Visually Guided Object Insertion Into Image

Adi Tsach & Aya Spira

Supervised by: Noam Rotstein & Roy Gantz

Table of contents

1

Introduction

Background

- Image editing has advanced significantly with the introduction of **text-conditioned** diffusion models such as DALL-E and Stable Diffusion.
- However, the reliance on textual descriptions can sometimes limit their adaptability in scenarios requiring more precise control or specific visual modifications.
- There are some techniques for **image-conditioned** diffusion models but they require user-provided input masks to designate the insertion point of an object within an image.

image-conditioned
(Paint By Example)

original image

mask

object image

text-conditioned
(Paint By Inpaint)

textual prompt

Motivation

- In our project, we leveraged the robust capabilities of a state-of-the-art text-conditioned diffusion model, eliminating the need for manual marking by replacing the textual input with an object image as the conditioning mechanism.
- This approach retains the model's generative strengths and realistic outputs while broadening its applicability to scenarios where a visual example is more intuitive or descriptive than text.

2

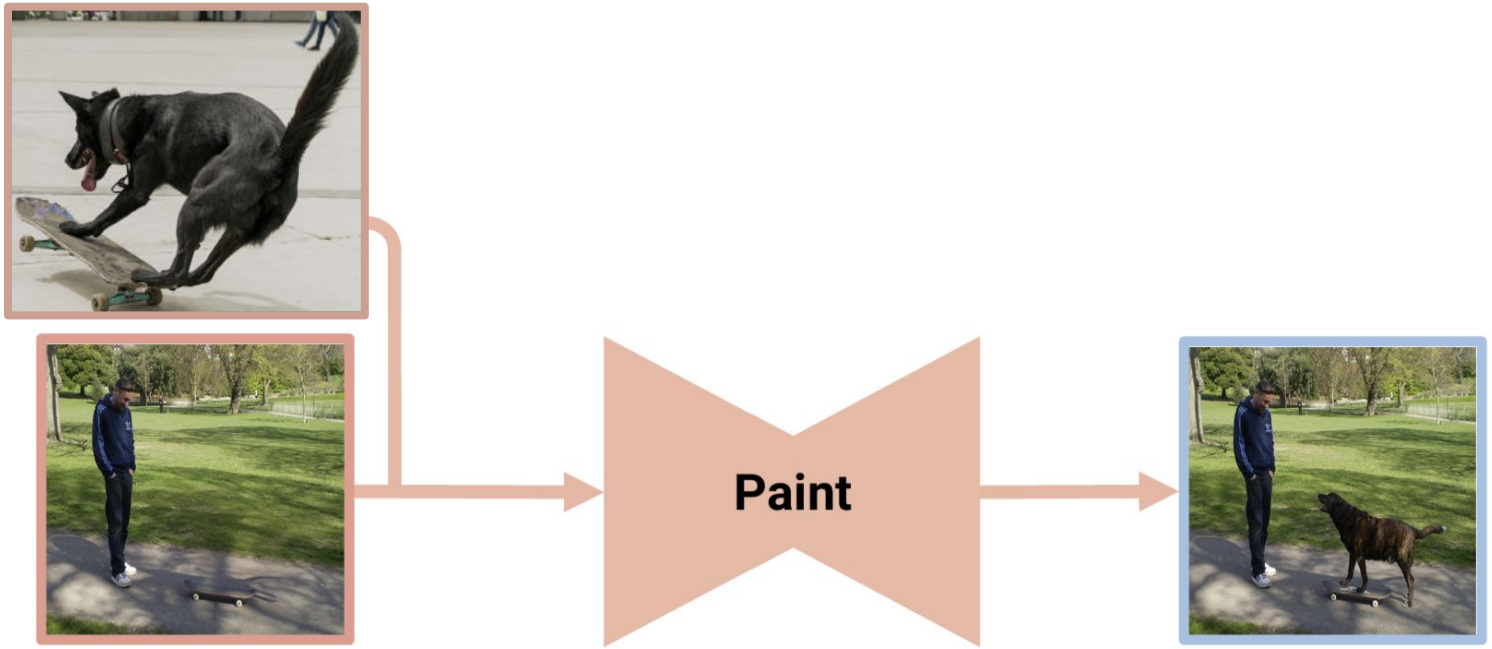
Project Goals

Project Goals

The primary objectives of our project:

1. Adapt a pretrained Model for Image-Based Conditioning.
2. Ensure High-Quality Integration of outputs.

Our Goal



Maybe more realistic goal

Pretrained text conditioned model performance – instruct_pix2pix

Prompt

“Add a dog”

