

```

1 import os
2 import json
3 import requests
4 import time
5 import sys
6 import shutil
7
8 import glob
9 import pickle
10 import torch
11 import safetensors.torch
12 import PIL
13 import numpy as np
14 import matplotlib.pyplot as plt
15 from torchvision import transforms
16
17

1
2
3 def ensure_folder_exists(folder_path):
4     if not os.path.exists(folder_path):
5         os.makedirs(folder_path) # 自動建立資料夾
6
7

1 prefixes = [ "LoRA_Trigger", "LoRA_noTrigger", "noLoRA_Trigger", "noLoRA_noTrigger" ]
2

1 def rename_files():
2     base_path = "..\\data\\input"
3     for p in prefixes:
4         folder_path = os.path.join(base_path, p)
5         ensure_folder_exists(folder_path)
6         counter = 0
7         for file_name in os.listdir(folder_path):
8             src = os.path.join(folder_path, file_name)
9             dst = os.path.join(folder_path, p + "_" + str(counter) + ".png")
10            os.rename(src, dst)
11            counter += 1
12

1 rename_files()

FileExistsError                                     Traceback (most recent call last)
Cell In[70], line 1
----> 1 rename_files()

Cell In[69], line 10, in rename_files()
    8 src = os.path.join(folder_path, file_name)
    9 dst = os.path.join(folder_path, p + "_" + str(counter) + ".png")
--> 10 os.rename(src, dst)
    11 counter += 1

FileExistsError: [WinError 183] 當檔案已存在時，無法建立該檔案。: '..\\data\\input\\LoRA_Trigger\\LoRA_Trigger_10.png' -> '..\\data\\input\\LoRA_Trigger\\LoRA_Trigger_2.png'

1 # read api from json file
2 api = {}
3 with open("anaysis with LoRA api.json", "r") as f:
4     api = f.read()
5 api = json.loads(api)
6
7 trigger_prompt = "zzMimikyu, solo, pokemon (creature), no humans, animal focus, mimikyu, "
8 object_prompt = "a cute rat"

```

```

1 class Config:
2     def __init__(self, model, LoRA, input_image, trigger_prompt, object_prompt, file_prefix, total_steps):
3         self.model = model
4         self.LoRA = LoRA
5         self.input_image = input_image
6         self.trigger_prompt = trigger_prompt
7         self.object_prompt = object_prompt
8         self.file_prefix = file_prefix
9         self.total_steps = total_steps
10        self.current_step = current_step
11
12    def add_step(self):
13        self.current_step += 1
14
15    def set_input_image(self, input_image):
16        self.input_image = input_image
17
18 config = Config(
19    model="dreamshaper_8.safetensors",
20    LoRA="Mimikyu_Pokemon_PDXL.safetensors",
21    input_image="input_image.png",
22    trigger_prompt=trigger_prompt,
23    object_prompt=object_prompt,
24    file_prefix=prefixes,
25    total_steps=20,
26    current_step=0
27 )

```

1 開始使用 AI 編寫或生成程式碼。

```

1
2
3 def modify_api(api, config, counter = 0):
4     api["prompt"]["3"]["inputs"]["ckpt_name"] = config.model
5     api["prompt"]["2"]["inputs"]["lora_name"] = config.LoRA
6     api["prompt"]["7"]["inputs"]["image"] = config.input_image
7     api["prompt"]["4"]["inputs"]["text"] = config.trigger_prompt + config.object_prompt
8     api["prompt"]["11"]["inputs"]["filename_prefix"] = f'{config.file_prefix}/{config.object_prompt}/'
9     api["prompt"]["15"]["inputs"]["filename_prefix"] = f'{config.file_prefix}/{config.object_prompt}/'
10    api["prompt"]["10"]["inputs"]["steps"] = config.total_steps
11    api["prompt"]["10"]["inputs"]["denoise"] = 1-config.current_step/ config.total_steps
12    api["prompt"]["1"]["inputs"]["start_at_step"] = config.current_step
13    api["prompt"]["1"]["inputs"]["end_at_step"] = config.current_step + 1
14    return api

```

```

1 import shutil
2 output_folder = "..\\..\\..\\myComfyUI\\output\\lab\\hspace"
3 for filename in os.listdir(output_folder):
4     file_path = os.path.join(output_folder, filename)
5     try:
6         if os.path.isfile(file_path) or os.path.islink(file_path):
7             os.unlink(file_path)
8         elif os.path.isdir(file_path):
9             shutil.rmtree(file_path)
10    except Exception as e:
11        print(f'Failed to delete {file_path}. Reason: {e}')

```

```

1 import shutil
2
3
4 generate_counter = 1
5
6 for prefix in prefixes:
7     config.file_prefix = prefix
8     object_prompt = "a cute rat"
9     config.object_prompt = object_prompt

```

```

10     for input_image in os.listdir(f"..\..\..\..\myComfyUI\input"):
11         if prefix not in input_image:
12             continue
13         print(f"{prefix} {object_prompt} {input_image}")
14         config.set_input_image(input_image)
15         config.current_step = 0
16         for i in range(20-1):
17             config.current_step = i
18             api = modify_api(api, config, generate_counter)
19             generate_counter += 1
20             send_request(api)
21             print(f"{prefix} {object_prompt} {i} done")

```

LoRA\_Trigger a cute rat LoRA\_Trigger\_0.png  
200  
{'prompt\_id': '2641f5ad-a417-4b3b-8065-8e62022aba7e', 'number': 2028, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 0 done  
200  
{'prompt\_id': 'c685179a-9489-4e18-ba71-6513577d8e7d', 'number': 2029, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 1 done  
200  
{'prompt\_id': 'b4f0d604-c663-4e3a-bdf9-451fc359521c', 'number': 2030, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 2 done  
200  
{'prompt\_id': '3bbfdab7-6097-46c5-9934-ce6a524283a8', 'number': 2031, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 3 done  
200  
{'prompt\_id': 'd194d65e-9c16-4f40-8965-ebdce967da35', 'number': 2032, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 4 done  
200  
{'prompt\_id': '7307cec1-5092-4643-add0-7b6b4473afa7', 'number': 2033, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 5 done  
200  
{'prompt\_id': '1032ed72-2d5a-4e04-8da6-842b399140f4', 'number': 2034, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 6 done  
200  
{'prompt\_id': 'b3ae58a1-8bb5-411b-b938-fa0fe341d183', 'number': 2035, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 7 done  
200  
{'prompt\_id': '4f3dbc80-4534-4515-bd84-b9a76312271e', 'number': 2036, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 8 done  
200  
{'prompt\_id': 'a5ba8b2d-af16-46c5-8f79-97d8188d0a7e', 'number': 2037, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 9 done  
200  
{'prompt\_id': 'e0957744-04eb-4997-91d9-bef562b4e0ea', 'number': 2038, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 10 done  
200  
{'prompt\_id': 'ccf8a222-e98a-40c5-b6ad-0a55fadfac99', 'number': 2039, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 11 done  
200  
{'prompt\_id': '90b73e65-7acb-4154-8fb9-e2bb44db3ca5', 'number': 2040, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 12 done  
200  
{'prompt\_id': 'fdb1b218-6caf-414a-bfc2-6481749a9cab', 'number': 2041, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 13 done  
200  
{'prompt\_id': '4f322e67-0f75-481d-8381-437b047bb7b9', 'number': 2042, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 14 done  
200  
{'prompt\_id': 'e372a8b1-02b1-4a04-afe9-0619cd3fbaab', 'number': 2043, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 15 done  
200  
{'prompt\_id': 'ebf40bbd-3ebe-4015-98d2-72a6637752aa', 'number': 2044, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 16 done  
200  
{'prompt\_id': 'fc9adb8d-35e6-4664-9c46-95acf7908a71', 'number': 2045, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 17 done  
200  
{'prompt\_id': '46a968c1-0461-42d5-92a8-7b2466e62146', 'number': 2046, 'node\_errors': {}}  
LoRA\_Trigger a cute rat 18 done

```

1 import json
2 import requests
3 def send_request(api):
4     # 3. 設定 API 的 URL 與 headers
5     url = 'http://127.0.0.1:8188/prompt'
6     headers = {'Content-Type': 'application/json'}
7
8     # 4. 發送 POST request 並傳送修改後的 JSON 資料
9     response = requests.post(url, headers=headers, data=json.dumps(api))
10
11    # 5. 輸出回應結果

