SAM: Simpson's Artistic Memory

Jinrui Lin, Haomin Mo, Huihan Yang, Rongxiao Qu

The Chinese University of Hong Kong, Shenzhen

April 23, 2023

DDA4210_Group_project.git

Contents

Introduction

Get a sneak peek: some generated pictures

Figure: Sample generated images

Get a sneak peek: our app

Intro to the app

- has integrated 3 models
- just for demonstration purpose
- detailed prompt lead to better generated pictures
- have a try!

Limitation of the current version

- The speed and resources are limited since we are using free cloud service.
- High capacity access are not supported.
- If you have been waiting for a long time, it means you are in queue.

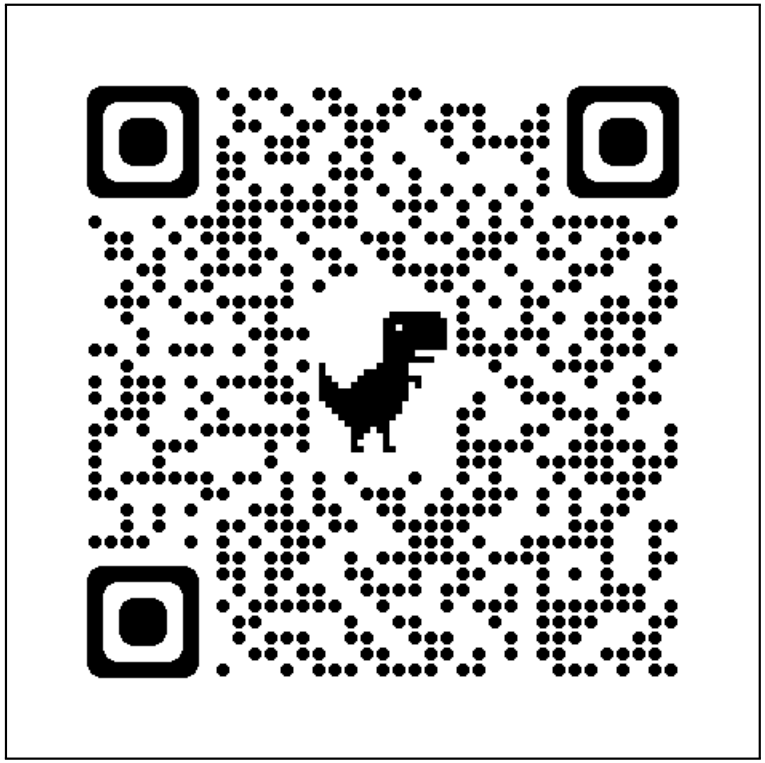


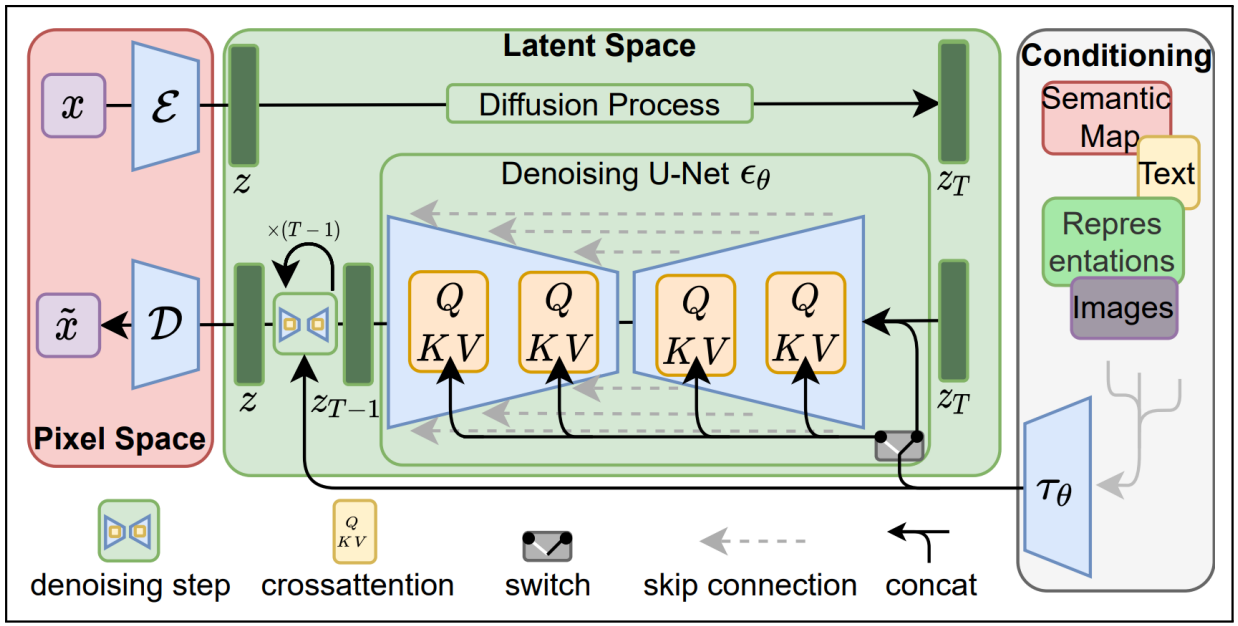
Figure: QR code for the app

Preview: What will we discuss?