Original_Image

Target_Image

3

Workflow

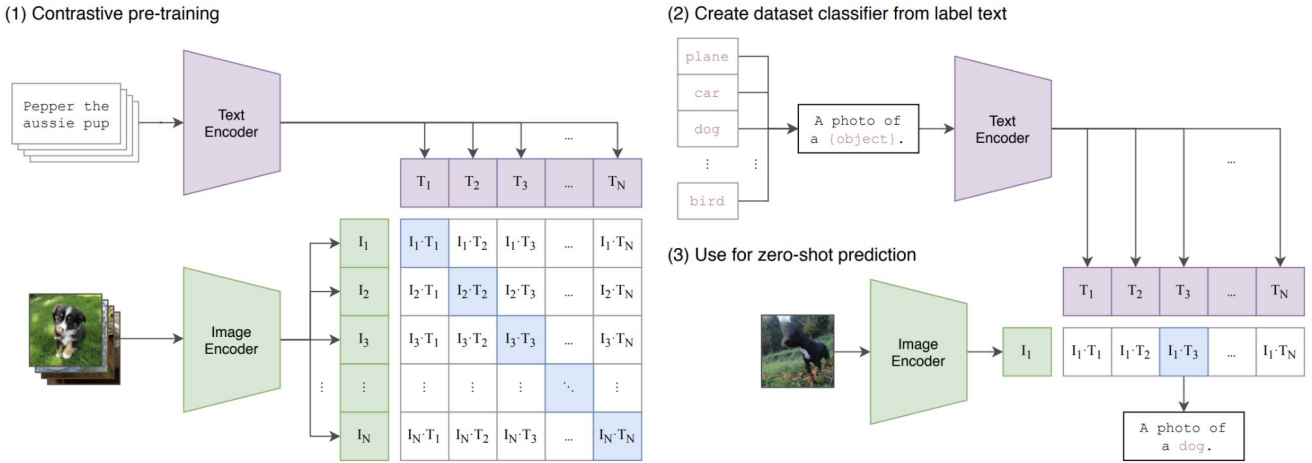
Existing Models

Paint by Inpaint: ([2404.18212] Paint by Inpaint: Learning to Add Image Objects by Removing Them First) - Text-conditioned diffusion models are designed to produce high-quality, realistic images. This approach aligns with our project's goal of generating logical results, ensuring objects are placed in a realistic size and position within the image.

Paint by Example: ([2211.13227] Paint by Example: Exemplar-based Image Editing with Diffusion Models) - Image editing based on an object image and a mask indicating where to insert the object. This methodology somewhat aligns with our project's goal of visual guidance rather than text-based input.

Existing Models

CLIP Image Encoder: The *CLIPVisionModel* is utilized for extracting high-quality visual embeddings from the object image. These embeddings are supposed to match the dimensionality of those produced by the *CLIPTextModel*, which are used as inputs to the U-Nets of text-conditioned diffusion models.



Existing Models

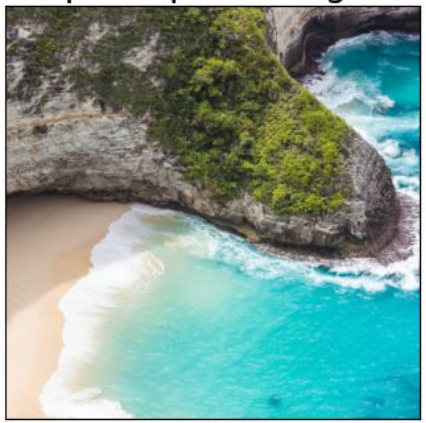
IP_Adapter: The IP Adapter is an newly addition to pix2pix model. It is a image+text based conditioning module designed to enable edits that are guided by both visual and textual prompts.

prompt

"best quality, high quality"

