Feed-Forward Guidance for Text-to-Image Diffusion Models

Swiss Federal Institute of Technology in Lausanne

Image and Visual Representation Lab

Yoann Lafore

Bachelor Semester Project

Supervisor:
Martin Nicolas Everaert

Autumn Semester 2023

Abstract

Guidance mechanisms such as Classifier Free Guidance [Ho and Salimans, 2022] achieved remarkable results at improving the generation quality of Latent Diffusion Models (LDMs). However, such techniques usually require to do additional forward/backward passes at inference time effectively making the generation slower. In order to eliminate this inference overhead, an idea is to distill the guidance into the LDM during the training. This has been experimented with Classifier Free Guidance in the following [Meng et al., 2023]. However, we propose that distillation can be performed for an arbitrary guidance technique. In the following, experimentation has been done on distilling guidance with CLIP as a classifier [Radford et al., 2021] as well as a blue tone guidance. We further demonstrate that introducing guidance at training time is possible and effective compared to performing it at inference time.

1 Introduction

Over the past 5 years, interest in text-to-image generation hasn't ceased to grow to the point where private companies have begun to develop their own model to sell their services. In particular, latent diffusion models (LDMs) such as Stable Diffusion [Rombach et al., 2022] have proven to be a powerful and expressive approach to tackle this problem. Indeed, these models are capable of generating high resolution images conditioned on various parameters such as text or even images.

However, even though the results achieved by LDMs are remarkable, those are far from being perfect and various kinds of problems can happen in the generated images such as artifacts on faces, incorrect number of fingers, etc. In particular, one of these issues resides in the model not correctly following the given prompt. This can lead to undesirable outputs that do not accurately reflect the initial intention of the user.

A common technique addressing this issue is the use of guidance to govern the generation at inference time. In particular, Classifier-Free guidance (CFG) [Ho and Salimans, 2022] and classifier guidance using the CLIP Model [Radford et al., 2021] have achieved promising results at improving quality and control of the generation. However, this comes at a certain cost. Indeed, the guidance is introduced at inference time which typically means that for each inference step, either multiple forward passes (CFG) or a backward propagation through the classifier model are required. This significantly extends the generation time and also heightens memory usage, thereby rendering image generation more energy-intensive, along with all the accompanying drawbacks.

This study aims to demonstrate that it is possible to distill the characteristics of any guidance method into an LDM model through proper fine-tuning. This approach allows trading off the inference overhead for a reasonable increase in training time. Essentially, more time is invested in refining the model, which results in improved time/memory performance during inference. This

could effectively reduce the costs of running an LDM model while maintaining a high-quality standard.
To achieve this goal, the focus is initially set on distilling classifier-free guidance into a Stable Diffusion model to assert the divergence in quality between the actual guidance and the fine-tuned model. Subsequently, a similar experiment is conducted for a classifier guidance technique: CLIP guidance. Finally, to further demonstrate that any arbitrary guidance method can be distilled into the model, experiments will be conducted on guidance that aims to shift generation towards blue tones, referred to as blue guidance.

2 Related Work

Diffusion models achieve impressive results in generating images [Ho et al., 2020]. These models essentially train to predict the noise in a given image, allowing for its gradual denoising. Subsequently, this process is applied starting from a random Gaussian distribution, enabling the generation of new images.

Latent Diffusion Models (LDMs) enable the efficient generation of high-resolution images. They utilize an auto-encoder to compute a latent representation of the image, allowing a smaller input size for the diffusion model. This approach makes it possible to generate large-sized images with reasonable computing power. A popular LDM referenced in this paper is Stable Diffusion [Rombach et al., 2022].

Classifier-Free Guidance is a technique that has shown remarkable results in improving prompt alignment during generation [Ho and Salimans, 2022]. One of its strengths lies in its straightforward implementation. Specifically, at each inference step, a linear combination of a conditional and an unconditional prediction is used to compute the final prediction. This approach encourages the generation to more closely follow the given prompt.

CLIP [Radford et al., 2021] is a model designed to evaluate the similarity between a caption and an image . It can be used as a classifier to provide guidance, thereby improving the quality and text alignment of generated images.

Classifier-free guidance distillation can be performed to generate high quality images while maintaining a low inference overhead [Meng et al., 2023].

3 Preliminaries

3.1 Latent Diffusion Models

Diffusion Models [Ho et al., 2020] work by predicting Gaussian noise injected into the data at a particular step.

Forward process: More concretely, they define a forward process that aims to drift the data toward a Gaussian distribution by gradually adding noise. This process is defined as follows (1):

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{\beta_t} \epsilon_t \tag{1}$$

Here, x_t depicts the noisy data at timestep t. Thus x_0 corresponds to the original data. Also, α_t and $\beta_t = 1 - \alpha_t$ describe the amount of noise added at timestep t. Finally, ϵ_t is sample from a normal distribution.

Reverse process: The reverse process aims to learn to generate data by reversing the noise addition process. This is where the model is trained. It will learn to predict the step added noise ϵ_t given the noisy data x_t. This is done gradually for each timestep allowing to gradually denoise the data. More formally, this process can be pictured as below (2):

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) \tag{2}$$

Here, $\epsilon_\theta(x_t, t)$ corresponds to the prediction of ϵ_t and $\bar{\alpha}_t = \prod_{i=0}^{t} \alpha_i$. This prediction is performed by the model with parameters θ given the noisy data x_t and the current timestep t.

