Memory-Efficient Fine-Tuning for Quantized Diffusion Model (Supplementary Material)

Hyogon Ryu, Seohyun Lim, and Hyunjung Shim

Korea Advanced Institute of Science and Technology (KAIST)
{hyogon.ryu, seohyunlim, kateshim}@kaist.ac.kr

A Implementation detail

All experiments were reproduced by our implementation based on `Diffusers`[1] library. For quantized checkpoint, we use q-diffusion[2]'s official checkpoint. Experiments regarding to Full-precision Dreambooth[3] and Custom Diffusion[4] are conducted based on `Diffusers` official implementation without any editing. Pseudo code for multi channel-wise scale update is available in Algorithm A.1

A.1 Hyperparameter

For single-subject generation inference, we utilize a guidance scale of 7.5 and set eta to 0, with a DDIM step of 50. For multi-subject generation, we adjust the parameters to a guidance scale of 5.0, eta of 1.0, and a DDIM step of 100. This configuration is for preserving the default setting.

In the case of multi-subject generation, prior loss is employed. However, for single-subject generation, prior loss is excluded, as the quantized model often fails to fine-tune for the target subject in almost cases.

Table A.1: Learning rate. The values of full precision are same as the default setting mentioned in the original paper, as discussed in the main paper.

Method	Full prec.	4bits	8bits
Dreambooth	5e-6	3e-5	3e-6
CustomDiffusion	1e-5	1e-5	1e-5

We used a batch size of 1 for Dreambooth and 2 for Custom Diffusion. We generated the images with train iteration 400 and 800, then selected the better

[1] `https://github.com/huggingface/diffusers`
[2] `https://github.com/Xiuyu-Li/q-diffusion`
[3] `https://huggingface.co/docs/diffusers/training/dreambooth`
[4] `https://huggingface.co/docs/diffusers/training/custom_diffusion`

one. For fair comparison, except for the learning rate, all hyperparameters are set to the same values for both the full precision and quantized models. Learning rates are displayed in Table A.1. We searched for the best setting for the baseline and then applied it to TuneQDM as well. Since we didn't search for the best settings for TuneQDM, there might be a possibility of slight performance improvement through hyperparameter search.

A.2 Metric

To measure subject fidelity, we evaluated DINO-I [4] and CLIP-I [2] scores, while for prompt fidelity, we measured CLIP-T scores. The CLIP encoder used ViT-B/32, and DINO-I utilized DINOv2 ViT-S/14. DINO, being trained via self-supervised methods, is known to measure differences well compared to the CLIP image encoder when given the similar type of subject.

A.3 Training loss

To fine-tune Stable Diffusion, we utilize the same loss function as employed in DreamBooth and Custom Diffusion. The loss is defined as the weighted sum of the prior-preservation loss and the simple diffusion loss. The loss function can be expressed as follows:

$$\mathcal{L} = \mathbb{E}_{z,c,\epsilon,t}[||\hat{\epsilon}_\theta(z,c) - \epsilon||^2] + \lambda\mathcal{L}_{\text{prior}}, \tag{1}$$

$$\mathcal{L}_{\text{prior}} = \mathbb{E}_{z_{\text{pr}},c_{\text{pr}},\epsilon,t}[||\hat{\epsilon}_\theta(z_{\text{pr}},c_{\text{pr}}) - \epsilon||^2]. \tag{2}$$

Here, $\mathcal{L}$ represents the total loss, $\mathcal{L}_{\text{prior}}$ denotes the prior-preservation loss, $\hat{\epsilon}_\theta(z,c)$ and $\hat{\epsilon}_\theta(z_{\text{pr}},c_{\text{pr}})$ are the generated noise vectors corresponding to the target images and prior examples, respectively. z and c represent the target image latents and text embeddings, z_{pr} and c_{pr} represent the latent and text embeddings for the prior examples, ϵ represents the ground truth noise vector, λ is a weighting coefficient, and $t \sim \mathcal{N}(1,T)$ represents the diffusion timestep.

By optimizing the aforementioned loss function during fine-tuning, the adapted diffusion model becomes capable of generating single and multi-subject images tailored to specific user preferences or input text prompts.

