AI-705 RECOMMENDATION SYSTEMS

INTERIOR DESIGN RECOMMENDER

Team: FrostByte

ABSTRACT

- Existing Text-to-Image generator models have shown tremendous progress.

- Traditional methods are limited to recommending existing designs. Generative AI allows users to envision entirely new design possibilities that align with their unique vision.

- However dataset specific to Interior designs are limited. The recommendations generated lack diversity.

- We propose training GAN’s on specifically interior design dataset, user prompts, including a detailed description of the design such as a particular style, furniture, color scheme etc.

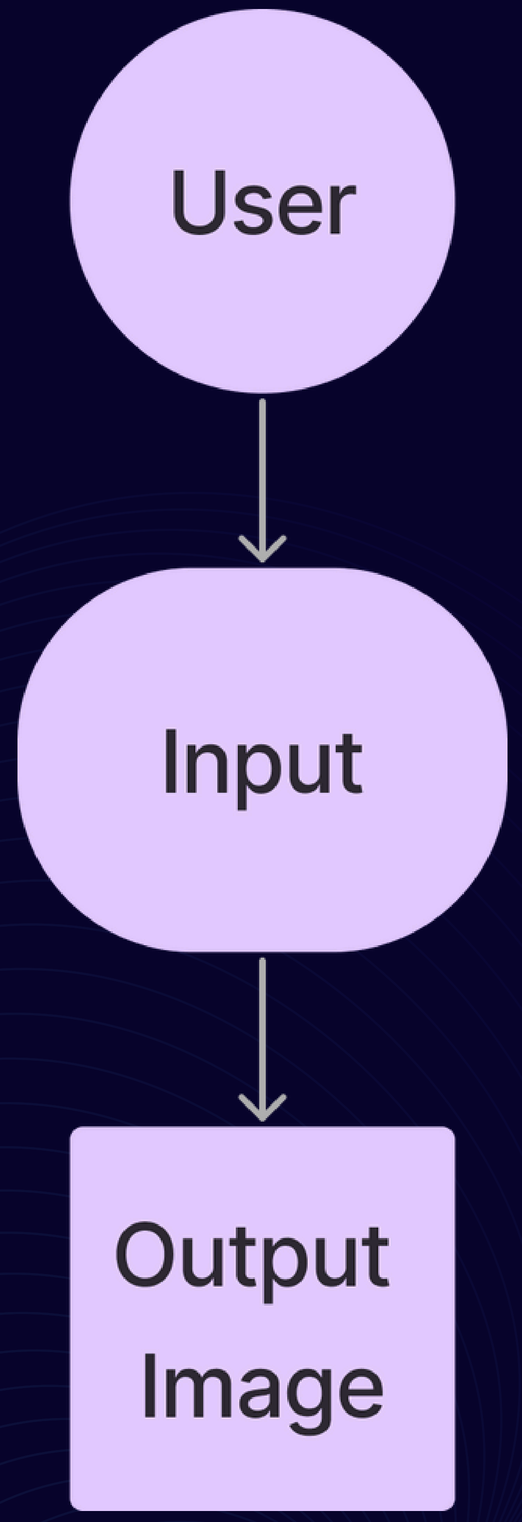
“Elegant living room vibes with stunning blue walls and a touch of sophistication from the gold lamp. ”

From Dataset

Using Gen AI

INTRODUCTION

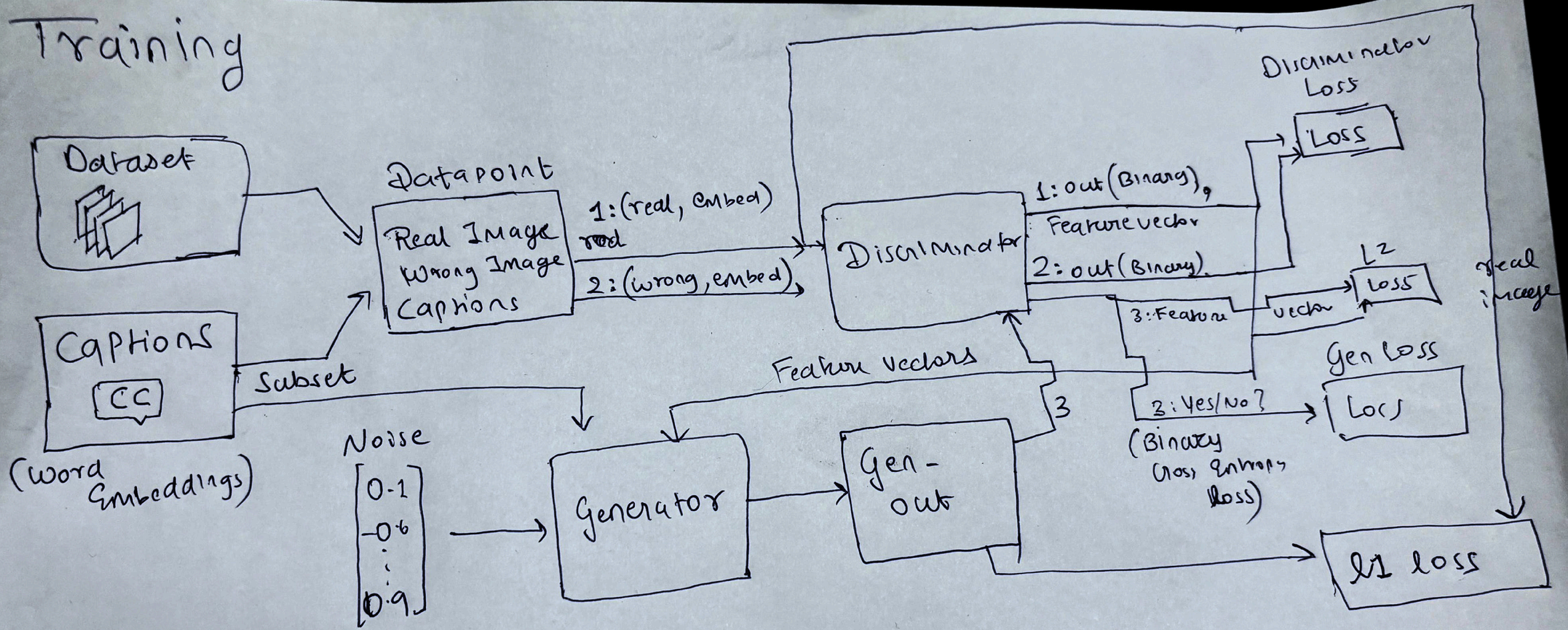
- Traditional GAN’s use a single step approach for training discriminator
- The discriminator typically receives the generated image independently of the text description. This can lead to scenarios where the generated image might be visually appealing but not actually depict the elements or style described in the text.
- Better understanding of Semantics(Style, lighting, furniture etc) will improve the recommendation.
- The two-step approach might lead to a more robust discriminator that can not only identify unrealistic images but also detect when a generated image visually contradicts the textual description.



