a <A female cyborg> style dog

Python编程与人工智能实践

应用篇：AI绘图

私炉模式（textual inversion，文本反演）

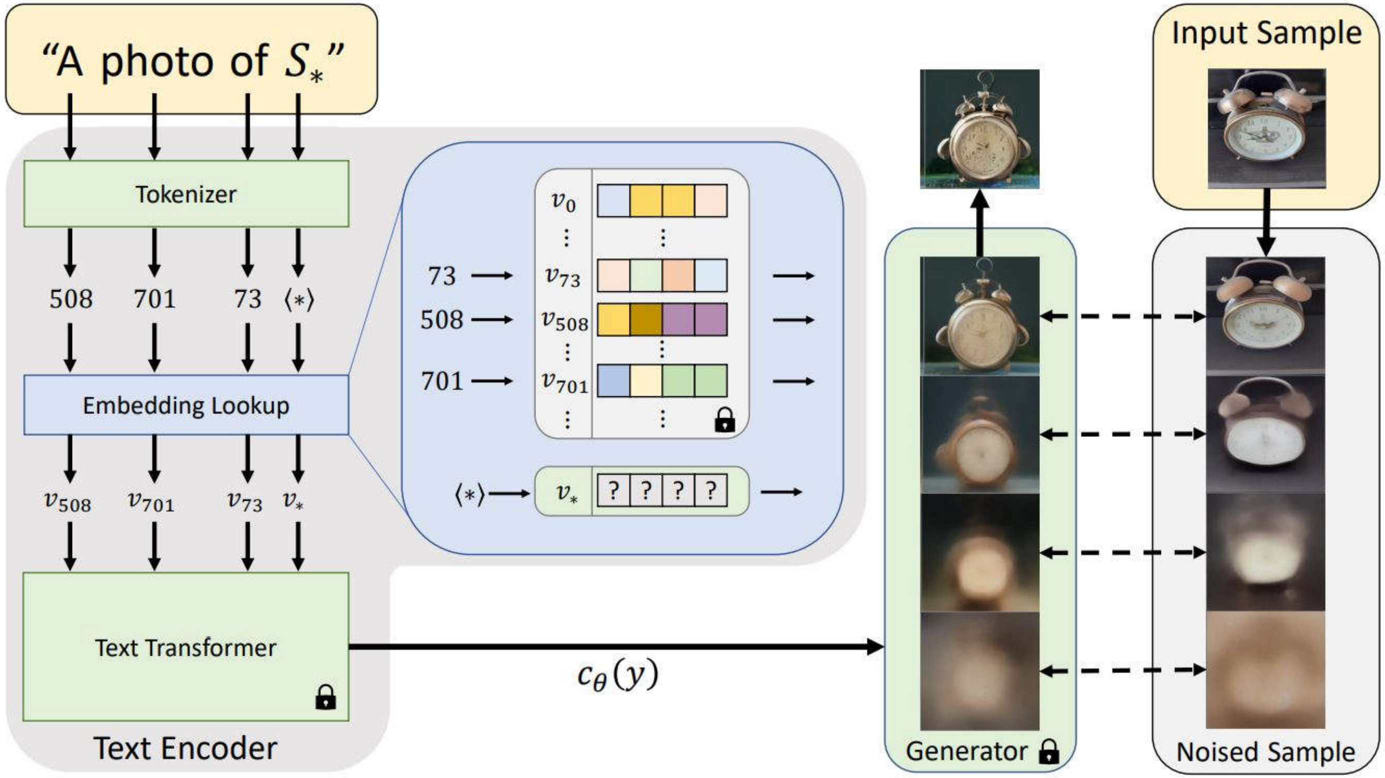
a <Marc_Allante> style dog

于泓
鲁东大学
信息与电气工程学院

2022.10.30

私炉模式

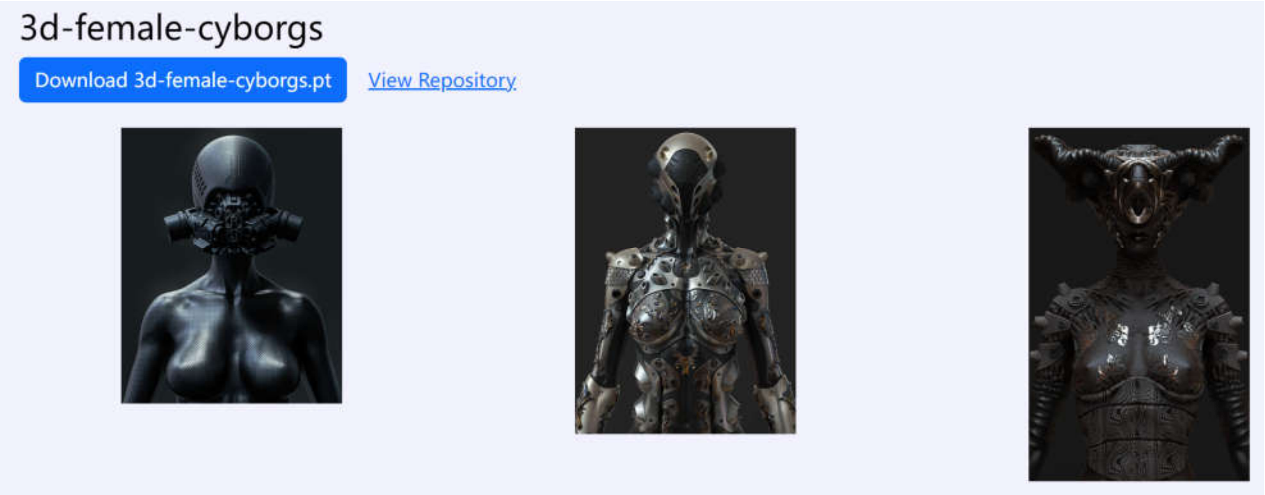
利用3-5张风格图像进行训练，将风格转换为一个词向量



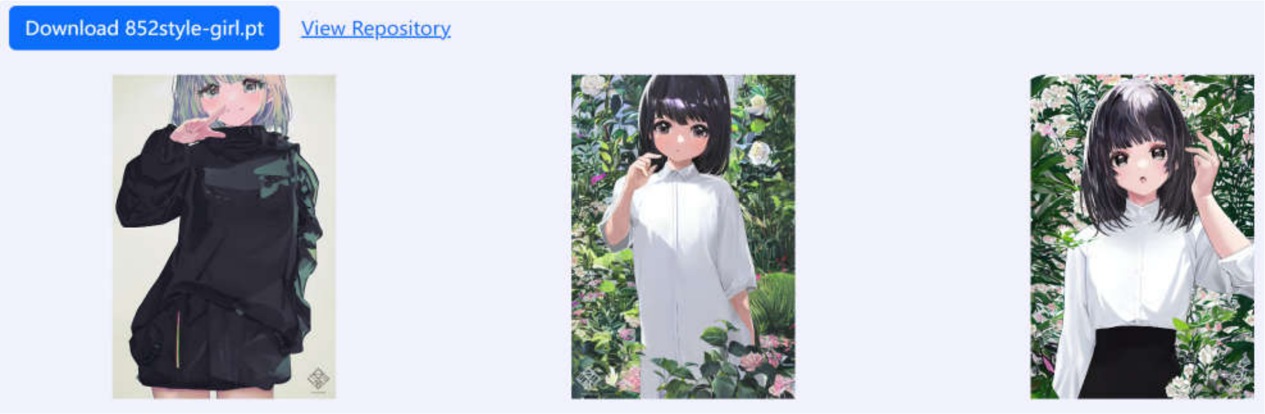
训练过程较慢

