UiA Fakultet for teknologi og realfag

IKT526 - Emerging AI technologies

Assignment 1: Fine-Tuning SD

Fall 2025

Philip André Haugen

October 12, 2025

1 Introduction

Stable Diffusion (SD) is a text-to-image generative model based on the Latent Diffusion Model (LDM) framework introduced by Rombach et al. [1]. LDMs apply diffusion models in the latent space of pretrained autoencoders. Cross-attention layers are used to enable conditioning on text or other modalities. SD v1.5 [2] is a checkpoint trained on the LAION-5B subset, mapping text embeddings from CLIP to a latent image space for high quality generation. It is a standard pretrained backbone for fine-tuning due to its stable training behavior and open source weights.

Training or adapting diffusion models such as SD through full fine-tuning is computationally intensive, as it requires updating all model parameters. Hu et al. [3] proposed Low-Rank Adaptation (LoRA) to address this issue by freezing pretrained weights and introducing trainable low-rank matrices within linear layers. This approach reduces the number of trainable parameters by several orders of magnitude while achieving performance comparable to full fine-tuning.

This assignment applies LoRA to SD v1.5 using the cartoon-blip-captions dataset. The objective is to analyse how the LoRA rank parameter affects performance, training time, memory usage, and final image quality. Two LoRA configurations with different ranks were trained and evaluated using CLIP similarity to measure alignment between generated images and their text prompts. The results provide insight into how LoRA capacity influences efficiency and stylistic adaptation when transferring Stable Diffusion to a cartoon domain.

2 Method

The implementation builds on the Exercise 3 notebook and open-source LoRA guide from GitHub [4]. The dataset contains cartoon images paired with BLIP-generated captions. A subset of 1,500 samples was used for training to balance generalization and training time/memory. Images were resized to $512 \cdot 512$ pixels, normalized to $[-1, 1]$, and paired with tokenized captions using the CLIP tokenizer.

SD v1.5 is the base model, with its U-Net, Variational Autoencoder (VAE), and text encoder loaded from pretrained weights. Only the U-Net was modified. LoRA adapters were injected into attention and feed-forward layers (`to_q`, `to_k`, `to_v`, `to_out.0`, `ff.net.0`, `ff.net.2`). Each adapter added two projection matrices, scaled by $\alpha \div r$. The A matrices were initialized with Kaiming uniform weights, and B matrices with zeros.

Two LoRA configurations were trained with rank 16 and rank 64, with alpha $\alpha = 2 \cdot \text{rank}$ as it was recommended by the lecturer. All other model components were frozen. Training used AdamW with a learning rate of 0.0005 and betas (0.9, 0.999) [4]. Batch size was 4, with mixed precision enabled via `torch.amp` [4]. The loss function was mean squared error (MSE) between predicted and target noise in the latent space. Each model trained for 10 epochs.

After training, LoRA weights were merged into the SD inference pipeline. Evaluation used 50 prompts divided into two categories: normal (concrete descriptions) and regular abstract (conceptual phrases). Images were generated with 30 denoising steps and a guidance scale of 7.5. CLIP similarity between prompts and images was used to measure alignment.

3 Results

Training performance for both configurations is shown in Figure 1. Rank 16 reached a lower final loss (0.1192) than rank 64 (0.1266) while requiring nearly identical runtime (4383 s vs. 4541 s) and memory (16.2 GB vs. 16.6 GB). Model size increased roughly linearly with rank, from 16 MB to 64 MB.

