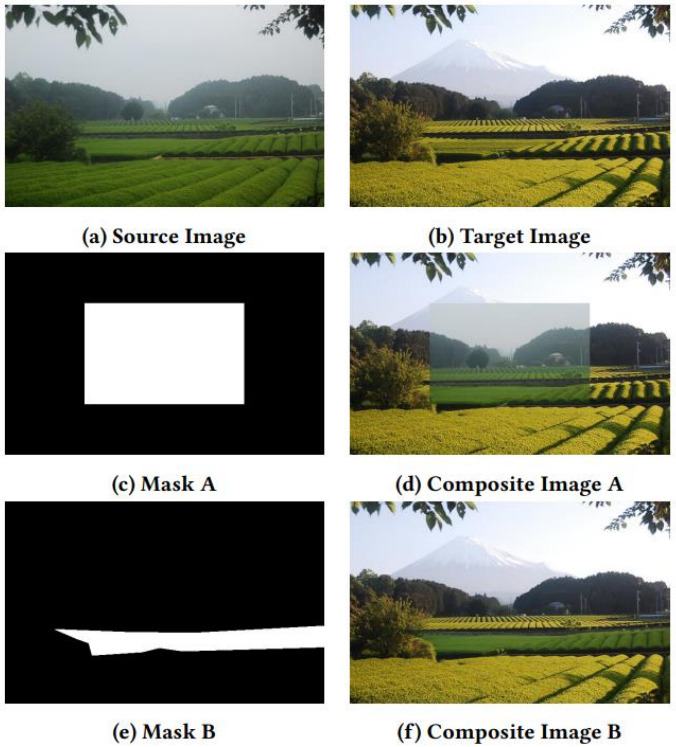
DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., & Aberman, K. (2023). Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 22500-22510).

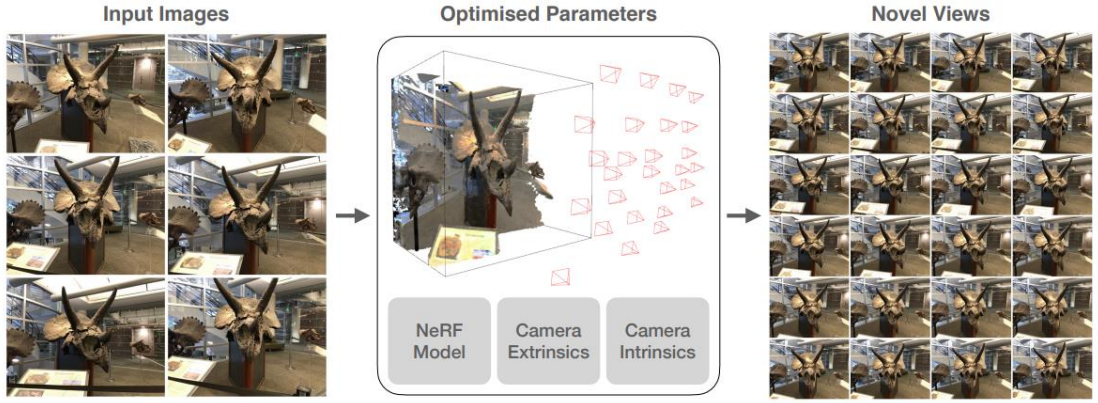
Related Work – Image composition

- Given subject new background[1]



- 물체와 배경 이미지들을 자연스럽게 자동으로 합성하는 모델
- 한계 : 입력한 모습으로만 합성 가능

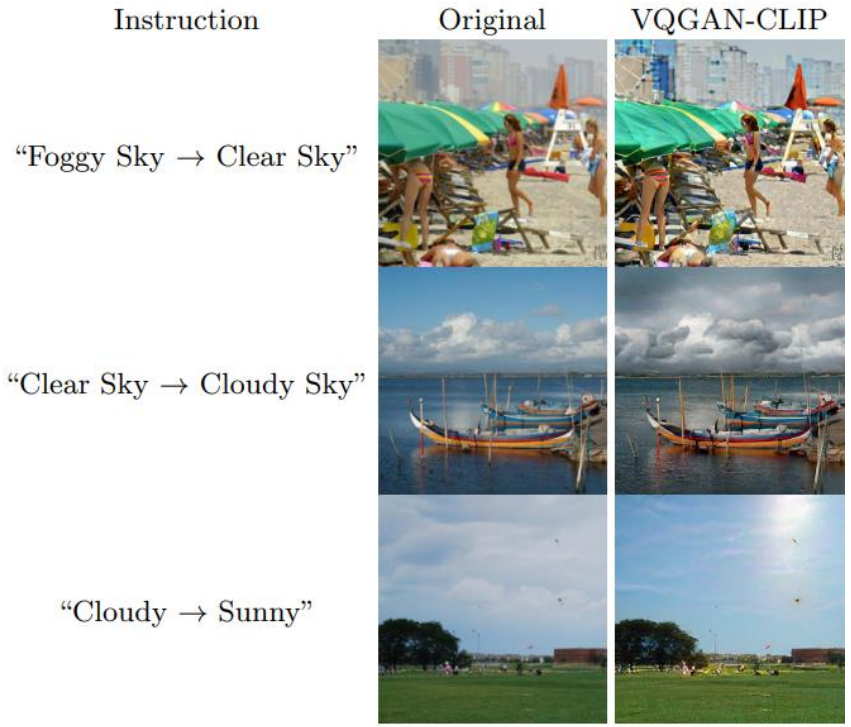
- 3D reconstruction[2]



- 다양한 방향의 이미지들을 통해 새로운 구도의 이미지를 만들어내는 모델
- 한계 : 일단 다양한 방향의 이미지들이 필요하며 배경이 기존 배경으로 고정됨