- Motivation and practical applications
  - DELL-E, Wen Xin, Midjourney
  - MAIN REASON: INTERESTING
- We work on the topic - stable diffusion
  - Text-to-image, SD fine-tuned, Simpson's Style
- We ask the following questions:
  - Dataset? (Blip caption/Manually)
  - Fine-tuned? (simple baseline/ LoRA / Dreambooth)
- Our answer: Manually, Simpson Artistic Memory
  - Only children make choices.
  - Each options may have its advantages and we try to combine them.

Background knowledge: Stable Diffusion Model and Fine-tune

Stable diffusion Architecture[3]

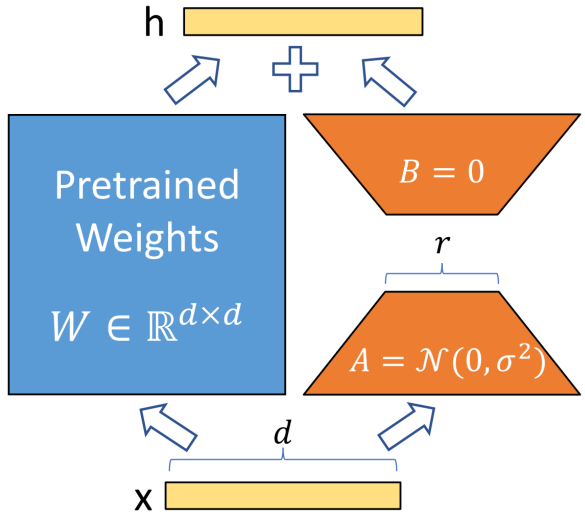


- Image Encoder: VAE Encoder
- Iteratively Adding Gaussian Noise (Diffusion Process)
- Text Encoder: CLIP ViT-L/14
- Denoising: U-Net
- Image Decoder: VAE Decoder
- Loss Function:

$$L_{LDM} := \mathbb{E}_{\varepsilon(x),y,\epsilon\sim\mathbb{N}(0,1),t}[||\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))||_2^2] \quad (1)$$

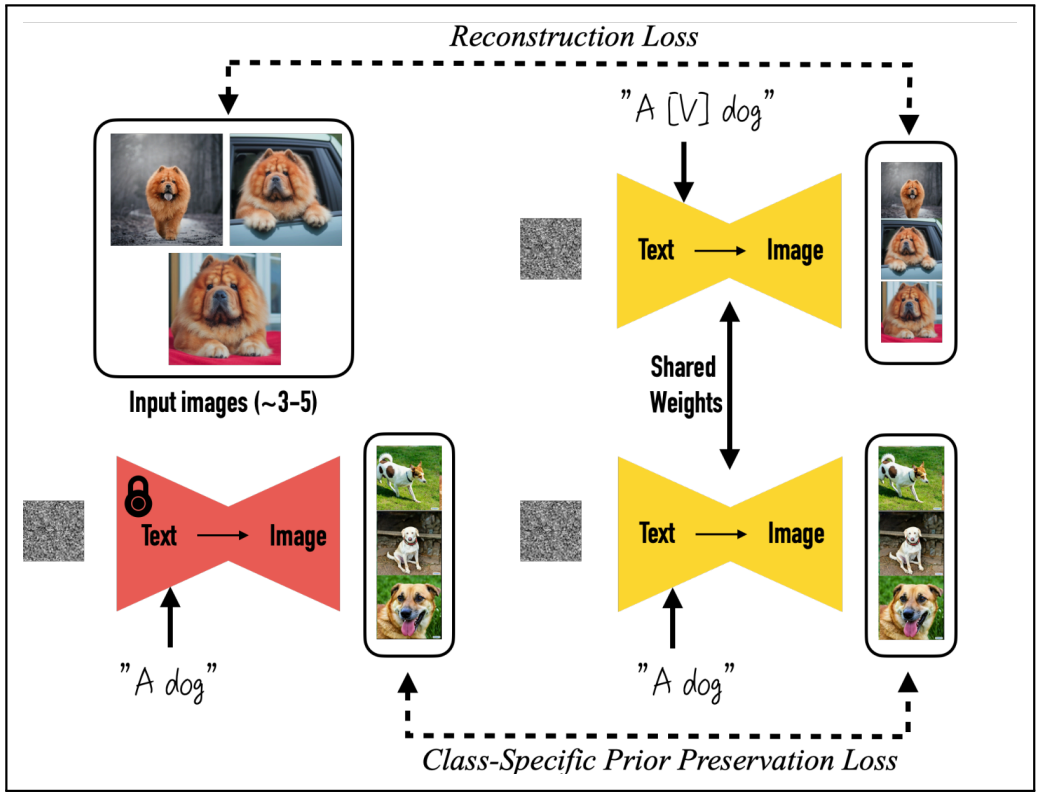
LoRA: Low-Rank Adaptation of Large Language Models[1]

