CS 236 Default Project: Stable Diffusion and Beyond

Stefano Ermon, Bryan Chiang, Sylvia Yuan

October 11, 2023

1 Preliminaries

Stable Diffusion [RBL+22] is a latent diffusion model that leverages text-to-image capabilities to synthesize high-fidelity photographic images from textual descriptions. Its open-source release in 2022 has catalyzed the generative AI revolution and illustrates how scaling up some of the approaches we saw in class (i.e. score-based models) can yield incredible results. Please see this blog post for a more detailed look at the internals.

The goal of the default project is to get you acquainted with several ways of customizing and controlling Stable Diffusion to take it to the next level for your own use case.

2 Vanilla fine-tuning (on a budget)

Sometimes the quality of Stable Diffusion just isn't good enough for your particular task. Fine-tuning involves updating the weights of the stable diffusion model to more closely capture the desired style and image characteristics of a specific dataset. This is a step that comes after the initial pretraining over billions of text-image pairs. Unfortunately, fine-tuning all the parameters can be quite GPU memory intensive and time-consuming for us GPU-Poor [Pat23].

To address this, we turn to a technique called LoRA (Low Rank Adaptation) [HSW+21] that enables efficient fine-tuning through low-rank weight updates. Since there are a smaller number of parameters to learn, LoRA is faster and significantly more memory efficient compared to fine-tuning all weights. While there are numerous techniques for parameter-efficient fine-tuning, LoRA has empirically worked well (i.e. almost matching full fine-tune performance) and has gained strong traction in the open-source community over the past year.

Full fine-tuning involves updating every weight matrix $W_\ell^0 \in \mathbb{R}^{d_1 \times d_2}$ in each layer ℓ into an arbitrary new weight matrix W_ℓ^{ft}. In contrast, LoRA constrains the new weight matrix to be of the form

$$W_\ell^{ft} = W_\ell^0 + AB^T \tag{1}$$

, where $A \in \mathbb{R}^{d_1 \times p}, B \in \mathbb{R}^{p \times d_2}$ and $p << d_1, d_2$. W_ℓ^0 is kept frozen and only the rank-p residual matrix AB^T is fine-tuned. The A, B matrices comprise the LoRA adapter and are injected into the base model at inference time. For Stable Diffusion, LoRA adapters are incredibly compact (on the order of MBs) and thus are easy to upload and share online.

3 Personalizing Generation (Subject-Specific Finetuning)

3.1 Textual Inversion

Text-to-image diffusion models offer users the ability to generate high quality images with natural language guidance. In practical applications, users may need more control over the generated images. For example, users may want to generate images of some specific concepts or compose objects into new scenes. Textual inversion [GAA+22] offers the means for this kind of creative freedom. In textual inversion, a string with a "placeholder" word is converted to tokens. Then the tokens are converted to embeddings, which are transformed to a conditioning code to guide the generative model. This embedding vector is optimized with a reconstruction objective against the placeholder word.

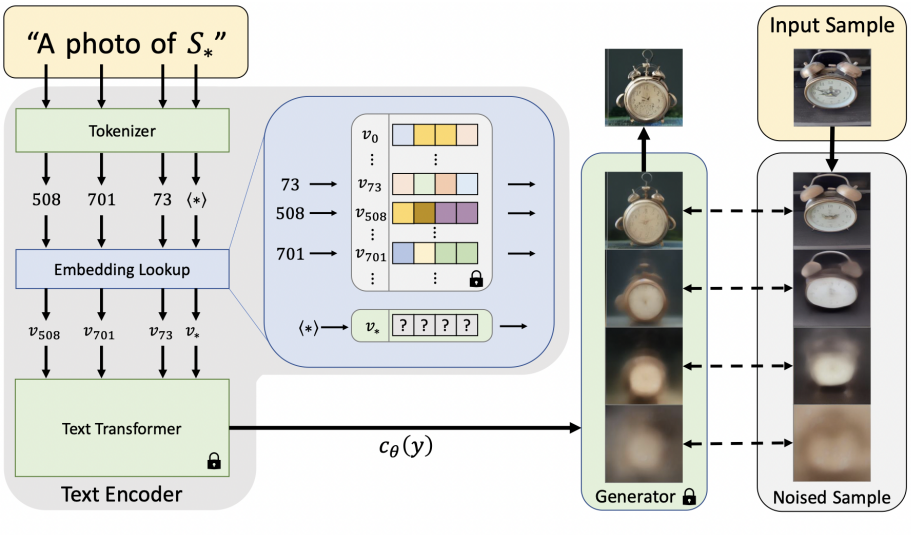

Figure 1: Textual inversion architecture, figure from [GAA+22].

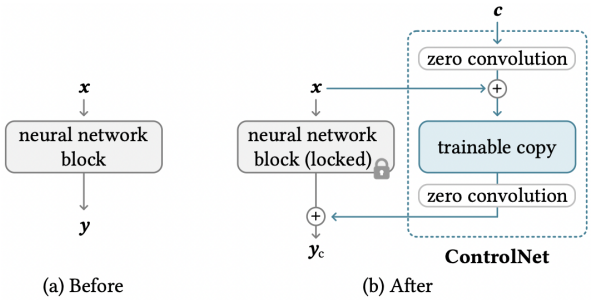

Figure 2: ControlNet architecture, figure from [ZRA23].