Related Work - Text-to-Image Editing and Synthesis

- GAN based Models[3]



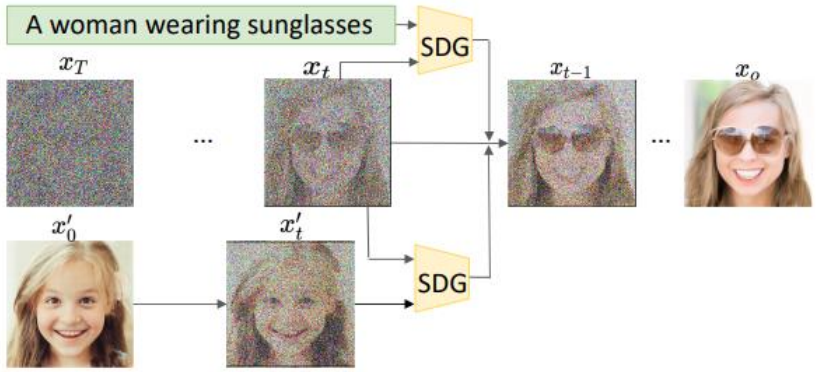
- VQGAN이 생성한 이미지의 embedding이 CLIP과 닮아가도록 학습하는 방법
- 텍스트를 활용해 실제 사람이 한듯한 편집 능력을 보여줌
- 하지만 다양한 객체들을 프롬프트로 입력하는 경우 성능이 급격히 떨어짐(단순한 변환만 가능)

- Diffusion based model[4]

- CLIP의 Text Embedding을 활용하여 이미지 생성
- 대규모 VLM 모델을 사용하기에, 텍스트를 활용해 처음 보는 다양한 데이터셋에서도 좋은 성능을 보임
- 하지만 정교한 수정이 어려우며 프롬프트로만 편집을 진행해야함

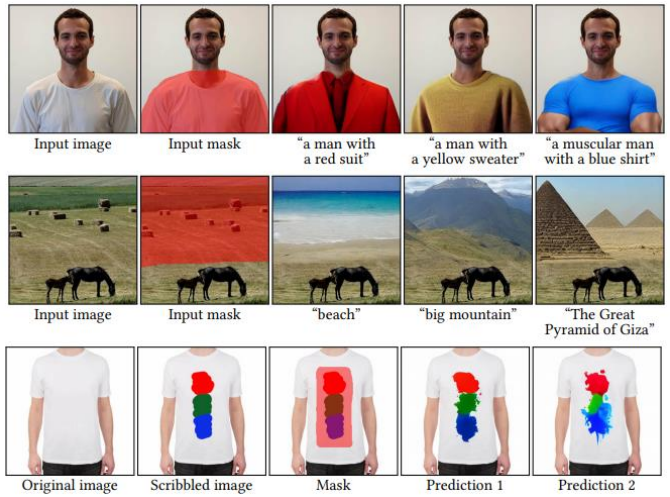
Related Work – Controllable Generative Models

- Text를 활용 [5]



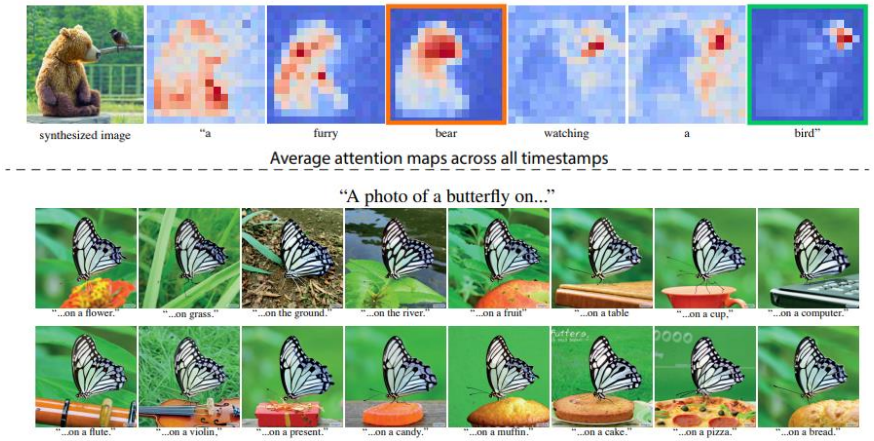
- Diffusion Network에서는 이미지를 깎기 위해 semantic 정보를 prompt로 입력
- Text semantic 정보만으로 이미지 수정이 이루어지면 원래 subject들의 특징이 변화됨

- Text & Mask 활용[6]



- 수정하고 싶은 부분만 mask를 하고 mask 부분이 자연스러워지도록 이미지를 깎는 방법
- Mask 내부의 structure, content를 고려하지 않는 문제가 발생

- Text & Attention Mask[7]