Thus, using the forward process, it is possible to train a model to predict the noise added at each timestep. Then, using this trained model and the reverse process starting from a random Gaussian noise, new data following the initial distribution can be be generated.
Latent Diffusion Models adhere to the same principle but have one major difference. First, an auto-encoder is trained to learn a latent representation z_t of the data x_t. Then, the diffusion model is fed with these latents. This approach significantly reduces the computational requirements and makes the generation of high-resolution images with diffusion models feasible.

3.2 Classifier Free Guidance

Classifier free guidance aims to improve the text alignment [Ho and Salimans, 2022]. In the context of text to image generation, this technique works as follows. At each timestep t, two noise predictions will be made: one conditioned on the given prompt $\epsilon_\theta(x_t, t, \text{"}prompt\text{"})$ and another unconditional $\epsilon_\theta(x_t, t, \text{""})$.

Then, to compute the final prediction, a linear combination of those terms is performed with the objective of *pushing* to prediction towards the conditional one. This process can be described as below (3).

$$\hat{\epsilon}_\theta(x_t, t, prompt) = \epsilon_\theta(x_t, t, "") + \alpha\,[\,\epsilon_\theta(x_t, t, "prompt") - \epsilon_\theta(x_t, t, "")\,] \tag{3}$$

Here, α is the guidance scale, it depicts how much is *pushed* the guidance. In particular, with a guidance scale of $\alpha = 1$ there is actually no guidance applied.

3.3 CLIP Guidance

CLIP is a model utilized to determine the correspondence between an image and its caption [Radford et al., 2021]. This correspondence is quantified by the CLIP score (the higher the score, the more accurately the caption describes the image). This model can be employed as a classifier to guide the generation of images. Specifically, given a noisy image x_t at timestep t, its CLIP score is first computed, then back-propagation is used to obtain the gradient. Finally, the gradient (multiplied by a scaling factor) is subtracted from the original x_{t-1} prediction to yield the guided one. More formally, this process can be described as shown in Equation (4):

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left[x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\left(\epsilon_\theta(x_t, t) + \gamma\sqrt{\bar{\beta}_t}\cdot\nabla_{x_t}\text{CLIP}_{\text{score}}(x_t, "prompt")\right)\right] \tag{4}$$

Here, γ represents the guidance scale, indicating the degree to which the process aims to improve the CLIP score. By employing this technique, the generated images are encouraged to attain a higher CLIP score, which, due to the intrinsic properties of CLIP, results in enhanced text-image alignment. This method can be generalized to any classifier guidance by taking the corresponding gradient (and adapting the sign depending on if the objective is to minimize or maximize the classifier score).

4 Distilling guidance into a diffusion model

In this section, a method for distilling guidance techniques into a Stable Diffusion model [Rombach et al., 2022] is proposed. The general approach is as follows. Firstly, a specific guidance technique is selected, along with a diffusion model into which the guidance will be distilled. The next objective is to generate a dataset that facilitates the fine-tuning of the model. Finally, using this newly generated dataset, the model is fine-tuned to incorporate the guidance.

4.1 Pre-processing the dataset

The initial step in the pre-processing phase involves selecting a dataset and a diffusion model. For this research, the model *miniSD* [LambdaLabs, 2022], a smaller version (256x256) of the

Stable Diffusion model [Rombach et al., 2022], is chosen. Concerning the dataset, a subset of *Laion Aesthetics 6.5 Plus* [Christoph Schuhmann, 2022] is selected, as this is the dataset on which the base version of Stable Diffusion was trained.

The dataset is then pre-processed in the following manner. For each image in the dataset, the latent representation is computed, and a random timestep is selected. Subsequently, the necessary passes through the model (and the classifier, if applicable) are performed to obtain information relevant to guidance. Finally, all these data are output as a sample for the new dataset.
Let's now see a more detailed description of this algorithm based on the guidance method used.

Classifier-Free Guidance: Regarding Classifier-Free Guidance [Ho and Salimans, 2022], the idea is to output the two noise predictions (conditional and unconditional). This way, during the finetuning, various guidance scales can be simulated by performing linear combinations with different factors. More concretely, the algorithm can be found below [1]:

Algorithm 1 Classifier-free pre-processing

for $z =$ Latent of an image **do**
  $t \leftarrow Rand(0, 1000)$
  $\epsilon_{step} \leftarrow$ Noise at timestep t
  $\tilde{z} \leftarrow$ Noised version at t according to (1)
  $\hat{\epsilon}_{\theta,cond} \leftarrow \text{UNET}(\tilde{z}, t, \text{"}caption\text{"})$
  $\hat{\epsilon}_{\theta,uncond} \leftarrow \text{UNET}(\tilde{z}, t, \text{""})$
  Output $\{\tilde{z},\ \hat{\epsilon}_{\theta,cond},\ \hat{\epsilon}_{\theta,uncond},\ t,\ \text{"}caption\text{"}\}$
end for

Classifier guidance: The pre-processing pipeline for guidance techniques that utilize a classifier differs slightly but follows a similar approach. In this case, the objective is to output the base noise prediction of the model as well as the gradient of the classifier loss function. This method ensures the ability to vary the guidance scale during fine-tuning. Specifically, the algorithm employed is as detailed in 2:

Algorithm 2 Classifier pre-processing

for $z =$ Latent of an image **do**
  $t \leftarrow Rand(0, 1000)$
  $\epsilon_{step} \leftarrow$ Noise at timestep t
  $\tilde{z} \leftarrow$ Noised version at t according to (1)
  $\hat{\epsilon}_{\theta} \leftarrow \text{UNET}(\tilde{z}, t, \text{"}caption\text{"})$
  $\nabla_{\tilde{z}}\text{Class.} \leftarrow$ Backward prop. of the classifier loss
  Output $\{\tilde{z},\ \hat{\epsilon}_{\theta},\ \nabla_{\tilde{z}}\text{Class},\ t,\ \text{"}caption\text{"}\}$
end for

4.2 Loss for fine-tuning the model

To fine-tune the model effectively, the loss function needs to be modified to incorporate the guidance aspect using the previously computed datasets. As a result, two distinct loss functions have been developed: one tailored for classifier-free guidance and the other specifically designed for classifier guidance. Detailed descriptions of these loss functions will now be provided.