3.2 DreamBooth

DreamBooth [RLJ+22] is another approach that allows us to personalize text-to-image models, enabling them to generate high-quality images of specific subjects in diverse contexts.

The method refines a pre-trained text-to-image diffusion model using around three to five images of a given subject. Initially, a low-resolution model is fine-tuned using input images paired with a text prompt, which includes a unique identifier and the subject's class (e.g., "A photo of a [T] dog"). Also, a class-specific prior preservation loss is applied in parallel, leveraging the model's existing semantic knowledge of the class. This process promotes the generation of diverse instances belonging to the subject's class using a class name in a text prompt (e.g., "A dog"). Subsequently, the model's super-resolution components are fine-tuned with pairs of low and high-resolution images from the input set to ensure the retention and high-fidelity of intricate subject details.

4 Conditional Generation with Diffusion Models: ControlNet

In some applications, users often need more fine-grained control over generated images. ControlNet[ZRA23] is a method to conditionally control pre-trained text-to-image diffusion models. The ControlNet architecture learns conditional inputs by locking the pre-trained large diffusion models and preserving their robust encoding layers trained from internet-sized datasets. Figure 2 shows an overview of ControlNet architecture. ControlNet enables controlling image generation with a multitude of control inputs, such as edge maps, depth maps, human pose, etc.. In this project, you will experiment with ControlNet.

Figure 3: Sample ControlNet results, figure from [ZRA23].

5 Tasks

For this project, you will explore diffusion models. We provide a list of starting steps below:

1. Choose a dataset and fine-tune Stable Diffusion on this dataset with LoRA.
2. Choose a few (3-5) images to experiment with personalized text-to-image generation. Try both textual inversion and DreamBooth.
3. Can you extend DreamBooth to do multi-subject generation?
4. Choose a category of current ControlNet control inputs and explore conditional generation with diffusions by experimenting with pre-trained ControlNet.
5. Apply diffusion models or ControlNet on data forms other than images.

Note that the amount of work required for groups of different sizes is different. For example, for a one person group, step 1 and 2 may be sufficient, but for group of larger sizes, you will need to explore extensions of these steps, such as steps 3-5 or your own ideas.

5.1 Training a LoRA

For the first task, you will train a LoRA over your own custom dataset. Collect your own dataset of around 100 images with a common theme. You can see this blog post for inspiration of what kind of themes the dataset can encapsulate, such as objects, styles, and concepts.

To do the fine-tuning, this repository is a good starting point. You are free to use any libraries and resources. To improve your results, try tweaking LoRA parameters such as the number of layers you choose to fine-tune, the dimension p, and the type of layers that you choose to apply LoRA to (linear, convolution, attention, etc.).

5.2 Training Textual Inversion

Come up with your own dataset of a specific subject (around 10 images). It could be any object, person, or animal.

Then, see the Textual inversion GitHub repository for instructions.

If due to GPU resources constraints, you want to use Colab for this step, consider looking into the diffusers library and using diffusers for Textual Inversion.

5.3 Training DreamBooth

Use the same subject dataset you created for the textual inversion part. Get started with the DreamBooth tutorial from HuggingFace. The tutorial combines DreamBooth with LoRA to conserve memory usage.

Compare DreamBooth against textual inversion with the same subject dataset. What kind of prompts work well with DreamBooth but not textual inversion, and vice versa? Does adding more data to the subject dataset improve quality? Does adding more diverse data (lighting, poses, etc.) improve quality?

5.4 Experimenting with ControlNet

Choose a category of ControlNet control input (e.g. edge maps, depth maps, sketches) and experiment with controlling generation with prompts of your choosing.

If due to GPU resources constraints, you want to use Colab for this section, also consider looking into the diffusers library and the StableDiffusionControlNetPipeline.

5.5 Evaluation

After you have trained your models, your job is to evaluate their performance. You should do this both quantitatively and qualitatively. You are responsible for selecting and designing evaluation metrics for each of your experiments. You should present quantitative and qualitative evaluation results.

Quantitative Evaluation You need to select and/or design evaluation metrics to assess the performance of your fine-tuned models. To get you started, here are some possible ideas on applicable metrics: 1). CLIP [RKH+21] similarity to evaluate prompt fidelity; 2) The kernel inception distance (KID)[BSAG18] measures the similarity between two sets of images and can be used to assess the distribution of your generated images and your training images; 3). For more ideas, you can check the Huggingface guide on evaluating diffusion models.

Qualitative Evaluation Qualitative evaluation is crucial in image generation tasks. As a starting point, image quality and prompt fidelity are important aspects in text-to-image generation models. One commonplace practice is to present figures containing your generation results in comparison to the baseline generation results.

References

[BSAG18] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[GAA+22] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[HSW+21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[Pat23] Dylan Patel. Google gemini eats the world "gemini". https://www.semianalysis.com/p/google-gemini-eats-the-world-gemini, August 2023.

[RBL+22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[RLJ+22] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.

[ZRA23] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.