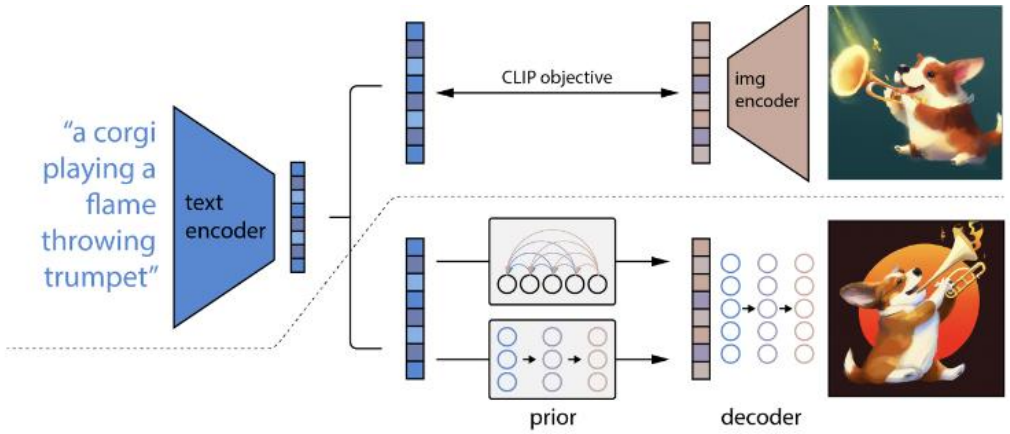


- 단어와 매칭되는 attention map을 활용해 prompt의 단어들이 어떤 local에서 주요하게 작용하는지 확인 가능
- 이를 활용해 이미지와 매칭되는 부분에 대해 세밀하게 수정 가능

3가지 방법 모두 기존의 이미지를 수정 하는 것에 포커스

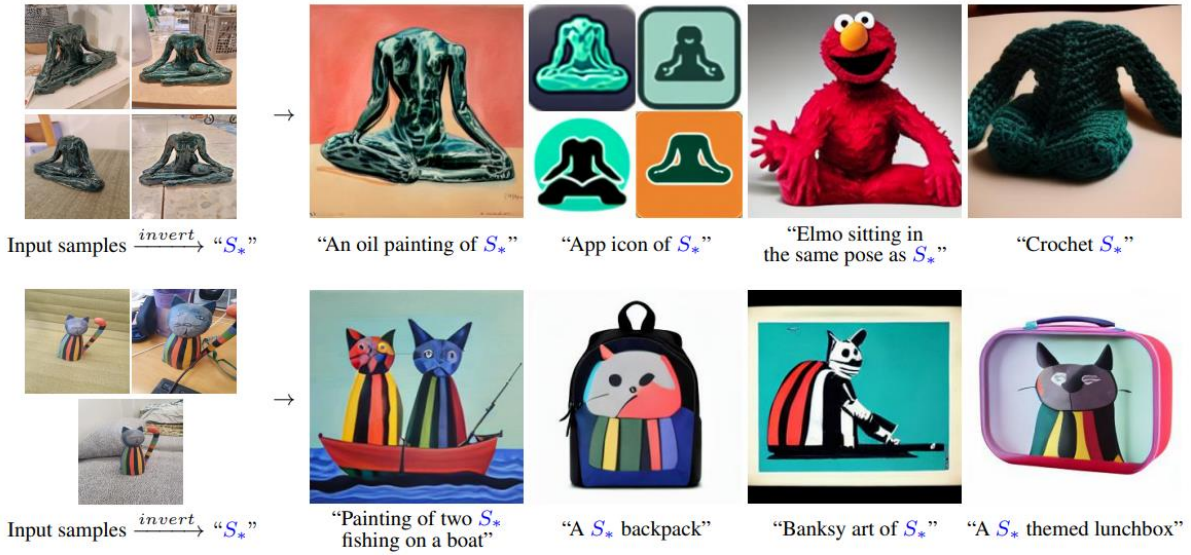
Related Work – Controllable Generative Models

- Inversion[4]



- 특정 latent vector와 유사하게 생성하는 방법
- Pretrained CLIP의 embedding을 바탕으로 학습 및 생성

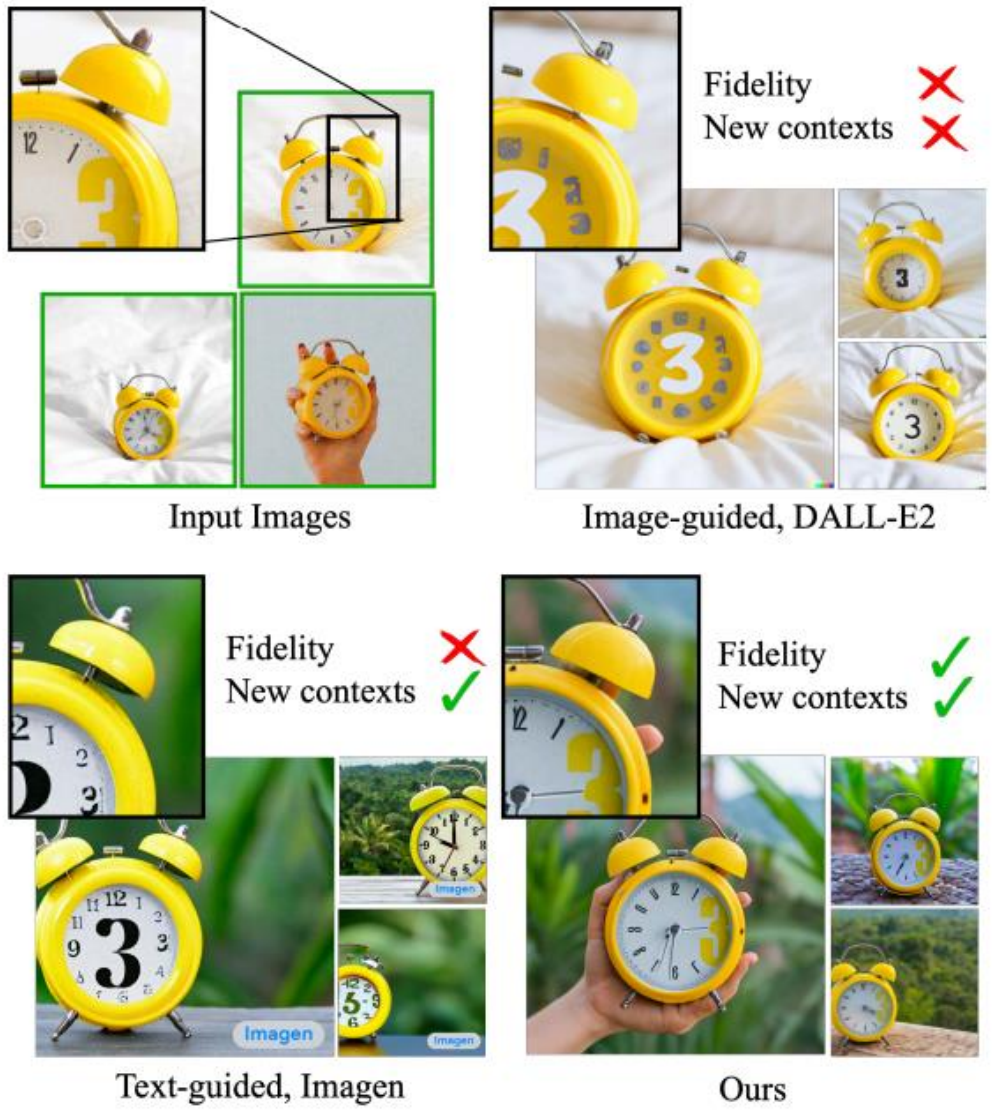
- 새로운 학습용 토큰을 Diffusion에게 제공[8]