Classifier-Free guidance: For classifier-free guidance, our approach involves calculating the *Mean Square Error* (MSE) between the prediction of the model undergoing fine-tuning and the prediction generated when applying classifier-free guidance. This leads us to the following loss function (5).

For a sample $\{\tilde{z},\ \hat{\epsilon}_{\theta,cond},\ \hat{\epsilon}_{\theta,uncond},\ t,\ \text{"}caption\text{"}\}$ of the dataset, a noise prediction ϵ_θ for $\tilde{z}$ by the model being finetuned and a guidance scale of α:

$$Loss = \text{MSE}\left(\epsilon_\theta,\ \hat{\epsilon}_{\theta,uncond} + \alpha\left(\hat{\epsilon}_{\theta,cond} - \hat{\epsilon}_{\theta,uncond}\right)\right) \tag{5}$$

Classifier guidance: Regarding classifier guidance, the approach is similar. The difference lies in the way the guidance is applied, leading to the following loss function for our training (6). For a sample $\{\tilde{z},\ \hat{\epsilon}_\theta,\ \nabla_{\tilde{z}}\text{Class},\ t,\ \text{"}caption\text{"}\}$ of the dataset, a noise prediction ϵ_θ for $\tilde{z}$ by the model being finetuned and a guidance scale of γ:

$$Loss = \text{MSE}\left(\epsilon_\theta,\ \hat{\epsilon}_\theta + \gamma \cdot \sqrt{\beta_t} \cdot \nabla_{\tilde{z}}\text{Class}\right) \tag{6}$$

4.3 Fine-tuning the model

Finally, to fine-tune the model, a sample is first taken from the new dataset, followed by a prediction of the noise from the latent in the sample. Subsequently, the loss is computed and back-propagated through the model. This process can be described by the high-level algorithm in [3].

Algorithm 3 Fine-tuning

```
for sample ∈ Pre-processed dataset do
    ε_θ ← UNET(sample.z̃)
    L ← Loss(ε_θ, sample)
    Back-propagate L
end for
```

5 Experiments

In this section, the conducted experiments are presented. Distillation is performed on three types of guidances to assess the variations based on the type of guidance distilled. The process begins with Classifier-Free guidance [Ho and Salimans, 2022] and then continues with the CLIP model [Radford et al., 2021]. Finally, experiments are conducted on blue loss guidance, inspired by [Whitaker, 2022]. The implementation was carried out using the *Diffusers* library [von Platen et al., 2022].

5.1 Classifier-Free guidance

Experiment: For this experiment, 20,000 samples were first generated using the method described in Section 4.1. The model was then fine-tuned with a guidance scale of 7.5 for 150,000 steps, maintaining a constant learning rate of $5 \cdot 10^{-6}$. Detailed training parameters can be found in Table 1.

# samples	20000
Guidance scale	7.5
Batch size	8
Epochs	60
Learning rate	$5 \cdot 10^{-6}$
GPUs	1 Rtx 3090
Training time	18h

Table 1: Training parameters for distilling classifier-free guidance.

Results: This fine-tuning process resulted in qualitative enhancements in the generated images when compared to the base model. Qualitative comparisons are available in Figure 1. Additional comparisons are presented in the appendix 8.3.

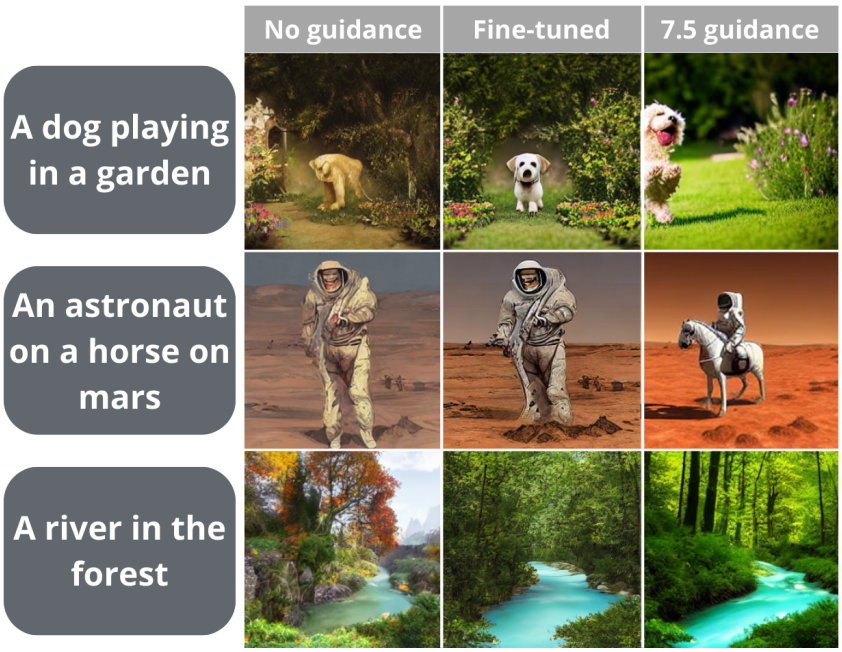

Figure 1: Qualitative comparison of the fine-tuned model against the base one with and without Classifier Free Guidance.

