Introduction

After the success of the latent diffusion model, text-to-image received much attention from research and application.
The active development of the field introduced a new image-generation task: generating new images consistent with reference images.
This allows the personalization of generative models.
As shown in the image, the reference images of a dog can be situated in an unseen context.

Algorithms that solve this task include DreamBooth, Textual Inversion, and some others.

However, this task lacks quantifiable evaluation metrics.

Problem Statement

There exist evaluation metrics used for general generative tasks such as CLIP-I, FID, and IS, but they all fall short of a holistic evaluation of the new generation task.
There are two factors to decide the quality of the generated image in this task.

1) Consistency of the subject introduced in the reference image
    a) “Input images”
2) Ability to generate new context introduced by text input
    a) “In the Acropolis”, “Swimming”, “In a dog house”, etc.

To tackle factor 1), cosine similarity between the reference and generated images is taken and quantified. Likewise, CLIP score is used to account for factor 2).
However, neither evaluates two factors as a whole.

The harmony of two factors in the evaluation metric is important.
In such a setting of a generation task, the tradeoff between factor 1) and 2) is not linear.
Thus, providing two separate quantities will not be sufficient to explain the superior quality of a generated image to another.

Knowing to rank the quality of generated images is important as the stochastic nature of the generative model sometimes creates low-quality images.
This problem can be mitigated by generating a batch and ranking by their qualities.
Thus, a novel evaluation metric that accesses a sweet spot in the tradeoff graph is needed.

Background Statement

As mentioned above, DreamBooth and Textual Inversion are proposed to generate images in a new context while maintaining consistency with reference images. They both learn the concept of a subject in the reference images (e.g. corgi dog) and encode it to the textual domain.
However, they differ in how they define the new concept.
DreamBooth trains a unique identifier that maps to the new concept and finetunes the entire model with reference images paired with basic prompts (e.g.“A photo of [V] dog”).
They keep semantic ability by adding a regularization term, reconstruction loss to images generated without the unique identifier (e.g. images generated by a prompt, “A photo of dog”)

Textual Inversion initializes a new embedding and optimizes the embedding directly to represent the new concept.
Likewise, they finetune on reference images paired with prompts (e.g. “A photo of S*”) and pass gradients to the new embedding, only trainable parameter.

As it does not change model parameters, the new embedding has potential to be tested on pretrained multimodal model such as CLIP.

In intuition, DreamBooth tweaks definitions of previously defined words to define a new word.
For example, consider the words “telephone” and “call”.
Before and after the word “telephone” was coined, the definition of the word “call” has changed a bit.
The definition extended to include “the act of contacting someone by phone”.

Textual Inversion, however, does not interfere with definitions of other words as only the new embedding is trained. In this case, consider the word “brunch”.
“brunch” is a simple composition of “breakfast” and “lunch”.
This does not impact definitions of any other words.

TODO: Not need context
From intuition, DreamBooth has better expressiveness when defining new concepts, and I posit this is why DreamBooth is more powerful
However, Textual Inversion is better suited to use for the proposed evaluation metric which I will describe in the next section.

Naive Proposed Method

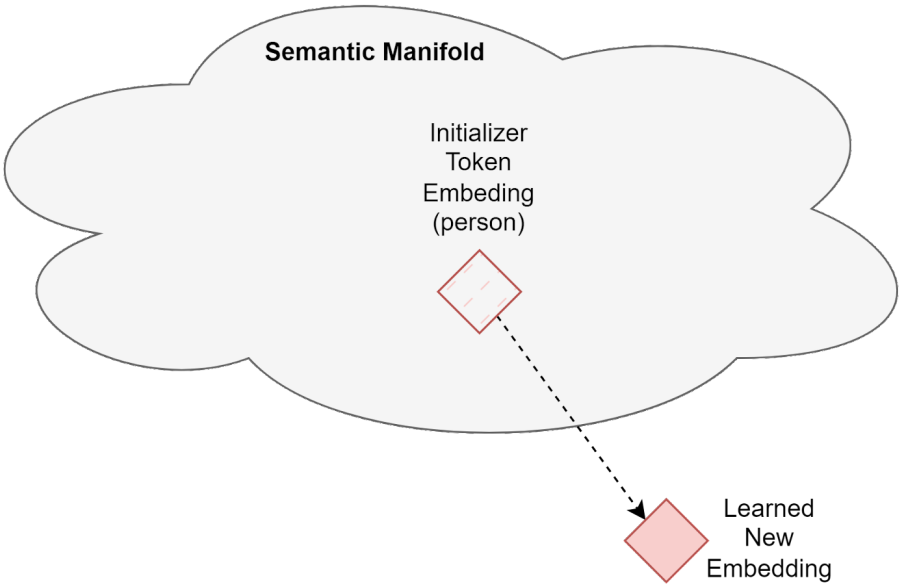
Often a text encoder of CLIP is used as a module to encode text for Stable Diffusion.
Thus, if a new embedding is trained with the text encoder untouched, it can be tested if the embedding is well-aligned to the reference image by checking cosine similarity.
Furthermore, it is crucial to test whether the embedding contains semantic information.