- Text to Image model을 Freeze 하고 text encoder 만 튜닝
- 새로운 토큰을 새로운 물체나 style로 치환하여 프롬프트 러닝 진행
- 그림처럼 초록 가부좌를 프롬프트로 쭉 설명하는 것이 아닌 S* 로 치환하여 학습 & style transfer
- Diffusion model 자체를 학습하는 것이 아니기에 다양성이 떨어진다 주장

Introduction

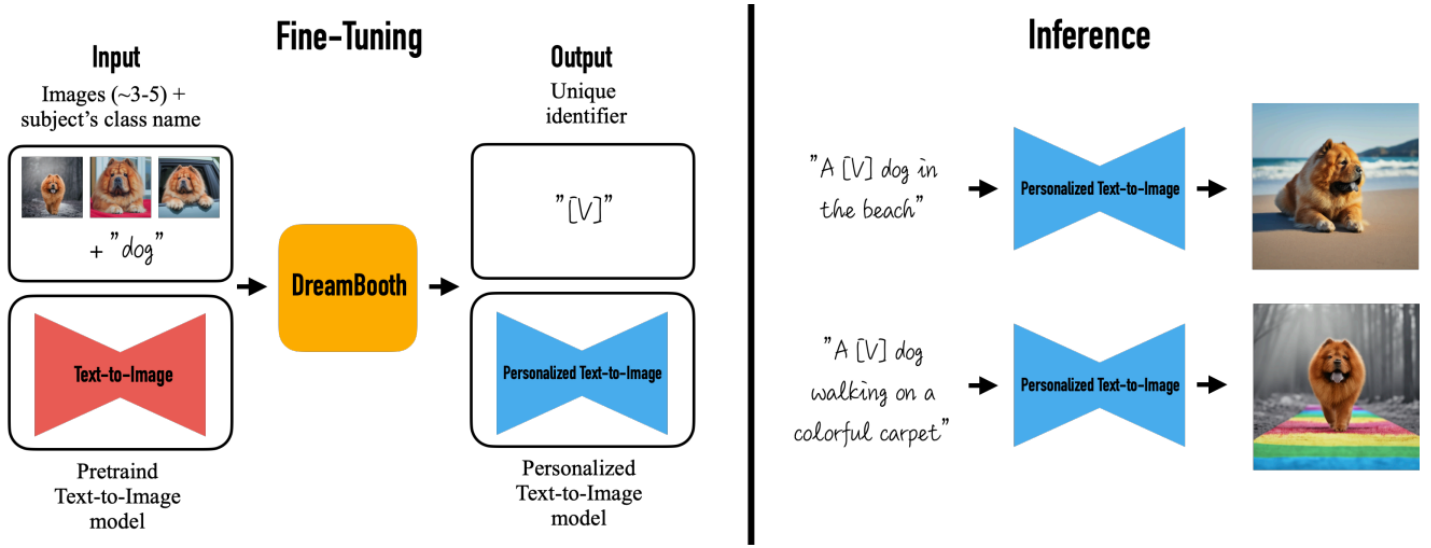
- Stable Diffusion 등장 이후 Diffusion 모델에 대한 관심 증가
- 해당 모델은 좋은 성능을 보이지만, “특정한 Object”를 지속적으로 생성하게 하는 것은 제한되어 있음
  - ❖ 오른쪽 그림과 같이 해당 시계를 input으로 넣었을 때, DALL-E2는 새로운 노란 시계를 그리고 있음
- 이를 해결하기 위해 여러 연구가 진행되었지만 input의 특징을 그대로 유지하거나, 자연스러운 배경을 생성에 어려움 겪음
  - ❖ 오른쪽 그림과 같이 해당 시계를 input으로 넣었을 때, Imagen(사전연구)는 그대로 노란 시계를 그리지만 원 시계가 짤린 형태이며, input object를 고려하여 배경을 생성하고 있지는 않음
- 모델을 personalization(Fine-tunning)을 하기에는 많은 데이터를 필요로 함



Method

제안하는 모델(DreamBooth)은

- 진정한 “personalization”을 제공 : input object 그 자체를 새로운 token sequence 와 연결지어 Diffusion model이 인식하도록 설정 (A [V] dog)
- Text 설명이 없는 단순 3~5개의 이미지만 사용하여 Fine-tunning
- 새로운 loss를 도입해 Fine-tunning시 발생하는 문제 해결
- Diffusion model처럼 범용성을 지닌 방법론