```

```
12     print(response.status_code)
13     print(response.json())
```

```
1 config2 = Config(
2     model="dreamshaper_8.safetensors",
3     LoRA="Mimikyu_Pokemon_PDXL.safetensors",
4     input_image="noLoRA_Trigger_5.png",
5     trigger_prompt=trigger_prompt,
6     object_prompt=object_prompt,
7     file_prefix=prefixs[0],
8     total_steps=20,
9     current_step=15
10 )
11 send_request(modify_api(api, config2, 18))
```

```
200
{'prompt_id': 'e6332703-ab44-4d67-8ab5-5be461ee987c', 'number': 2027, 'node_errors': {}}
```

```
1 hspace_folder = "..\\..\\..\\..\\myComfyUI\\output\\lab\\hspace"
2 data_folder = "..\\data"
3
4 for prefix in os.listdir(data_folder):
5     for prompt in os.listdir(os.path.join(data_folder, prefix)):
6         for image_name in os.listdir(os.path.join(data_folder, prefix, prompt)):
7             for file_name in os.listdir(os.path.join(data_folder, prefix, prompt, image_name)):
8                 if '@' in file_name:
9                     index = file_name.split('@')[1]
10                    print(index)
11                    hspace_file = os.path.join(hspace_folder, index+".pkl")
12                    destination = os.path.join(data_folder, prefix, prompt, image_name)
13                    # if hspace file exist
14                    if os.path.exists(hspace_file):
15                        shutil.move(hspace_file, destination)
16                    else:
17                        print(f"{hspace_file} not exist")
18
19
20
21
22
23
```

```
609
...\\..\\..\\myComfyUI\\output\\lab\\hspace\\609.pkl not exist
619
...\\..\\..\\myComfyUI\\output\\lab\\hspace\\619.pkl not exist
620
...\\..\\..\\myComfyUI\\output\\lab\\hspace\\620.pkl not exist
621
622
623
624
625
626
627
610
611
612
613
614
615
616
617
618
628
638
639
640
641
642
643
644
645
646
629
630
```

```
631
632
633
634
635
636
637
647
657
658
659
660
661
662
663
664
665
648
649
650
651
652
653
654

1 class read_file():
2     def __init__(self, file_path):
3         pass
4
5     def load_hspace_file(filepath):
6         """讀取 .pkl 檔案，並將 numpy.ndarray 轉換為 numpy array"""
7         with open(filepath, 'rb') as f:
8             arr = pickle.load(f)
9         return arr
10
11    def load_latent_file(filepath):
12        """利用 safetensors.torch 讀取 .latent 檔案，並取出 'latent_tensor'"""
13        sd = safetensors.torch.load_file(filepath)
14        return sd['latent_tensor'].numpy()
15
16    def load_image_file(filepath):
17        """讀取圖片檔案，並轉換為 numpy array"""
18        img = PIL.Image.open(filepath)
19        return np.array(img)
20

1 experiment_data = {}
2 for prefix in os.listdir(data_folder):
3     experiment_data[prefix] = {}
4     for prompt in os.listdir(os.path.join(data_folder, prefix)):
5         experiment_data[prefix][prompt] = {}
6         for image_name in os.listdir(os.path.join(data_folder, prefix, prompt)):
7             experiment_data[prefix][prompt][image_name] = {}
8             experiment_data[prefix][prompt][image_name]["hspace"] = []
9             experiment_data[prefix][prompt][image_name]["image"] = []
10            experiment_data[prefix][prompt][image_name]["latent"] = []
11
12            start_index = 999999999
13            for file_name in os.listdir(os.path.join(data_folder, prefix, prompt, image_name)):
14                if '@' in file_name:
15                    print(file_name)
16                    index = int(file_name.split('@')[1])
17                    start_index = min(start_index, index)
18
19
20            for i in range(20-1):
21                hspace_file = os.path.join(data_folder, prefix, prompt, image_name, f"{start_index+i}")
22                image_file = os.path.join(data_folder, prefix, prompt, image_name, f"{i}@{start_index}")
23                latent_file = os.path.join(data_folder, prefix, prompt, image_name, f"{i}_00001.lat")
24
25                if os.path.exists(hspace_file) and os.path.exists(image_file) and os.path.exists(latent_file):
26                    experiment_data[prefix][prompt][image_name]["hspace"].append(read_file.load_hspace_file(hspace_file))
27                    experiment_data[prefix][prompt][image_name]["image"].append(read_file.load_image_file(image_file))
28                    experiment_data[prefix][prompt][image_name]["latent"].append(read_file.load_latent_file(latent_file))
```

```

29         else:
30             if not os.path.exists(hspace_file):
31                 print(f"{hspace_file} not exist")
32             if not os.path.exists(image_file):
33                 print(f"{image_file} not exist")
34             if not os.path.exists(latent_file):
35                 print(f"{latent_file} not exist")
36
37
38

```

0@609@\_00001\_.png  
 10@619@\_00001\_.png  
 11@620@\_00001\_.png  
 12@621@\_00001\_.png  
 13@622@\_00001\_.png  
 14@623@\_00001\_.png  
 15@624@\_00001\_.png  
 16@625@\_00001\_.png  
 17@626@\_00001\_.png  
 18@627@\_00001\_.png  
 1@610@\_00001\_.png  
 2@611@\_00001\_.png  
 3@612@\_00001\_.png  
 4@613@\_00001\_.png  
 5@614@\_00001\_.png  
 6@615@\_00001\_.png  
 7@616@\_00001\_.png  
 8@617@\_00001\_.png  
 9@618@\_00001\_.png  
 0@628@\_00001\_.png  
 10@638@\_00001\_.png  
 11@639@\_00001\_.png  
 12@640@\_00001\_.png  
 13@641@\_00001\_.png  
 14@642@\_00001\_.png  
 15@643@\_00001\_.png  
 16@644@\_00001\_.png  
 17@645@\_00001\_.png  
 18@646@\_00001\_.png  
 1@629@\_00001\_.png  
 2@630@\_00001\_.png  
 3@631@\_00001\_.png  
 4@632@\_00001\_.png  
 5@633@\_00001\_.png  
 6@634@\_00001\_.png  
 7@635@\_00001\_.png  
 8@636@\_00001\_.png  
 9@637@\_00001\_.png  
 0@647@\_00001\_.png  
 10@657@\_00001\_.png  
 11@658@\_00001\_.png  
 12@659@\_00001\_.png  
 13@660@\_00001\_.png  
 14@661@\_00001\_.png  
 15@662@\_00001\_.png  
 16@663@\_00001\_.png  
 17@664@\_00001\_.png  
 18@665@\_00001\_.png  
 1@648@\_00001\_.png  
 2@649@\_00001\_.png  
 3@650@\_00001\_.png  
 4@651@\_00001\_.png  
 5@652@\_00001\_.png  
 6@653@\_00001\_.png  
 7@654@\_00001\_.png  
 8@655@\_00001\_.png  
 9@656@\_00001\_.png  
 0@666@\_00001\_.png

```

1 # turn experiment_data into numpy array
2 for prefix in experiment_data:
3     for prompt in experiment_data[prefix]:
4         for image_name in experiment_data[prefix][prompt]:
5             experiment_data[prefix][prompt][image_name]["hspace"] = np.array(experiment_data[prefix][|]
6             experiment_data[prefix][prompt][image_name]["image"] = np.array(experiment_data[prefix][|]
7             experiment_data[prefix][prompt][image_name]["latent"] = np.array(experiment_data[prefix][|]

1 print(experiment_data.keys())
2 print(experiment_data['LoRA_noTrigger'].keys())
3 print(experiment_data['LoRA_noTrigger']['a cute rat'].keys())
4 print(experiment_data['LoRA_noTrigger']['a cute rat']['LoRA_noTrigger_0.png'].keys())