input image

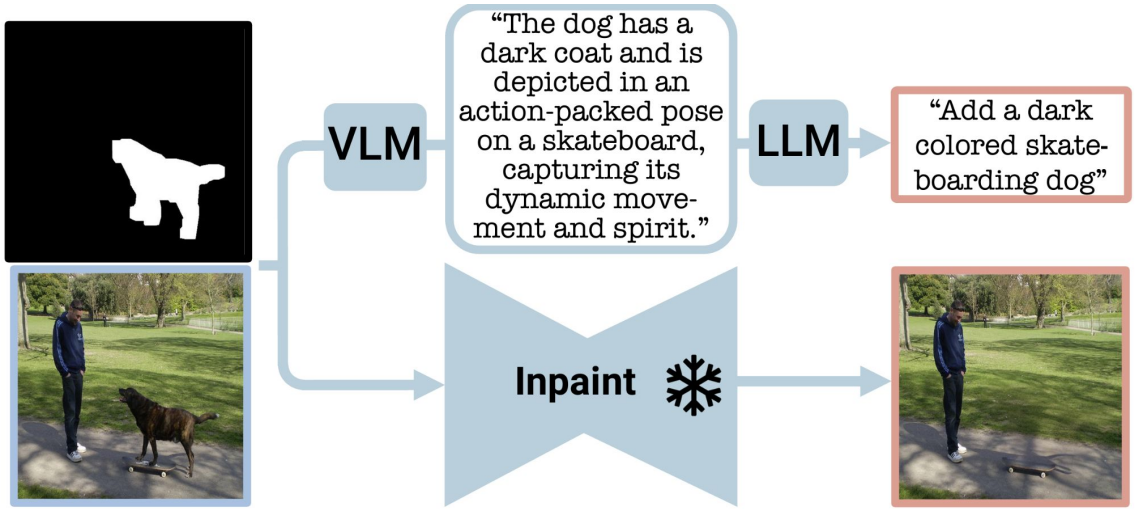
ip-adapter image

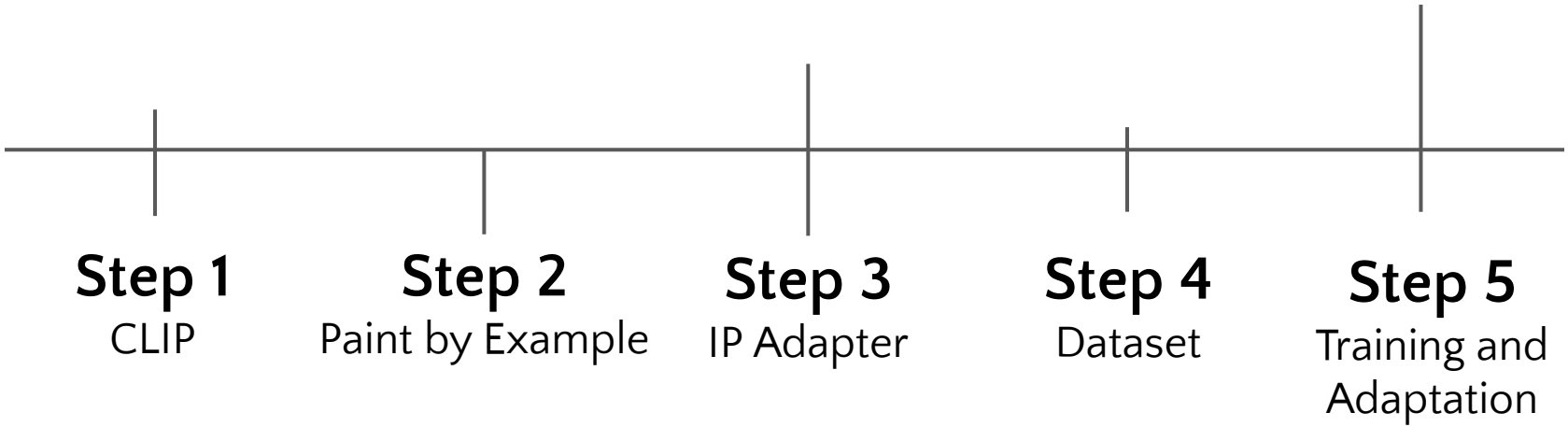


result

Existing Datasets

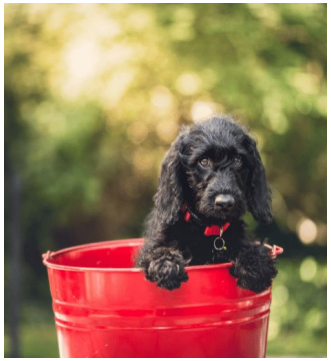
PIPE (Text-Guided Image Editing): A large-scale dataset for mask-free, instruction-based image editing. Created by using a Stable Diffusion-based inpainting model and segmentation datasets to generate realistic image pairs.

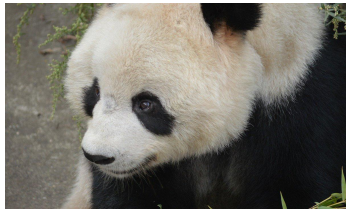


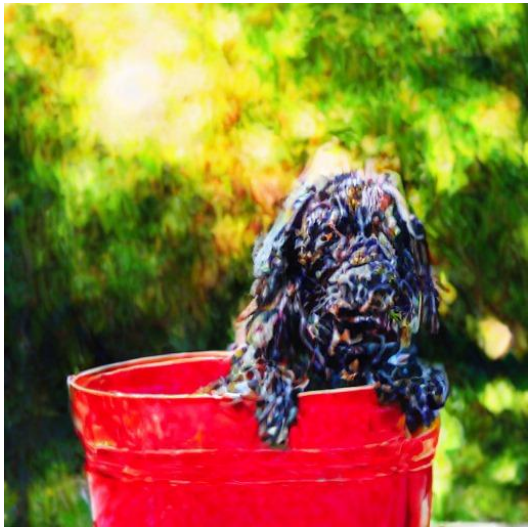

Step 1
CLIP
Step 2
Paint by Example
Step 3
IP Adapter
Step 4
Dataset
Step 5
Training and Adaptation

Step 1 - CLIP

- First attempt at implementation.
- We trained with all layers frozen to establish a performance baseline. This allowed us to validate the feasibility of using object images for conditioning by replacing *CLIPTextModel* with *CLIPVisionModel* before fine-tuning additional components.
- In the inference phase, the original image and object image are provided as inputs, and the model generates the edited result based on the refined embeddings.
- **Failure):**

