modified-FGD

I. Improvement

A. *Idea*

Instead of using a fixed guidance strength (δ) throughout the diffusion process, we can adaptively adjust the δ based on the differences between the generated image and the target guide structure at every step. This would benefit us in several ways -

- *Better Results* - It would eliminate the need of running the process multiple times for various values of δ in order to get a perfect image.
- *Efficient Convergence* - This approach can help the model converge more quickly to an optimal solution by preventing excessive deviations from the desired structure during generation itself.

B. *Pseudocode*

Algorithm 1 Adaptive Delta Update for Diffusion Process

Input: step t image $\hat{x}_t$, guide image x_g
Parameters: similarity threshold τ, scaling factors α and β, minimum delta δ_{min}, maximum delta δ_{max}
Set Initial Delta: $\delta_o = (\delta_{\text{min}} + \delta_{\text{max}})/2$
for each diffusion step t **do**
  Compute similarity score *(yet to decide the similarity metric)*: $\text{sim}(\hat{x}_t, x_g)$
  if $\text{sim}(\hat{x}_t, x_g) < \tau$ **then**
    Increase delta: $\delta_{t+1} = \min(\delta_{\text{max}}, \delta_t \cdot (1 + \alpha))$
  else
    Decrease delta: $\delta_{t+1} = \max(\delta_{\text{min}}, \delta_t \cdot (1 - \beta))$
  end if
  Update latent: $\hat{x}_{t+1} = \hat{x}'_{t+1} + \delta_{t+1} \cdot \vec{d}_g(\hat{x}'_{t+1})$
end for
Output: Generated image structure $\hat{x}_0$ aligned with guide image x_g

C. *Parameters*

All parameters are same as before, but we need not mention *Guidance Strength* - $\boldsymbol{\delta}$, from now onwards. Instead we have a new parameter and a batter one to handle things, **Similarity Threshold** - $\boldsymbol{\tau}$, for setting the extent of guidance strength in our adaptive delta approach

II. Results

A. *Variation of generated image with* ***similarity threshold parameter*** τ -

- As we have eliminated the *Guidance Strength* - δ parameter due the fact of it being too raw and difficult to decide for good results. We will now perform a similar analysis for the *Similarity Threshold* - τ parameter.
- Examples have been generated using $\sigma_s = 3$, $\sigma_v = 0.3$ and $t_{end} = 15$ as parameters, while varying τ from 0.05 to 0.3
- We can see that as we increase the *similarity threshold* the generated image becomes more of guide image oriented and yet giving exceptional results across all the values.

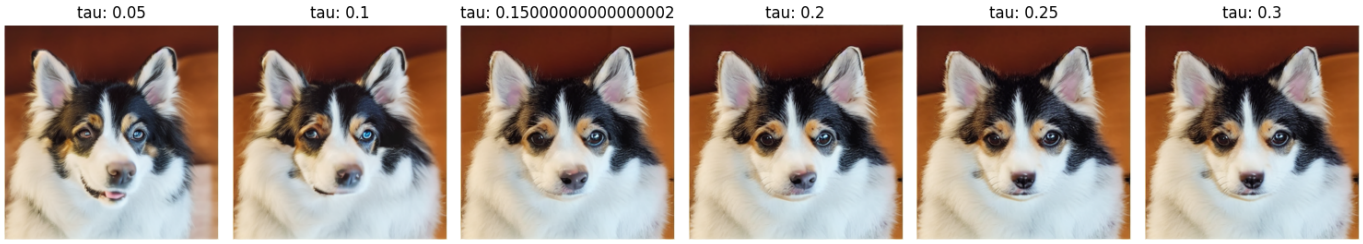
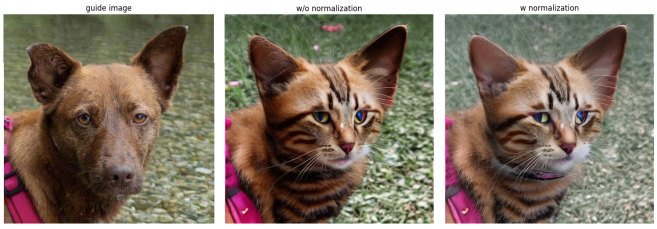
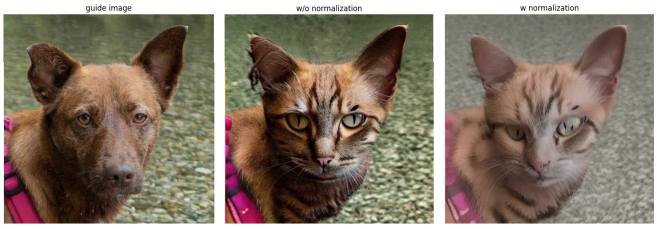