B Additional results

B.1 Quantitative Results

Table A.2 and A.3 present the quantitative results for each task, evaluated using DINO-I, CLIP-I, and CLIP-T scores. While the differences in CLIP-T scores are negligible, significant differences exist between TuneQDM and the baseline in terms of DINO-I and CLIP-I scores. However, as mentioned in the main paper, measuring subject- and prompt fidelity using DINO and CLIP scores is inaccurate. Therefore, it is necessary to evaluate through qualitative results and user studies.

Table A.2: Quantitative Comparison of single-subject generation. TuneQDM* initializes the multi-channel-wise scale from $\mathcal{N}(0, 0.01)$.

Method	Bits(W)	Size	# Params	DINO-I	CLIP-I	CLIP-T
Full prec.	32	3.20GB	859M	0.431	0.746	0.316
Baseline	4	0.40GB + 1.32MB	0.33M	0.519	0.787	0.313
TuneQDM	4	0.40GB + 2.48MB	0.62M	0.551 (+6.16%)	0.802 (+1.91%)	0.306 (−2.23%)
Baseline	8	0.80GB + 1.32MB	0.33M	0.581	0.824	0.300
TuneQDM	8	0.80GB + 2.48MB	0.62M	0.584 (+0.52%)	0.830 (+0.73%)	0.298 (−0.67%)
TuneQDM*	8	0.80GB + 2.48MB	0.62M	0.578 (−0.52%)	0.816 (−0.97%)	0.307 (+2.33%)

Table A.3: Quantitative Comparison of multi-subject generation. TuneQDM* initializes the multi-channel-wise scale from $\mathcal{N}(0, 0.01)$.

Method	Bits(W)	Size	# Params	DINO-I	CLIP-I	CLIP-T
Full prec.	32	3.20GB	859M	0.345	0.706	0.304
Baseline	4	0.40GB + 1.32MB	0.33M	0.275	0.677	0.314
TuneQDM	4	0.40GB + 2.48MB	0.62M	0.276 (+0.36%)	0.675 (−0.30%)	0.317 (+0.96%)
Baseline	8	0.80GB + 1.32MB	0.33M	0.330	0.704	0.286
TuneQDM	8	0.80GB + 2.48MB	0.62M	0.329 (−0.30%)	0.708 (+0.57%)	0.295 (+3.15%)
TuneQDM*	8	0.80GB + 2.48MB	0.62M	0.329 (−0.30%)	0.705 (+0.14%)	0.293 (+2.45%)

B.2 Explanation about full precision's DINO-I, CLIP-I score

The DINO-I and CLIP-I scores are easily influenced by some components unrelated to subject fidelity. In Fig A.1., despite both the (a) and (b) images effectively reflecting the features of the subject, there are significant differences in the DINO-I and CLIP-I scores. This difference occurred because the similarity of the background and subject's pose to the reference image had an effect on the score. In the case of the full precision model, various components unrelated to the prompt (*e.g.* background or subject pose) exhibited diversity, resulting in lower scores compared to the quantized model, as illustrated in the table. Thus, evaluating whether the subject's features are well-reflected through CLIP-I and DINO-T scores is hard. Therefore, as repeatedly mentioned, it is essential to focus on qualitative results or conduct a user study to evaluate the performance accurately.

B.3 Inference speed

Our method focuses on memory efficiency through weight-only quantization. When examining its impact on inference speed, two aspects must be considered. First, quantizing weights to 4 bits reduces the cost of memory allocation on the GPU to $\frac{1}{4}$. However, the overhead of the dequantization process will slow down the operations such as matrix multiplication. Therefore, to increase inference speed through weight-only quantization, it is essential to verify if the actual

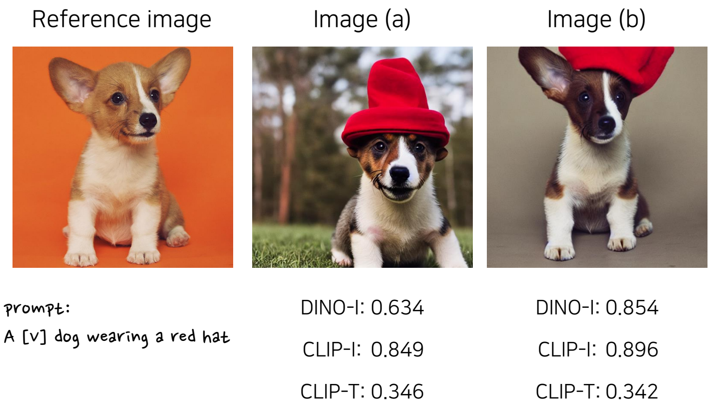


Fig. A.1: Limitation of subject-fidelity metrics

speed improvement occurs by balancing memory and computational efficiency. As noted by other studies [3], considering the increasing size of recent models and the batch sizes in practical use scenarios are often 1 or 2, weight-only quantization can indeed be expected to improve inference speed.

