Fine Tuning and Efficient Quantization for Optimization of Diffusion Models

Gurneet Singh and Pranjal Kumar *

Posted Date: 4 February 2025

doi: 10.20944/preprints202502.0138.v1

Keywords: diffusion models; generative ai; fine-tuning; text-to-image models

Article

Fine Tuning and Efficient Quantization for Optimization of Diffusion Models

Gurneet Singh [1] and Pranjal Kumar [2]

[1] Student Researcher, School of Computer Science and Engineering, Lovely professional University, Phagwara, Punjab
[2] Professor, Department of Cognitive Computing, School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab
* Correspondence: pran04146@gmail.com; Tel.: +91-8637511985

Abstract: Diffusion models have emerged as a critical component in the domain of deep generative models for image synthesis. However, the substantial computational demands of their practical deployment limit the viability of developing next-generation machine learning algorithms. This paper presents an optimization strategy that includes quantization, fine-tuning, and inference techniques to improve the effectiveness of diffusion models. It also discusses identifying and reducing particular training bottlenecks. By means of thorough testing and assessment, the suggested enhancements are methodically compared with baseline models. While preserving similar image quality, the optimization methods show improved computational efficiency. This advancement makes it easier to create diffusion models that are more precise and scalable, enabling wider uses in computer vision and related domains.

Keywords: diffusion models; generative ai; fine-tuning; text-to-image models

I. Introduction

Significant progress has been made in the field of generative artificial intelligence in recent years, especially in the area of image synthesis. This field demonstrates its potential in a wide range of applications by allowing the creation of visuals from a variety of input types. However, these models require a significant amount of computational power to generate new data. Training transformer-based techniques that can generate high-resolution images with realistic features, natural scenes, and a variety of textures is still a difficult and time-consuming process that often takes days or even months, even with the most advanced hardware [1]. These models also require significant guidance & few shot techniques in order to provide state-of-the-art results [28].

Based on denoising score models, diffusion models are constructed by combining several denoising autoencoder combinations [2,21]. These models are generative frameworks that are based on likelihood. They iteratively add noise to input data during inference, then denoise and refine the samples to produce new data. These models outperform conventional techniques when they are carefully scaled in conjunction with computational resources [3].

Diffusion models have shown promise in producing audio and music in addition to text and graphic applications [4]. Nevertheless, despite their remarkable potential, diffusion models' computational requirements are especially noticeable when working with big datasets and high-resolution imagery, which restricts their applicability in real-world situations. This study presents methods to maximize diffusion model training, increasing the models' effectiveness.

- Model architecture selection
- Hyper-parameter impact while training
- Parallelization and Hardware Acceleration with GPUs
- Model Quantization and Compression

 doi:10.20944/preprints202502.0138.v1

II. Background and Related Work

The origins of diffusion probabilistic models (DPM) can be traced back to the work of J. Sohl-Dickstein et al. in 2015 [5]. A forward process and a reverse process are the two separate processes that these models use to function. The forward process adds noise to the data gradually until it is totally distorted. By maximizing the evidence lower bound (ELBO), the reverse process, which is parameterized to reconstruct the data, learns to reverse this destruction and aligns with the forward process's joint distribution [6,31]. DPMs employ a Markov diffusion chain to map data to a sample distribution. Denoising Diffusion Probabilistic Models (DDPM) were created to improve and optimize DPMs by building upon this framework [7]. Furthermore, by adding random Gaussian noise to the data of different intensities and training the model with score functions obtained from noisy data distributions, score-based generative models (SGM) improve DDPM [8,22]. Interestingly, J. Ho and colleagues showed that SGM and DDPM are equivalent when the optimization targets are the same [7]. Pascal Vincent also proposed a novel method to denoise the autoencoders without the use of a second derivative in his work [23]. Subsequent advancements in DDPM and SGM facilitated their application to music generation. To produce high-fidelity audio, for example, N. Chen et al. [9] investigated diffusion models. Starting with Gaussian noise signals, the method gradually improves them according to certain sampling circumstances. By bridging the gap between autoregressive and non-autoregressive models, this approach achieves fidelity and high-quality audio synthesis. A similar feat was achieved by Noise2Music models, which generates music from the text [24].