Furthermore, to obtain a more quantitative interpretation of these results, the CLIP score and the FID score of the fine-tuned model can be computed. To calculate the CLIP score, 200 prompts from the *Drawbench* dataset [Imagen Research Team, 2022] were used, generating three images per prompt. For the FID score, a subset of 600 images from the *COCO2017* validation dataset [Lin et al., 2014] was utilized.

These scores can also be computed for the base model with different Classifier-Free Guidance scales. This computation enables an evaluation of where the fine-tuned model stands in relation to the CLIP/FID curve. Figure [2] presents the CLIP versus FID plot, indicating the position of the fine-tuned model on it.

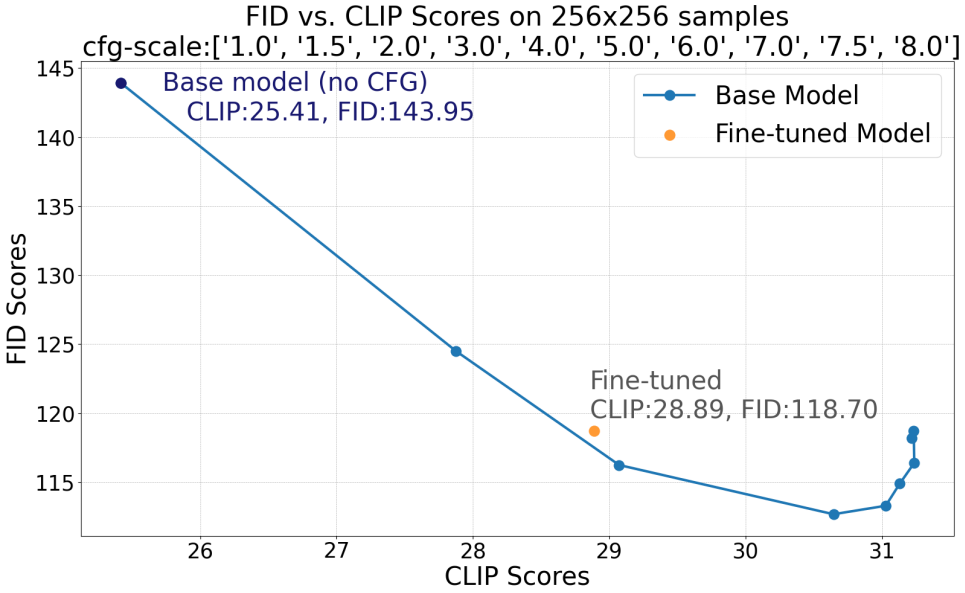

Figure 2: CLIP against FID scores for various classifier free guidance scales. Data can be found in Annex 8.1.

The Figure [2] illustrates the effectiveness of our fine-tuning approach which significantly enhances both FID and CLIP scores when compared to the base model. Importantly, this technique efficiently circumvents the requirement of a dual forward passes in each inference iteration, typically associated with classifier-free guidance.

Performances: In comparing the computational performances of the fine-tuned model against the base model with Classifier-Free Guidance, the primary focus was on two critical factors: memory consumption and time per iteration. To assess these metrics, measurements of time and memory usage were conducted for image generation using both models. The results of these measurements can be found in Figures [3] and [4].

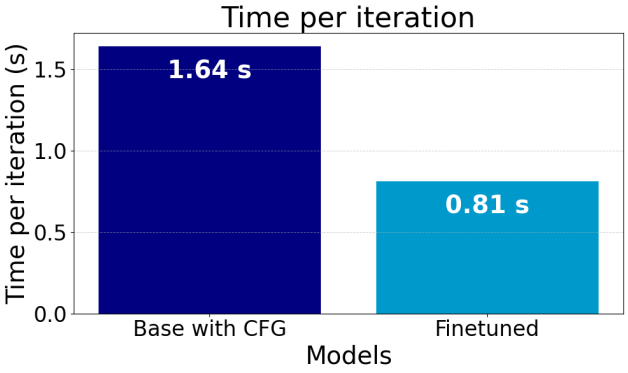

Figure 3

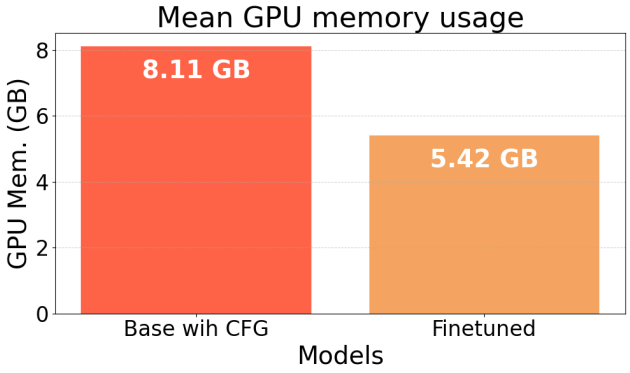

Figure 4

Figure 5: Mean time per iteration and memory usage for the finetuned model and the base model with 7.5 guidance. Performed on one Rtx 3090, batch of 90 images.