Since implementing custom kernels for all layers of Stable Diffusion is challenging, we created a simple benchmark to test the inference time specifically for the linear layers where TuneQDM was applied. For multiplication operations, we used the GEMM kernel and conducted experiments on an A6000 GPU. As shown in Table A.4, both the baseline and TuneQDM were faster compared to full-precision and half-precision settings. However, the additional multiplication operations made TuneQDM slightly slower than the baseline.

Table A.4: Inference speed comparison.

Method	Bits(W)	Time
full prec.	32	15.60 s
half prec.	16	8.88 s
Baseline	4	6.94 s
TuneQDM	4	6.99 s

B.4 Additional qualitative results

Fig. A.3 A.4, A.5, and A.6 respectively represent the qualitative results of single-subject generation with an 8-bit quantized model, multi-subject generation with 4-bit quantized model, and multi-subject generation with an 8-bit quantized model.

In Fig. A.4, it can be seen that TuneQDM produces images that reflect both the subject and prompt better than the baseline. In particular, in rows 1, 4, and 5, the prompt is reflected much more harmoniously than in the baseline. While generating images that reflect the content of the prompt, as seen in the rightmost example in row 1, unnatural images can also be generated, but TuneQDM generates such unnatural images less frequently. In the case of row 6, both TuneQDM and the baseline did not produce satisfactory results.

For multi-subject generation, the overall quality of the generated images is unsatisfactory. This was influenced by the poor performance of the Full Precision model. Except for the cases where the cat was used (rows 1 and 2 in Fig. A.5 and A.6), our experiments did not produce satisfactory results even when the full precision model was used for fine-tuning. We conducted experiments with the original codebase without any modifications when fine-tuning the full precision model.

Fig. A.5 shows the results of multi-subject generation using a 4-bit quantized model. TuneQDM tends to be intermediate between Full Precision and the baseline. However, significant differences occur in cases where the presence or absence of subjects changes between the full precision model and the quantized model, as shown in rows 4 and 5.

Fig. A.6 shows the results of multi-subject generation using an 8-bit quantized model. Similar to the 4-bit results, the 8-bit results show a similar trend. In particular, in rows 1 and 2, TuneQDM shows better performance than the baseline, and in the remaining rows, TuneQDM produces images closer to Full Precision than the baseline.

C User study details

We conducted a survey with a total of 86 questions to 45 participants. The survey focused on subject fidelity and prompt fidelity, comparing the baseline and TuneQDM to determine the preferred method. 56 questions were about single-subject generation, and 30 questions were about multi-subject generation. Baseline and TuneQDM were compared using the same configuration. An example of the survey is shown in Fig A.2.

D Discussions

D.1 Low-bits settings

Our approach was generally more effective at 4 bits than at 8 bits. As the low-bit setting decreased, the capacity of the quantized model decreased, and the performance improvement achievable with our approach was greater. This is because our goal is ultimately to increase the training capacity of the model by providing denoising roles and applying multi-channel-wise scale update methods.

D.2 Limitations

It has been observed that the performance of multi-subject generation is significantly lower compared to single-subject generation. This appears to be due to inherent limitations in stable diffusion. Previous research [1] has shown that stable diffusion does not effectively process images of multiple concepts. As a result, the limitations observed in multi-subject image generation persisted even in quantized models, and overcoming them is difficult even with our approach.