For the experiment, I trained a new embedding with a few Jim Carrey images.
Then, I tested the embedding in multiple settings to evaluate these two factors that define the quality of generated images.

1) {“A photo of [Jim Carrey]”} - {Jim Carrey image, Jim Carrey look alike image, Miley Cyrus image, Backpack image}
    a) This checks whether the embedding learned the new concept.
    b) The expected outcome is to have high cosine scores in order of Jim Carrey image, Jim Carrey look alike image, Miley Cyrus image, Backpack image
2) {“A photo of [Jim Carrey]”, “A photo of a white man with big expression, brown hair”} - {Jim Carrey image}
    a) Harder problem than 1).
    b) The expected outcome is to have high cosine scores in order of “A photo of [Jim Carrey]”, “A photo of a white man with big expression, brown hair”
3) {“A photo of [Jim Carrey]”, “A photo of [Jim Carrey] doing ballet”} - {Jim Carrey image, Jim Carrey ballet image}
    a) This checks whether the embedding maintains semantic information.
    b) The expected outcome is to have high cosine scores in order of “A photo of [Jim Carrey]”, “A photo of [Jim Carrey] doing ballet”

TODO: With thorough experiment, need to fill in the experiment result

It is observed that the new embedding does not maintain semantic information.
A hypothesis is that the new embedding has converged outside a semantic manifold.



This provides a possible explanation to the poor result.
In intuition, “brunch” as a word is defined without any semantic signal such as “I eat brunch on Saturday.”, “Pancake is the must-have menu for brunch.”, but by mere alphabets.
Then, it is difficult to tell how to use the word “brunch” in what context (e.g. I brunch to you → makes no sense!).
The loss of semantic information is observed in generated images with the embedding.
If we add context to the embedding “A photo of [Jim Carrey] dancing with a girl”, many of the generated images are similar to the output of “A photo of [Jim Carrey].”, missing the context information.

Textual inversion is finding one of many representations by reversing a many-to-one problem.
Thus, to find one representation holding semantic information, we need to introduce a constraint.

Proposed Method

Textual Inversion initializes an embedding to learn with an initializer token.
Then, it only optimizes the embedding with the diffusion objective.

$$\mathbb{E}\left[||\epsilon - \epsilon_\theta(x_t, t, c_{new})]\right]||_2^2$$

I propose to add KL regularization similar to DPOK Appendix A.2.
Lemma needed for proof is similar to DPOK.

$$\beta KL(p(x_{t-1}|x_t, c_{new})||p(x_{t-1}|x_t, c_{init}))$$

KL divergence penalizes for the output of new embedding having different distribution to the initializer tokens.
This prevents the new embedding deviating from the semantic manifold.

The final loss is the equation below.

$$\mathbb{E}\left[||\epsilon - \epsilon_\theta(x_t, t, c_{new})]||_2^2 + \beta KL(p(x_{t-1}|x_t, c_{new})\|p(x_{t-1}|x_t, c_{init})\right)$$

With the proposed objective function, train the new embedding and run through the same experimental setting as Naive approach.

4) {“A photo of [Jim Carrey]”} - {Jim Carrey image, Jim Carrey look alike image, Miley Cyrus image, Backpack image}
    a) This checks whether the embedding learned the new concept.
    b) The expected outcome is to have high cosine scores in order of Jim Carrey image, Jim Carrey look alike image, Miley Cyrus image, Backpack image
5) {“A photo of [Jim Carrey]”, “A photo of a white man with big expression, brown hair”} - {Jim Carrey image}
    a) Harder problem than 1).
    b) The expected outcome is to have high cosine scores in order of “A photo of [Jim Carrey]”, “A photo of a white man with big expression, brown hair”
6) {“A photo of [Jim Carrey]”, “A photo of [Jim Carrey] doing ballet”} - {Jim Carrey image, Jim Carrey ballet image}
    a) This checks whether the embedding maintains semantic information.
    b) The expected outcome is to have high cosine scores in order of “A photo of [Jim Carrey]”, “A photo of [Jim Carrey] doing ballet”

TODO: Through Result should be run and shared here

Now, why does this seem to work? When training Stable Diffusion, images paired with captions are used. This maps features from text encoder to specific distribution of images.
For example, any features related to the word “person” are unique to the distribution of images that generates a person. In other words, it is very unlikely that random features map to the distribution of images.

With the constraint, I find out that training textual inversion with KL Divergence constraint is more stable and maintains semantic information in CLIP space.