Project Report

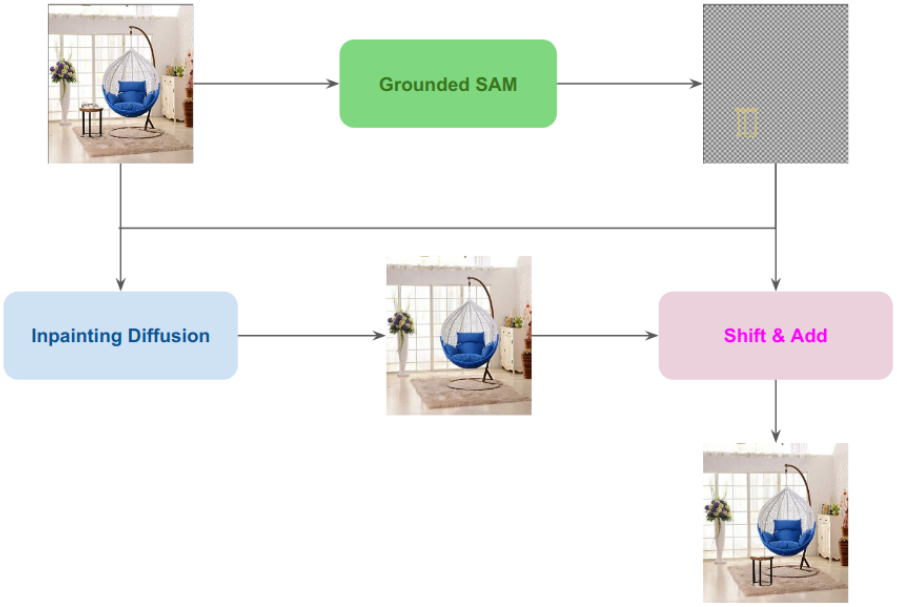
↗ Parent item | Avataar

Overview

This project covers updating visual location of an object (defined through text). The focus is on maintain realism and semantics while shifting the object by (x,y) pixels.

In this report, we illustrate successful examples (achieved through tinkering the core pipeline), followed with description of Failure Scenarios (and their potential solutions). We finally conclude the report with a few suggestions on potential avenue for propriety generative models.

The adapted pipeline (as shown in the figure below) employs Gounded Sam + Inpainting-Diffusion to perform the required task.



I. Success Cases

Bagpack

Initial **In-Painted** **Shifted (to left)**



Stool

Initial **In-Painted** **Shifted (to infront of the chair)**



Wall Hanging

Initial **In-Painted** **Shifted (to the right and above)**



II. Failure Scenarios

Segmentation Mask Issues

Problem → Grounded SAM not identifying the correct object (or creating ideal mask). It either leaves residual pixels (non-marked) of the target object, or includes part of some other object into the mix.

This ends up causing In-Painting diffusion pipeline to pick up draw up weird

Potential Solutions:

Solution 0 → Dialation of the segmentation boundary (Implemented in this project).

- Essentially, dialate the segmentation mask to cover surrounding few pixels (to ensure no pixels of that particular object) are left unsegmented.
- Example:

1. *Solution 1* → Automated improvement approach:
    a. Using "Yolo" (Object Detection) + Ferret (Grounded description) pipeline to generate better descriptions for the object ⇒ Fed into Grounded-SAM
2. *Solution 2* → Fine-tuning the model to improve performance:
    a. SceneGraph descriptions extracted using VLMs (like LLAVA and Ferret) to then be fed for fine-tuning better version of Grounded SAM.
    b. Would also help induce "positional & semantic awareness" in form of prompts like "Wall painting on the left of the curtains and above the chair"

In-Painting Not Reliable

Problems:

1. *Problem 1* → Not all of the pixels are segmented out perfectly, and they end up causing the in-filling task to fill with random objects

2. *Problem 2* → Maybe the model just doesn't get the context of the background well enough

Solutions:

1. *Solution 1* (automated detection) → Put in a Regional Object Detection check (using maybe Grounding Dino or something else) to check whether there's an object present in the bounding box region for the original object.
2. *Solution 2* → Re-Ranker kind of approach:
   a. Extract 5 different variations of the "inpainted image". With Re-Ranker trained on evaluating which image has actually inpainted correctly.
   b. Would be interesting to try out an approach similar to Self-Consistency from LLM space, but I suspect that would actually degrade performance, because it is more likely for the model to produce more wrong examples than correct ones.

Failure Examples:

Created Image (with Shifted Object) Semantically Incorrect

Problems:

1. *Problem 1* → The repositioned location of the object isn't always semantically meaningful. A good solution would make minor adjustments to the image (in the diffusion process), to return a semantically correct image with the user's specifications included.
2. *Problem 2* → The basic shifting ends up causing issues if the original image did not contain complete object. Ideally, a diffusion based conditional genertion pipeline should be able to "complete" the incomplete object.

Solution:

Automated Detection Pipeline (of Semantic Incongruity):

1. Heuristic based approach →
    a. Depth model to extract the depth of items.
    b. Bunch of rules to ensure no abrupt changes in the depth, etc.
2. VLM based →
    a. Simple prompting → (on SOTA VLMs) like "what's wrong with this image?" followed by "create sharp descriptions of updates needed (in object positions) to make it more realistic"
    b. Fine-tuning LLAVA → Purposefully corrupt the image to create [Corrupted Image <> Correction Text] pairs to fine-tune LLAVA (or similar) models on.

Diffusion Re-Generation Pipeline (to fix Images):

- Re-feading into the Diffusion pipeline with updated prompt describing the "required adjustment to the image"

Failure Examples:

III. Future Directions

1. *"Sam-Like" scribble-point-bb based Diffusion Foundation model* to make adjustments to the input image (without having to describe it)
    a. Included task capabilities → Scribble to [Add, Delete, Rotate, Move, complete, change (with same object), etc.] while ensuring the semantics don't break.

2. *Image Sharpening* → Better models tuned on sharpening images while taking specific prompts on region to improve upon