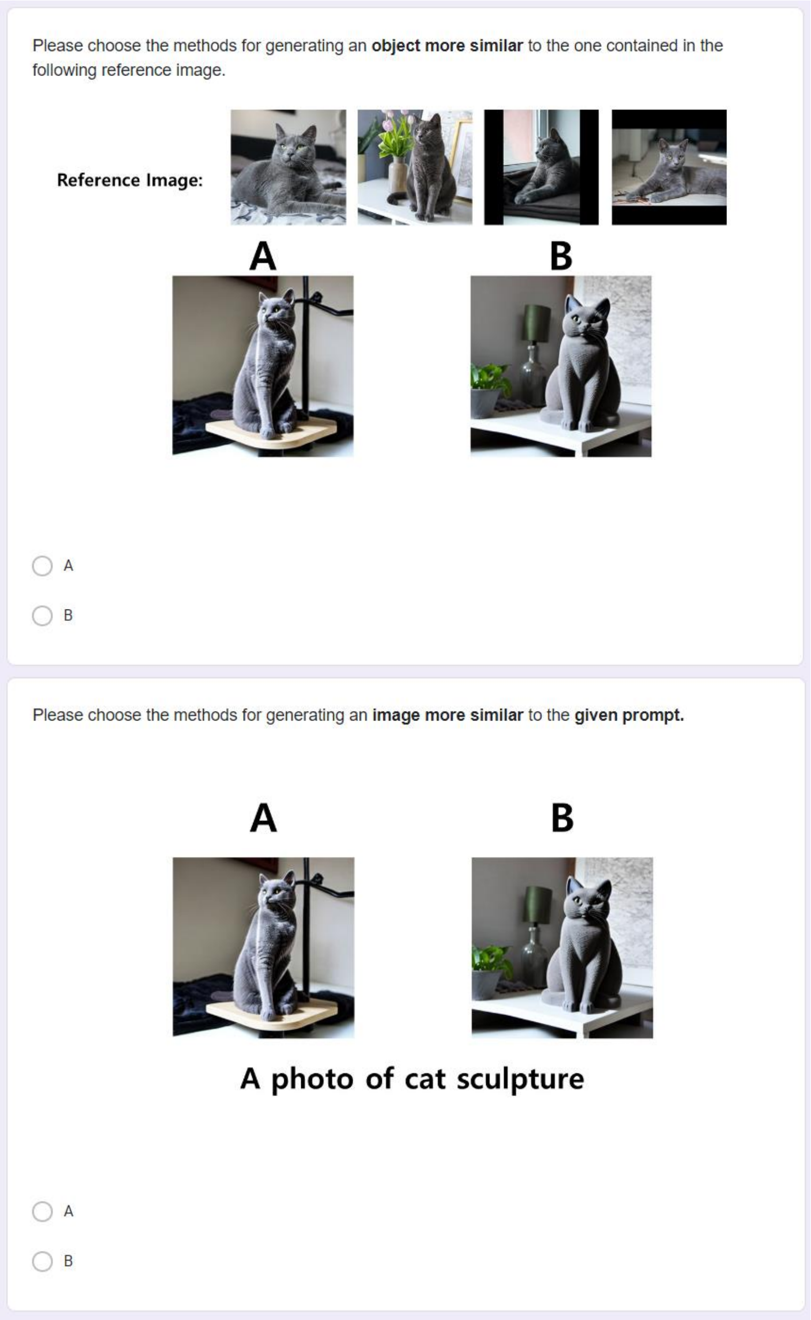
D.3 future work

The application of prior preservation loss did not yield satisfactory results. It appeared that the capacity of the quantized model was insufficient to learn new concepts while preserving the prior. There is a need to explore methods that facilitate effective tuning while maintaining the prior.

After fine-tuning the quantized model, even with the same seed, the resulting images differed from those of the full precision model. Considering other parameter-efficient fine-tuning methods that produce similar images to full fine-tuning even after fine-tuning completion using the same seed, our approach seems to fine-tune in a somewhat different manner compared to fine-tuning the full precision model. Research into methods to fine-tune such that the results of fine-tuning the full precision model and the quantized model are similar is warranted.