Method

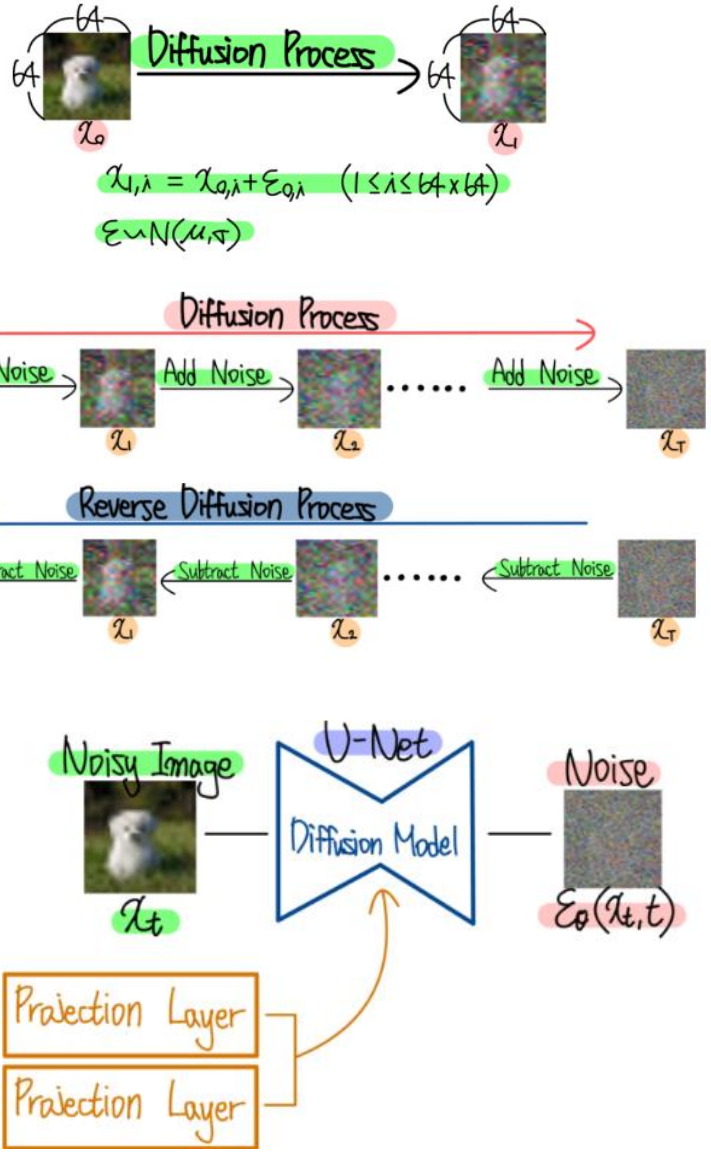
제안하는 모델은 Diffusion 모델(DDPM) 베이스[9]

- DDPM은 매 step마다 각 픽셀마다 첨가된 Noise를 예측하는 모델
- Forward Diffusion Process :이미지에 고정된 정규 분포로 생성된 Noise가 더해지는 과정
- Reverse Diffusion Process : 생성된 Noise를 이미지에서 제거하여 입력 이미지와 유사한 확률 분포로 만드는 과정
- Input : 임의의 Noisy가 추가된 이미지, 몇 번째 프로세스 인지, 혹은 prompt, 특정 클래스 정보 등도 가능
- Loss는 실제 노이즈와 Diffusion Model이 예측한 Noise가 같아지도록 하는 목표함수