OUR MODEL

- Training:
  - At core we are using a Modified Deep Convolutional Generative Adversarial Network.
  - Reason we chose this is we have a large dataset(95000 images), to effectively train a Deep Convolutional Model.
  - The Novelty lies in the Modification of the DCGAN. We train the Discriminator in a 2-step process.
  - This increases the understanding of our Model to more effectively gauge the underlying semantics of the Word Embeddings.

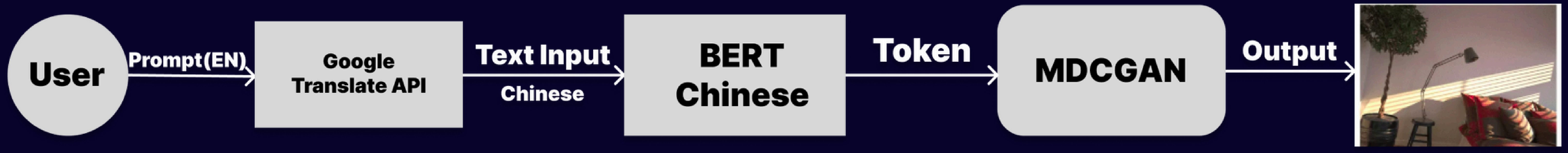
Training



Dataset
Datapoint
Real Image
Wrong Image
Captions
Captions
CC
(word Embeddings)
Subset
1:(real, embed)
~~red~~
2:(wrong, embed)
Discriminator
1: out (Binary),
Feature vector
2: out (Binary).
3: Feature
vector
Discriminator Loss
Loss
L2
loss
Gen Loss
Loss
3: Yes/No?
(Binary Cross Entropy loss)
Feature vectors
3
Noise
0.1
-0.6
0.9
Generator
Gen-out
real image
l1 loss

OUTPUT GENERATION

- One of the biggest challenges we faced were finding a dataset with captions.
- Only could find one that too with Chinese captions.
- One solution was to run through Translate API, but only limited calls are available.
- Hence, we decided to use the BERT Model trained in Chinese.
- Convert User Input(English) to Chinese then feed into BERT to get the word embeddings.



Timeline

Our ideation phase

	Dataset	Translate	Pre-trained?	Mods
Aim	Dataset	Translate	Pre-trained?	Mods
Problems	• Images ... no captions!	• Dataset with Chinese Captions	• Training of GANs takes huge amount of time	• Increasing model complexity
Solution	• Feature extraction, Text Representation. • Caption generation	• Free Translate API, limited calls	• Use of pre-trained GANs .. **but might not give good output for our dataset**.	• Building up on the existing GAN model from scratch.

Why DCGANs?

- **Preferences over Diffusion:** Diffusion models are notorious for being data hungry and even more heavier to train. GANs which only require a single forward pass through the generator network, inference in diffusion models is two to three orders of magnitude slower.

- **Focus on Spatial Information:** The convolutional layers in DCGANs are adept at processing the spatial relationships between pixels in images as compared to a simple Neural Network.

- **Specificity for Interior Design:** Our model is specifically trained on interior design elements like furniture types, spatial constraints, and functional considerations.

Why Not Traditional Recommendation Algorithms?

- A very baseline approach would've been to compare the user prompt with the image captions and use traditional recommendation algorithms such as cosine similarity.

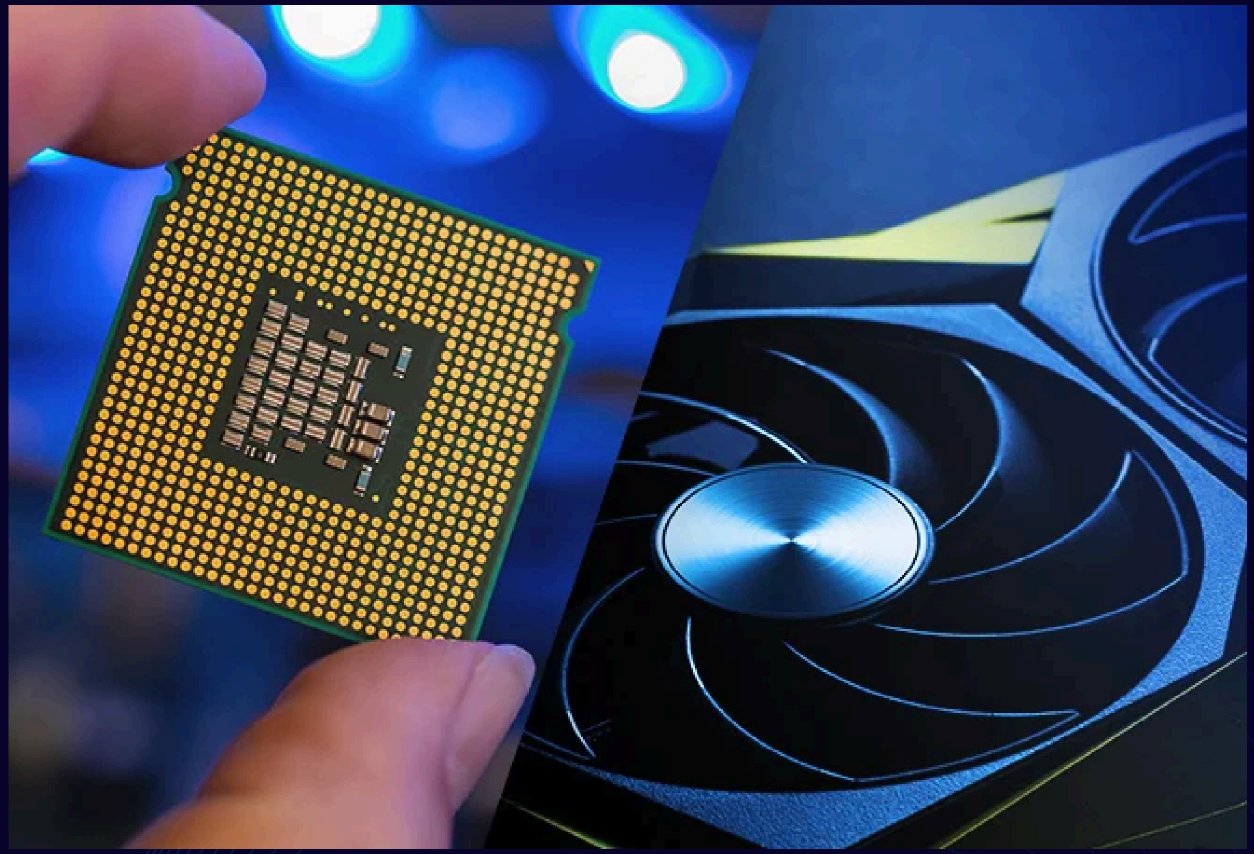
- We take the user prompt feed it to a Language Model (such as BERT) to convert them into tokens, do the same process for the image captions and using simple distance metric to compare the embeddings, then display the corresponding image.

