memory savings under low precision [15]. Although PTQ4DiT does not provide explicit compression ratios, its PTQ-based design substantially reduces activation and model memory requirements.

In terms of efficiency, ViDiT-Q leverages low-bit operators and optimized kernels to achieve 1.4–1.7× acceleration in training and inference [15]. PTQ4DiT, by contrast, requires no retraining or fine-tuning, making it attractive for rapid deployment with minimal time cost.

Method classification Through the above analysis, we classify the applicable scenarios of the two methods and summarize them in Table 5 below.

Table 5. Summary of applicable scenarios for quantization method on image generation

Application Scenarios	Recommend Approach	Typical devices	Reasons
Offline, no training resources	PTQ4DiT	Flying drones, mini robot GPUs	Zero training cost, simple deployment, and minimal requirements.
Real-time scenarios, latency-sensitive tasks	ViDiT-Q	AR, VR, interactive filters	Mixed-precision with optimized kernels improves latency.
Memory-constrained environments	ViDiT-Q	Embedded GPU, IoT SoCs	Significant memory compression performance compared to PTQ4DiT

Short-term development cycle, maintainable scenarios with robustness	PTQ4DiT	Modular embedded devices, Industrial control equipment	Zero training request and few pressure for Additional system support.
--	---------	--	---

Current Challenges and Solution Admitting the limitations of current methods above, Table 6 gives their statements and tries to give possible solutions for quantization in the field of image generation.

Table 6. Current challenges and solutions for quantization on image generation.

Challenges	Challenges Details	Solutions
Limited hardware support for ultra-low-bit inference	Current NPUs/NNAPI stacks have insufficient support for 4-bit weights or mixed precision, which directly affects both methods' performance	Develop inference kernels optimized for low-bit operations. Tailor quantization operators for mobile NPUs and energy-constrained devices.
Quantization-induced artifacts	Especially in multi-frame video generation, quantization may lead to temporal inconsistencies, which can be problematic for stability-sensitive applications.	Enhance stability of quantized generation. Explore decoupling quantization from training, with special attention to PTQ-based methods for generative models.

3 Conclusion

This study provides a comprehensive analysis of quantization methods for Transformer-based models across three domains—image classification, large language models, and image generation—under the constraints of edge deployment. By classifying and comparing state-of-the-art techniques, the paper highlights their strengths, limitations, and application scenarios. It also identifies current challenges in terms of accuracy retention, robustness, and hardware adaptability, while suggesting practical directions for improvement. These findings aim to guide both academic research and industrial practices, ultimately facilitating the real-world deployment of large Transformer models on edge devices.

References

1. Roumeliotis, K.I., and Tselikas, N.D.: ChatGPT and Open-AI Models: A Preliminary Review. *Future Internet*, 15, 192 (2023).

2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I.: Attention is all you need. In: *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 5998–6008. Curran Associates, Long Beach, USA (2017).
3. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., and Amodei, D. et al.: Language models are few-shot learners. In: *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, pp. 1877–1901. Curran Associates, Vancouver, Canada (2020).
4. Lin, J., Hwu, W.M., and Hubara, I.: A survey of quantization methods for efficient neural network inference. In: *Proceedings of the 5th International Workshop on Systems for ML*, pp. 1–8. ACM, Virtual (2021).
5. Stanković, L., and Mandić, D.: Convolutional neural networks demystified: A matched filtering perspective-based tutorial. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(6), 3614–3628 (2023).
6. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., and Adam, H. et al.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2704–2713. IEEE, Salt Lake City, USA (2018).
7. Hinton, G., Vinyals, O., and Dean, J.: Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 1-9 (2015).
8. Kim, N.J., Lee, J., and Kim, H.: HyQ: Hardware-Friendly Post-Training Quantization for CNN-Transformer Hybrid Networks. In: *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI-24)*, pp. 4291–4299. IJCAI, Jeju, South Korea (2024).
9. Choi, D., and Kim, H.: GradQ-ViT: Robust and Efficient Gradient Quantization for Vision Transformers. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-25)*, vol. 39, pp. 11245–11253. AAAI Press, Philadelphia, USA (2025).
10. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J.: Measuring Massive Multitask Language Understanding. In: *International Conference on Learning Representations (ICLR 2021)*. OpenReview, Virtual (2021).
11. Lin, J., and Xu, L.: AWQ: Activation-Aware Weight Quantization for On-Device LLM Compression and Acceleration. In: *Proceedings of the 41st International Conference on Machine Learning (ICML-24)*, pp. 14562–14578. PMLR, Vienna, Austria (2024).
12. Liu, Z., Chen, J., Sun, T., and He, Y.: SpinQuant: LLM Quantization with Learned Rotations. In: *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, pp. 45211–45225. Curran Associates, New Orleans, USA (2024).
13. Yao, Y., Zhang, H., Deng, C., Zhou, H., and Li, X.: LLM-QAT: Data-Free Quantization Aware Training for Large Language Models. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 857–869. ACL, Miami, USA (2024).
14. Wu, J., Wang, H., Shang, Y., Shah, M., and Yan, Y.: PTQ4DiT: Post-training quantization for diffusion transformers. In: *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, pp. 62732–62755. Curran Associates, New Orleans, USA (2024).
15. Zhao, T., Fang, T., Huang, H., Liu, E., Wan, R., Soedarmadji, W., and Wang, Y. et al.: ViDiT-Q: Efficient and Accurate Quantization of Diffusion Transformers for Image and Video Generation. *arXiv preprint arXiv:2406.02540*, 1-31 (2024).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