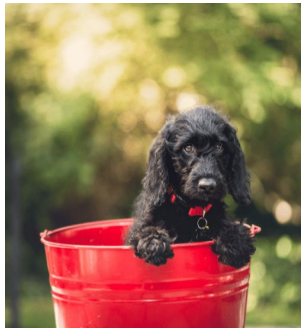


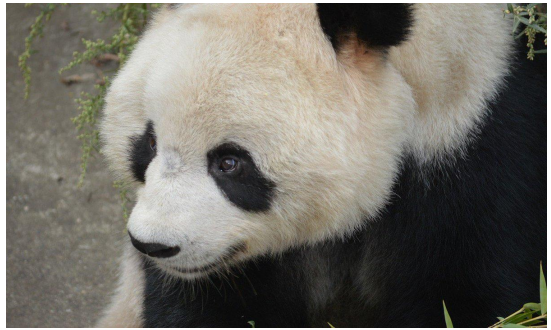


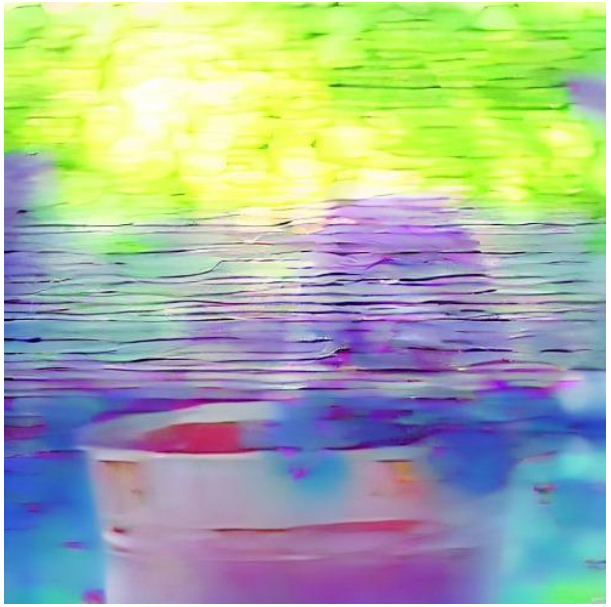


Step 2 - Paint by Example

- We tried assimilating our existing model with the trained Image Encoder of Paint by Example.
- Here too we trained with all layers frozen to establish a performance baseline. This allowed us to validate the feasibility of using example images for conditioning before fine-tuning additional components.
- **Failure :(**



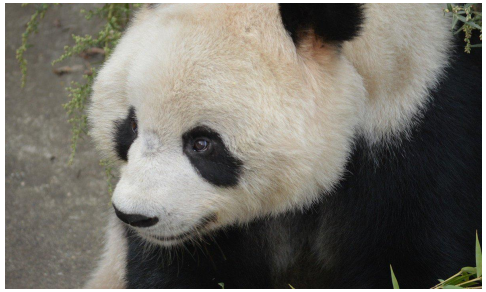


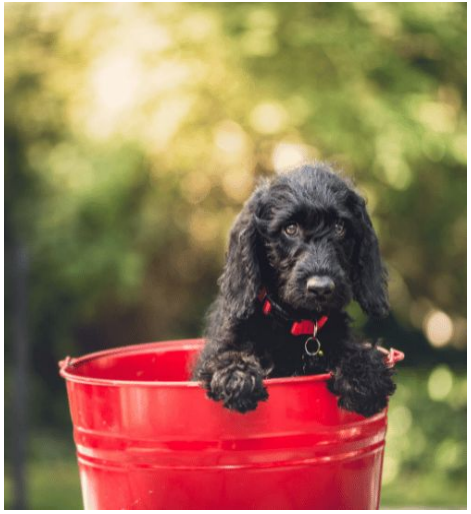


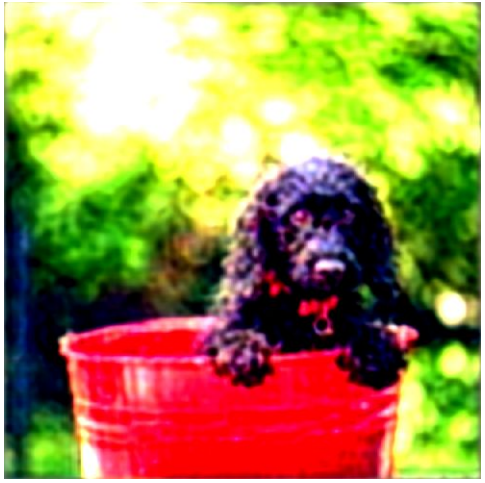
Step 3 - IP_Adapter

- We attempted to utilize the IP_Adapter module in our pre-trained model in order to allow for Image+text conditioned guidance.
- Lack of ability at capturing semantic meanings from example images
- **Failure**

“Add a panda”