- Our model is an improvement over it such that the baseline model will only capture overall feature similarity and not the deeper nuances of User Preference.

LIMITATIONS

01 Resouces: 500 datapoints with 4 layers and 25 epochs took 8 hours with GTX 1650 TI 8GB RAM GPU

02 Increasing the complexity of the CNNs used in Generator, which again causing a spike in run-time

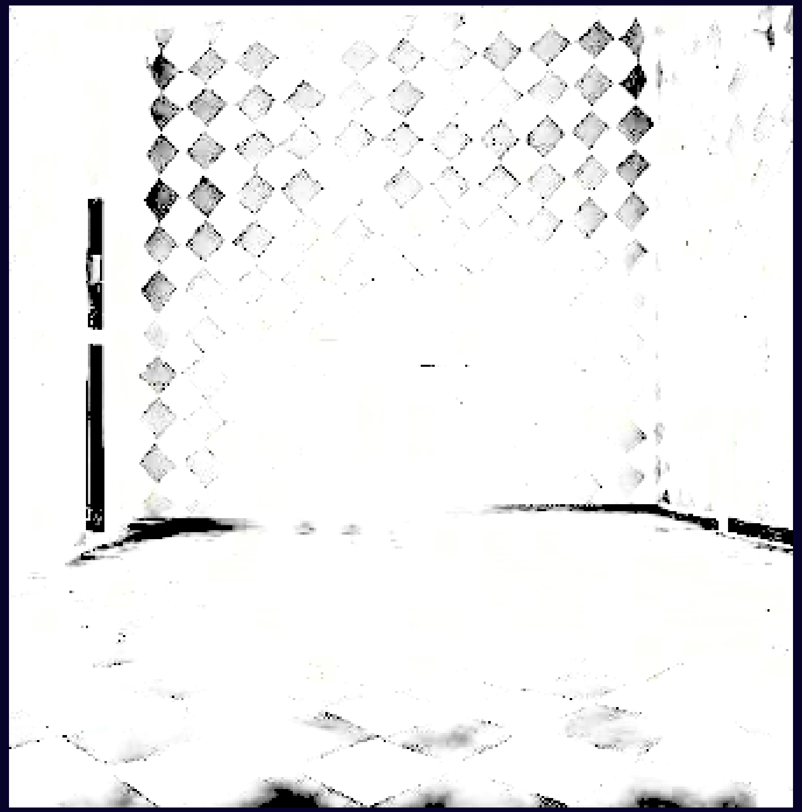


OUTPUTS

Prompt: “A simple black and white themed room with no furniture.”