```

```

5 print(len(experiment_data['LoRA_noTrigger']['a cute rat']['LoRA_noTrigger_0.png']['hspace']))
6 print(len(experiment_data['LoRA_noTrigger']['a cute rat']['LoRA_noTrigger_0.png']['image']))
7 print(len(experiment_data['LoRA_noTrigger']['a cute rat']['LoRA_noTrigger_0.png']['latent']))

→ dict_keys(['LoRA_noTrigger', 'LoRA_Trigger', 'noLoRA_noTrigger', 'noLoRA_Trigger'])
dict_keys(['a cute rat'])
dict_keys(['LoRA_noTrigger_0.png', 'LoRA_noTrigger_1.png', 'LoRA_noTrigger_10.png', 'LoRA_noTrigger_11.png', 'LoRA_noTrigger_12.png'])
dict_keys(['hspace', 'image', 'latent'])

19
19
19

◀ ━━━━━━ ▶

1 # save experiment data
2 with open('experiment_data.pkl', 'wb') as f:
3     pickle.dump(experiment_data, f)

1 # load experiment data
2 with open('experiment_data.pkl', 'rb') as f:
3     experiment_data = pickle.load(f)
4

1 # replace last image data with random data
2 for prefix in experiment_data:
3     for prompt in experiment_data[prefix]:
4         last_image_name = list(experiment_data[prefix][prompt].keys())[-1]
5         print(experiment_data[prefix][prompt][last_image_name]["image"].shape)
6         print(experiment_data[prefix][prompt][last_image_name]["latent"].shape)
7         print(experiment_data[prefix][prompt][last_image_name]["hspace"].shape)
8         print("*"*10)
9         # as random data
10        experiment_data[prefix][prompt][last_image_name]["image"] = np.random.rand(19, 512, 512, 3)
11        experiment_data[prefix][prompt][last_image_name]["latent"] = np.random.rand(19, 1, 4, 64, 64)
12        experiment_data[prefix][prompt][last_image_name]["hspace"] = np.random.rand(19, 2, 1280, 8, 8)
13
14        second_last_image_name = list(experiment_data[prefix][prompt].keys())[-2]
15        # as zero data
16        experiment_data[prefix][prompt][second_last_image_name]["image"] = np.zeros((19, 512, 512, 3))
17        experiment_data[prefix][prompt][second_last_image_name]["latent"] = np.zeros((19, 1, 4, 64, 64))
18        experiment_data[prefix][prompt][second_last_image_name]["hspace"] = np.zeros((19, 2, 1280, 8, 8))
19

→ (19, 512, 512, 3)
(19, 1, 4, 64, 64)
(19, 2, 1280, 8, 8)
*****
(19, 512, 512, 3)
(19, 1, 4, 64, 64)
(19, 2, 1280, 8, 8)
*****
(19, 512, 512, 3)
(19, 1, 4, 64, 64)
(19, 2, 1280, 8, 8)
*****
(19, 512, 512, 3)
(19, 1, 4, 64, 64)
(19, 2, 1280, 8, 8)
*****
```

## 1 開始使用 AI 編寫或生成程式碼。

```

1 def normalize_experiment_data(experiment_data):
2     normalized_data = {}
3     for prefix in experiment_data.keys():
4         normalized_data[prefix] = {}
5         for prompt in experiment_data[prefix].keys():
6             normalized_data[prefix][prompt] = {}
7             for image_name in experiment_data[prefix][prompt].keys():
8                 normalized_data[prefix][prompt][image_name] = {}
9                 for key in ["latent", "image", "hspace"]:
10                     data = experiment_data[prefix][prompt][image_name][key]
11                     min_val = np.min(data)
12                     max_val = np.max(data)
```

```

13             normalized_data[prefix][prompt][image_name][key] = (data - min_val) / (max_val - m
14     return normalized_data
15
16 normalized_experiment_data = normalize_experiment_data(experiment_data)
17 experiment_data = normalized_experiment_data

```

1 開始使用 AI 編寫或生成程式碼。

```

1 print(experiment_data.keys())
2 print(experiment_data['LoRA_noTrigger'].keys())
3 print(experiment_data['LoRA_noTrigger']['a cute rat'].keys())
4 print(experiment_data['LoRA_noTrigger']['a cute rat']['LoRA_noTrigger_0.png'].keys())
5 print(len(experiment_data['LoRA_noTrigger']['a cute rat']['LoRA_noTrigger_0.png']['hspace']))
6 print(len(experiment_data['LoRA_noTrigger']['a cute rat']['LoRA_noTrigger_0.png']['image']))
7 print(len(experiment_data['LoRA_noTrigger']['a cute rat']['LoRA_noTrigger_0.png']['latent']))

8 dict_keys(['LoRA_noTrigger', 'LoRA_Trigger', 'noLoRA_noTrigger', 'noLoRA_Trigger'])
dict_keys(['a cute rat'])
dict_keys(['LoRA_noTrigger_0.png', 'LoRA_noTrigger_1.png', 'LoRA_noTrigger_10.png', 'LoRA_noTrigger_11.png', 'LoRA_noTrigger_12.png'])
dict_keys(['latent', 'image', 'hspace'])
19
19
19

```

```

1 import numpy as np
2
3 def euclidean(p1, p2):
4     return np.linalg.norm(np.array(p1) - np.array(p2))
5
6 def frechet_recursive(ca, P, Q, i, j):
7     if ca[i, j] > -1:
8         return ca[i, j]
9     elif i == 0 and j == 0:
10        ca[i, j] = euclidean(P[0], Q[0])
11    elif i > 0 and j == 0:
12        ca[i, j] = max(frechet_recursive(ca, P, Q, i-1, 0), euclidean(P[i], Q[0]))
13    elif i == 0 and j > 0:
14        ca[i, j] = max(frechet_recursive(ca, P, Q, 0, j-1), euclidean(P[0], Q[j]))
15    elif i > 0 and j > 0:
16        ca[i, j] = max(min(frechet_recursive(ca, P, Q, i-1, j),
17                            frechet_recursive(ca, P, Q, i-1, j-1),
18                            frechet_recursive(ca, P, Q, i, j-1)),
19                            euclidean(P[i], Q[j]))
20    else:
21        ca[i, j] = float('inf')
22    return ca[i, j]
23
24 def frechet_distance(P, Q):
25    ca = np.ones((len(P), len(Q))) * -1
26    return frechet_recursive(ca, P, Q, len(P)-1, len(Q)-1)
27
28
29

1 # see as different step
2
3 # calculate the L2 distance of latent, image, hspace between each image and the first image
4 # also calculate the L1 distance of hspace between each image and the first image
5
6 def calculate_euclidean_distance(experiment_data):
7     analysis_data = {}
8     for prefix in experiment_data.keys():
9         analysis_data[prefix] = {}
10        for prompt in experiment_data[prefix].keys():
11            analysis_data[prefix][prompt] = {}
12            first_image_name = ""
13            for image_name in experiment_data[prefix][prompt].keys():

```

```
14         if first_image_name == "":
15             first_image_name = image_name
16             print(first_image_name)
17             continue
18         anaysis_data[prefix][prompt][image_name] = {"latent": [], "image": [], "hspace": []}
19         for key in ["latent", "image", "hspace"]:
20             for i in range(0, 20-1):
21                 distance = euclidean(experiment_data[prefix][prompt][image_name][key][i], expre
22                 anaysis_data[prefix][prompt][image_name][key].append(distance)
23     return anaysis_data
24 anaysis_data = {"L2_step": {}, "frechet_step": {}, "L1_step": {}}
25 anaysis_data["L2_step"] = calculate_euclidean_distance(experiment_data)
```