Algorithm A.1 Pseudo-Code for multi channel-wise scale update applied on Linear layer, PyTorch-like

```
class TQLinear(nn.Module):
    def __init__(self, QuantParam: Dict[str, Dict[str, torch.tensor]], weight: torch.tensor,
          bias):
        ’’’
        :param in_features: size of each input sample
        :param out_features: size of each output sample
        :param weight: weight tensor (quantized : dtype should be int)
        :param bias: bias tensor
        :param kwargs: other parameters

        :param QuantParam: load from quantized checkpoint
        ’’’
        super(TQLinear, self).__init__()

        self.weight = weight
        if bias != None:
            self.bias = bias
        else:
            self.register_parameter(’bias’, None)

        self.delta = QuantParam[’delta’]
        self.zero_point = QuantParam[’zero_point’]
        self.n_bits = QuantParam[’n_bits’]
        self.sym = QuantParam[’sym’]

        self.delta = nn.Parameter(self.delta)
        self.double_delta = nn.Parameter(torch.ones((1, self.weight.shape[1])))
        torch.nn.init.normal_(self.double_delta, mean=1.0, std=0.1)

    def forward(self, input, *args, **kwargs):
        return F.linear(input, (self.weight-self.zero_point) * self.delta * self.double_delta
              , self.bias)
```



Please choose the methods for generating an **object more similar** to the one contained in the following reference image.

○ A

○ B

Please choose the methods for generating an **image more similar** to the **given prompt.**