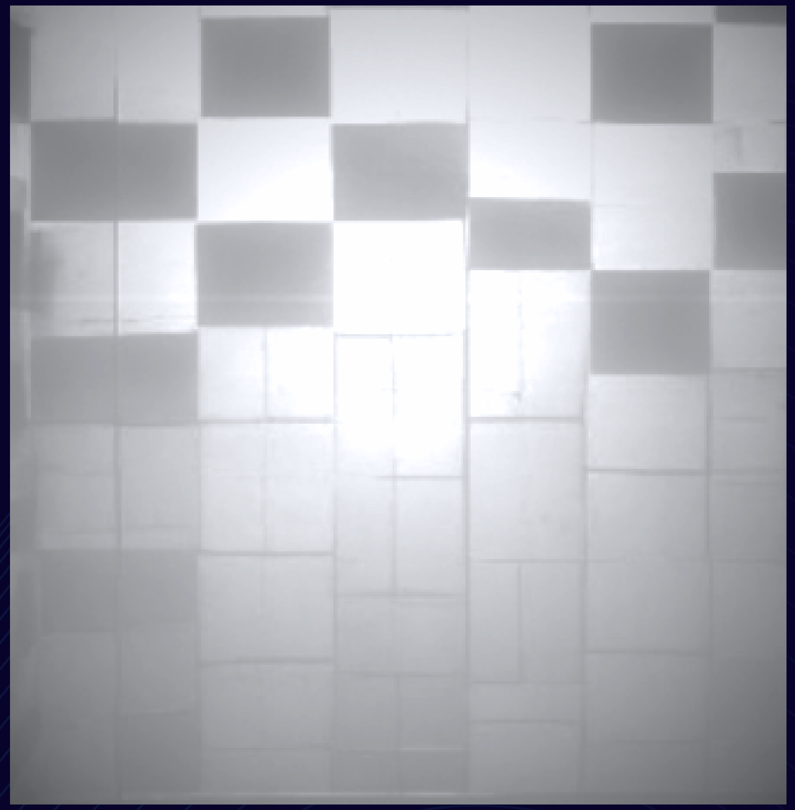
10 epochs, 500 samples

Runtime: 3hrs



25 epochs, 500 samples

Runtime: 8hrs

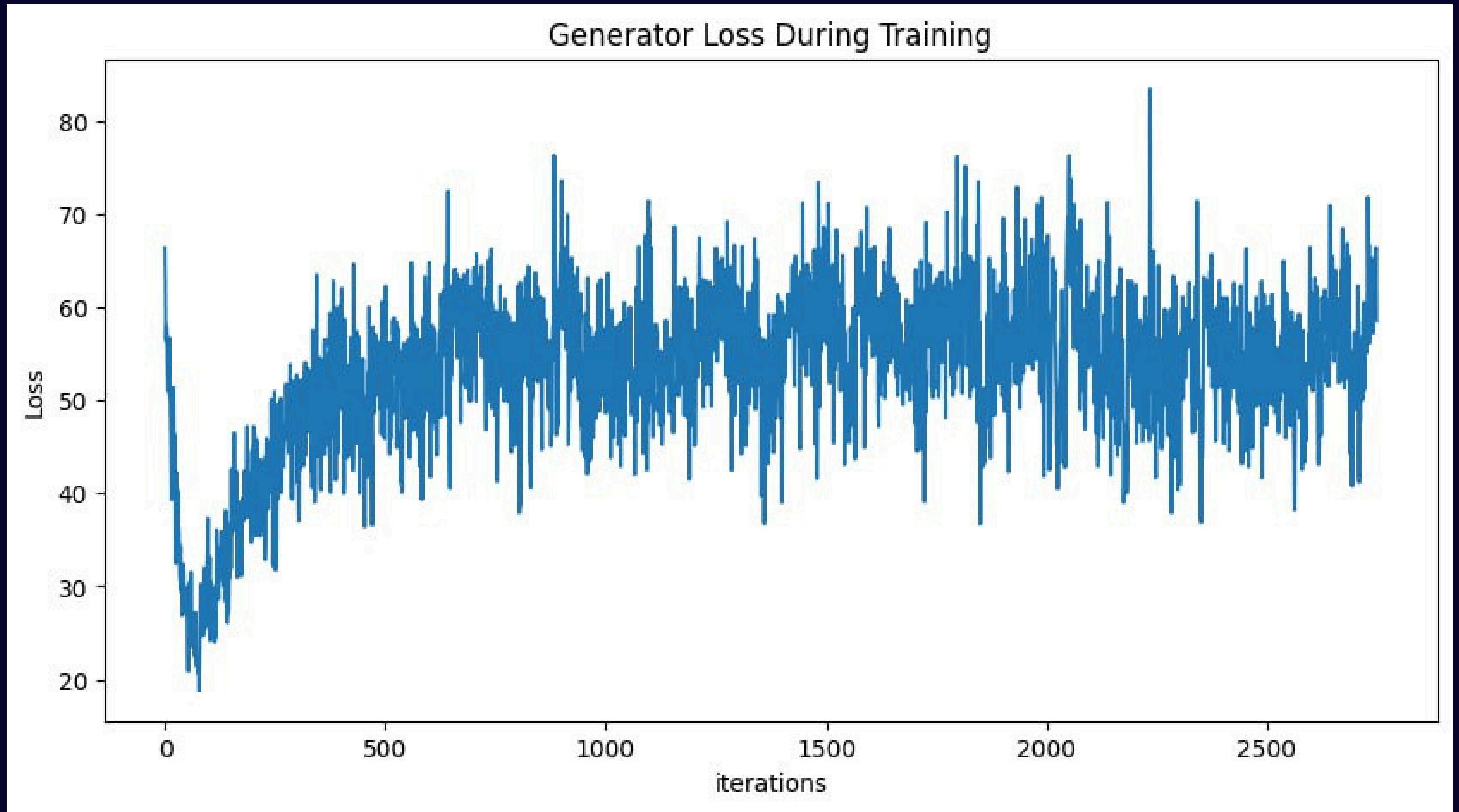