- **Low-rank adaptation**: LoRA focuses on adapting a low-rank LoRA attention layer instead of fine-tuning the entire model.
- Representing weights W_i with a low-rank representation: $W_i' = W_i + \Delta W_i$, with $W_i, \Delta W_i \in \mathbb{R}^{d\times k}$, where $i \in q, k, v$ represents the query/key/value in the self-attention module.
- LoRA attention weights: $W_i' = W_i + B_i A_i$. Here, $B_i \in \mathbb{R}^{d\times r}$, $A_i \in \mathbb{R}^{r\times k}$, $r \ll min(d, k)$ are the low-rank factors to be learned during fine-tuning.
- Parameterization of LoRA weights



Dreambooth[4]

- Dreambooth Representation:



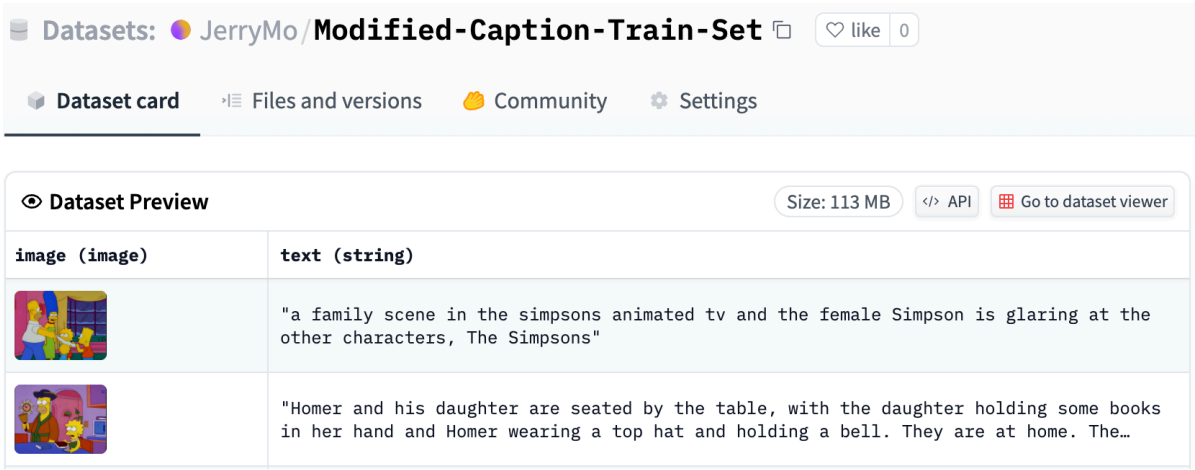
- Incorporate images of the same class to learn the subject concept
- Prior preservation loss is used to avoid overfitting and language-drift
- Loss Function:

$$\mathbb{E}_{\mathrm{x},\mathrm{c},\epsilon,\epsilon',t}[\omega_t||\hat{\mathrm{x}}_\theta(\alpha_t\mathrm{x}+\delta_t\epsilon,c)-\mathrm{x}||_2^2+\lambda\omega_{t'}||\hat{\mathrm{x}}_\theta(\alpha_{t'}\mathrm{x}_{pr}+\delta_{t'}\epsilon',c_{pr})-\mathrm{x}_{pr}||_2^2] \quad (2)$$

Data Processing

Dataset overview

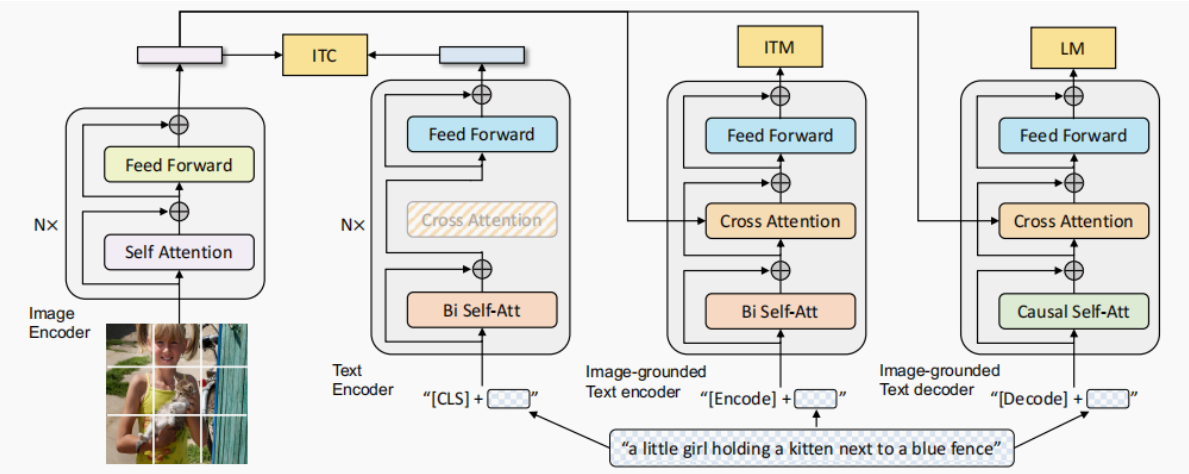
- Blip Captioned Dataset - 2500 Images.
  - Link: Datasets at Hugging Face · Cartoon-Blip-Captions
- Manually Annotated Dataset - 1000 Images.
  - Link: Datasets at Hugging Face · Modified-Caption-Train-Set
- Manually Annotated Dataset For DreamBooth - 100 Images.
  - Link: Datasets at Hugging Face · DB-Simpsons-Dataset



- Open source and free to use.