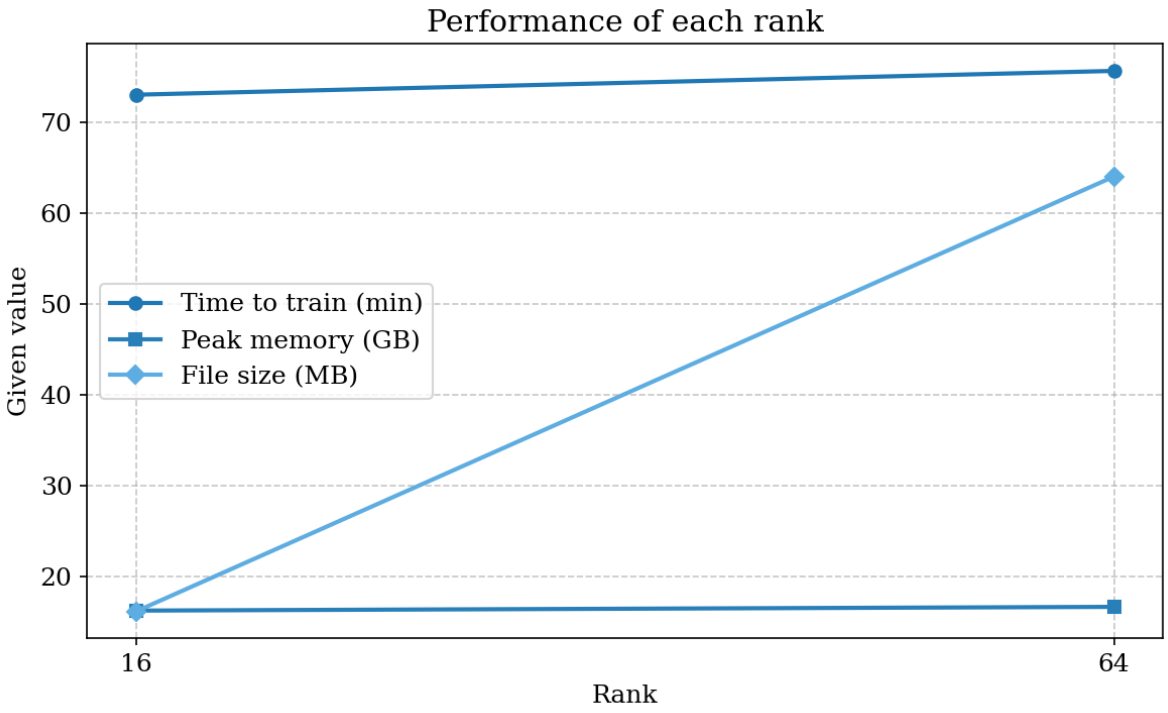

Figure 1: Runtime, peak GPU memory, and LoRA file size for rank 16 and rank 64.

The smoothed loss curves in Figure 2 show stable and consistent convergence for both ranks.