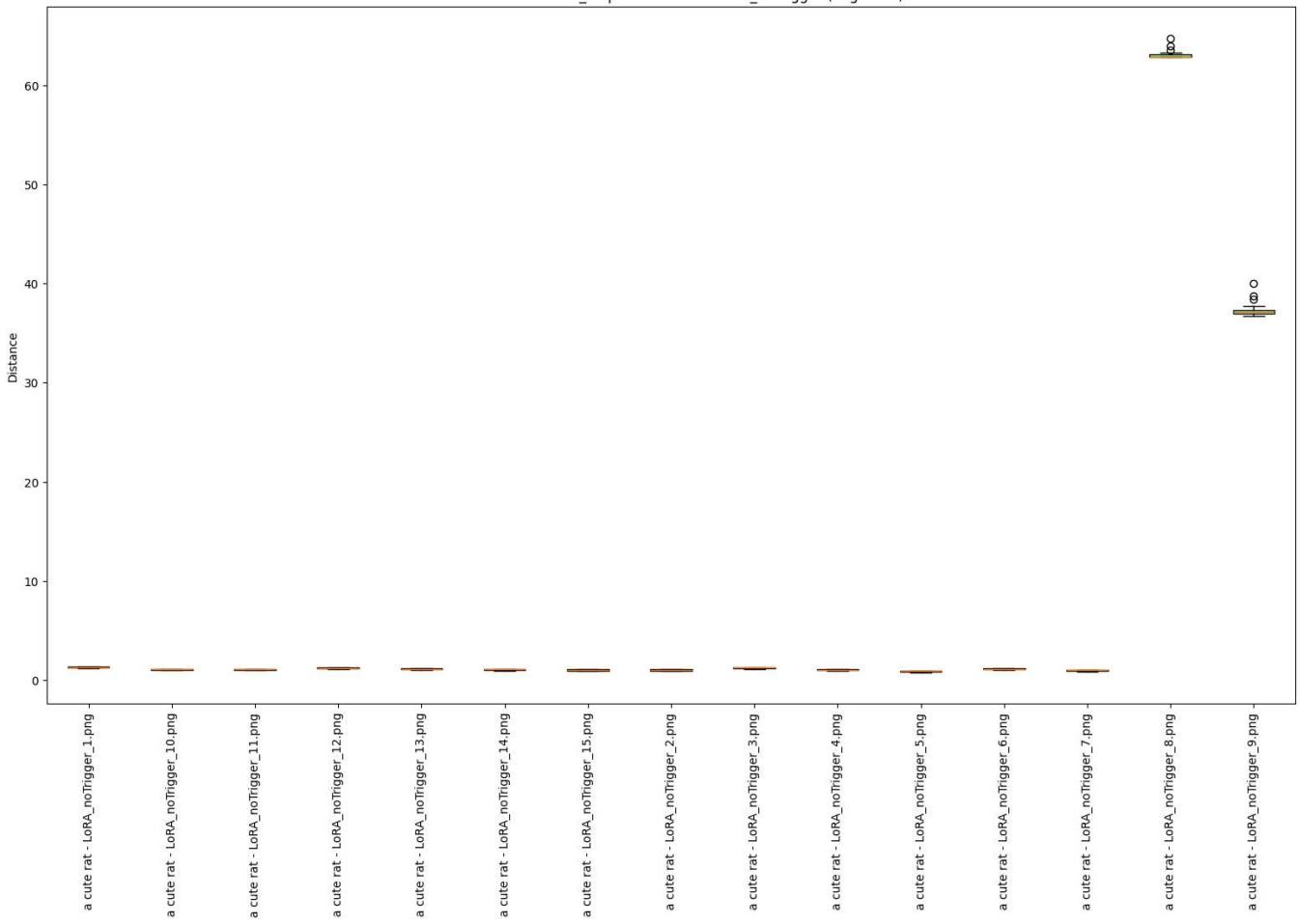
```
↳ LoRA_noTrigger_0.png
LoRA_Trigger_0.png
noLoRA_noTrigger_0.png
noLoRA_Trigger_0.png
```

```
1 print(anaysis_data["L2_step"].keys())
2 print(anaysis_data["L2_step"]['LoRA_noTrigger'].keys())
3 print(anaysis_data["L2_step"]['LoRA_noTrigger']['a cute rat'].keys())
4 print(anaysis_data["L2_step"]['LoRA_noTrigger']['a cute rat']['LoRA_noTrigger_1.png'].keys())
5 print(len(anaysis_data["L2_step"]['LoRA_noTrigger']['a cute rat']['LoRA_noTrigger_1.png']['hspace']))
6 print(len(anaysis_data["L2_step"]['LoRA_noTrigger']['a cute rat']['LoRA_noTrigger_1.png']['image']))
7 print(len(anaysis_data["L2_step"]['LoRA_noTrigger']['a cute rat']['LoRA_noTrigger_1.png']['latent']))
8
9
10 dict_keys(['LoRA_noTrigger', 'LoRA_Trigger', 'noLoRA_noTrigger', 'noLoRA_Trigger'])
11 dict_keys(['a cute rat'])
12 dict_keys(['LoRA_noTrigger_1.png', 'LoRA_noTrigger_10.png', 'LoRA_noTrigger_11.png', 'LoRA_noTrigger_12.png', 'LoRA_noTrigger_13.png'])
13 dict_keys(['latent', 'image', 'hspace'])
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
```