50 epochs, 2000 samples

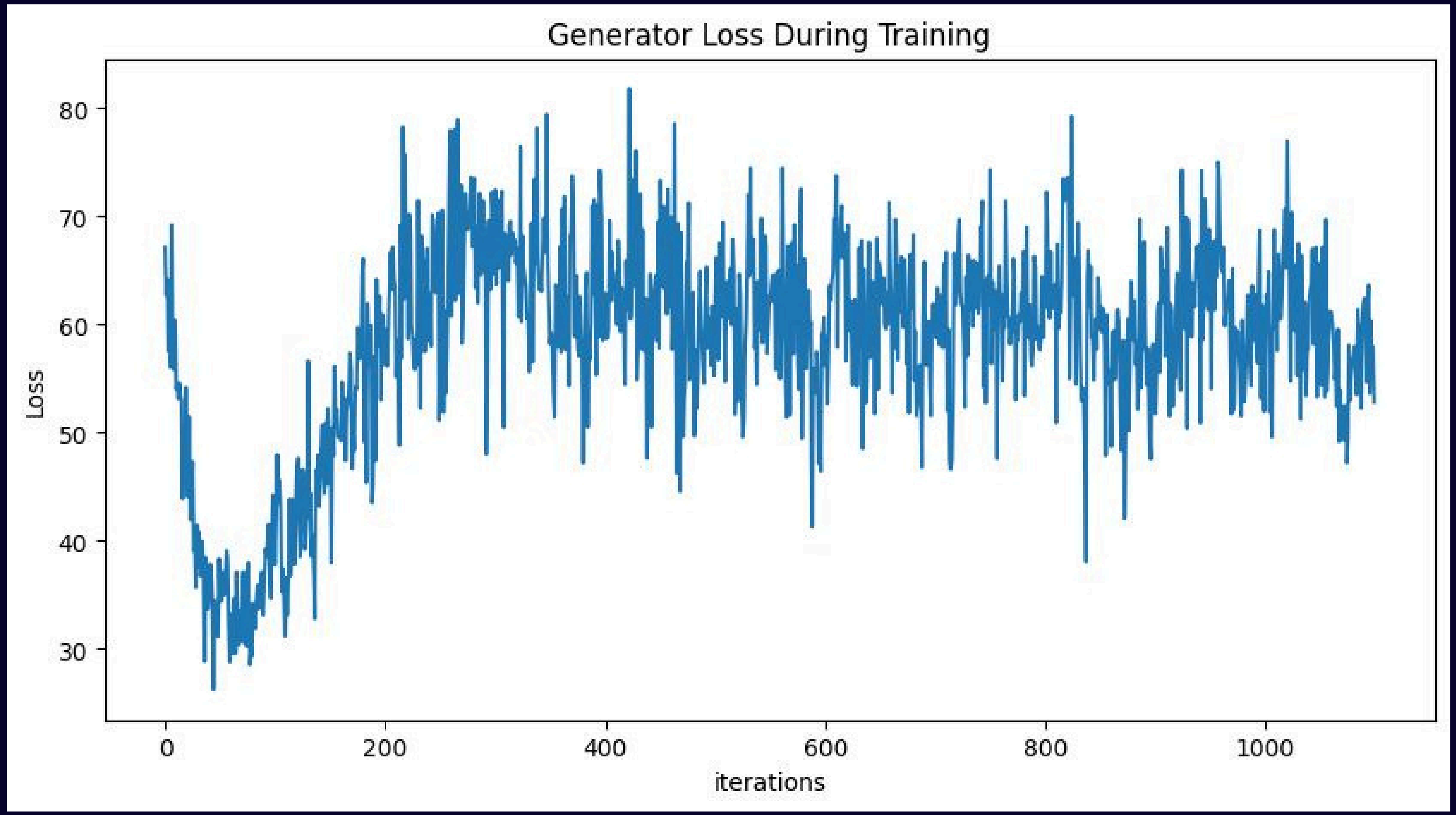
Runtime: 40 hrs

Evaluation Metrics And Benchmarks

Generator Loss



500 samples

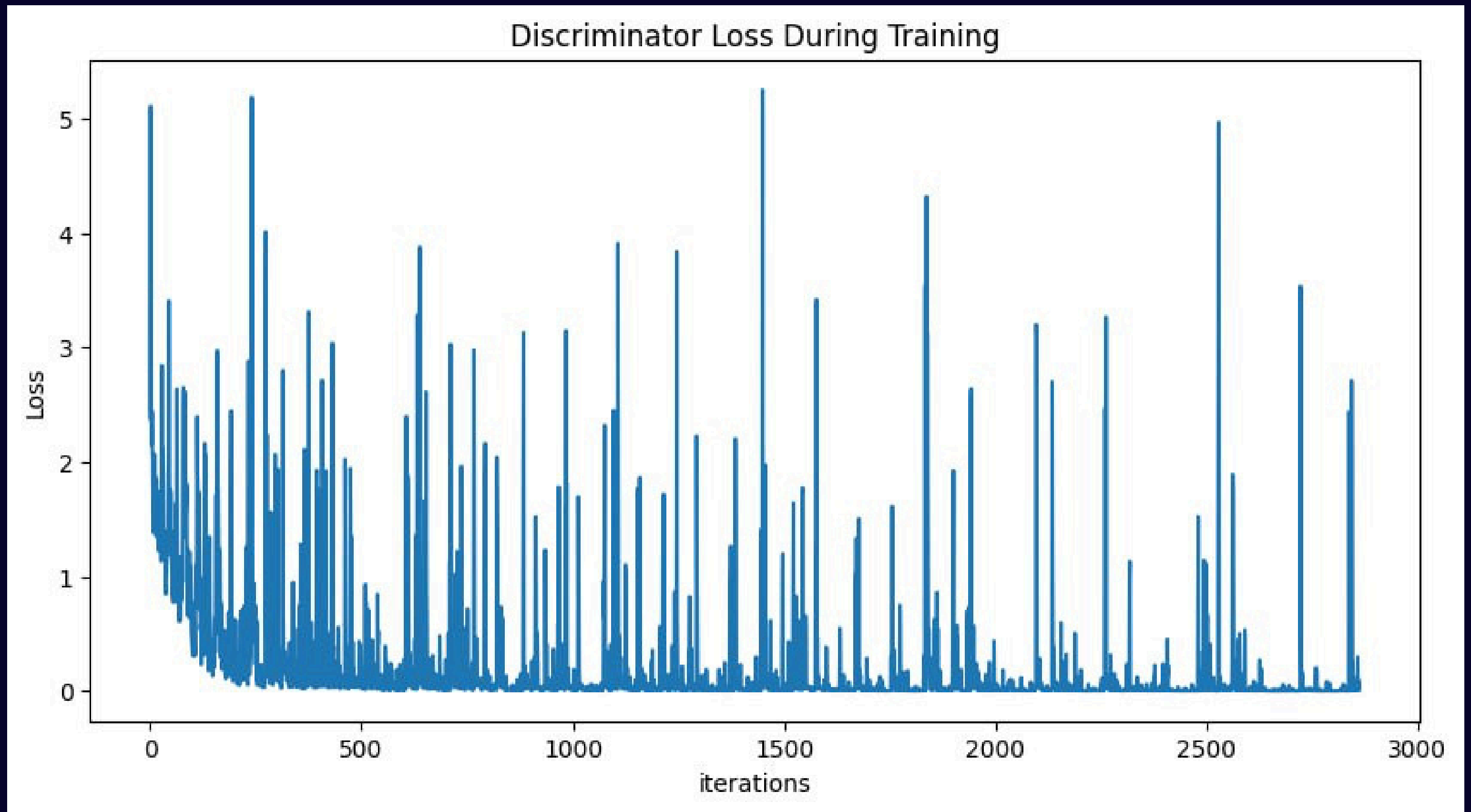