Figure [3] demonstrates that the fine-tuned model has effectively halved the generation time. This reduction is expected due to the fact that only a single forward pass per iteration is performed, as opposed to two.

The GPU memory usage (Figure [4]) for the fine-tuned model has been nearly halved compared to the base model with Classifier-Free Guidance. This reduction aligns with expectations and is primarily attributable to the elimination of the need to retain the unconditional latent predictions on the GPU.

5.2 CLIP guidance

Experiment: Here, the process began by selecting 10,000 samples as per Section 4.1. The model was then fine-tuned with a guidance target of 300 for 105,000 steps, using a learning rate of $1 \cdot 10^{-6}$. All the parameters are detailed in Table [2]. It is noteworthy to remark that different guidance scale schedules were tested but did not prove to be as effective as constant guidance training.

# samples	10000
Guidance scale	300
Batch size	6
Epochs	50
Learning rate	$1 \cdot 10^{-6}$
GPUs	1 Rtx 3090
Training time	9h

Table 2: Training parameters for distilling CLIP guidance.

Results: Once again, this method resulted in qualitative enhancements in the generated images. Indeed, Figure [6] presents several qualitative outputs of the fine-tuned model for comparison with the base model. More comparisons can be found on Appendix 8.4.

Figure 6: Qualitative comparison of the fine-tuned model against the base one with and without CLIP guidance (300 guidance scale).

To provide a more quantitative assessment of the fine-tuned model's capabilities, the CLIP/FID scores curve was computed. This approach enables the determination of the effectiveness of the fine-tuning when applied in conjunction with Classifier-Free Guidance. In Figure [7], the CLIP/FID score curves for both the base and the fine-tuned models are presented, encompassing a range of Classifier-Free Guidance scales. This comparison aims to illustrate the impact of the fine-tuning technique when the fine-tuned model is coupled with various levels of Classifier-Free Guidance.

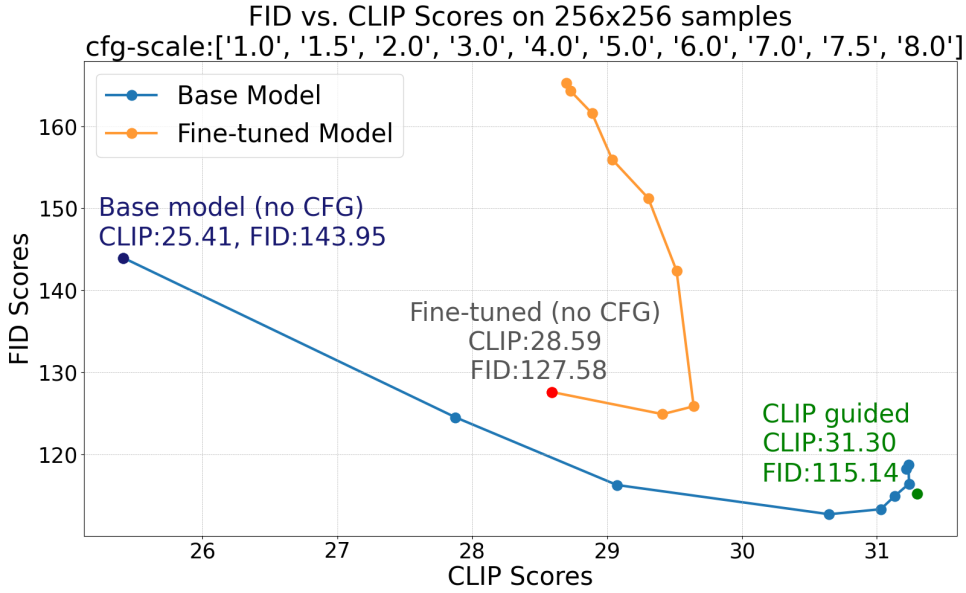

Figure 7: CLIP against FID scores of the base model and of the fine-tuned one for various Classifier-Free Guidance scales. Data can be found in Annex 8.2.

Figure [7] reveals that applying Classifier-Free Guidance [Ho and Salimans, 2022] to the fine-tuned model yields some improvements in the scores at lower scales. However, these enhancements are modest, and beyond a scale factor of 2.0, the scores begin to decline. This indicates that while the fine-tuned model shows quality improvements over the base model, it compromises the potential benefits of Classifier-Free Guidance. Essentially, this highlights a trade-off: opting for a faster model with moderate scores while applying limited classifier-free guidance, or choosing a model that requires guidance for optimal performance but achieves higher scores.

Performances: To put in perspective the computational benefits of our approach, the focus was set on the memory usage and time per iteration. Measurements were performed on the base model with 300 CLIP guidance as well as on the fine-tuned version. The subsequent Figures [8][9] resume those measurements.