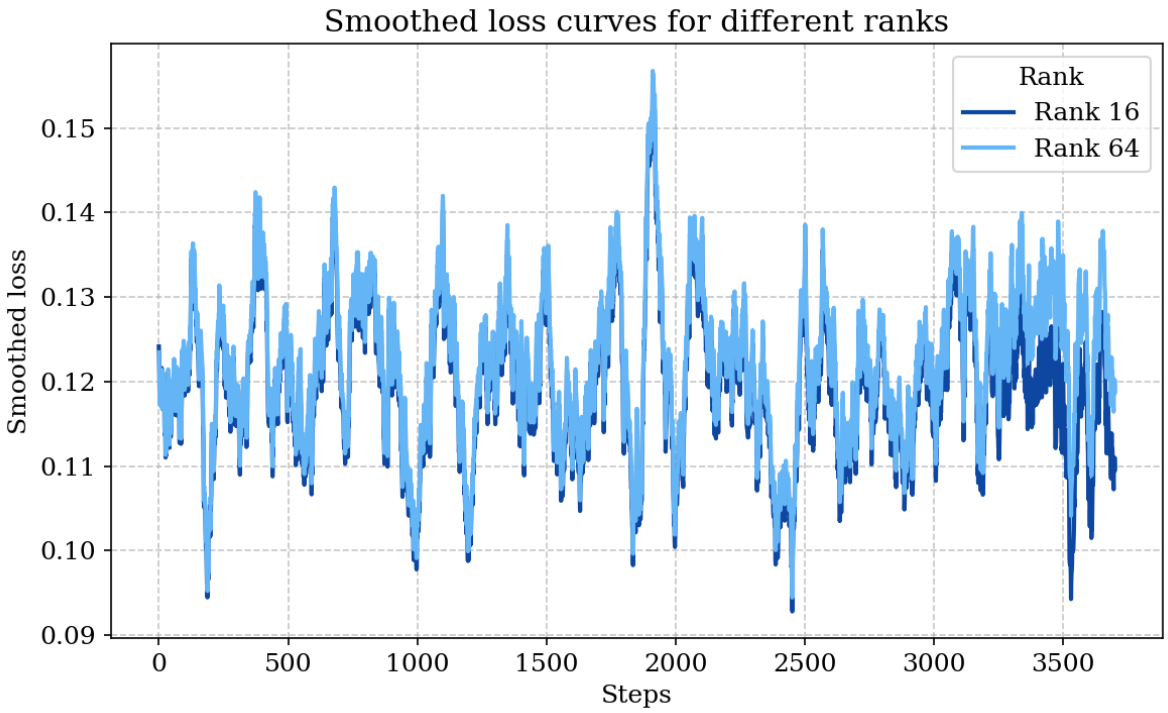

Figure 2: Smoothed training loss curves for rank 16 and rank 64 across ten epochs.

Quantitative evaluation using CLIP similarity is shown in Table 1. Rank 16 achieved higher mean CLIP

scores across both categories, with 29.26 for normal prompts and 23.38 for regular abstract ones. Rank 64 scored lower at 19.79 and 22.11, respectively. Standard deviations were moderate, ranging from 1.5 to 3.5.

Table 1: Mean and standard deviation of CLIP scores by LoRA rank and prompt category.

Configuration	**Mean CLIP**	**Std. Dev.**
Rank 16 (normal) – $\alpha = 32$	29.2631	3.4643
Rank 64 (normal) – $\alpha = 128$	19.7907	2.8496
Rank 16 (abstract) – $\alpha = 32$	23.3837	2.4608
Rank 64 (abstract) – $\alpha = 128$	22.1148	1.5294

The example in Figure 3 compares outputs from the baseline model and the LoRA rank 16 version for a normal prompt. The LoRA model preserves overall structure while applying a cartoon-like simplification, reducing visual noise and softening color gradients.

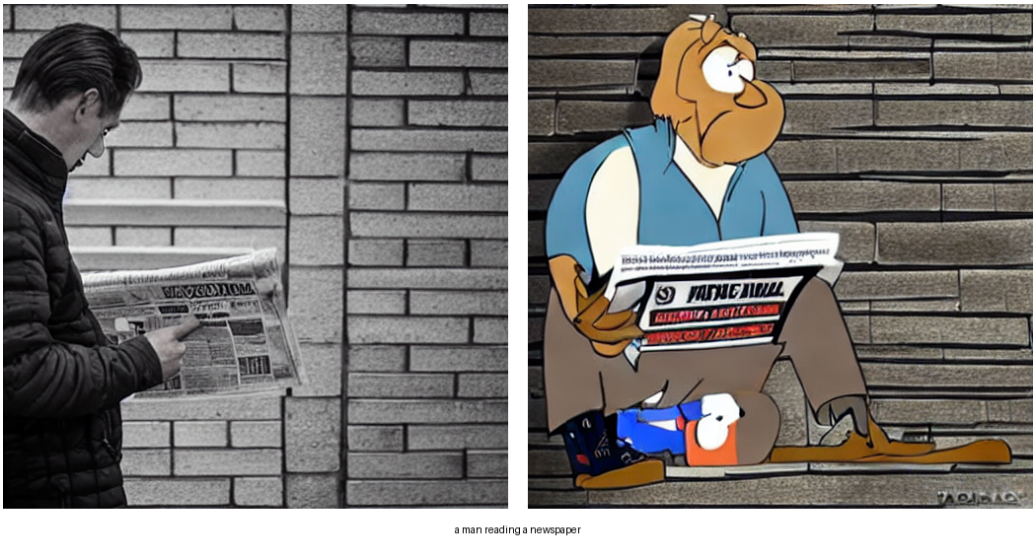

Figure 3: SD v1.5 vs. LoRA rank 16 outputs on the normal prompt “a man reading a newspaper”.

In contrast, the example in Figure 4 shows the rank 64 model producing distorted or incoherent shapes.