○ A

○ B

Fig. A.2: example of the survey

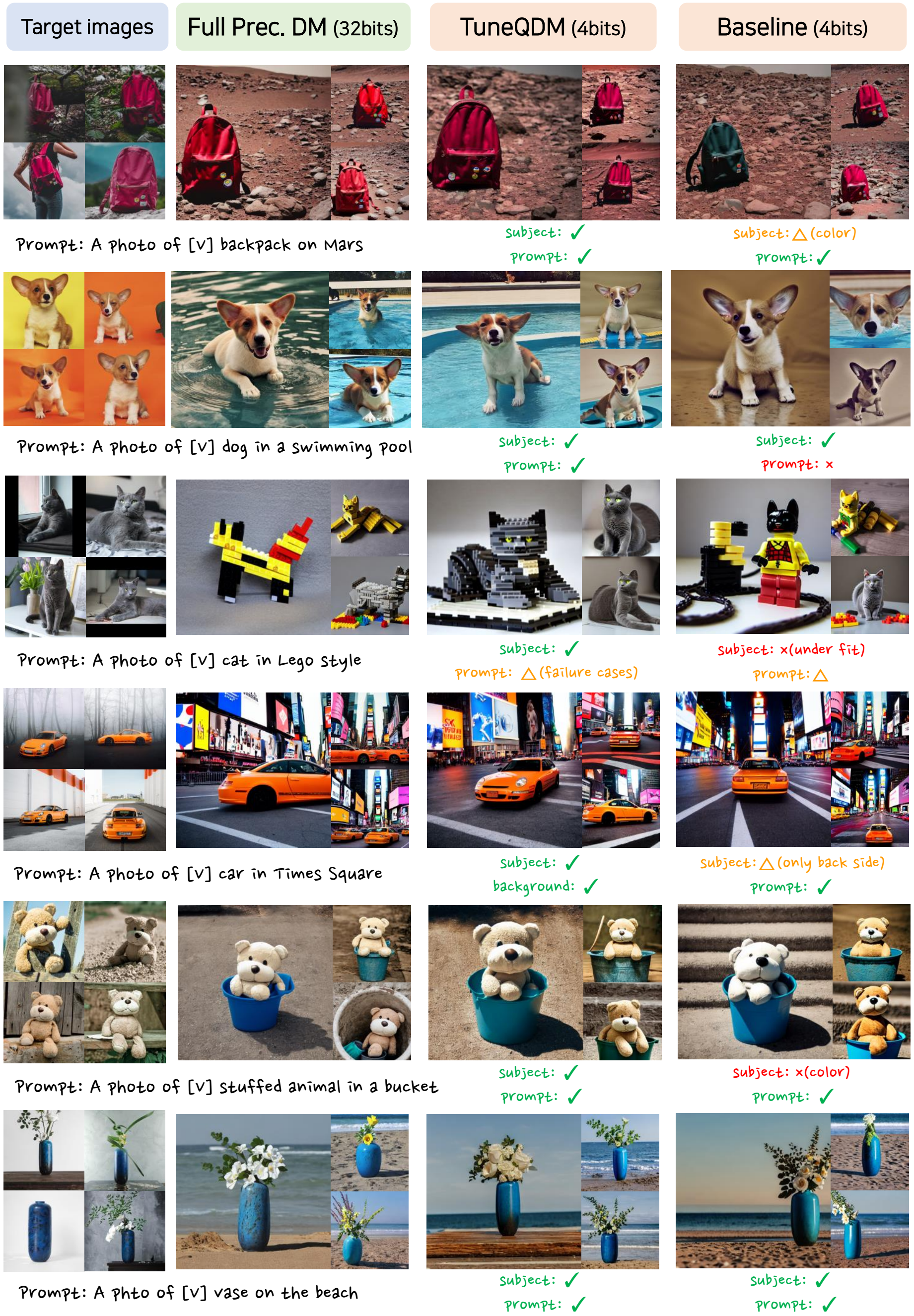


Fig. A.3: Qualitative results of single-subject generation, 4bits

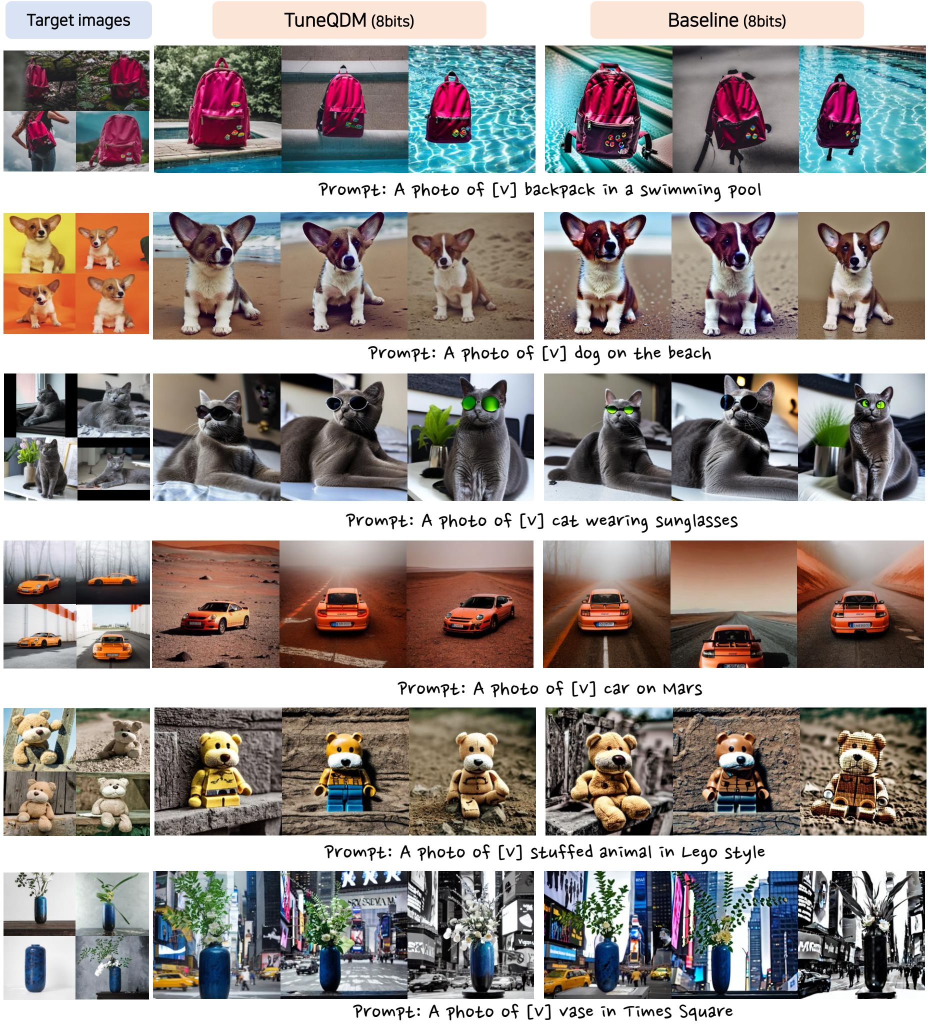

Target images
TuneQDM (8bits)
Baseline (8bits)
Prompt: A photo of [V] backpack in a swimming pool
Prompt: A photo of [V] dog on the beach
Prompt: A photo of [V] cat wearing sunglasses
Prompt: A photo of [V] car on Mars
Prompt: A photo of [V] stuffed animal in Lego style
Prompt: A photo of [V] vase in Times Square