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def plot_boxplot(anaysis_data, key, distance_type="L2_step"):
5     color = {'L2_step': '#99FF99', 'frechet_step': '#99CCFF', 'L1_step': '#FF9999', 'cosine_similarit
6     for prefix in anaysis_data.keys():
7         data_to_plot = []
8         labels = []
9         for prompt in anaysis_data[prefix].keys():
10             for image_name in anaysis_data[prefix][prompt].keys():
11                 data = anaysis_data[prefix][prompt][image_name][key]
12                 data_to_plot.append(data)
13                 labels.append(f'{prompt} - {image_name}')
14
15     plt.figure(figsize=(19.2, 10.8))
16     box = plt.boxplot(data_to_plot, labels=labels, vert=True, patch_artist=True)
17     for patch in box['boxes']:
18         patch.set_facecolor(color)
19
20     avg_distance = np.mean([np.mean(data) for data in data_to_plot])
21     plt.ylabel('Distance')
22     plt.title(f'Box Plot of {key} {distance_type} distances for {prefix} (Avg: {avg_distance:.2f})')
23     plt.xticks(rotation=90)
24     plt.show()
25
26 plot_boxplot(anaysis_data["L2_step"], 'latent', "L2_step")
27 plot_boxplot(anaysis_data["L2_step"], 'image', "L2_step")
28 plot_boxplot(anaysis_data["L2_step"], 'hspace', "L2_step")