Processing - Blip Captioned Dataset

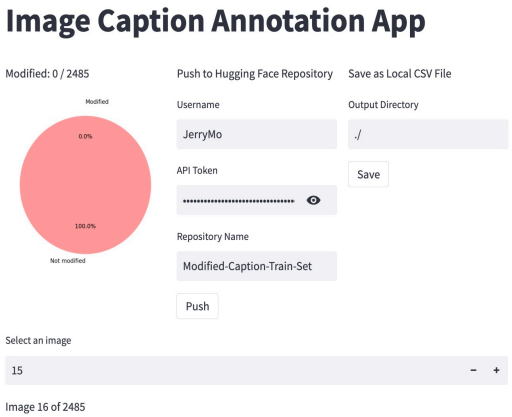
BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation [2]

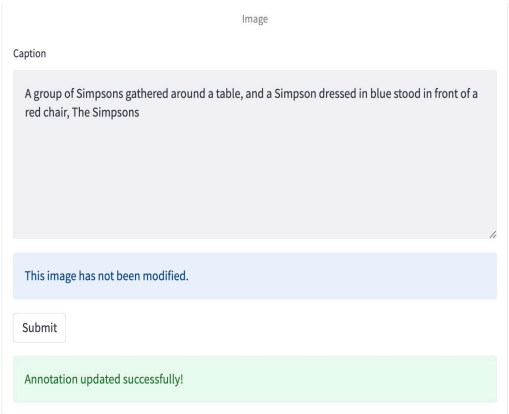


- Usage: **Image captioning**, Open-ended visual question answering, Image-text matching, Multimodal / unimodal feature extraction.
- We use BLIP model to caption **Simpsons** images from website Frinkiac.com
- **Pros**: Save time and effort compare with manually captioning.
- **Cons**: Difficult to handle complex scenarios.

Processing - Manually Annotated Dataset

- To improve the image annotation accuracy, we create a dataset with image captioned manually, based on the BLIP dataset.
  - Challenge: Unable to use annotation tools, like LabelHub or LabelBox.
  - We create an **Image Caption Annotation App** by streamlit.





- If you are interested in this app, please access the code from our GitHub repo.
  - Link: GitHub · sd-annotation-app

Processing - Manually Annotated Dataset for Dreambooth

- For Dreambooth model, we create a separate training dataset.
  - Small sample size is needed.
  - Describe the content of each image **in great detail**.
- An example image-caption from dataset:

- **Caption**: A trapezoid-shaped screen in the sky, and on the screen is a middle-aged man with grey hair. He is wearing a white shirt and a blue jacket, with a purple tie. He is looking down and has his hands open. Below him is a sea of people, all looking up at the man on the screen. The background shows a clear blue sky with white clouds, and in the distance, there is a forest. In daytime.

Model:SAM

Pipeline

- Our Model: SAM
  - Improvements
  - Results
- Comparisons with Previous Models
  - Simple Fine-tune
  - LoRA
  - Dreambooth
- Conclusion
  - Innovation
  - Limit

Remark

- Pretrained Model "Stable Diffusion v1-4":
  - Able to generate photo-realistic images.
  - Lacks ability to generate Simpson's style picture.

Figure: A family in the forest at night.

Figure: A landscape of towering icebergs in a sea, with the aurora borealis painting the sky.

- GPU: Fine-tuned on RTX3090

SAM: Point to Notice

- **Basic Method**: Dreambooth
- **Improvement1**: Utilize LoRA on Dreambooth
- **Improvement2**: Unfreeze U-Net with detailed prompts given
- **Improvement3**: Introduce negative prompts and negative examples

Basic Method: Dreambooth

- Unique identifier "Asim" - only one token.
- **Pros**: Efficient on transferring painting style with small amount of pictures required(Examples are generated after 30 pictures with 30 epochs)

Figure: A scene of Shenzhen city at sunset with buildings

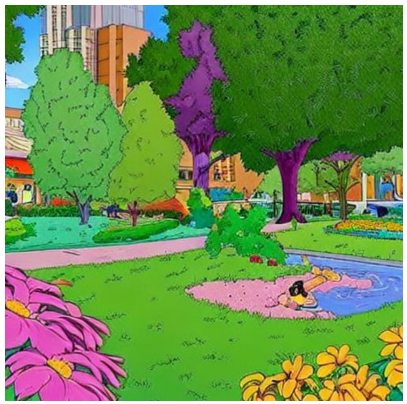


Figure: A scene of Shenzhen garden at daytime

- **Cons**:
  - Large storage space and slow training time
  - Language drifting
  - Loss of details

Based on the problems, we give our solutions by improvement1, improvement2 and improvement3.