실제 Noise

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]$$

모델이 예측한 Noise

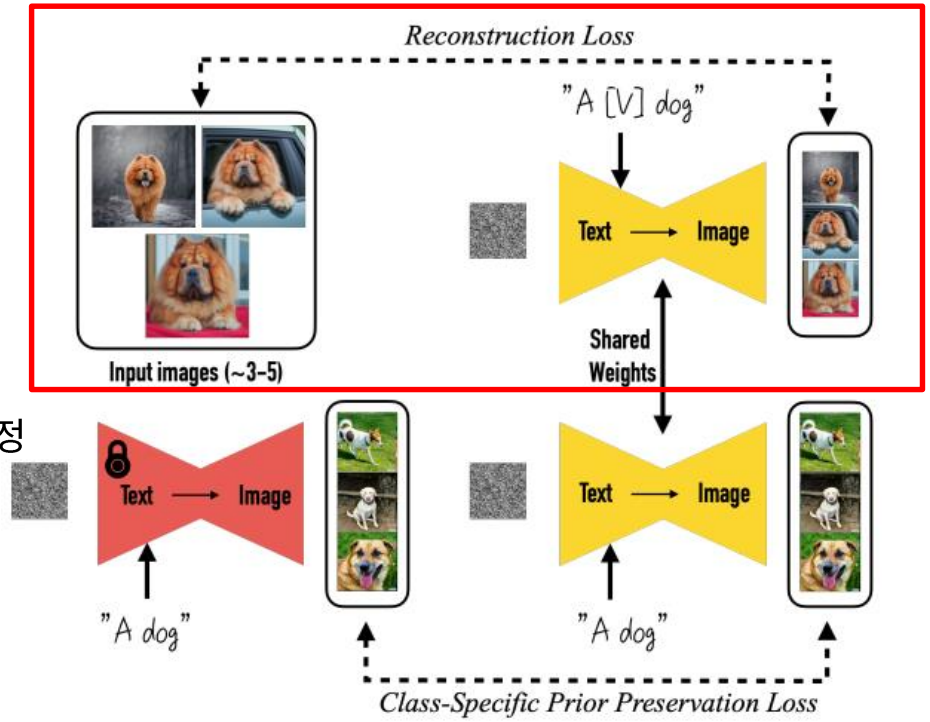


Method

적은 데이터셋 만으로 fine-tunning을 진행하기 위해 다음과 같은 과정을 진행

- Input 이미지와 “A [Identifier] [class noun]” 형태의 dictionary 형태로 재학습 진행
- Identifier는 3-4개 토큰으로 쪼개지는 말이 안되는 이상한 단어를 사용(Rare-token Identifier)
  - 스페이스 없이, 쭉 이어쓴 랜덤유니코드3개 문자가 합쳐진 문자열(xxy5syt00)
  - T5-XXL 토크나이저 기준 대략 5000번대~1000번대 토큰들을 사용
  - 즉, input objec와 mapping 할 “신규 토큰 " 을 넣어주되, 해당 토큰은 실제 사용되지 않아 기존에 학습한 language 임베딩에 영향을 주지 않아야 함
- 하지만 해당 방법만 사용한 경우,
  1. language drift 발생 : large text corpu에 대해 학습한 언어 모델이 이후 특정 task를 위해 fine-tunned 될 때 기존에 학습했던 syntactic & semantic knowledge 를 잊음
  2. 기존 class noun에 대해 학습된 특정 instance 만 생성하는 다양성 감소

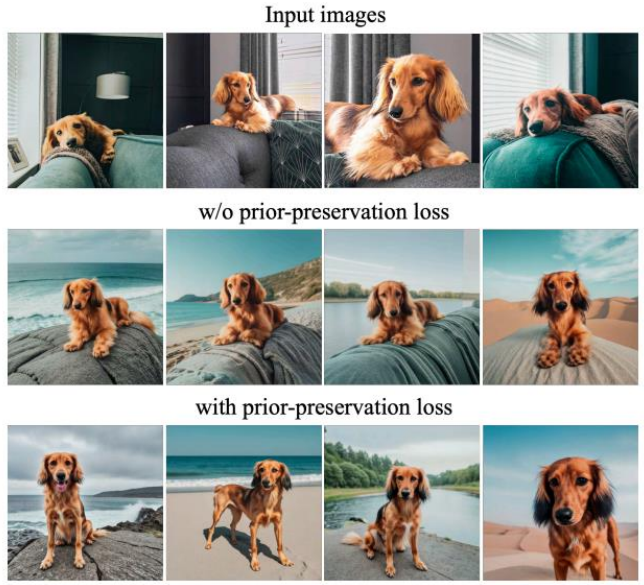
Designing Prompts for Few-Shot Personalization



Method

앞선 문제를 해결하기 위해 Class-specific prior preservation loss 도입

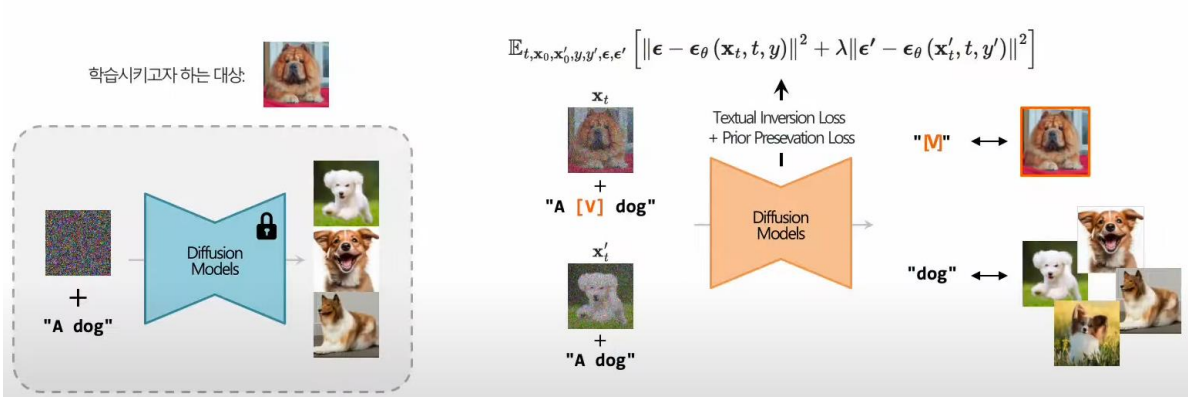
- 목적 : class의 정보를 잊지 않기 위해 class의 samples만으로 학습한 loss 추가
- Fine tunning 할 이미지에 Diffusion model이 만들어낸 class noun 이미지들을 학습 데이터로 추가
- 모델로 만들어낸 이미지들을 다시 학습에 사용해, class에 대한 지식을 유지시키며 새로 fine-tunning 하는 부분에 대해서 과하게 포커스가 가는 것을 방지



기존의 Diffusion Loss

$$\mathbb{E}_{\mathbf{x},\mathbf{c},\boldsymbol{\epsilon},\boldsymbol{\epsilon}',t}[w_t \|\hat{\mathbf{x}}_\theta(\alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}, \mathbf{c}) - \mathbf{x}\|_2^2 + \lambda w_{t'} \|\hat{\mathbf{x}}_\theta(\alpha_{t'} \mathbf{x}_{\text{pr}} + \sigma_{t'} \boldsymbol{\epsilon}', \mathbf{c}_{\text{pr}}) - \mathbf{x}_{\text{pr}}\|_2^2], \quad (2)$$