2000 samples

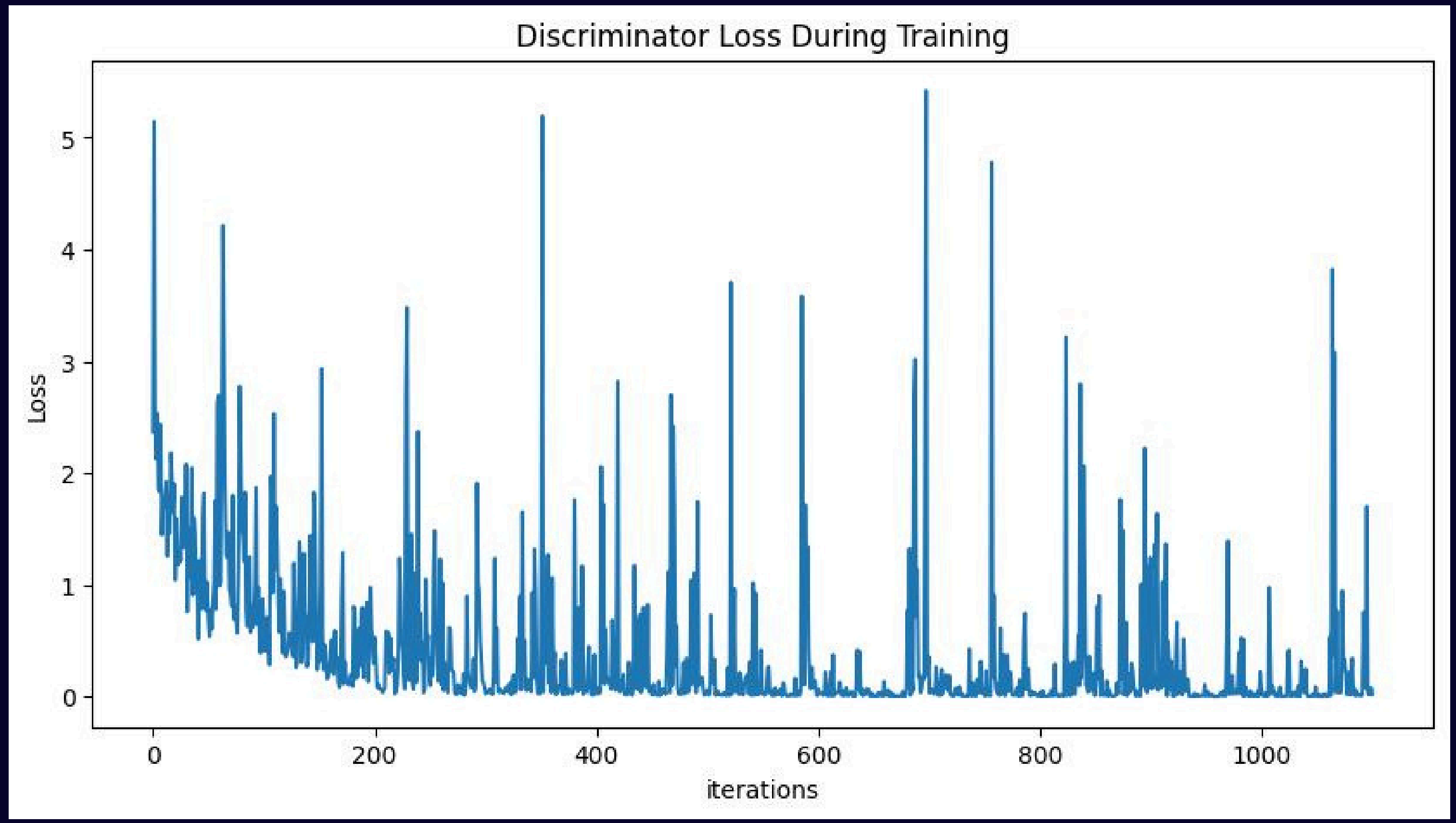
As we can see the generator has started to somewhat converge using a larger number of samples