Improvement1: Utilize LoRA on Dreambooth

Analysis on Speed and Storage Problem

- Dreambooth: Fine-tune all the parameters. Requires less pictures.
- LoRA: Fine-tune additional low rank parameters. Requires more pictures.
- Compared on 100 pictures and 80 epochs
  - Space: LoRA - 5MB Dreambooth - 5G
  - Time: LoRA - 1.3h Dreambooth - 5h

Method

- Utilize the idea of LoRA into Dreambooth.
- Freeze all the parameters in Dreambooth.
- Train the additional cross attention layer in the U-Net.

Motivation

- Reduce training time of Dreambooth while requiring less pictures then LoRA.
- "Attention is all you need".

Improvement2: Unfreeze U-Net with detailed prompts given

Analysis on Language Drifting Problem

- As training continues, the model gradually forgets the meaning of one word.
- Example

Figure: A character reading in the City at night

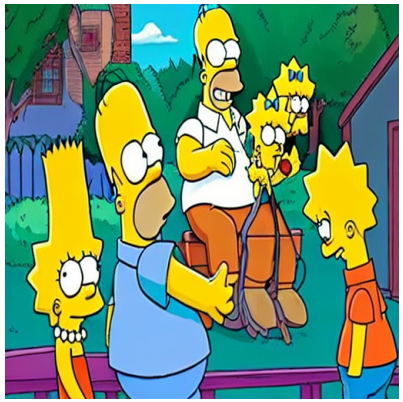


Figure: A man riding a horse in the day time

- Pretrained model forgets the meaning of face and horse.
- Generate every face as sad face. Paints two heads of two kids when the word instruction is the head of horse.
- Because of the only Prompt "Asim".

Improvement2: Unfreeze U-Net with detailed prompts given

Method

- Give detailed prompt with format to every picture.
- The start of detailed prompt should be "Asim".
- E.g."Asim. a [closeup?] of a [emotional expression] [race] [X year old] [man / woman / etc.], with [hair and makeup style], wearing [clothing style] while [doing] near [nearby objects],[outside / inside] with [objects / color] in the background,in [time period]."
- Unfreeze the last layer of U-Net.

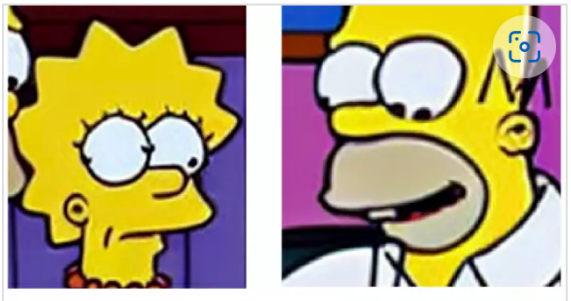
Motivation

- Detailed prompt of every picture reduces the effects of unique identifier.
- Using same and unique identifier in every prompt guarantees the consistent painting style.
- The release of last layer of U-Net enlarges the probability of generating multiple pictures giving same prompts.

Improvement3: Introduce negative prompts and examples

Analysis on Detailed Issues

- Inaccurate on detailed part painting like eyes.
- Only negative prompts help little on this issue.
- Example: (sometimes) have multiple eyeballs in one eye.



Method

- Label the bad picture with the same prompt that generated the picture plus negative [part].
- E.g. original prompt + negative eyes.
- Inference stage: prompt + no negative eyes.

Motivation

- Encourage model learn more specifically about the detail structure.
- Kind of intuition from GAN (with label as discriminator).

SAM: Evaluation Metrics

Three evaluation metrics are adopted to evaluate the pictures generated from different perspectives.

Fréchet Inception Distance

- Goal: Compare the similarity between generated picture and real picture.
- Method: Using the coding layer's vectors' distribution(assumed Gaussian) from Inception Model to calculate Fréchet Distance

$$d^2\left((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)\right) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \mathrm{Tr}\left(\boldsymbol{C} + \boldsymbol{C}_w - 2\left(\boldsymbol{C}\boldsymbol{C}_w\right)^{1/2}\right) \quad (3)$$

where $\boldsymbol{m}, \boldsymbol{C}$ are the mean and covariance of vectors from generated picture, $\boldsymbol{m}_w, \boldsymbol{C}_w$ are the mean and covariance of vectors from real picture.

- Setup:
  - Choose Inception-V3 model trained on ImageNet and pooling layer's distribution.
  - Calculate the average FID with 10 prompts with 10 images each.

SAM: Evaluation Metrics

Three evaluation metrics are adopted to evaluate the pictures generated from different perspectives.

Crossed Eyes Ratio

- Goal: Compare the negative prompts influences on crossed eyes
- Method:

$$C_{eyes} = \frac{1}{N}\Sigma_{i=1}^{N} 1(crossed\_eyes) \tag{4}$$

- Setup: Calculate the average crossed eyes ratio with 10 prompts with 10 images each.