In 2021, OpenAI unveiled the autoregressive model DALL-E [10], which used large datasets to produce text-based descriptions from images. However, this method has drawbacks like high processing costs and sequential error accumulation. Models such as CogView and NUWA were among the later developments [10]. To improve generative modeling capabilities, CogView, for example, uses a tokenizer driven by a 4-billion-parameter transformer. One efficient strategy for improving generative models involves projecting images into low-dimensional spaces, training models on this representation, and scaling outputs back up. Stable Diffusion [1] , VQ-Diffusion [2], and DALL-E 2 [12] are notable models that employ this technique. Stable Diffusion, developed by R. Romach et al., incorporates robust pre-trained autoencoders to improve visual fidelity. When it is released, state-of-the-art performance is made possible by the addition of cross-attention layers, which further enhances its creative capability. Developed by S. Gu et al., VQ-Diffusion uses a modified DDPM in conjunction with vector-quantized variational autoencoders (VQ-VAE) to produce latent representations. The model may effectively handle complicated scenarios by reducing unidirectional bias and using a mask-and-replace strategy inside the latent space. Its high efficiency is also a result of reparameterization, which makes it a formidable competitor in text-to-image generative modeling. Rate Adaptive VQ-VAE have also been proposed to address the issue of scalability and efficiency of the autoencoders and VAEs [25].

A. Ramesh and colleagues presented a two-phase method for capturing both visual semantics and style [13]. An encoder in the first stage creates image embeddings based on CLIP, and a decoder in the second stage creates an image conditioned on these embeddings. This technique has been successful in creating picture variations that preserve the original image's style and semantic content. Both diffusion and autoregressive models were used in their research; the publicly available version of DALL-E 2 was developed using this methodology. Another model, Imagen, developed by C. Saharia et al. [13], placed a greater emphasis on photorealism. Imagen first uses large language models (LLMs) to comprehend the textual context before producing visuals. It's interesting that the model did well on the COCO dataset even though it wasn't used for training [33]. In order to evaluate performance more thoroughly, the authors also suggested DrawBench, a benchmark for analyzing image creation models over a range of parameters. Q-Diffusion, which was first presented by [14], offers ways to optimize problems like sluggish inference and large memory usage. Through the use of split shortcut optimization approaches and timestamp-aware calibration, Q-Diffusion enables models to be quantized to 4-bit accuracy while preserving robust performance and excellent outcomes.

 doi:10.20944/preprints202502.0138.v1

a. Lora Based Fine-Tuning

The model is first trained using LoRA adapters. The reduces the need of extensive retraining. LoRA works by injecting trainable rank decomposition matrices into each layer of the Transformer architecture [15,32]. LoRA also freezes other pre-trained layers and weights. This significantly reduces the number of trainable parameters, which affects the resource consumption directly [29,30].

b. Model Quantization

Although LoRA allows easy fine-tuning of models, the model size after fine-tuning can still be huge. This is where n-bit quantization helps. Quantization is an essential technique for reducing the model's overall size, memory usage and improving inference speed. Typical models are trained on 32-bit or 16-bit floating-point numbers. For models with billions on parameters, this can take up huge space and resource. Hence, dynamic 8-bit quantization was applied, reducing the weights stored in model to 8-bit integer during inference. Dynamic quantization applies the 8-bit conversion only during the inference phase, leaving the training phase unaffected [18,26]. This selective quantization preserves model accuracy while lowering latency, making it suitable for real-time applications [27].

c. Image Upscaling

In order to prepare the LoRA trained model inferencing faster, the Image Resolution was kept low. However, in order to provide signifcant quality image as final output, an Image Upscaler was used. Traditionally, GANs were used for scaling images to super-resolution [16,20]. However, another catgeory of upscalers, which are dedicated to image-diffusion models, help to enhance the image by understanding the context and then adding finer details to image. StableDiffusionUpscale Pipeline was used from HuggingFace. This module works together with stable diffusion models and is built for the purpose of upscaling images by a factor of 4, without loss in quality [17].

III. Dataset Used

A suitable dataset must also be used to train the pipeline and get efficient results at the end. Hence, the vanilla Stable Diffusion model was tested on various prompts first and them the domain of "realistic-face images" was selected as a weak area for the diffusion model. The dataset is publicly available on HuggingFace [19]. The plus point of this specific dataset is the generalized caption for images, such as specifying gender, face qualities, expressions, and items/figures in the image. A sample of images and prompts from the dataset has been shown in Figure 1.

Figure 1. Sample images from the dataset.