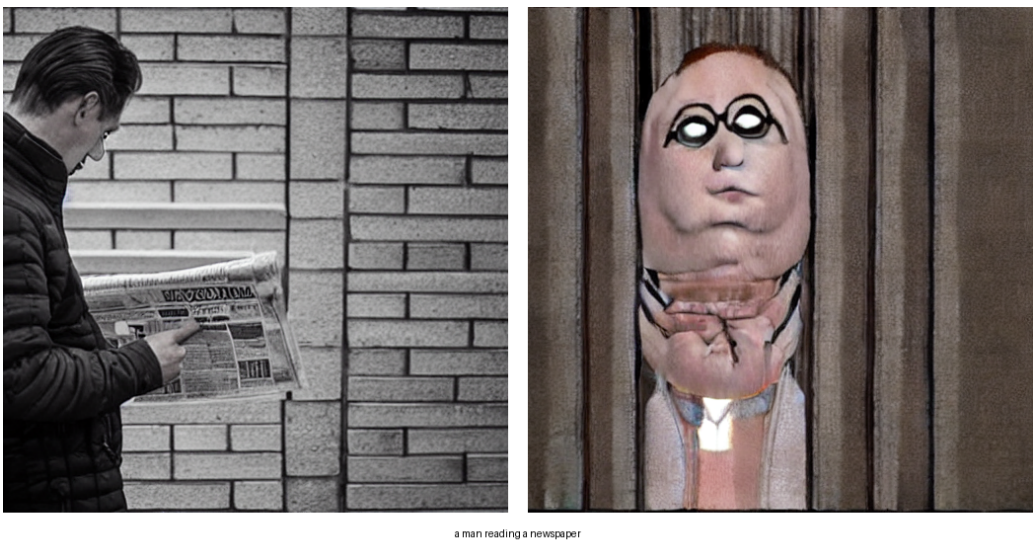

Figure 4: SD v1.5 vs. LoRA rank 64 outputs on the normal prompt “a man reading a newspaper”.

For abstract prompts, the example Figure 5 shows that the rank 16 LoRA often produces figurative human-focused interpretations, while the baseline model outputs literal representations.

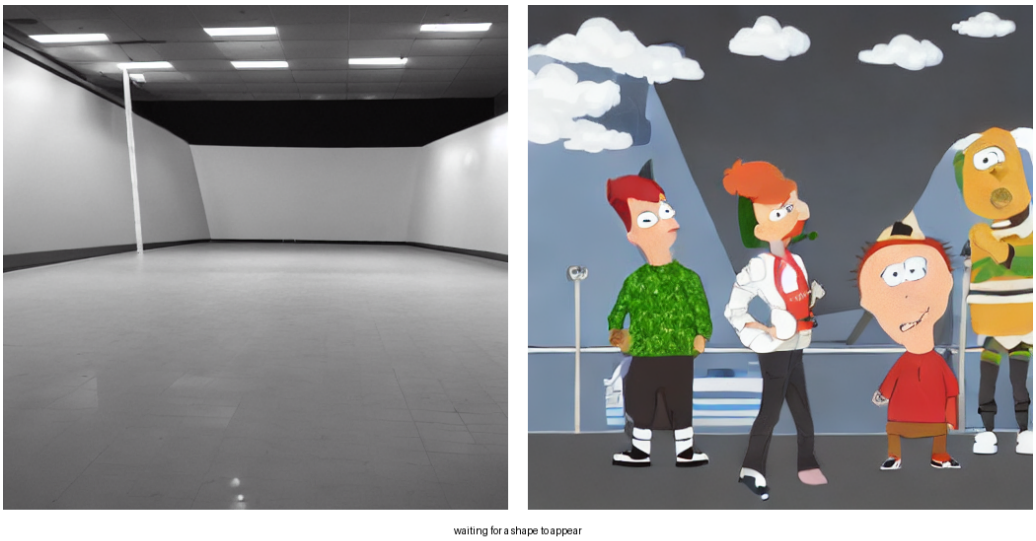

Figure 5: SD v1.5 vs. LoRA rank 16 outputs on the abstract prompt “waiting for a shape to appear”.

The rank 64 LoRA, with an example shown in Figure 6, maintains some of this semantic focus but introduces disjointed or uncanny imagery, with fragmented faces and inconsistent geometry.

Figure 6: SD v1.5 vs. LoRA rank 64 outputs on the abstract prompt “waiting for a shape to appear”.