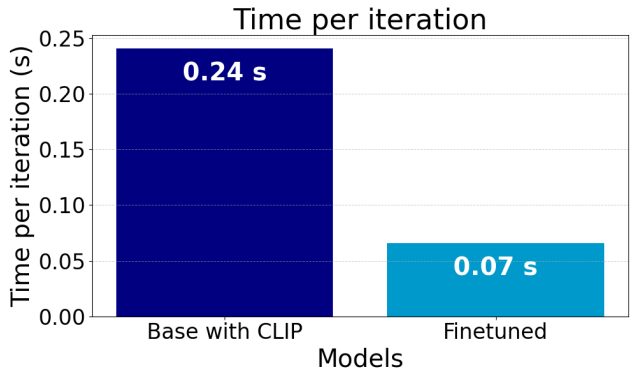

Figure 8

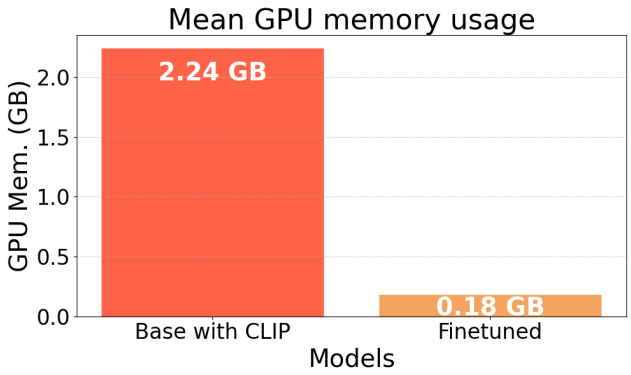

Figure 9

Figure 10: Mean time per iteration and memory usage for the finetuned model and the base model with 300 CLIP guidance. Performed on one Rtx 3090, batch of one image.

Regarding the time per iteration, the results revealed that the fine-tuned model was almost 4 times faster. This emphasizes the fact that the back-propagation done in CLIP guidance is a demanding process that sacrifices a lot of the generation time.
For the GPU memory usage on the other hand, the gap is even bigger with the fine-tuned model taking an order of magnitude less memory during the generation. The key difference in performance is once again largely attributed to the back-propagation process. For computing the final gradient, back-propagation must temporarily store intermediate gradients, which are roughly the size of the model itself. This requirement significantly increases the memory usage, explaining the observed gap in performance.

5.3 Blue guidance

Experiment: The final experiment aimed to fine-tune the model using an arbitrary guidance technique. For this purpose, guidance that encourages the generation to drift toward blue tones was selected, inspired by [Whitaker, 2022]. The model was then trained using parameters detailed in Table [3].

# samples	5000
Guidance scale	100
Batch size	8
Epochs	30
Learning rate	$5 \cdot 10^{-6}$
GPUs	1 Rtx 3090
Training time	2h18

Table 3: Training parameters for distilling the blue tone guidance.

Applying our method, the model was able to learn to apply blue tones to the generated images. Qualitative results put this behavior in perspective in Figure [11].

Figure 11: Blue tone of the images at different steps of the training.

As evident in Figure [11], the model rapidly assimilated the blue tone guidance. The qualitative results demonstrate that after 7,500 steps, there's minimal change in the generation, indicating that the model almost completely mastered this specific blue tone guidance swiftly. This rapid learning curve is a promising sign for the applicability of this method to arbitrary guidance functions.

6 Discussions

6.1 Trade-off of distillation

The results shown for Classifier free guidance [5.1] and CLIP guidance [5.2] demonstrate the ability of this distillation method to improve the quality of the generation over the base model. Indeed, in both cases, the fine-tuned model resulted in a better score on the CLIP/FID map. Moreover, it is important to emphasize on the size of the dataset that was used. Indeed, the models were trained with only few samples (less than 20k) for a short time (less than 60 epochs). It is likely that with further training, the fine-tuned model could have improve even more.

However, this distillation comes to a certain cost. Indeed, as seen on Figure [7], the reaction of the fine-tuned model to the added application of Classifier-Free Guidance [Ho and Salimans, 2022] resulted in lower scores for the maximum on the CLIP/FID map than simply performing Classifier-Free Guidance. This indeed highlights the trade-off that is present here. The quality has improved over the base model but it sacrifices the addition of other guidance techniques at inference time. This must be taken into account when using this method.

6.2 Arbitrary guidance distillation

The example of blue tone guidance presented in [5.3] revealed some encouraging results about distilling an arbitrary function into the model. The quick adaptation of the model opens the door to interesting perspectives. Indeed, this method could be used with more interesting classifier such as one validating the number of fingers on a hand, or one detecting artifacts on faces or even a text recognizer to improve the quality of images with text (which is a challenging task as of today). Thus, this adaptability suggests the potential to steer the model towards solving a variety of issues in image generation, tailored to specific requirements or objectives.

6.3 Perspective use

The potential of this methods appears in the case where inference time is critical or when the quantity of generated images prevails over the individual quality. In such situations, spending more time in the training (time that is reasonable) could reveal itself to be more interesting than applying inference guidance. Indeed, simply using the base model would result in bad quality images and using inference guidance would not obey the constraints or significantly increase the GPU cost. Thus, our method reveals itself to be particularly useful in such circumstances.

7 Conclusion

Overall, these researches revealed some of the potential of guidance distillation ([5.1], [5.2]). Substantial quality enhancements were achieved compared to the base model, and while it doesn't surpass the performances of standard guidance, the reduced computational demand, which can be several magnitudes lower, sets distillation apart as a distinct and valuable approach. Furthermore, the potential ability of distilling any kind of guidance technique makes this method an ideal choice to fine-tune a model for some specific needs. This is further reinforced by the quick learning ability shown in [5.3] that offers this technique.
Finally, these researches focused on distilling a single guidance function into the model. An interesting direction lies in distilling multiple guidances in a single model, leading to the possibility of creating a model capable of focusing on various aspects of its generation based on the asked image. This could effectively lead to a great improvement in the coherence and detailing of text to image generation.

References

[Christoph Schuhmann, 2022] Christoph Schuhmann (2022). Laion aesthetics 6.5 plus dataset. https://laion.ai/blog/laion-aesthetics/.