IV. Methodology

By combining image upscaling for improved output quality, quantization for optimal resource utilization, and fine-tuning for domain-specific knowledge adaptation, the suggested methodology combines the benefits and minimizes the drawbacks of multiple approaches. This framework preserves the essential features of previously trained diffusion models while functioning efficiently in resource-constrained settings.

Stable-Diffusion-v1-4 from HuggingFace served as the foundational model for the experiment. Low-Rank Adaptation (LoRA), which incorporates rank-decomposition matrices into the Transformer architecture, was used for fine-tuning. By freezing the majority of pre-trained layers, this method lowers computational costs without sacrificing generative performance by reducing the

 doi:10.20944/preprints202502.0138.v1

number of trainable parameters. For fine-tuning, a domain-specific dataset devoted to realistic face generation was used, selected for its comprehensive captions that describe facial features, expressions, and situations. This made it easier to adapt the model to new domains without a lot of retraining. The training and selection of hyper-parameters was done carefully to preserve the high-quality generative capabilities of the base model. Multiple iterations of the following hyper-parameters were adjusted to get the best set of results:

- Image Resolution and Augmentation
- Training Steps, Epochs, and batch size
- Learning Rate and Max Gradient Normalization
- Warmup steps and Learning Rate Scheduler

During inference, dynamic 8-bit quantization was used to convert floating-point weights to 8-bit integers in order to further improve computational efficiency. This preserved model accuracy while drastically lowering memory consumption and inference latency. In order to reduce computational demands, the model also produced low-resolution outputs during inference, which were then improved using the Stable Diffusion Upscaler. This super-resolution pipeline minimized quality degradation by using contextual content understanding to reconstruct high-fidelity images. The proposed pipeline has been represented in Figure 2.

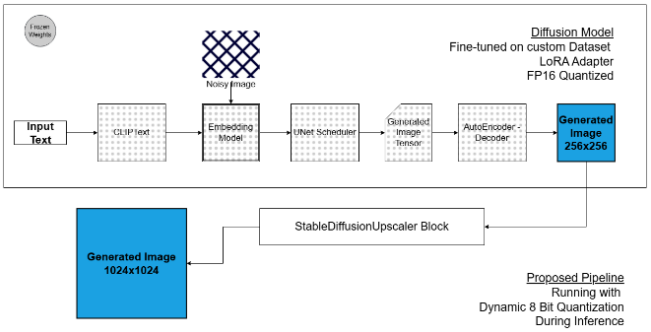

Figure 2. Proposed Methodology.

All things considered, the multi-stage optimization pipeline successfully strikes a balance between output quality, scalability, and computational efficiency, improving the applicability of diffusion models for real-world scenarios. After generating the pipeline, it was tested on multiple prompts and compared against Vanilla models, results of which have been compiled in Section 4.

V. Results

The proposed method was evaluated on a combination of qualitative and quantitative metrics. The results demonstrate the effectiveness of the proposed pipeline, in terms of model efficiency, inference time, and image quality. The results highlight significant improvements in computational efficiency, memory utilization, and output quality.

- **Generated Image Samples**: A range of prompts were used to create representative images in order to assess the effectiveness of the optimized diffusion model. These examples show how the model can generate excellent, aesthetically pleasing results in a variety of challenging situations. Due to upscaling, the image samples were output in a resolution of 1024x1024, with preserved image quality. Figure 3 shows some sample outputs generated from the pipeline.

 doi:10.20944/preprints202502.0138.v1

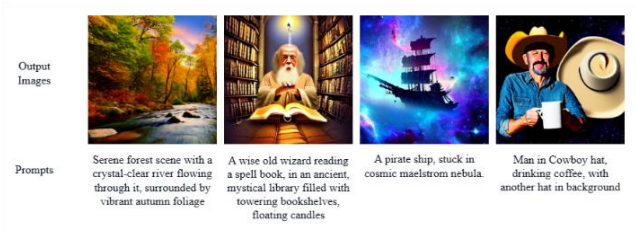

Figure 3. Sample output Images.