4 Discussion

The results show a clear difference in how LoRA rank affects training stability, text-image alignment, and visual quality when adapting SD v1.5 to a cartoon style. Both the rank 16 and rank 64 configurations trained successfully without instability, but their behavior during evaluation reveals that higher rank does not necessarily produce more coherent outputs. This observation aligns with recent findings that increasing LoRA rank can lead to higher memorization ratios without corresponding gains in generative quality. For example, in controlled experiments on the Imagenette dataset by Gu et al. [5], higher-rank

LoRA models showed up to an 8.77% memorization ratio under class-conditioned prompts, compared to 0.00% for lower-rank LoRA, while full fine-tuning reached 46.89% under the same conditions. These results indicate that higher LoRA capacity may capture specific patterns in the dataset more aggressively, increasing overfitting even when training remains stable.

The training losses remained stable across all epochs, which indicates that the learning rate, optimizer, and data pipeline were well-tuned. However, the model with rank 64 ended with a higher loss than rank 16 even though it contained more trainable parameters. This suggests that simply increasing LoRA capacity does not improve generalization under fixed conditions. The small difference in runtime and memory use also shows that rank scaling mainly affects parameter storage rather than how efficiently the model learns. The increase in file size between the two models follows directly from the number of parameters growing with rank.

The CLIP-based evaluation supports this pattern. The rank 16 model achieved higher similarity scores for both normal and abstract prompts, indicating stronger text–image alignment, while the rank 64 model performed worse across categories. Similar trends have been reported in a study by Soboleva et al. [6] where increasing rank led to higher overfitting, particularly under longer noise schedules, without measurable gains in alignment quality. In their experiments, LoRA at rank 64 achieved comparable image fidelity but lower CLIP similarity ≈ 0.23 than regularized variants ≈ 0.26, suggesting that excessive rank can distort pretrained attention representations.

The qualitative results reinforce the numerical results. The rank 16 configuration consistently produced cartoon-like versions of SD's original images while maintaining a recognizable form. It simplified details, reduced visual noise, and replaced realistic textures with flat, stylized color regions. These changes show that the LoRA learned to modify the style and color distribution of the base model without damaging its ability to represent structure. In comparison, the rank 64 model often produced images that were visually unstable. The results were distorted, with exaggerated features and a general loss of spatial coherence. This behavior suggests that a large LoRA rank can disrupt the balance between layers in the U-Net and cause the attention mechanism to misalign visual features.

The abstract prompts also showed differences between the two ranks. With rank 16, the model tended to interpret conceptual text through figurative and human-centered imagery, showing a degree of semantic grounding inherited from the original CLIP encoder. In contrast, the baseline SD model often interpreted abstract prompts literally, for example by inserting motivational written text into the image. The rank 64 model preserved some of this human focus but produced disjointed or uncanny results, including deformed faces and irregular geometry. These distortions are consistent with overfitting or excessive LoRA influence on attention activations.

Bibliography

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," *arXiv preprint arXiv:2112.10752*, 2022. DOI: 10.48550/arXiv.2112.10752.

[2] R. Rombach, A. Blattmann, P. Esser, B. Ommer, S. AI, and Runway, *Stable diffusion v1-5*, https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5, Accessed: 2025-10-09, 2022.

[3] E. J. Hu et al., "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021. DOI: 10.48550/arXiv.2106.09685.

[4] Haoming, *All-in-one stable diffusion guide*, https://github.com/Haoming02/All-in-One-Stable-Diffusion-Guide, Version corresponding to WebUI v1.10.1 and Forge v2.0.1, released February 2025. Licensed under CC-BY-SA-4.0., 2025.

[5] X. Gu, C. Du, T. Pang, C. Li, M. Lin, and Y. Wang, "On memorization in diffusion models," *arXiv preprint arXiv:2310.02664*, 2025. DOI: 10.48550/arXiv.2310.02664.

[6] V. Soboleva, A. Alanov, A. Kuznetsov, and K. Sobolev, "T-lora: Single image diffusion model customization without overfitting," *arXiv preprint arXiv:2507.05964*, 2025. DOI: 10.48550/arXiv.2507.05964.