Evaluation Metrics And Benchmarks

Discriminator Loss



500 samples

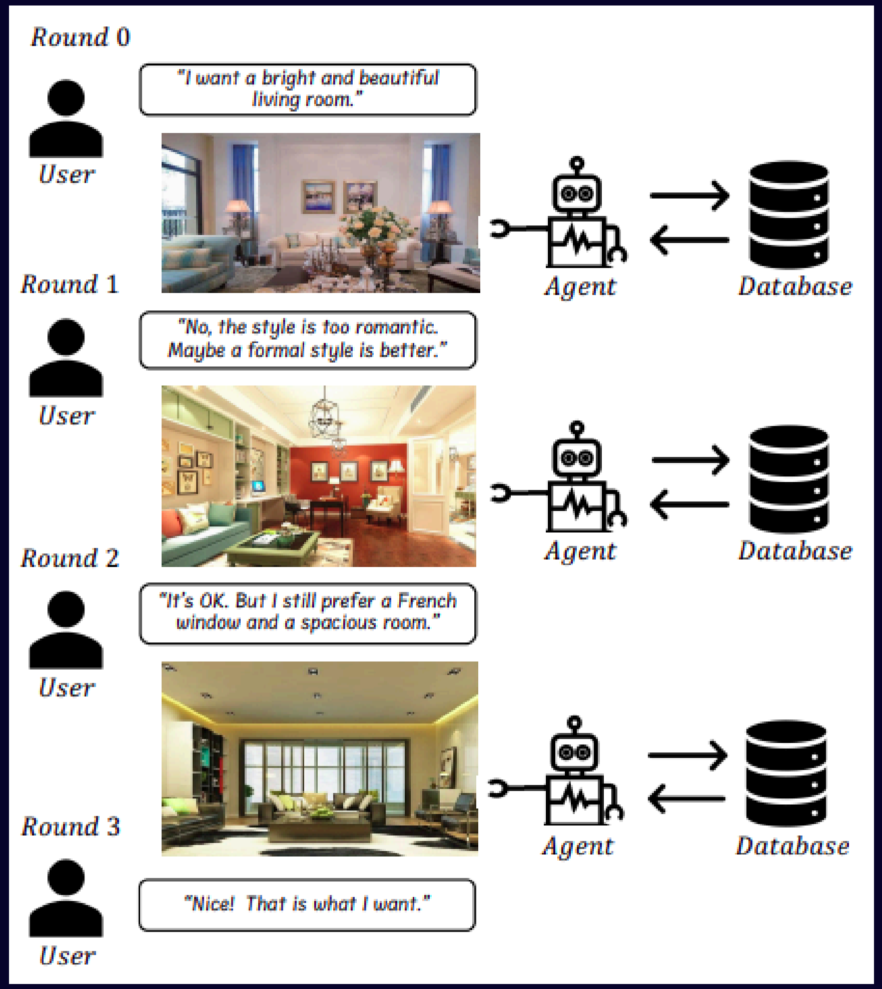


2000 samples

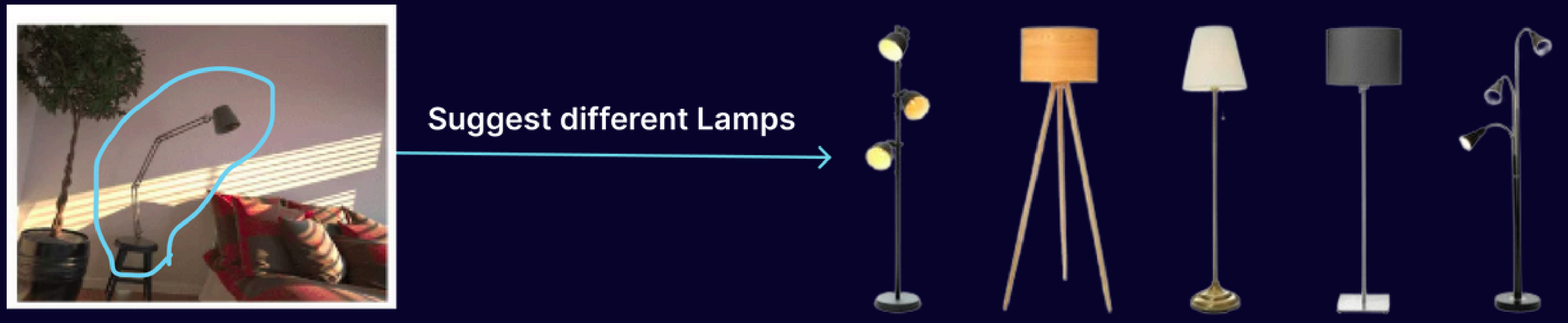
However the Discriminator Loss still doesn’t seem to increase as it should indicating more training time with larger number of samples is required.

FUTURE SCOPE

- Interactive Interior Design System: Incorporates Reinforcement Learning Algorithms which incorporate user rating while prompting. Making image generation an iterative process.



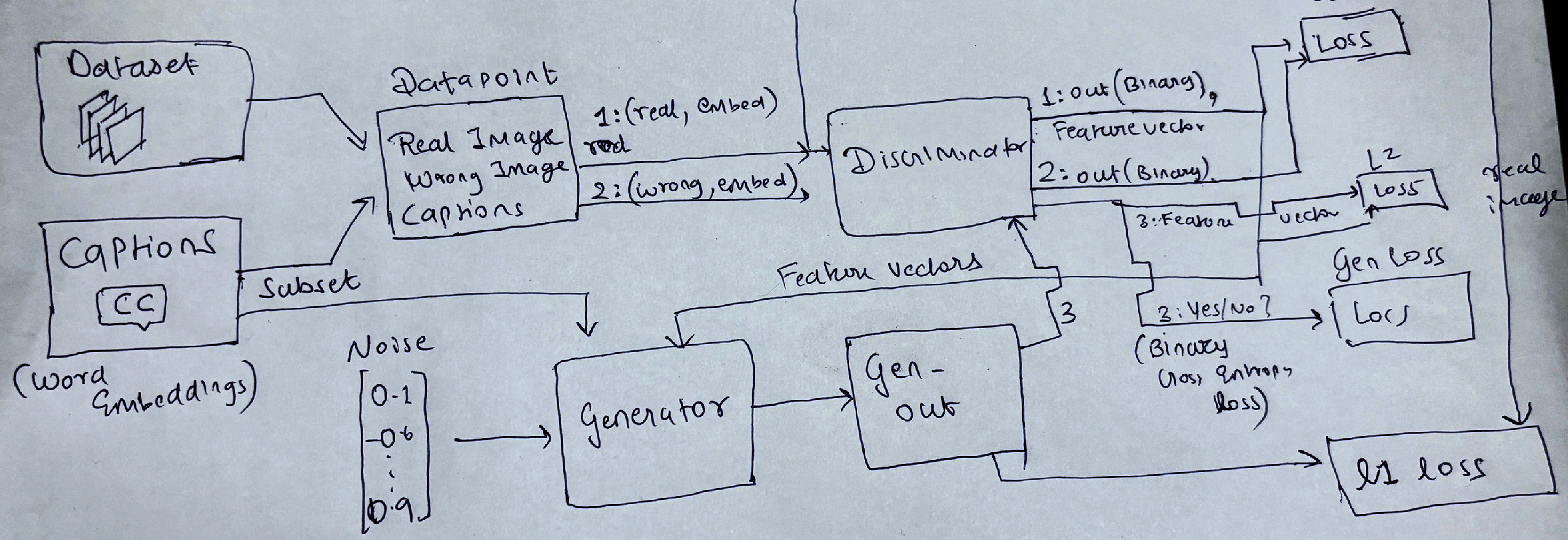
- Addition of AI tool called Interactive Image Editing with Text Guidance.