Fig. A.4: Qualitative results of single-subject generation, 8bits

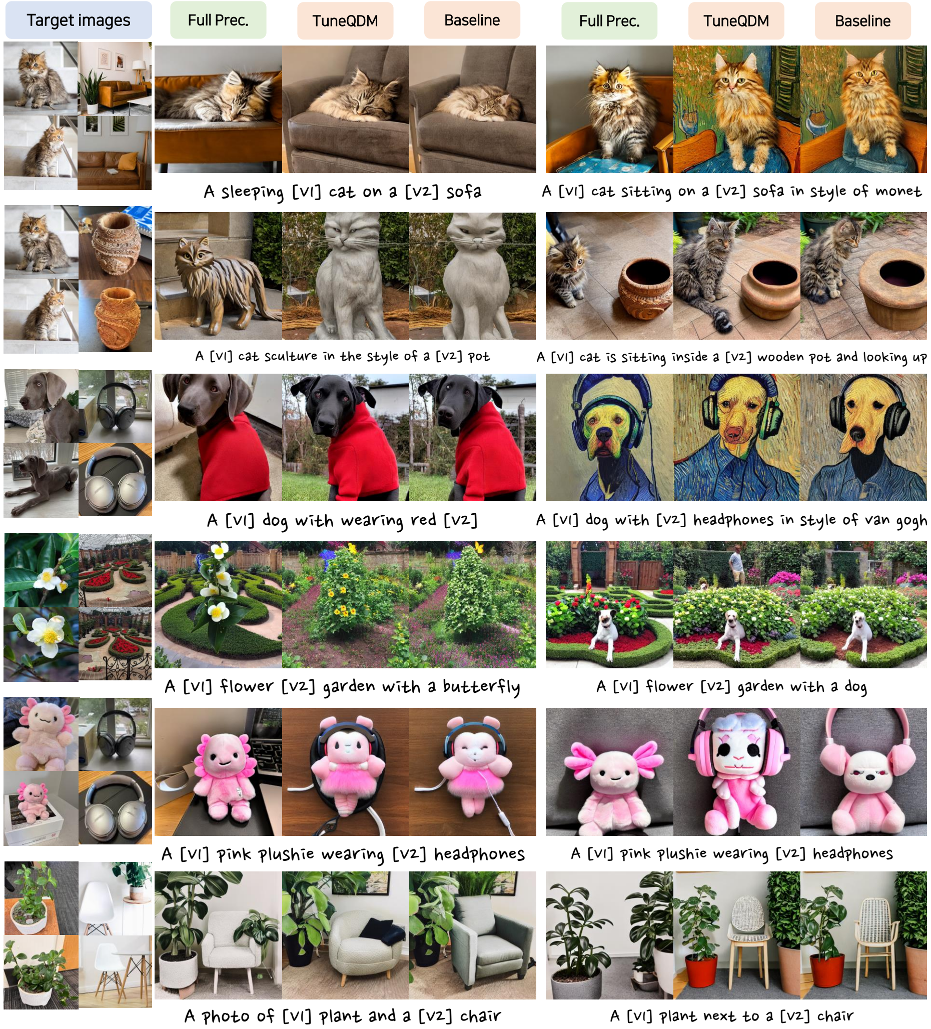

Target images
Full Prec.
TuneQDM
Baseline
Full Prec.
TuneQDM
Baseline
A sleeping [V1] cat on a [V2] sofa
A [V1] cat sitting on a [V2] sofa in style of monet
A [V1] cat sculture in the style of a [V2] pot
A [V1] cat is sitting inside a [V2] wooden pot and looking up
A [V1] dog with wearing red [V2]
A [V1] dog with [V2] headphones in style of van gogh
A [V1] flower [V2] garden with a butterfly
A [V1] flower [V2] garden with a dog
A [V1] pink plushie wearing [V2] headphones
A [V1] pink plushie wearing [V2] headphones
A photo of [V1] plant and a [V2] chair
A [V1] plant next to a [V2] chair