[Ho et al., 2020] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models.

[Ho and Salimans, 2022] Ho, J. and Salimans, T. (2022). Classifier-free diffusion guidance.

[Imagen Research Team, 2022] Imagen Research Team (2022). Drawbench dataset. https://imagen.research.google/.

[LambdaLabs, 2022] LambdaLabs (2022). lambdalabs/minisd-diffusers. https://huggingface.co/lambdalabs/miniSD-diffusers.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2014). Microsoft coco: Common objects in context. *arXiv preprint arXiv:1405.0312*.

[Meng et al., 2023] Meng, C., Rombach, R., Gao, R., Kingma, D. P., Ermon, S., Ho, J., and Salimans, T. (2023). On distillation of guided diffusion models.

[Radford et al., 2021] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision.

[Rombach et al., 2022] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695.

[von Platen et al., 2022] von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., and Wolf, T. (2022). Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers.

[Whitaker, 2022] Whitaker, J. (2022). Grokking stable diffusion. https://colab.research.google.com/drive/1dlgggNa5Mz8sEAGU0wFCHhGLFooW_pf1?usp=sharing.

8 Appendix

8.1 Classifier-free guidance distillation quantitative results

This section provides the data from which the Figure [2] has been created.

Guidance	CLIP	FID
1.0 (= no guidance)	25.41	143.9
1.5	27.88	124.5
2.0	29.07	116.2
3.0	30.20	111.8
4.0	30.65	112.7
5.0	31.02	113.3
6.0	31.13	114.9
7.0	31.24	116.4
7.5	31.23	118.7
8.0	31.22	118.2

Table 4: CLIP/FID score of the base model for various Classifier-Free guidance scales.

Model	CLIP	FID
Base (no guidance)	25.41	143.9
Fine-tuned	28.89	118.7
Target (7.5 cfg)	31.23	118.7

Table 5: CLIP/FID scores comparison of the fine-tuned model distilled for Classifier-Free Guidance.

8.2 CLIP guidance distillation quantitative results

This section provides the data from which the Figure [7] has been created. The CLIP/FID of the base model for various Classifier-Free Guidance scales can be found on Table [4].

Guidance	CLIP	FID
1.0 (= no guidance)	28.59	127.6
1.5	29.41	124.9
2.0	29.64	125.8
3.0	29.71	133.8
4.0	29.51	142.4
5.0	29.31	151.2
6.0	29.04	156.0
7.0	28.89	161.6
7.5	28.73	164.3
8.0	28.70	165.3

Table 6: CLIP/FID scores of the CLIP fine-tuned model for various Classifier-Free guidance scales.

Model	CLIP	FID
Base (no guidance)	25.41	143.9
Fine-tuned	28.59	127.58
Target (300 CLIP guidance)	31.2	115.4

Table 7: CLIP/FID scores comparison of the fine-tuned model distilled for CLIP Guidance.

8.3 Classifier-free guidance qualitative results

In this section, the performance of the fine-tuned model is compared across various types of tasks: landscapes (Figure 12), structures (Figure 13), human faces (Figure 14), musical instruments (Figure 15), and artistic styles (Figure 16).

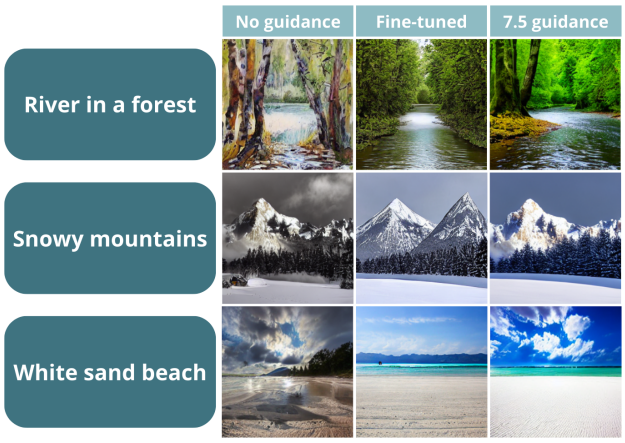

Figure 12: Comparison on landscapes of the distilled model for class. free guidance against the base one.

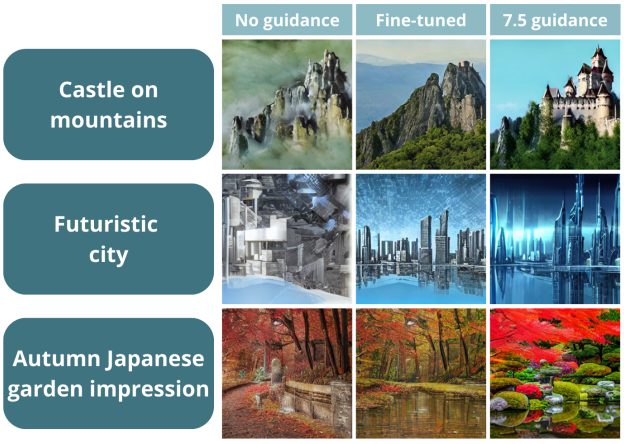

Figure 13: Comparison on structures of the distilled model for class. free guidance against the base one.