Class-specific prior preservation loss



Experiment

Dataset

- 30 subjects, including unique objects and pets such as backpacks, stuffed animals, dogs, cats, sunglasses, cartoons, etc
- 25 evaluation prompts
  - 10 recontextualization(prompt를 바꿨어도 이를 활용해 이미지가 바뀌는지 확인)
  - 10 accessorization(기존 이미지에 액세서리를 추가하는 prompt)
  - 5 property modification prompts for live subjects/pets(기존 이미지에서 대상을 바꾸는 prompt)
- CLIP과 DINO를 평가 metric(subject fidelity, prompt fidelity)으로 사용
  - CLIP-I[10] : 생성된 이미지와 실제 이미지 간의 CLIP 임베딩의 평균 코사인 유사도를 측정해 얼마나 유사한지 측정
  - CLIP-T[10] : 프롬프트와 이미지 간의 CLIP 임베딩의 평균 코사인 유사도 측정
  - DINO[11] : 생성된 이미지와 실제 이미지 간의 ViT-S/16 DINO 임베딩의 평균 코사인 유사도를 측정

Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```

Experiment

- 기존 연구와 비교해서 딥러닝 모델 및 실제 사람이 비교한 결과 제안하는 방법(Stable Diffusion에 Dreambooth로 튜닝)이 가장 좋은 성능을 보이고 있음

Method	DINO ↑	CLIP-I ↑	CLIP-T ↑
Real Images	0.774	0.885	N/A
DreamBooth (Imagen)	**0.696**	**0.812**	**0.306**
DreamBooth (Stable Diffusion)	0.668	0.803	0.305
Textual Inversion (Stable Diffusion)	0.569	0.780	0.255

Table 1. Subject fidelity (DINO, CLIP-I) and prompt fidelity (CLIP-T, CLIP-T-L) quantitative metric comparison.

Method	Subject Fidelity ↑	Prompt Fidelity ↑
DreamBooth (Stable Diffusion)	**68%**	**81%**
Textual Inversion (Stable Diffusion)	22%	12%
Undecided	10%	7%

Table 2. Subject fidelity and prompt fidelity user preference.

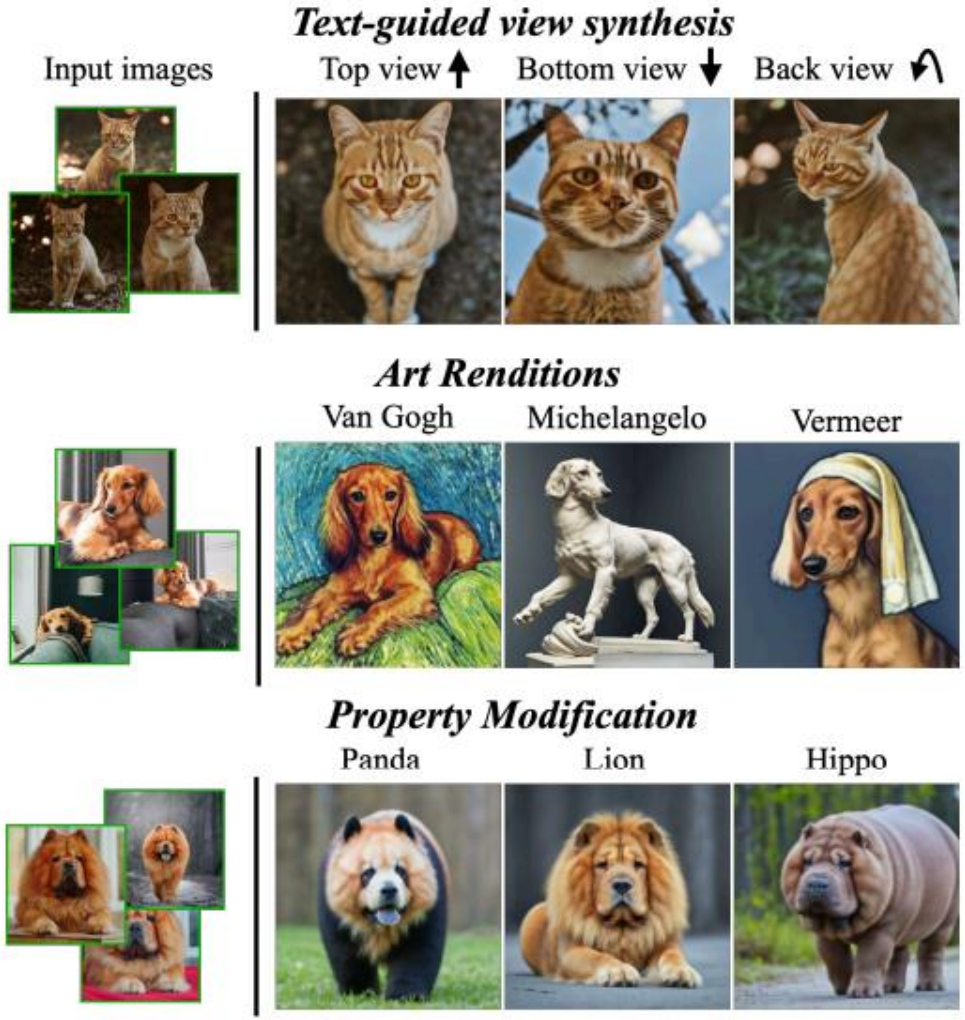
Figure 4. **Comparisons with Textual Inversion [20]** Given 4 input images (top row), we compare: DreamBooth Imagen (2nd row), DreamBooth Stable Diffusion (3rd row), Textual Inversion (bottom row). Output images were created with the following prompts (left to right): "a [V] vase in the snow", "a [V] vase on the beach", "a [V] vase in the jungle", "a [V] vase with the Eiffel Tower in the background". DreamBooth is stronger in both subject and prompt fidelity.