```

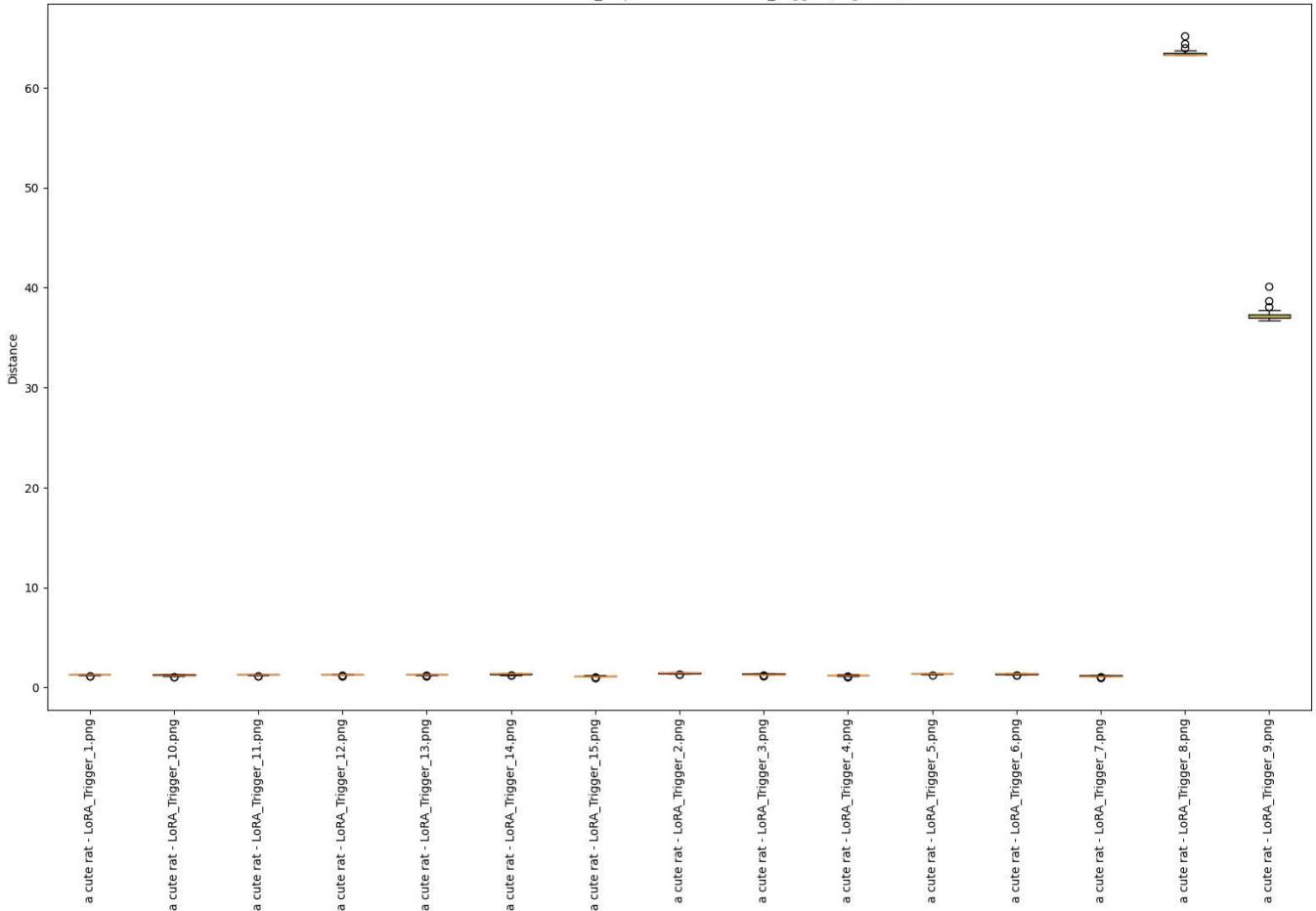
C:\Users\9373k\AppData\Local\Temp\ipykernel\_9684\2151108629.py:16: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() box = plt.boxplot(data\_to\_plot, labels=labels, vert=True, patch\_artist=True)

Box Plot of latent L2\_step distances for LoRA\_noTrigger (Avg: 7.66)

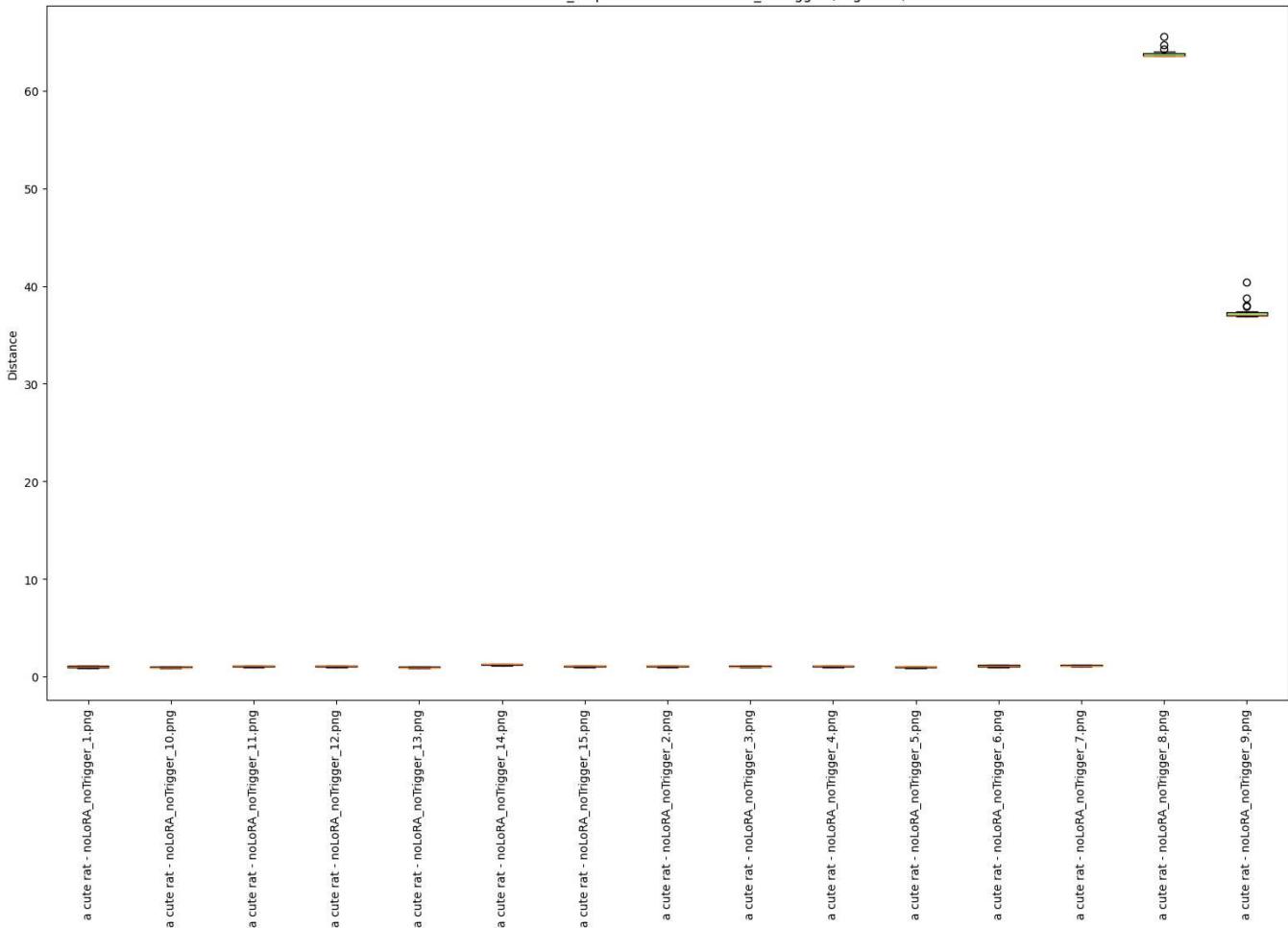


C:\Users\9373k\AppData\Local\Temp\ipykernel\_9684\2151108629.py:16: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() box = plt.boxplot(data\_to\_plot, labels=labels, vert=True, patch\_artist=True)

Box Plot of latent L2\_step distances for LoRA\_Trigger (Avg: 7.83)

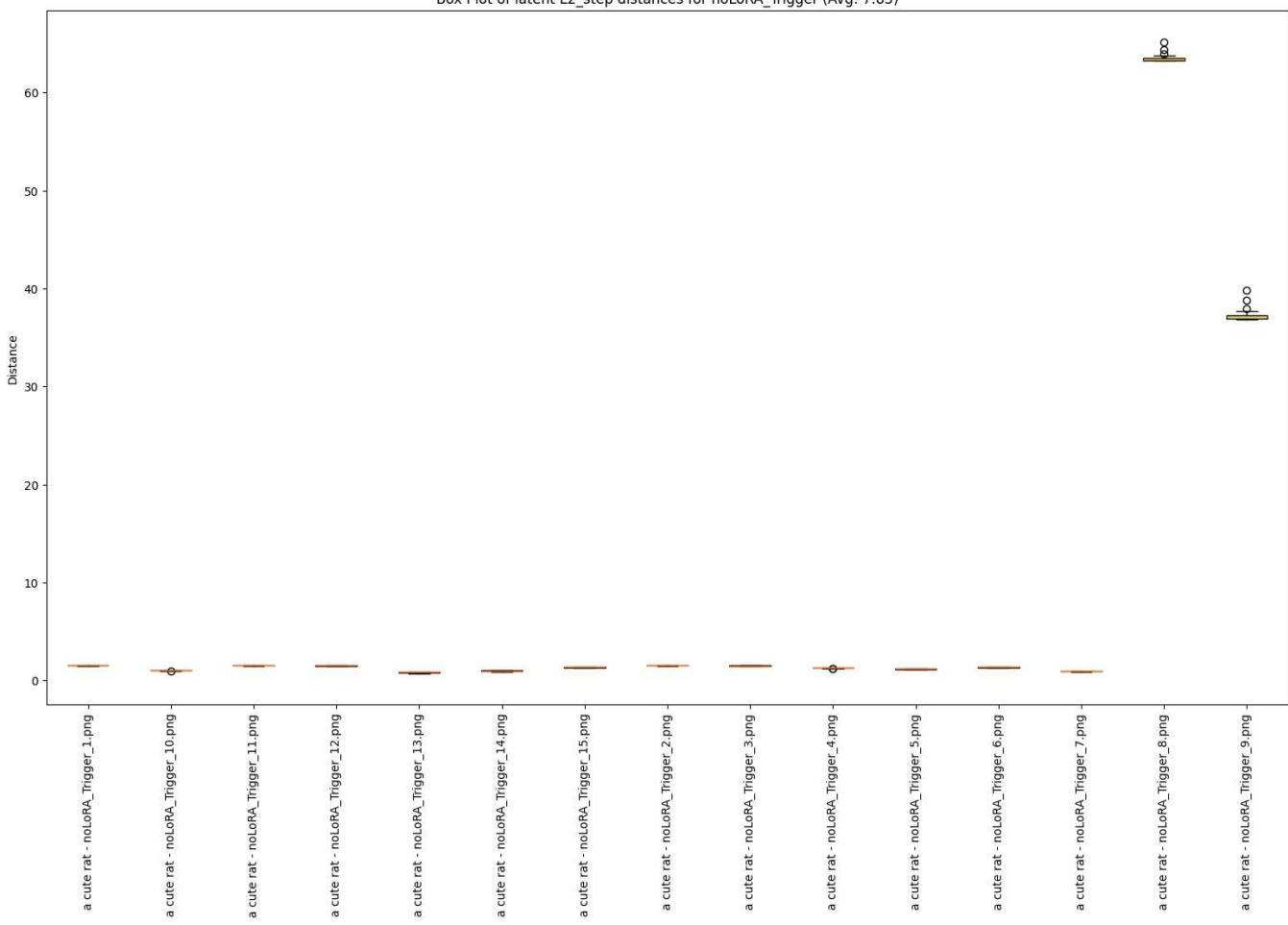


C:\Users\9373k\AppData\Local\Temp\ipykernel\_9684\2151108629.py:16: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() box = plt.boxplot(data\_to\_plot, labels=labels, vert=True, patch\_artist=True)



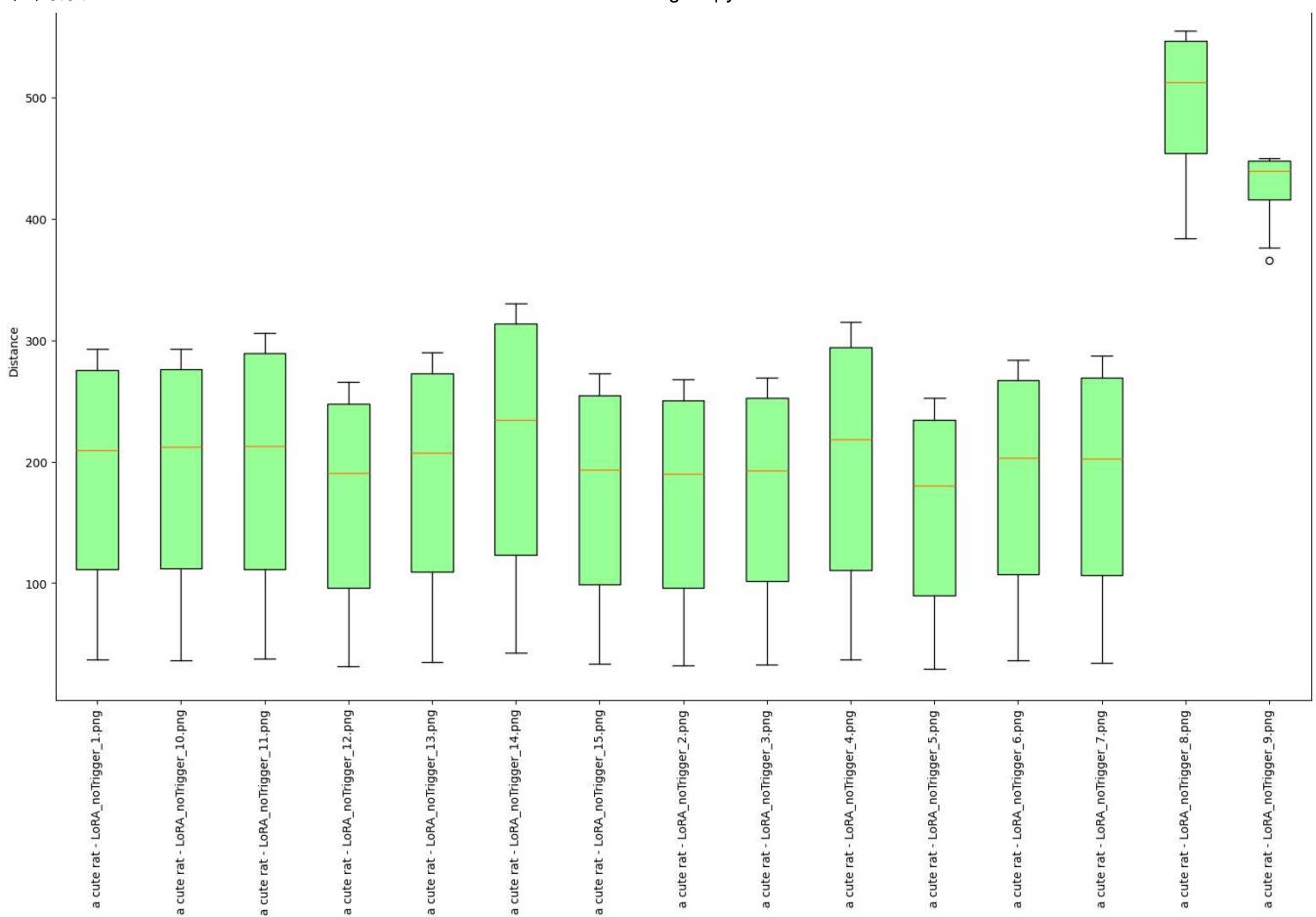
```
C:\Users\9373k\AppData\Local\Temp\ipykernel_9684\2151108629.py:16: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() is deprecated. Use patch_artist=True instead.
```

Box Plot of latent L2\_step distances for noLoRA\_Trigger (Avg: 7.85)

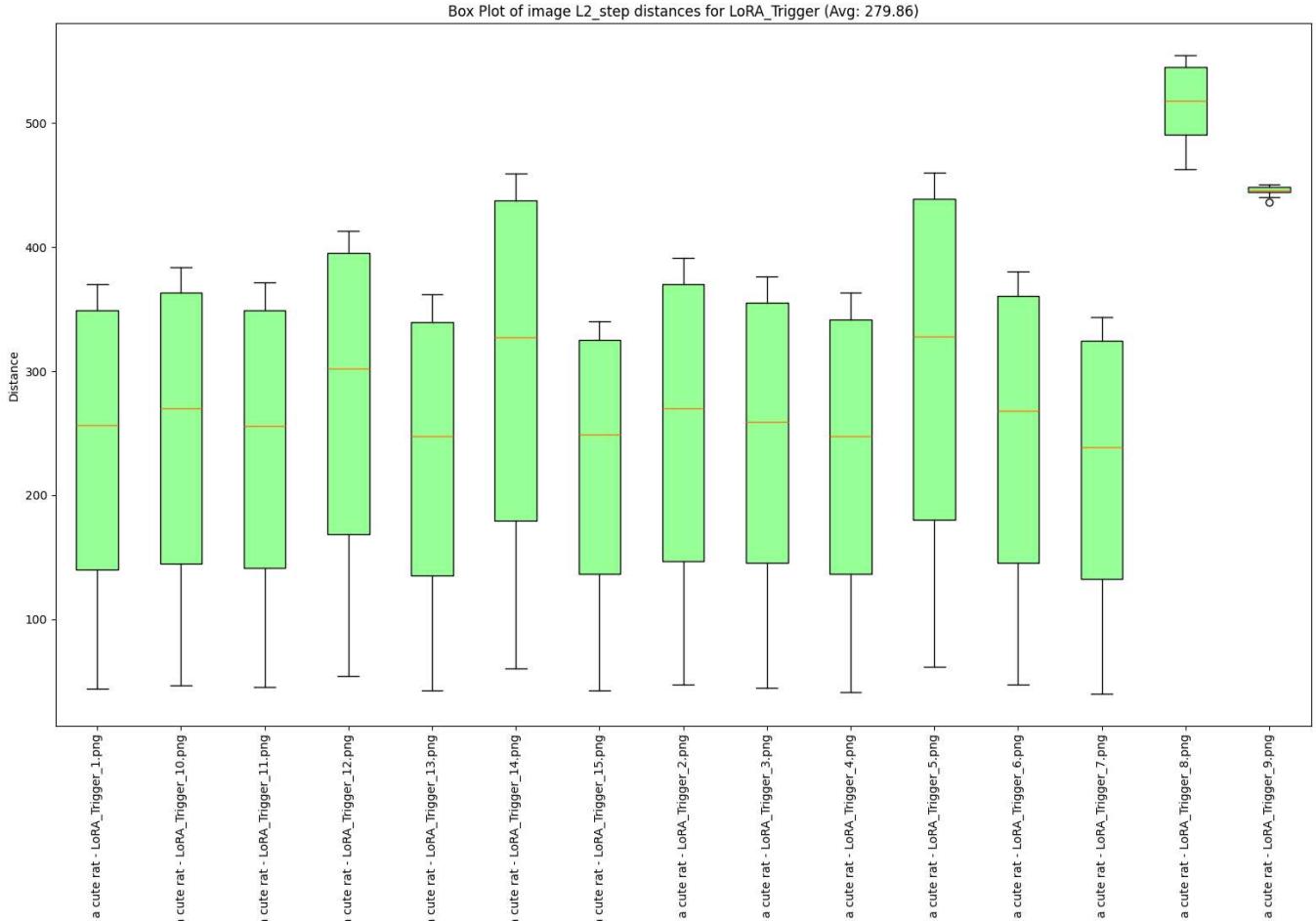