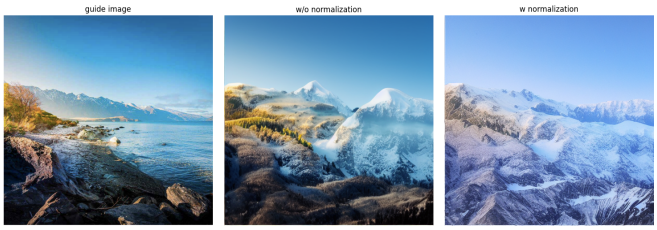
Fig. 1. Variation of generated image with τ - Guide Image: cat.png; Prompt: 'a photo of a dog'



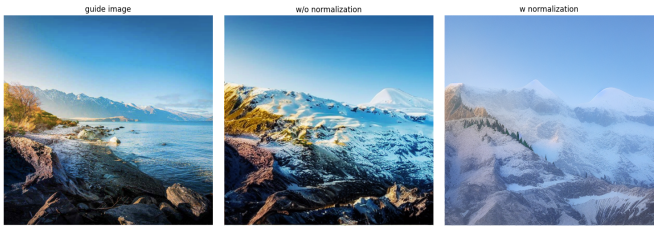
(a) **modifiedFGD** - a photo of a cat



(b) **FGD** - a photo of a cat



(c) **modifiedFGD** - a photo of a snowy mountain



(d) **FGD** - a photo of a snowy mountain

Fig. 2. Effect of normalization on the generated image - Guide Image 1: dog.png; Prompt 1: 'a photo of a cat'; Guide Image 2: landscape.png; Prompt 2: 'a photo of a snowy mountain'

*B. Effect of **normalization** on the generated image -*

- For the images with normalization prompts, the results for modified FGD are shown in the below Fig. 2
- Our modified FGD generates slightly better results that the original source, it traces the boundaries and the structure of the guide image better and overlaps it on the prompt very smoothly.

C. Variation of generated image with t_{end} parameter -

- We can see that as we increase the t_{end} while keeping the δ constant the generated image becomes moves away from the guide image and more towards the given prompt, due to increased iterations.
- The results for our modifiedFGD are way much better than that of FGD for low values of t_{end} parameter.

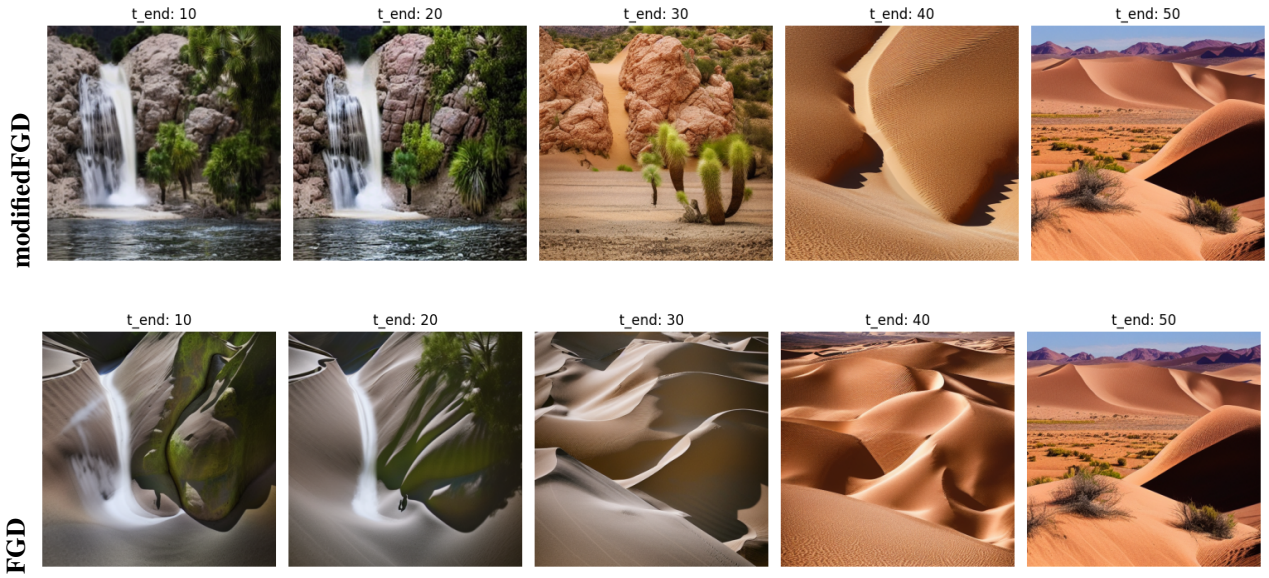


Fig. 3. Variation of generated image with t_{end} - Guide Image: waterfall.png; Prompt: 'a photo of a desert'. The first row shows us the results for our modifiedFGD and the second row shows the results for the source FGD implementation.

$\implies$ Most of the implementations and results described in this section can be reconstructed / regenerated by following the jupyter notebook [modifiedFGD.ipynb] procedurally in Google Colab using T4 GPU.