Step 4 - Dataset Creation

- No other choice but to train the model to enhance the quality of the embeddings we use for conditioning, for that we needed data.
- Utilized PIPE and PIPE_mask datasets to create a dataset with each sample being a set of three images: original_image, object_image and target_image
- Created an initial dataset of 42,000 triplets: https://huggingface.co/datasets/Aya168/project_from_PIPE/
- Later on (optimization step) we increased it to 108,000 triplets: https://huggingface.co/datasets/ADT1999/project_from_PIPE_extended

object_Image

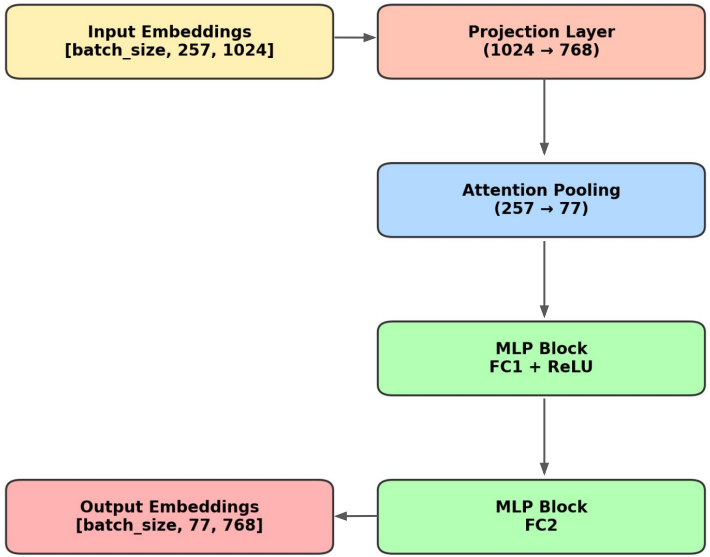
original_Image

target_Image

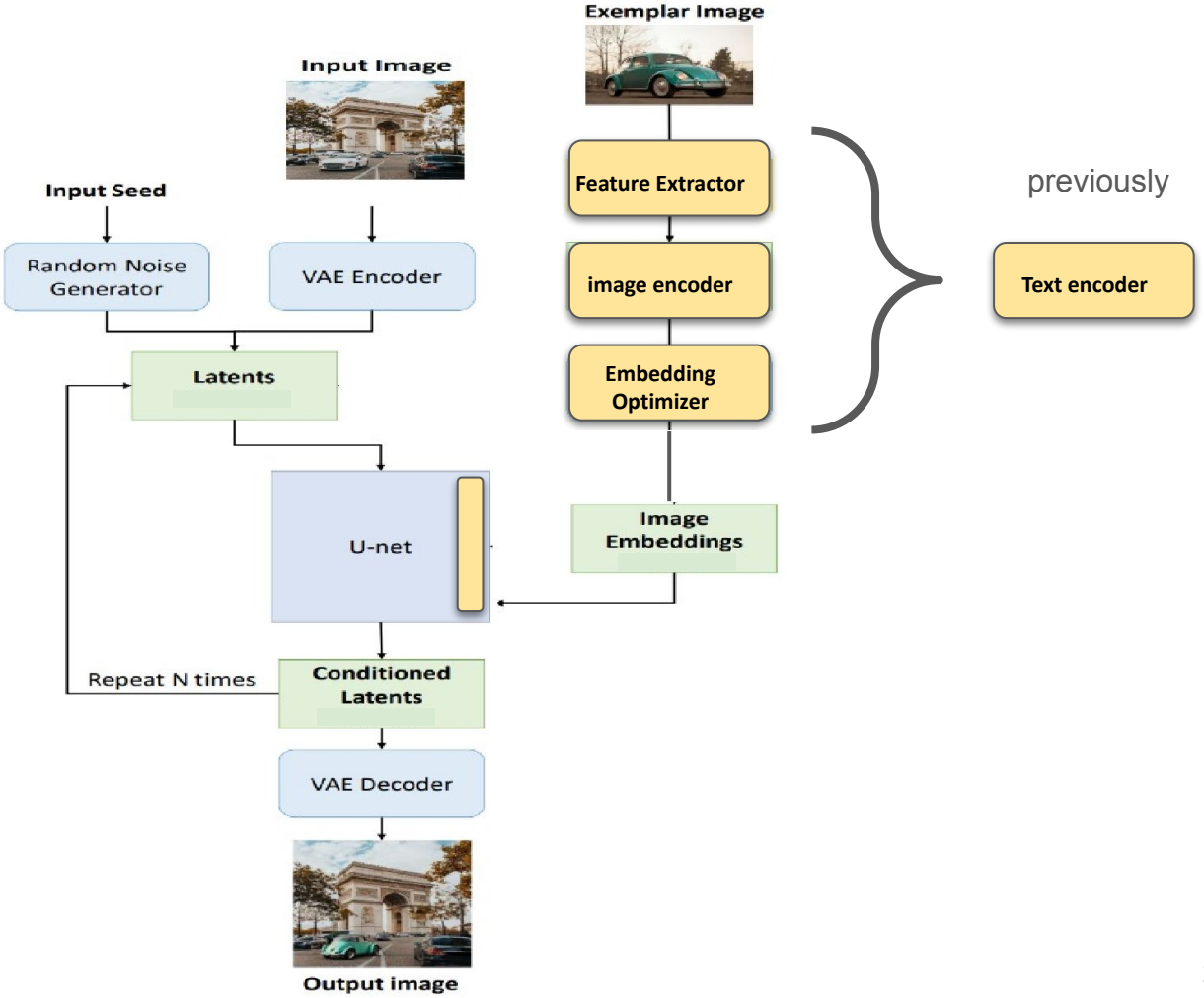
Step 5 – Training & Adaptation