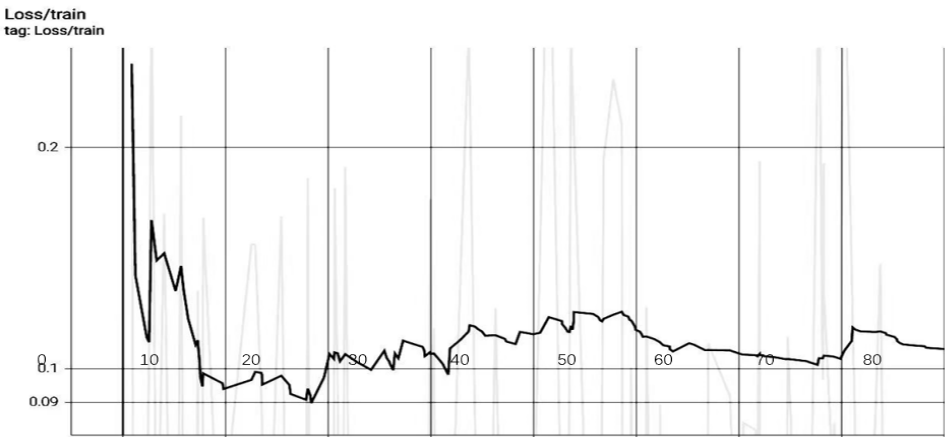
Language Drifting Measurement

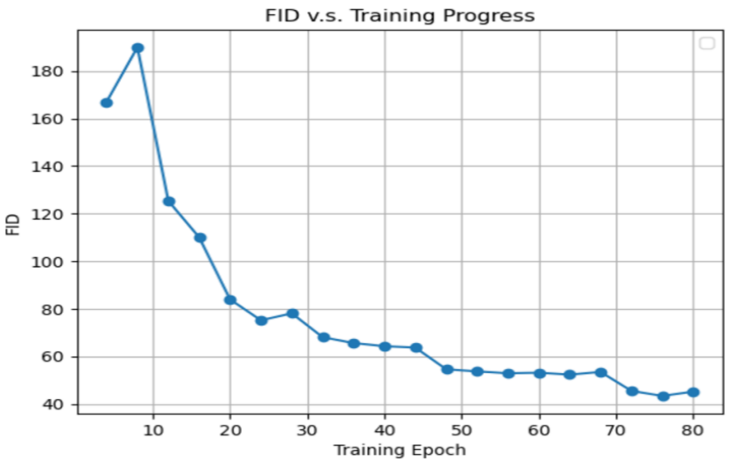
- Goal: Compare the language drifting situation after fine-tuning.
- Method: Using the pretrained text encoder and image encoder to get embedding vectors.

$$LDM = \frac{\boldsymbol{v_p} \cdot \boldsymbol{v_t}}{||\boldsymbol{v_p}||_2 ||\boldsymbol{v_t}||_2} \tag{5}$$

- Setup: Calculate average LDM with 10 prompts with 10 images each.

SAM: Training Results & Analysis





Trained on 100 pics with 80 epochs. Recorded epoch: 5, 30, 60, 70, 80

SAM: Training Results & Analysis

- The loss decreases fast at first and converges.
- The FID increases quickly at first. (unfreeze of last layer of U-Net and detailed prompts)
- The result of training loss and FID shows trained model is satisfying.
- Crossed Eye Ratio decreases from 0.58(without improvement3) to 0.32(with improvement3)
- LDM increases from 0.69(without improvement2) to 0.87(with improvement2)
- Training time: 2.1h; Storage: 1.4G