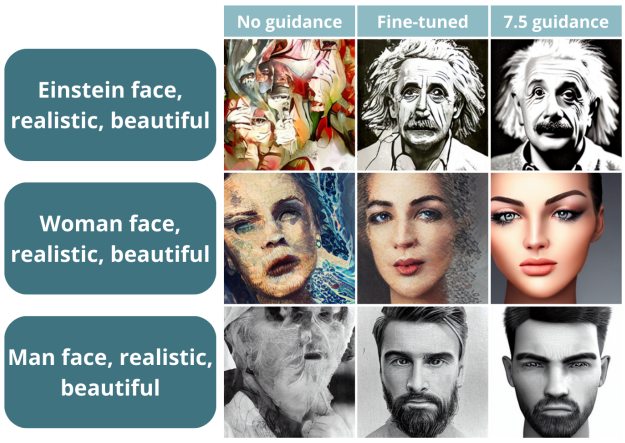

Figure 14: Comparison on human faces of the distilled model for class. free guidance against the base one.

Figure 15: Comparison on musical instruments of the distilled model for class. free guidance against the base one.

Figure 16: Comparison on artistic styles of the distilled model for class. free guidance against the base one.

8.4 CLIP guidance qualitative results

In this section, the performance of the fine-tuned model is compared across various types of tasks: landscapes (Figure 17), structures (Figure 18), human faces (Figure 19), musical instruments (Figure 20), and artistic styles (Figure 21).

Figure 17: Comparison on landscapes of the distilled model for CLIP guidance (300 guidance scale) against the base one.

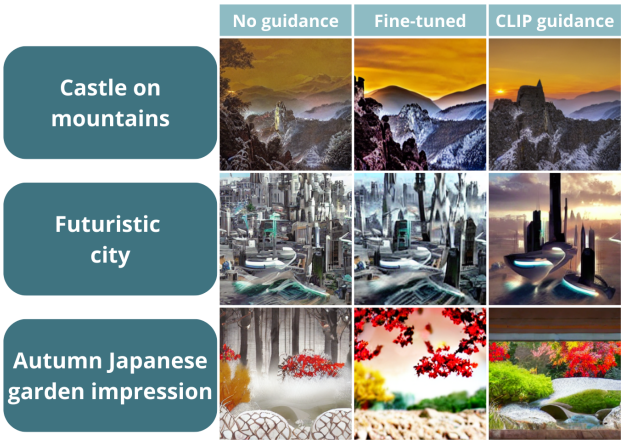

Figure 18: Comparison on structures of the distilled model for CLIP guidance (300 guidance scale) against the base one.

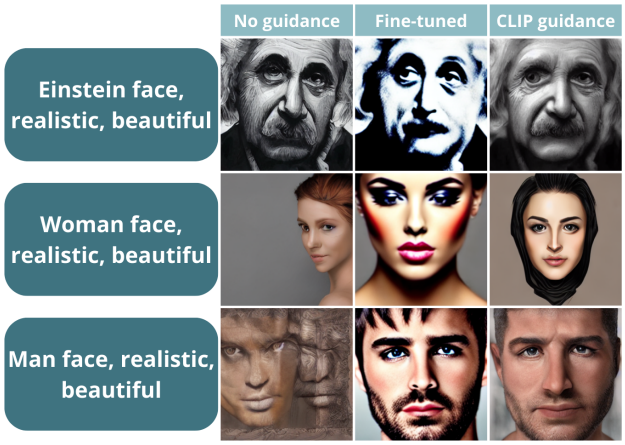

Figure 19: Comparison on human faces of the distilled model for CLIP guidance (300 guidance scale) against the base one.

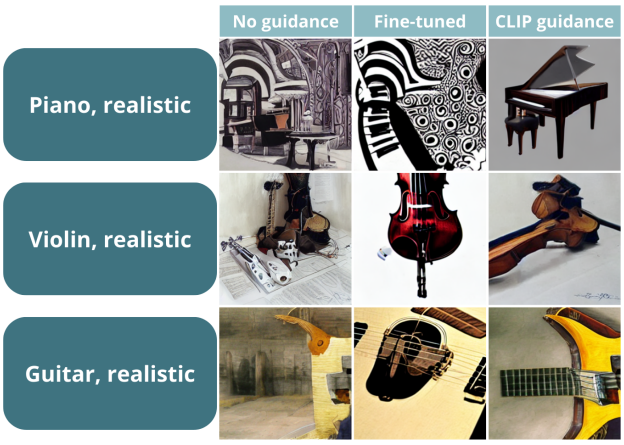

Figure 20: Comparison on musical instruments of the distilled model for CLIP guidance (300 guidance scale) against the base one.

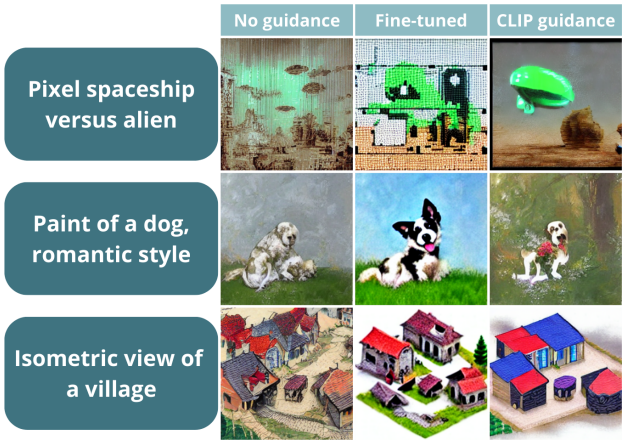

Figure 21: Comparison on artistic styles of the distilled model for CLIP guidance (300 guidance scale) against the base one.