- We use the *CLIPImageEncode*r to generate embeddings from the object image. These embeddings capture semantic information of the object image that are crucial for guiding the edits.
- To enhance the quality of the embeddings, we applied initially **two additional linear layers** after the image encoder (*embedding_optimizer*).
- Results with not much distinguished semantic meaning
- We then improved our design:

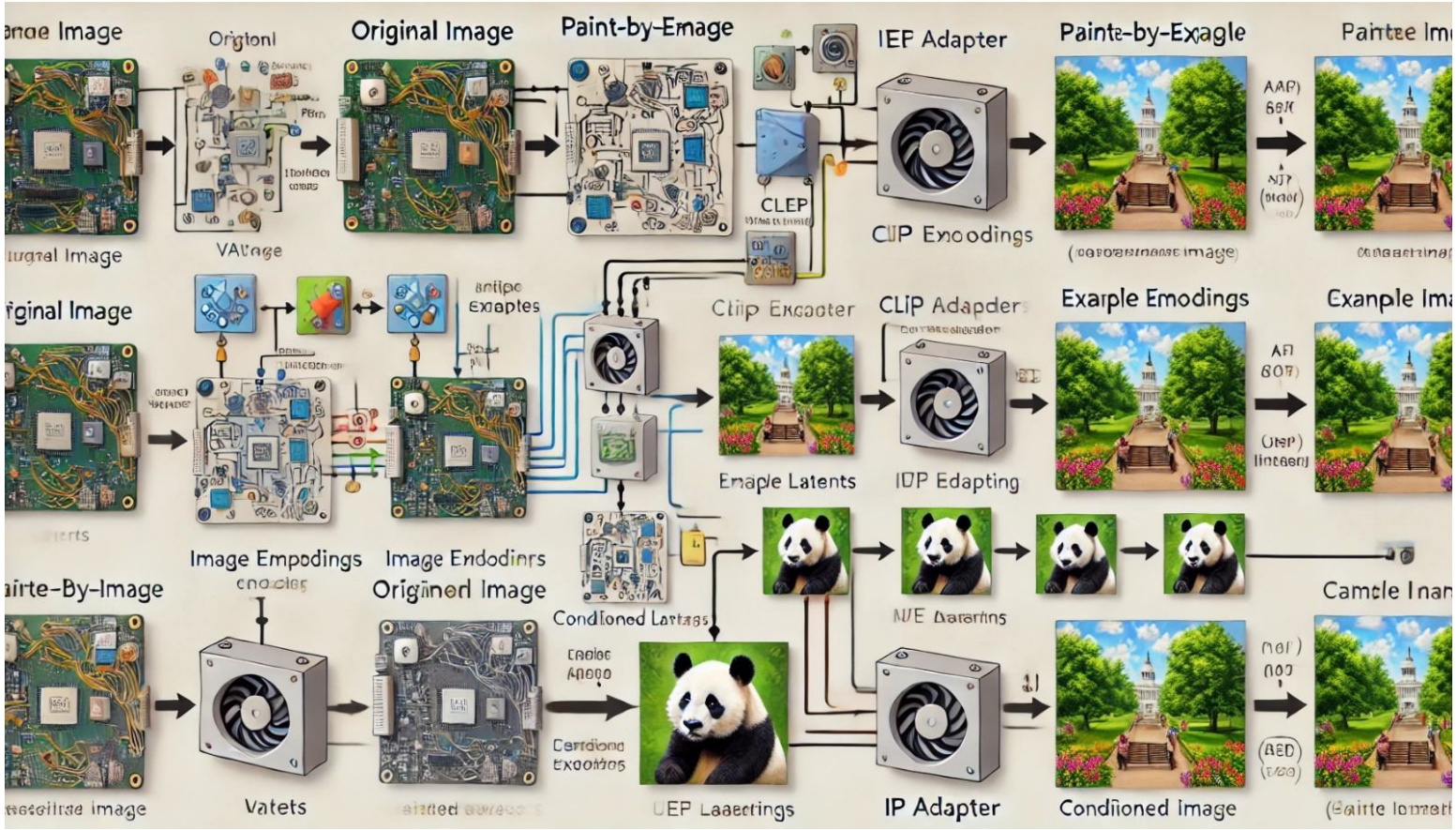
Embedding Optimizer



Architecture Overview:
1. Input embeddings of shape [batch_size, 257, 1024].
2. Projection layer reduces dimensions (1024 → 768).
3. Attention pooling reduces sequence length (257 → 77).
4. MLP block with FC1, ReLU, and FC2 layers.
5. Output embeddings of shape [batch_size, 77, 768].