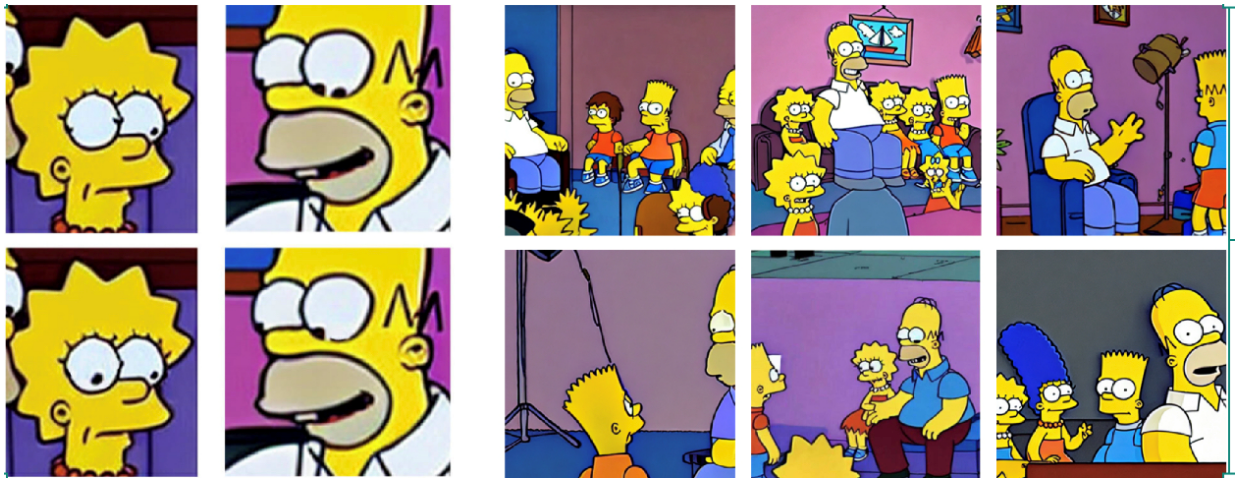


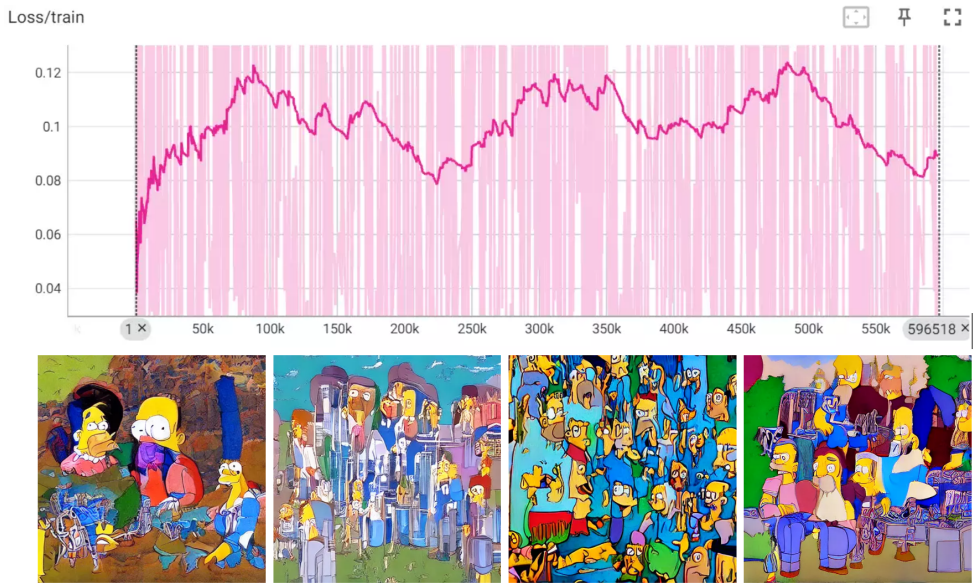
Figure: Small Examples on the Crossed Eyes Ratio and LDM

Previous Model: Simple Fine-tuning

Definition:
Directly tune all the parameters on training dataset without any techniques.

Results:

- Long training times: 6.9hours (100pics/80epoch)
- Large storage requirement: >6G/Checkpoint
- Sensitive training loss: Easy to diverge
- Catastrophic Forgetting Phenomenon



Training Loss and Generated Picture

Comparisons:

- SAM requires less time and storage space.
- SAM are more easily to tune with significantly small FID.

Previous Model: LoRA

- **Results A**:
  - Short training time: 1.3 hours (100pics/80epochs)
  - Small storage requirement: 5MB
  - Requires many pictures (1000pics)
  - Relatively high quality: Average FID - 67.92
- **Results B**:
  - Unable to draw Simpson style on prompts that not appear in the training data
  - Simply paste the cartoon character in the scene

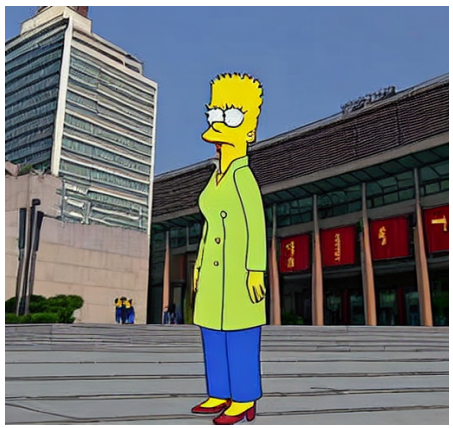


Figure: A woman stand by the Chinese National Congress

Figure: A solider with gun stand in trench in daytime

Figure: A man stand in the downtown of Shanghai

Previous Model: LoRA

Comparison on SAM

- **Result A**:
  - SAM takes more time given same training steps. (less pics)
  - SAM requires more storage space.
  - SAM has higher quality: Average FID - 42.63.
- **Result B**:
  - SAM could draw Simpson style on prompts that not appear in training data.

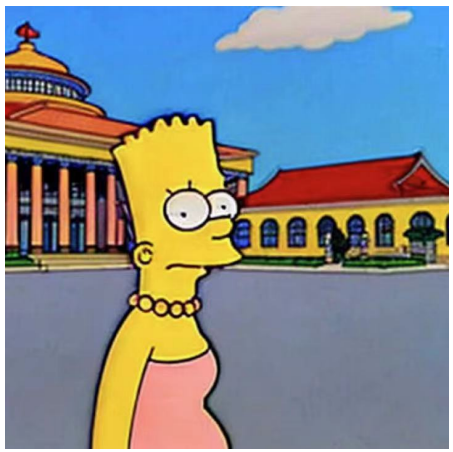


Figure: A woman stand by the Chinese National Congress

Figure: A solider with gun stand in trench in daytime

Figure: A man stand in the downtown of Shanghai

Previous Model: Dreambooth

- **Result A**:
  - Nice style painting
  - Require less data and epoch: 30 pics+30 epoch
  - Medium training time: 5 hours (100pics/80epoch)
- **Result B**:
  - Easily overfit: Every person generated has same facial expression.
  - Language drift: Easy to forget the pre-trained word's meaning- Similarity: 0.69.