Application

- recontextualization(prompt를 바꿨어도 이를 활용해 이미지가 바뀌는지 확인) 확인
  - 다양한 배경에 대해 생성 가능
  - 프롬프트에 어울리는 이미지를 재구성(차주전자로 차를 따르는 이미지)

Application

- 그 외 확인
  - Text guide view synthesis(특정 object의 unseen view를 자연스럽게 잘 생성할 수 있는지)
  - Art Rendition(화풍을 복사할 수 있는지)
  - Property modification(기존 이미지에서 대상을 바꾸는 prompt)



Limitation

- 그림에서 표현된 한계점
  - 프롬프트 맥락과 다른 이미지 생성
    - ❖ contex에 대해 weak prior가 있는 경우
    - ❖ Object 유지와 배경 생성을 동시에 하는 경우
  - 입력 이미지의 특성이 유지되지 않음
  - 입력 데이터에 오버피팅 되어 비슷한 이미지 생성

- 추가적인 한계점
  - Diffusion 모델이 잘 알고 있는 object에 영향이 큼
    - ❖ 몇몇 대상이 다른 대상에 비해 학습이 빠름(개, 고양이)
    - ❖ 희귀한 입력 object들에 대해서 불가능
  - 모델 prior에 기반해 생기는 hallucination이 발생
    - ❖ 독특한 prompt를 입력해도 일반적인 모습으로 회귀
    - ❖ 프롬프트가 복잡하고 추상적일수록 해석 능력이 떨어짐

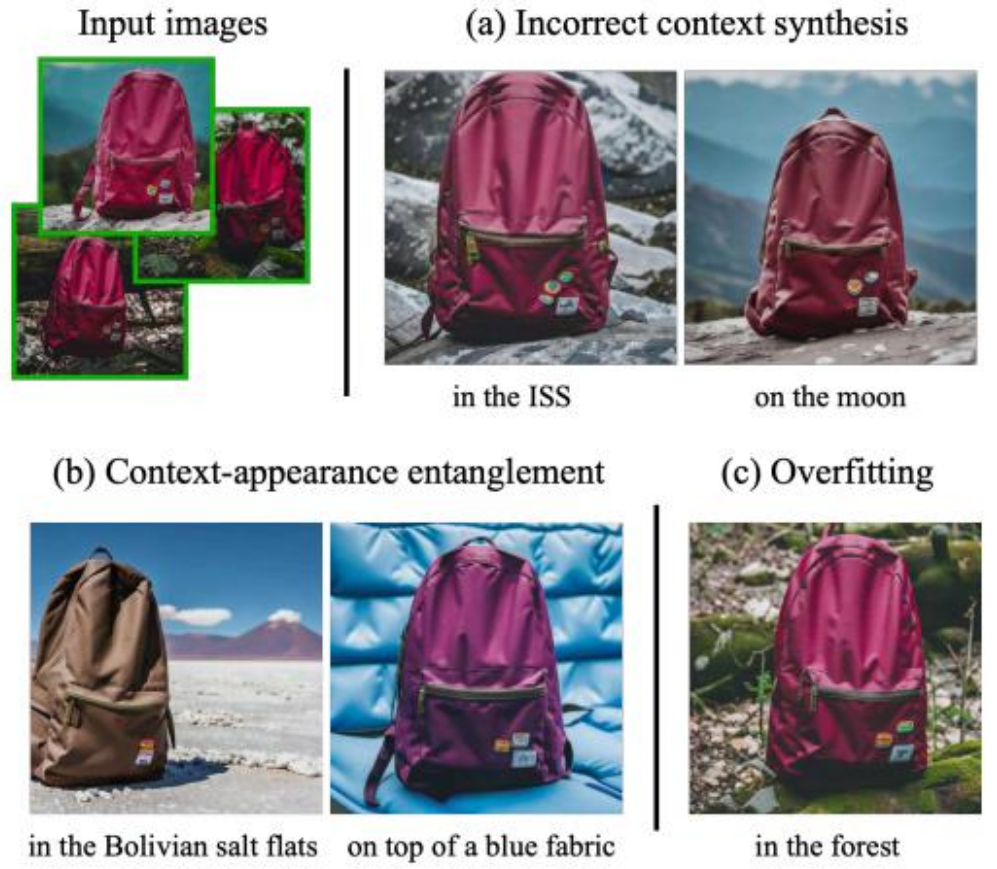


Figure 8. **Failure modes.** Given a rare prompted context the model might fail at generating the correct environment (a). It is possible for context and subject appearance to become entangled (b). Finally, it is possible for the model to overfit and generate images similar to the training set, especially if prompts reflect the original environment of the training set (c).

Reference

1. Wu, H., Zheng, S., Zhang, J., & Huang, K. (2019, October). Gp-gan: Towards realistic high-resolution image blending. In Proceedings of the 27th ACM international conference on multimedia (pp. 2487-2495).
2. Wang, Z., Wu, S., Xie, W., Chen, M., & Prisacariu, V. A. (2021). NeRF--: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064.
3. Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., & Raff, E. (2022, October). Vqgan-clip: Open domain image generation and editing with natural language guidance. In European Conference on Computer Vision (pp. 88-105). Cham: Springer Nature Switzerland.
4. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125, 1(2), 3.
5. Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More control for free! image synthesis with semantic diffusion guidance. 2021.
6. Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. arXiv preprint arXiv:2206.02779, 2022.
7. Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626, 2022.
8. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., & Cohen-Or, D. (2022). An image is worth one word: Personalizing text-to-image generation using textual inversion. arXiv preprint arXiv:2208.01618.
9. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. Advances in neural information processing systems, 33, 6840-6851.
10. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763). PMLR.
11. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 9650-9660).