```
C:\Users\9373k\AppData\Local\Temp\ipykernel_9684\2151108629.py:16: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() is deprecated. Use patch_artist=True instead.
```

Box Plot of image L2\_step distances for LoRA\_noTrigger (Avg: 222.09)

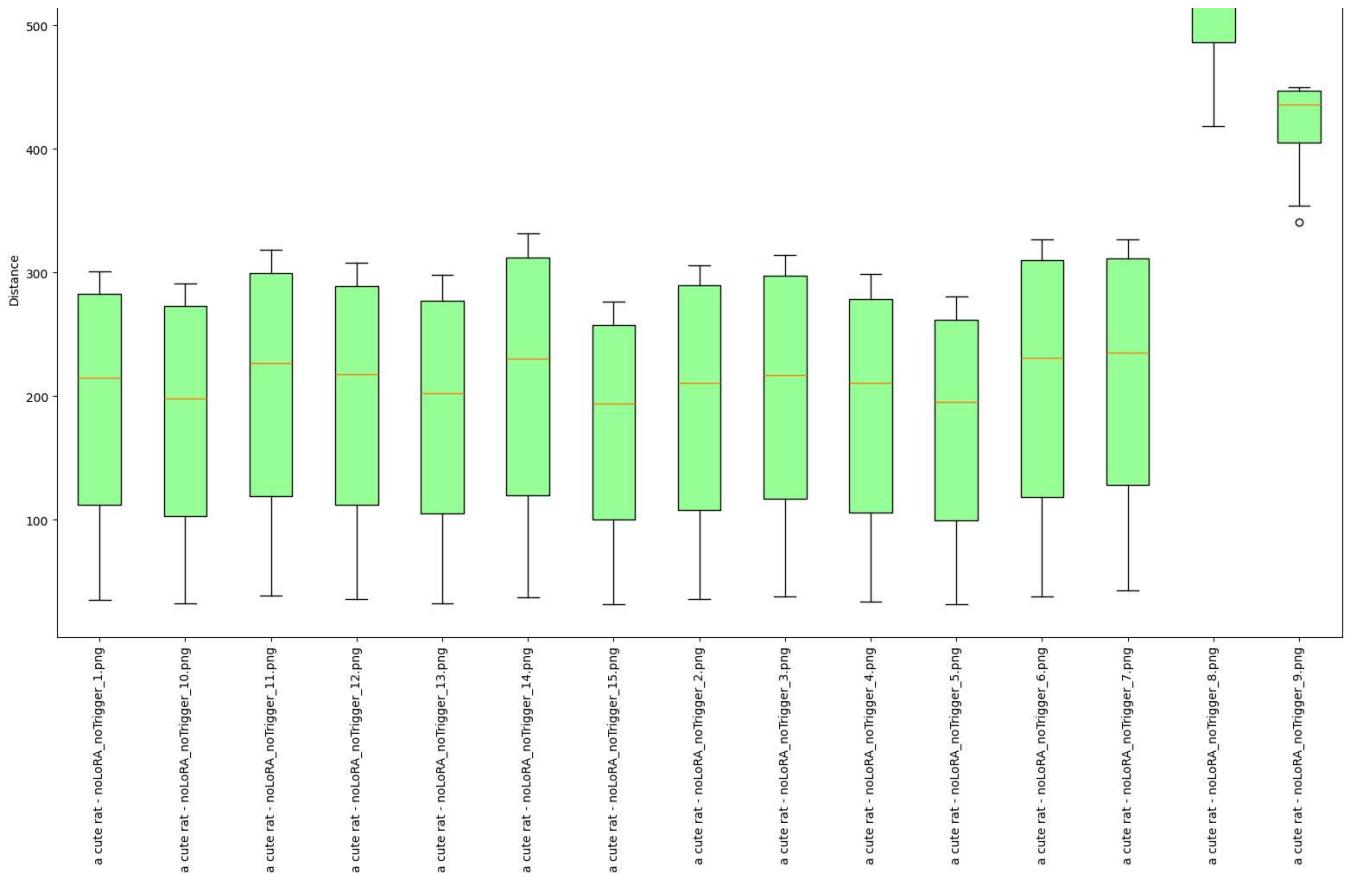


C:\Users\9373k\AppData\Local\Temp\ipykernel\_9684\2151108629.py:16: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() box = plt.boxplot(data\_to\_plot, labels=labels, vert=True, patch\_artist=True)

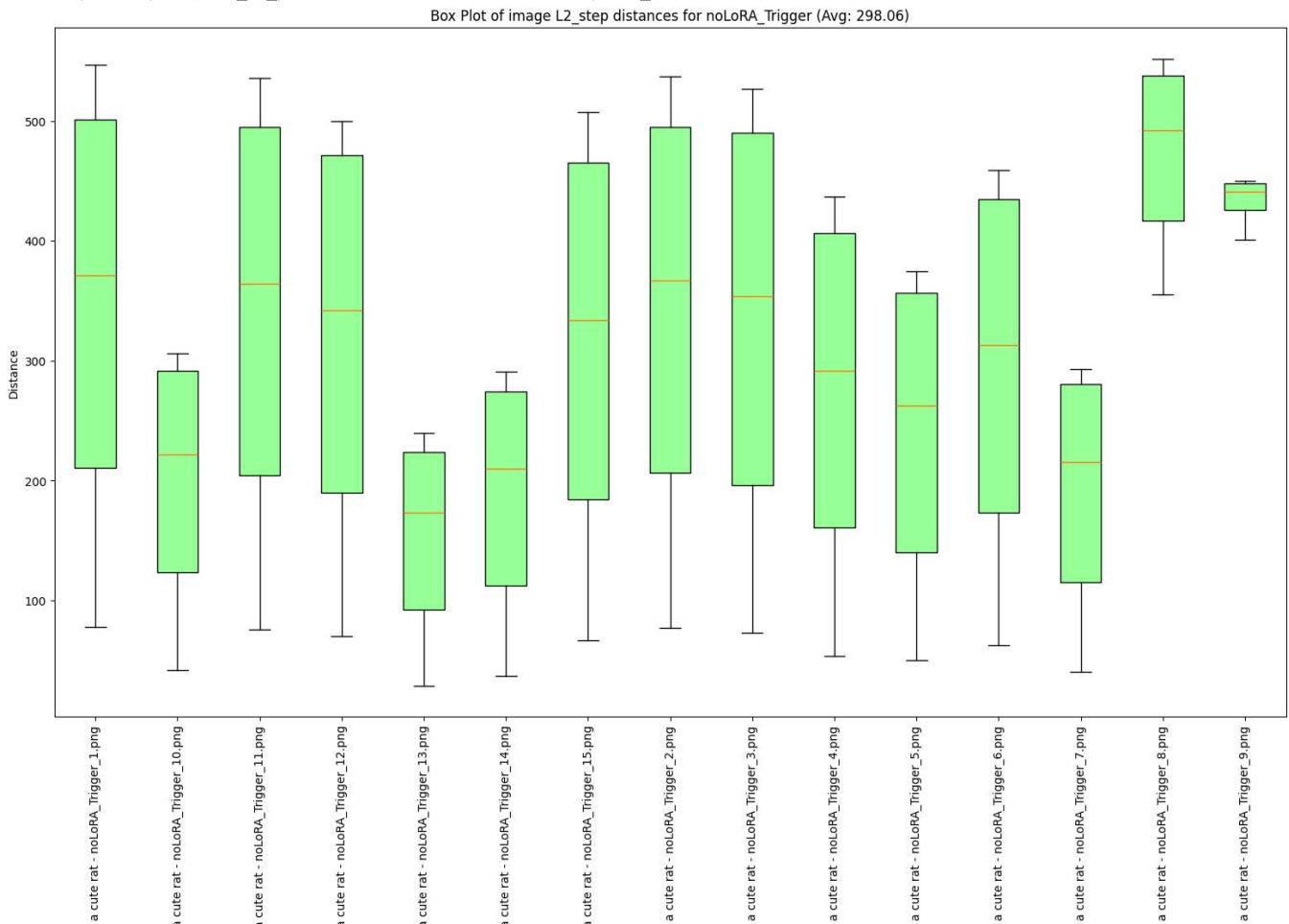


C:\Users\9373k\AppData\Local\Temp\ipykernel\_9684\2151108629.py:16: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() box = plt.boxplot(data\_to\_plot, labels=labels, vert=True, patch\_artist=True)

Box Plot of image L2\_step distances for noLoRA\_noTrigger (Avg: 231.97)

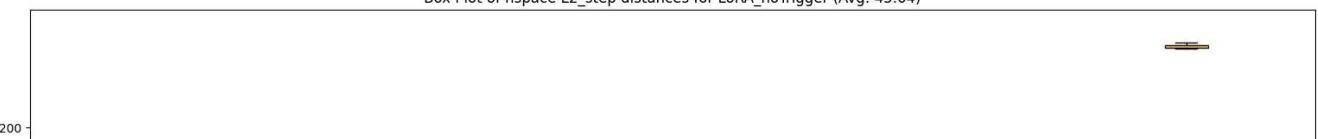


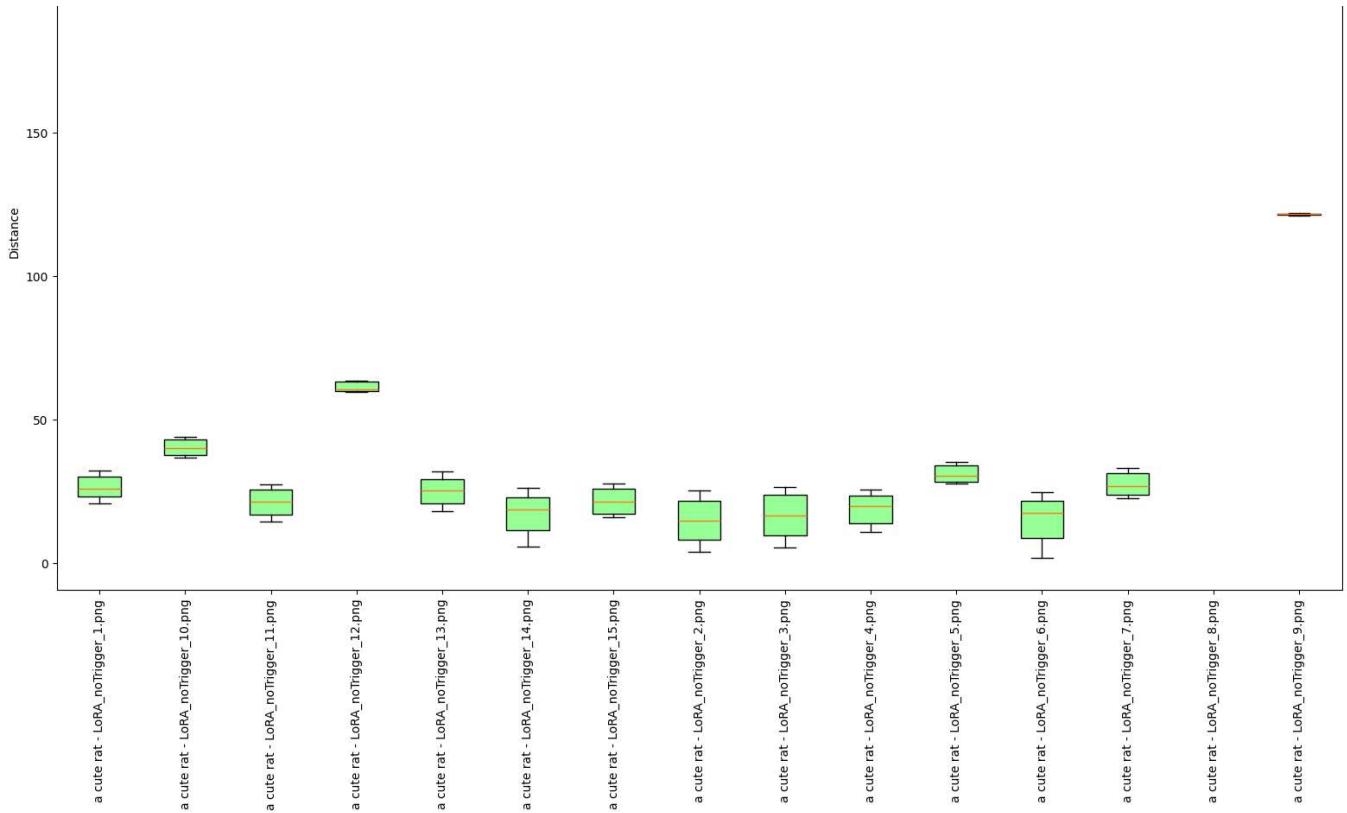
C:\Users\9373k\AppData\Local\Temp\ipykernel\_9684\2151108629.py:16: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() is deprecated. box = plt.boxplot(data\_to\_plot, labels=labels, vert=True, patch\_artist=True)