- **Memory Utilization and Inference Speed:** During the inference of model, the team kept a track of resource utilization. It was noted that dynamic quantization significantly reduced memory footprint and accelerated inference. Reduction from fp16 to int8 was the correct decision taken while building the pipeline, in order to create an efficient pipeline.
- Memory Consumption: The complete fine-tuned fp16 model when loaded on Colab T4 took around 6.1 GB of constant memory space for inference. While the int8 quantized model took nearly half memory, at 3.5 GB to inference the images.
- Inference Time: The inference time was measured between 3 models - default, fine-tuned fp16, and int8 quantized model. Alongside this, the models were also trained on a combination of different inference steps, with values as 50, 25 and 10, with a constant guidance scale of 20. The results have been compiled in Figure 4.

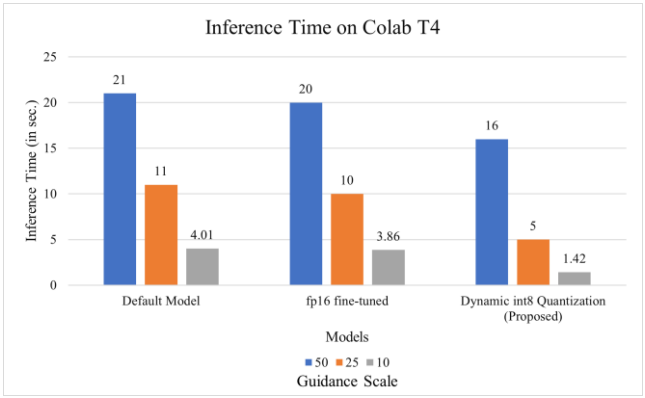

Figure 4. Inference Time of different Models with different Guidance steps.

The prompts used for different guidance steps are:

1. Realistic Crocodile wearing a sweater
2. A alpaca made of colorful building blocks, cyberpunk.
3. Realistic Astronaut, Behance HD, riding horse on, with cosmic malestom in back.

- **Training Efficiency**: The LoRA-based fine-tuning pipeline achieved efficient training. The training loss graph shows that the model reached a stable training loss at higher steps. The training loss was monitored using 'Weights & Baises' and is shown in Figure 5.

 doi:10.20944/preprints202502.0138.v1

Figure 5. Training Loss while Fine-tuning.

VI. Conclusion and Future Scope

A thorough optimization methodology for diffusion models has been effectively described in this work, addressing important issues such high processing requirements, memory inefficiencies, and lengthy inference times. The suggested method proved its potential to improve computational efficiency and model output quality by combining dynamic quantization, Low-Rank Adaptation (LoRA), and sophisticated image upscaling approaches.

The framework's ability to preserve the generative fidelity of the base models while drastically cutting down on latency and memory utilization was validated by experimental findings. This efficiency-performance balancing demonstrates how the optimized pipeline can be applied to real-world situations where computational resources are frequently limited. The results of this investigation help to push the boundaries of generative AI, especially in the area of image synthesis.

Building on this work, future research could go in a number of exciting areas. Expanding the suggested optimization techniques to more general fields, such video synthesis or 3D content creation, is one possible avenue to confirm the method's adaptability and scalability. Furthermore, investigating hybrid designs that integrate diffusion models with other generating frameworks, such transformers or GANs, may take advantage of the complementing advantages of these methods to produce better outcomes. Additional developments in quantization methods, including hardware-aware optimizations or mixed precision quantization, may make deployment on a variety of platforms—including edge devices—even more effective.

Note: Gurneet Singh has developed the manuscript and Pranjal Kumar has supervised the work.

References

1. Ramesh et al., “Zero-Shot Text-to-Image Generation,” International Conference on Machine Learning, pp. 8821–8831, Jul. 2021.
2. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models” 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10674–10685, Jun. 2022, doi: 10.1109/cvpr52688.2022.01042.
3. Vaswani et al., “Attention is All you Need,” arXiv (Cornell University), vol. 30, pp. 5998–6008, Jun. 2017.
4. P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: a generative model for music,” arXiv (Cornell University), Jan. 2020, doi: 10.48550/arxiv.2005.00341.
5. J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep Unsupervised Learning using Nonequilibrium Thermodynamics,” International Conference on Machine Learning, pp. 2256–2265, Jul. 2015.
6. J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, “Cascaded diffusion models for high fidelity image generation,” May 30, 2021. https://arxiv.org/abs/2106.15282
7. J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” arXiv.org, Jun. 19, 2020. https://arxiv.org/abs/2006.11239
8. Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” arXiv (Cornell University), vol. 32, pp. 11895–11907, Sep. 2019.