Fig. A.5: Qualitative results of multi-subject generation, 4bits

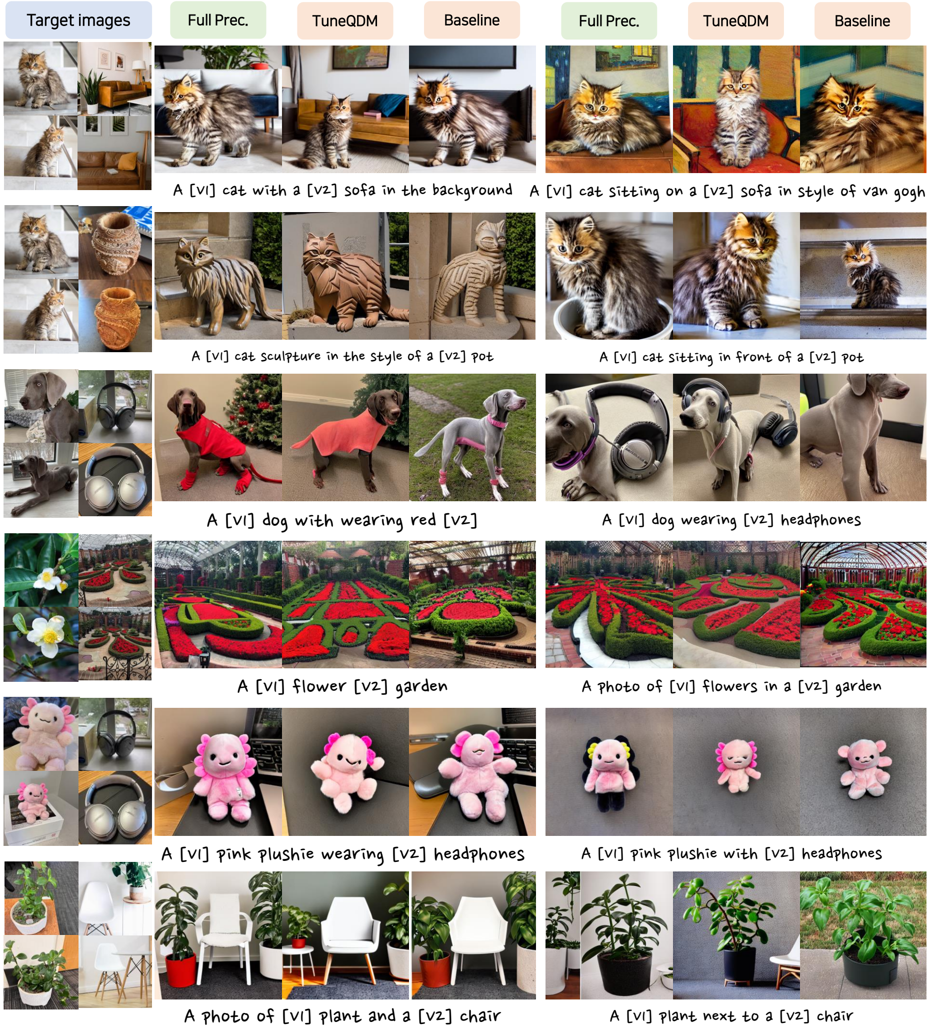

Target images
Full Prec.
TuneQDM
Baseline
Full Prec.
TuneQDM
Baseline
A [V1] cat with a [V2] sofa in the background
A [V1] cat sitting on a [V2] sofa in style of van gogh
A [V1] cat sculpture in the style of a [V2] pot
A [V1] cat sitting in front of a [V2] pot
A [V1] dog with wearing red [V2]
A [V1] dog wearing [V2] headphones
A [V1] flower [V2] garden
A photo of [V1] flowers in a [V2] garden
A [V1] pink plushie wearing [V2] headphones
A [V1] pink plushie with [V2] headphones
A photo of [V1] plant and a [V2] chair
A [V1] plant next to a [V2] chair

Fig. A.6: Qualitative results of single-subject generation, 8bits

References

1. Feng, W., He, X., Fu, T.J., Jampani, V., Akula, A., Narayana, P., Basu, S., Wang, X.E., Wang, W.Y.: Training-free structured diffusion guidance for compositional text-to-image synthesis. arXiv preprint arXiv:2212.05032 (2022)
2. Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., Choi, Y.: CLIPScore: A reference-free evaluation metric for image captioning. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 7514–7528. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.595, https://aclanthology.org/2021.emnlp-main.595
3. Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.M., Wang, W.C., Xiao, G., Dang, X., Gan, C., Han, S.: Awq: Activation-aware weight quantization for on-device llm compression and acceleration. Proceedings of Machine Learning and Systems **6**, 87–100 (2024)
4. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023)