Exemplar Image
Input Image
Input Seed
Random Noise Generator
VAE Encoder
Feature Extractor
image encoder
Embedding Optimizer
previously
Text encoder
Latents
U-net
Image Embeddings
Repeat N times
Conditioned Latents
VAE Decoder
Output image

how it felt


Original Image
Paint-by-Image
IEP Adapter
Paint-by-Example
CLIP Encodings
Original Image
Image Embeddings
Image Embeddings
Original Image
Image-By-Image
IP Adapter
Conditioned Image

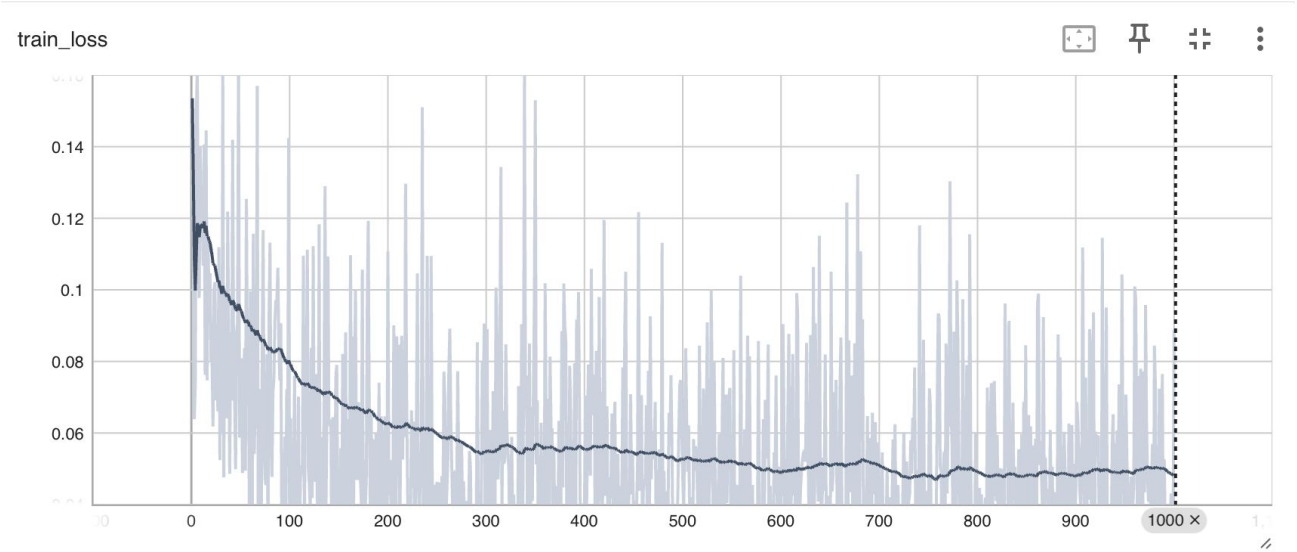
Examples

4

Enhancements

Improved Training

Optimize hyperparameters: batch size, training steps, learning rate, grad accumulation steps, dropout probability, lr scheduler & data augmentation