Figure: A character reading in the City at night

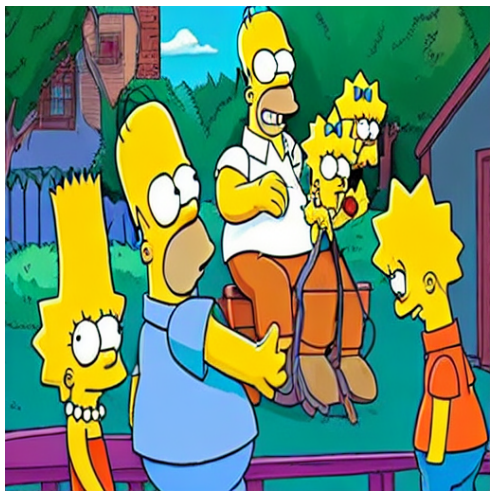


Figure: A man riding a horse in the day time

Previous Model: Dreambooth

Comparison on SAM

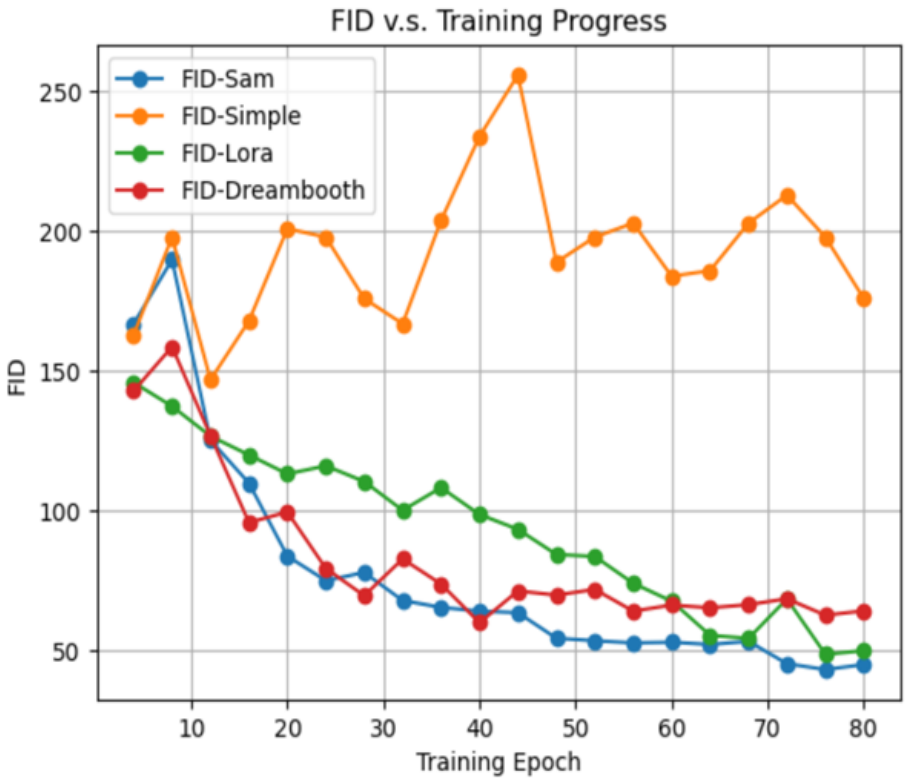
- **Results A**:
  - SAM requires more pictures and more detailed prompts input.
  - SAM takes less time given same training steps: 2.1h(100 pictures+80 epoch). (more pics)
- **Results B**:
  - Alleviate over-fitting: Generate various facial expression given same prompts.
  - Alleviate language drifting: Similarity-0.87, increase about 18%.

Figure: A character reading in the City at night

Figure: A man riding a horse in the day time

Total Comparison

- Compare four models using three evaluation metrics.



Models	Crossed Eye Ratio	LDM
Simple-tune	Nan	0.03
Dreambooth	58.7%	0.69
Lora	46.2%	0.81
SAM	32.9%	0.87

Conclusion

Innovation in SAM

- Combine LoRA method and dreambooth method to accelerate training rate and save storage space.
- Utilize detailed prompts with unfreezing last layer of U-Net to alleviate language drifting issue.
- Introduce negative prompts and negative pictures to deal with crossed eyes problem.
- Our work may shed lights on future fine-tuning work, especially in the area of text2image.

Limits and Future

- Fail to develop a systematic way to solve the detail issue.
  - There may exists some regularization techniques.
  - More general solutions are needed.
- Time and storage requirement could be deduced more.
- Fail to find a efficient way to deal with generating text correctly.

Thanks for Listening!

Figure: Wishes To You from Our Simpson Model!

[1] 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.

[2] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

[3] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. 2022 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2022.

[4] 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. *arXiv preprint arXiv:2208.12242*, 2022.