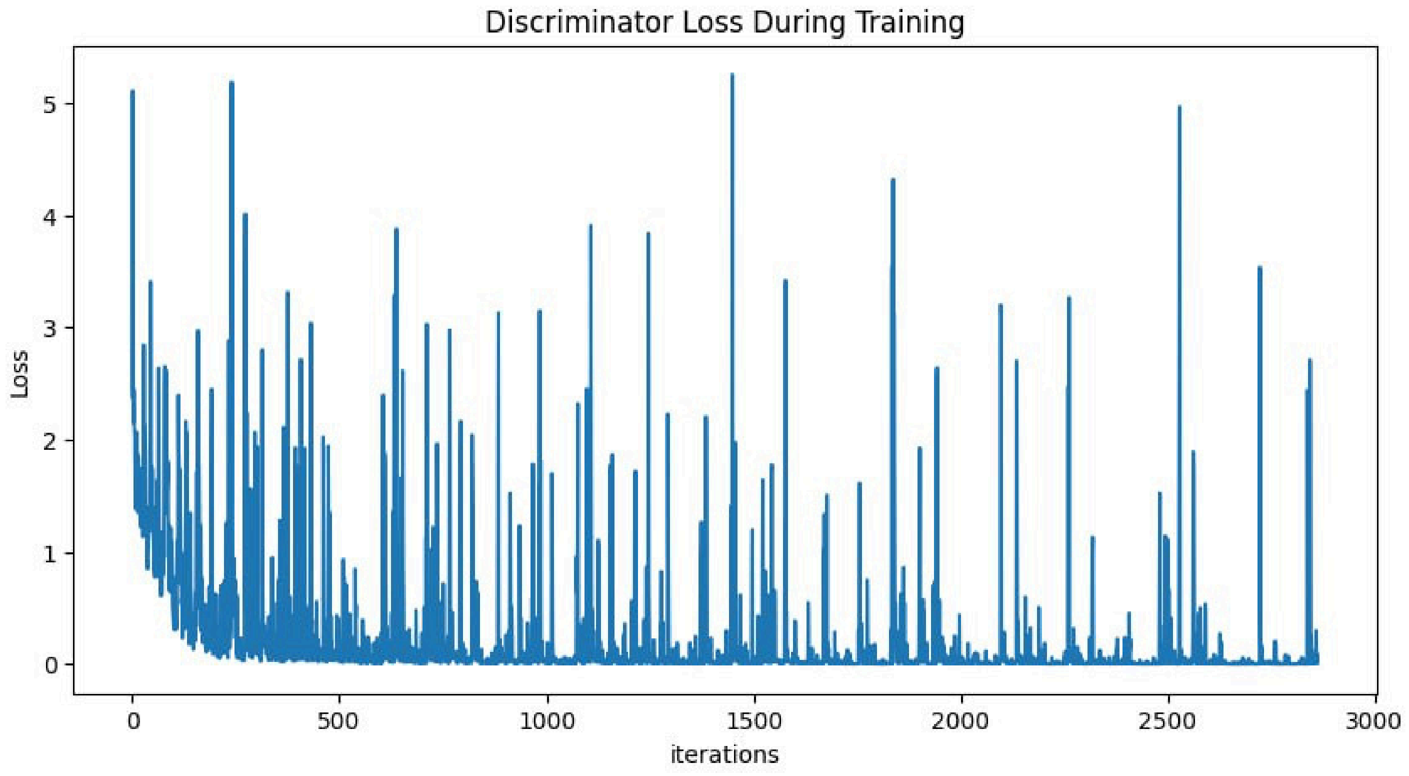
- Generate same image of room from 2 different views. [Inspiration: Text2Room]

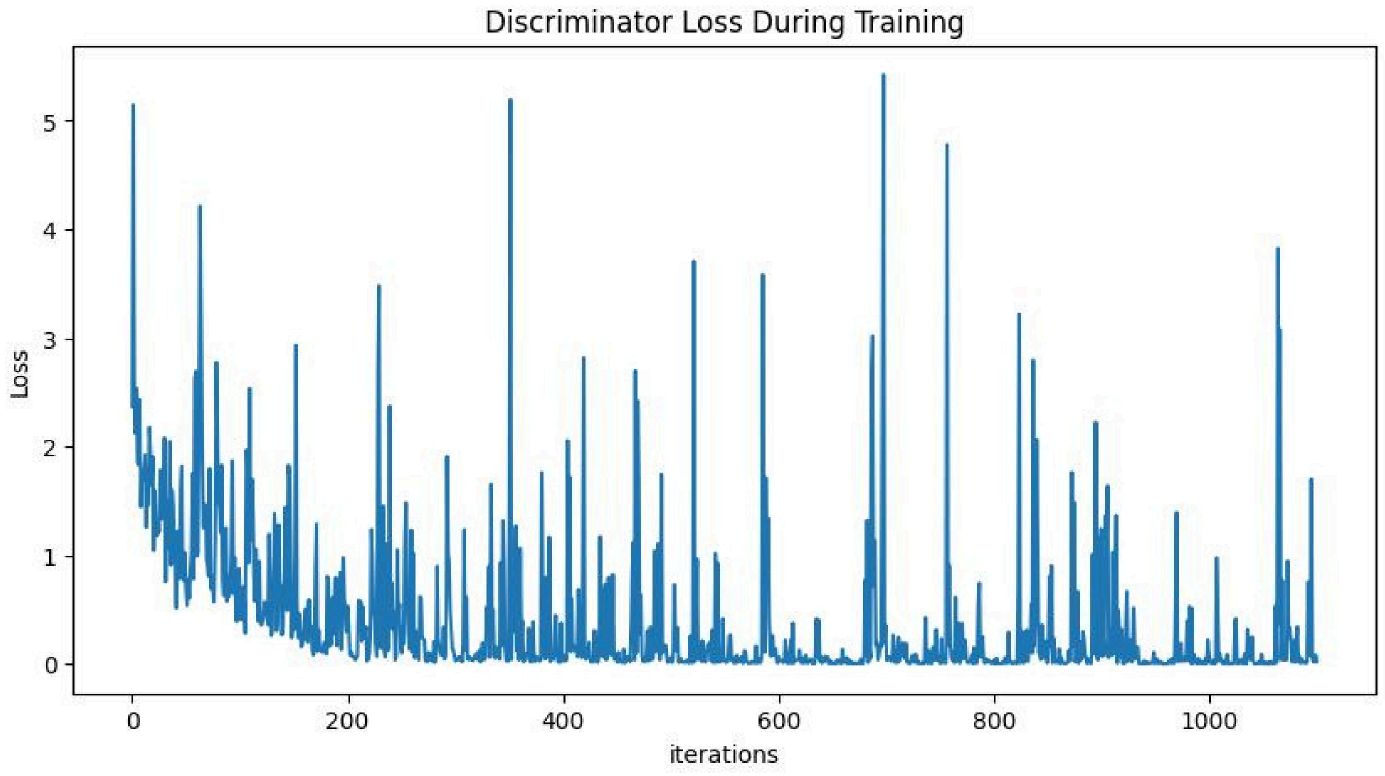
THANK YOU

Training


Dataset
Captions
CC
(word Embeddings)
Subset
Datapoint
Real Image
Wrong Image
Captions
1:(real, embed)
2:(wrong, embed)
Discriminator
1: out (Binary),
Feature vector
2: out (Binary).
3: Feature vector
Discriminator Loss
Loss
L2
loss
real image
Gen Loss
Loss
3: Yes/No?
(Binary Cross Entropy loss)
Feature vectors
3
Noise
0.1
0.6
0.9
Generator
Gen-out
l1 loss

Discriminator Loss





Generator Loss