9. N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "WaveGrad: Estimating gradients for waveform Generation," arXiv.org, Sep. 02, 2020. https://arxiv.org/abs/2009.00713
10. M. Ding et al., "CogView: Mastering Text-to-Image Generation via Transformers," Neural Information Processing Systems, vol. 34, Dec. 2021.
11. S. Gu et al., "Vector Quantized diffusion model for Text-to-Image synthesis," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10686–10696, Jun. 2022, doi: 10.1109/cvpr52688.2022.01043.
12. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents," arXiv.org, Apr. 13, 2022. https://arxiv.org/abs/2204.06125
13. Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding," arXiv (Cornell University), Jan. 2022, doi: 10.48550/arxiv.2205.11487.
14. X. Li et al., "Q-Diffusion: Quantizing diffusion models," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2023, doi: 10.1109/iccv51070.2023.01608.
15. E. J. Hu et al., "LORA: Low-Rank adaptation of Large Language Models," arXiv.org, Jun. 17, 2021. https://arxiv.org/abs/2106.09685
16. T. Dettmers, M. Lewis, S. Shleifer, and L. Zettlemoyer, "8-bit optimizers via block-wise quantization," arXiv (Cornell University), Jan. 2021, doi: 10.48550/arxiv.2110.02861.
17. Ledig et al., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 105–114, Jul. 2017, doi: 10.1109/cvpr.2017.19.
18. Huggingface Super-resolution https://huggingface.co /docs/diffusers/ /pipelines/stable_diffusion/upscale
19. Huggingface "roborovski/celeba-faces-captioned · Datasets at Hugging Face." https://huggingface.co/roborovski/celeba-faces-captioned
20. T. Karras, S. Laine, and T. Aila, "A Style-Based generator architecture for generative adversarial networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 12, pp. 4217–4228, Jan. 2020, doi: 10.1109/tpami.2020.2970919.
21. D. P. Kingma and M. Welling, "Auto-Encoding variational Bayes," arXiv (Cornell University), Jan. 2013, doi: 10.48550/arxiv.1312.6114.
22. Y. Song and S. Ermon, "Improved techniques for training Score-Based generative models," arXiv (Cornell University), Jan. 2020, doi: 10.48550/arxiv.2006.09011.
23. P. Vincent, "A Connection Between Score Matching and Denoising Autoencoders," in Neural Computation, vol. 23, no. 7, pp. 1661-1674, July 2011, doi: 10.1162/NECO_a_00142.
24. Q. Huang et al., "Noise2Music: Text-conditioned Music Generation with Diffusion Models," arXiv (Cornell University), Jan. 2023, doi: 10.48550/arxiv.2302.03917.
25. J. Seo and J. Kang, "RAQ-VAE: Rate-Adaptive Vector-Quantized Variational Autoencoder," arXiv (Cornell University), May 2024, doi: 10.48550/arxiv.2405.14222.
26. Y. He, L. Liu, J. Liu, W. Wu, H. Zhou, and B. Zhuang, "PTQD: Accurate Post-Training Quantization for Diffusion Models," arXiv (Cornell University), Jan. 2023, doi: 10.48550/arxiv.2305.10657.
27. S. Jung et al., "Learning to quantize deep networks by optimizing quantization intervals with task loss," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019, doi: 10.1109/cvpr.2019.00448.
28. T. B. Brown et al., "Language Models are Few-Shot Learners," Neural Information Processing Systems, vol. 33, pp. 1877–1901, May 2020.
29. S. Hayou, N. Ghosh, and B. Yu, "LORA+: efficient low rank adaptation of large models," arXiv (Cornell University), Feb. 2024, doi: 10.48550/arxiv.2402.12354.
30. Y. Mao et al., "A survey on LoRA of large language models," Frontiers of Computer Science, vol. 19, no. 7, Dec. 2024, doi: 10.1007/s11704-024-40663-9.
31. Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," arXiv (Cornell University), Jan. 2019, doi: 10.48550/arxiv.1907.05600.
32. R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating Long Sequences with Sparse Transformers," arXiv (Cornell University), Jan. 2019, doi: 10.48550/arxiv.1904.10509.

 doi:10.20944/preprints202502.0138.v1

33. T.-Y. Lin et al., "Microsoft COCO: Common Objects in context," arXiv (Cornell University), Jan. 2014, doi: 10.48550/arxiv.1405.0312.