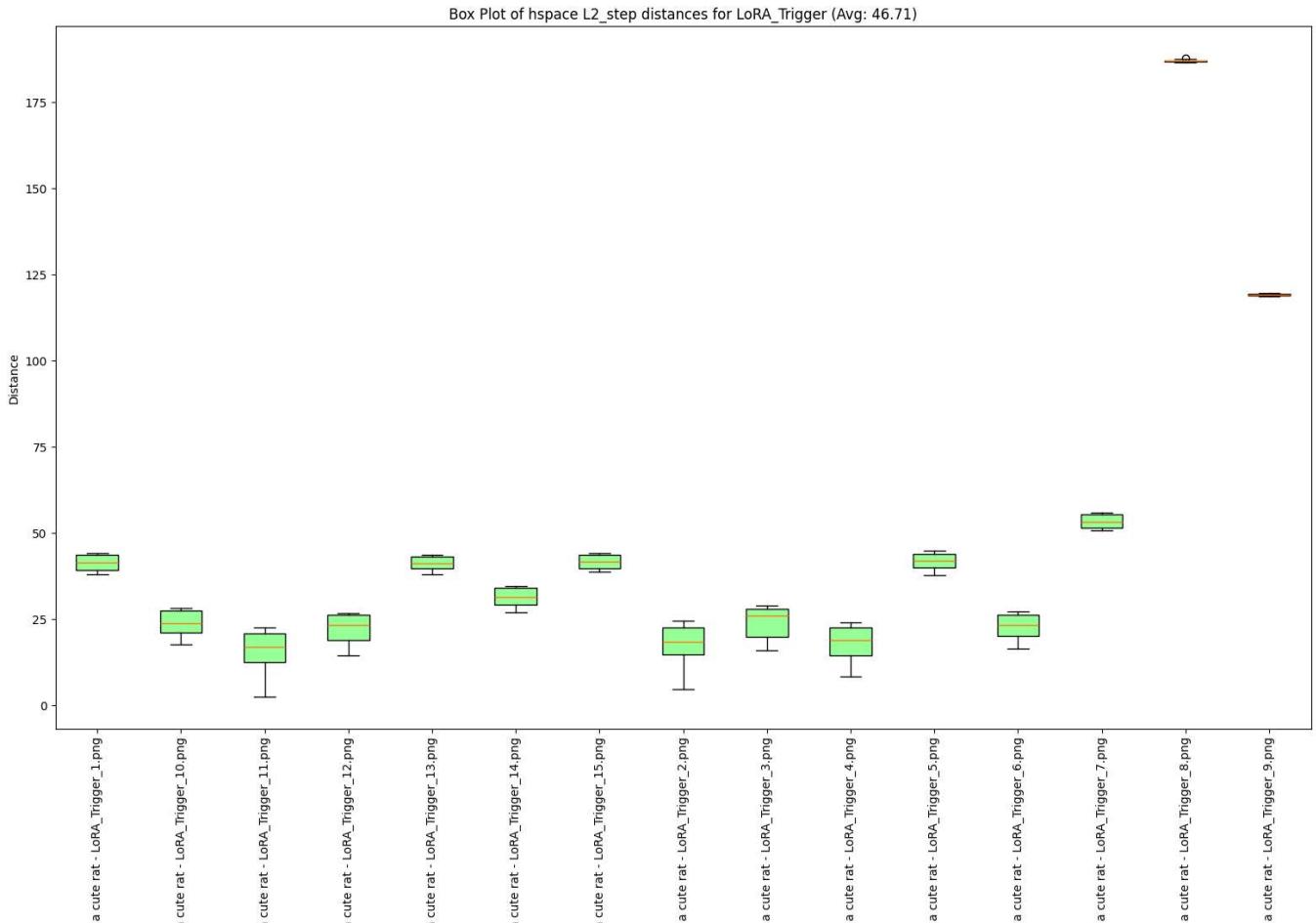
C:\Users\9373k\AppData\Local\Temp\ipykernel\_9684\2151108629.py:16: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() is deprecated. box = plt.boxplot(data\_to\_plot, labels=labels, vert=True, patch\_artist=True)

Box Plot of hspace L2\_step distances for LoRA\_noTrigger (Avg: 45.64)





```
C:\Users\9373k\AppData\Local\Temp\ipykernel_9684\2151108629.py:16: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() is deprecated. Use 'label' instead.
  box = plt.boxplot(data_to_plot, labels=labels, vert=True, patch_artist=True)
```



```
C:\Users\9373k\AppData\Local\Temp\ipykernel_9684\2151108629.py:16: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() is deprecated. Use 'label' instead.
  box = plt.boxplot(data_to_plot, labels=labels, vert=True, patch_artist=True)
```

Box Plot of hspace L2\_step distances for noLoRA\_noTrigger (Avg: 44.77)