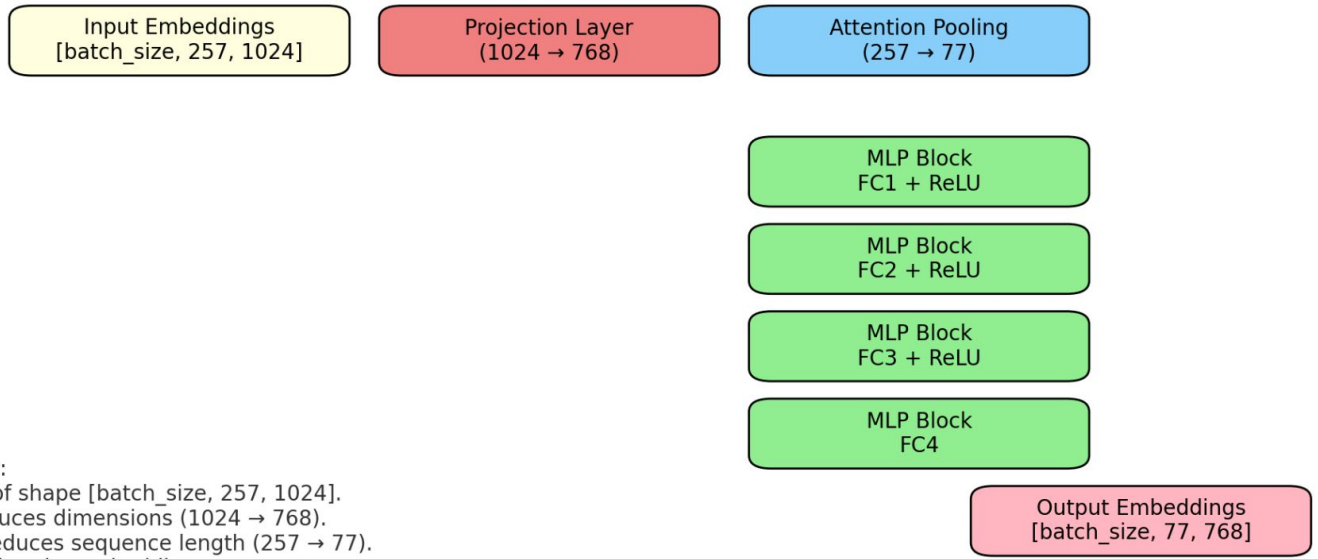


Improved Dataset

Expand Dataset and Extend Training: Incorporate a larger and more diverse dataset to improve the model's generalization capabilities. Longer training with varied examples yielded better performance and reduced overfitting to specific types of input images (108,093 triplets).

https://huggingface.co/datasets/ADT1999/project_from_PIPE_extended

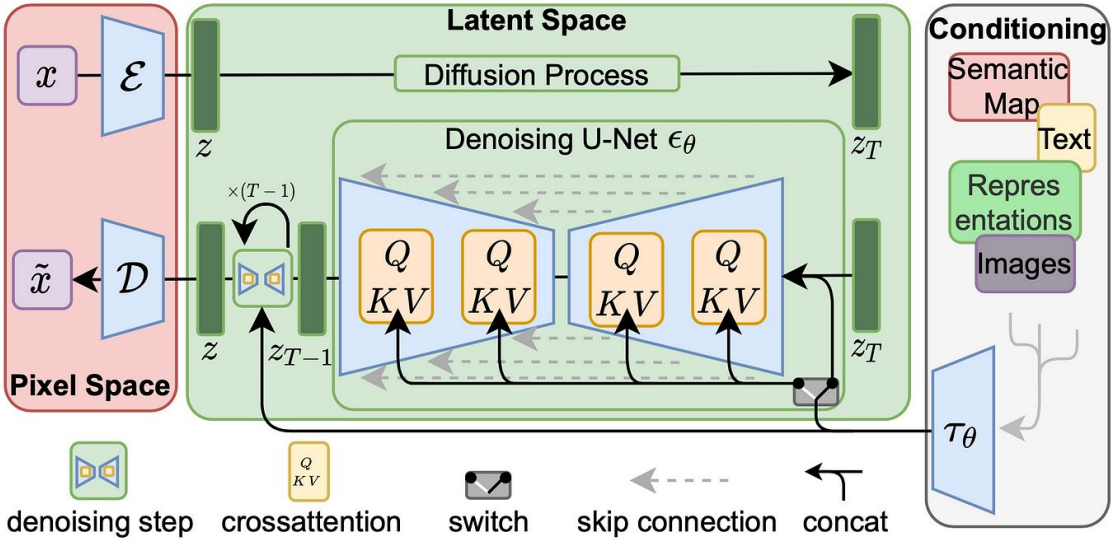
Improved Embedding Optimizer



Architecture Overview:
1. Input embeddings of shape [batch_size, 257, 1024].
2. Projection layer reduces dimensions (1024 → 768).
3. Attention pooling reduces sequence length (257 → 77).
4. Four MLP blocks refine the embeddings.
5. Output embeddings of shape [batch_size, 77, 768].

UNET

Train Cross Attention Layers in UNet: Training the **cross attention layers** in the UNet architecture to better accommodate the image embeddings.



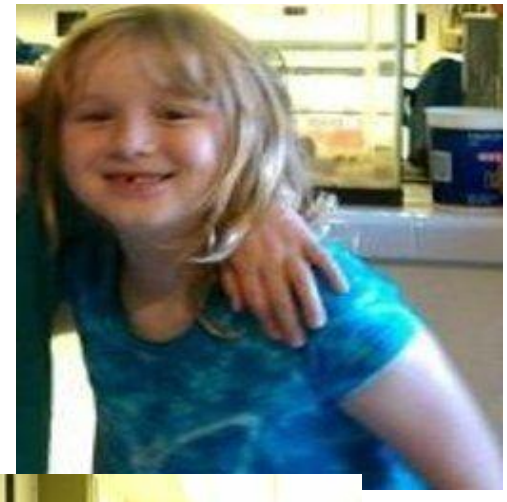
Examples

+

=

+

=




MARYELLEN
Shaughnessy


Shaughnessy


slambox


slambox


JB 007


slambox

5

Conclusion

Limitations

- Semantics “bridging” text and image
- Classifier free guidance – optimal settings are unique per case
- Expand training resources and improve training dataset to improve generalization
- Explore an alternative approach to our task, mask generation through segmentation models trained on PIPE sourced data.

Conclusion

- By replacing textual prompts with visual examples, we offer a new dimension of flexibility and usability.
- Our experiments have shown that the foundational model, combined with careful adjustments and enhancements, can handle this shift.
- With continued development, we believe our approach has potential for applications in creative industries, content creation, and beyond, where intuitive and visually driven editing tools are highly valued.

Thank you for listening!