https://cyberes.github.io/stable-diffusion-textual-inversion-models/

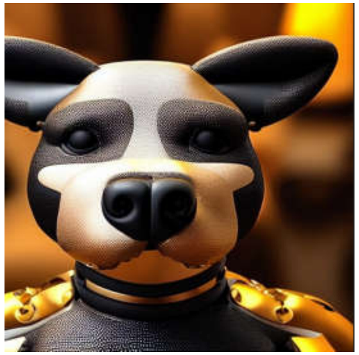
一些训练好的 词向量



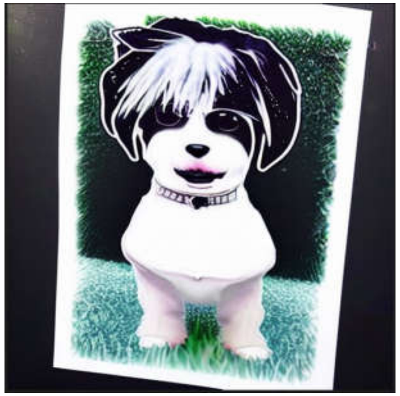
3d-female-cyborgs

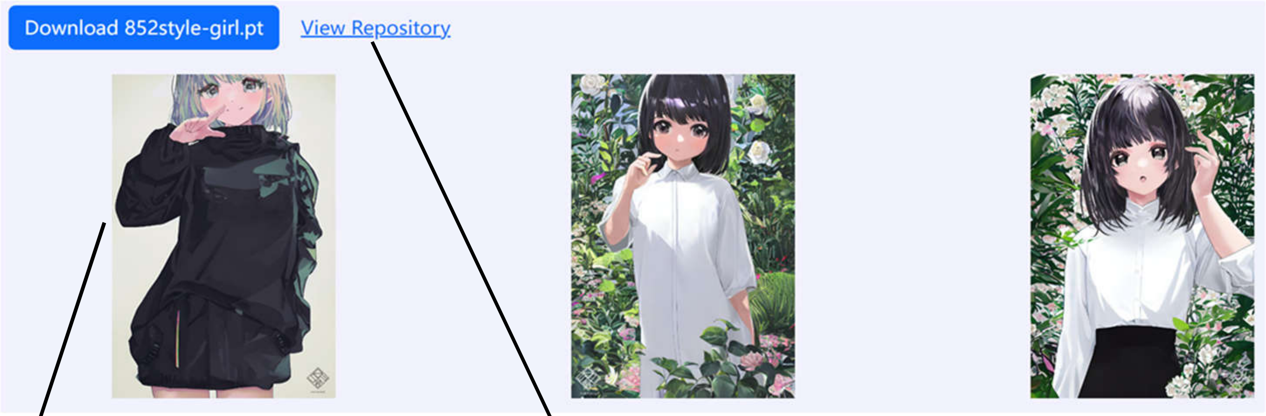


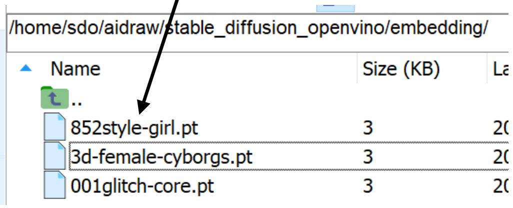
a **<A female cyborg>** style dog



" **<852style-girl>** style dog"

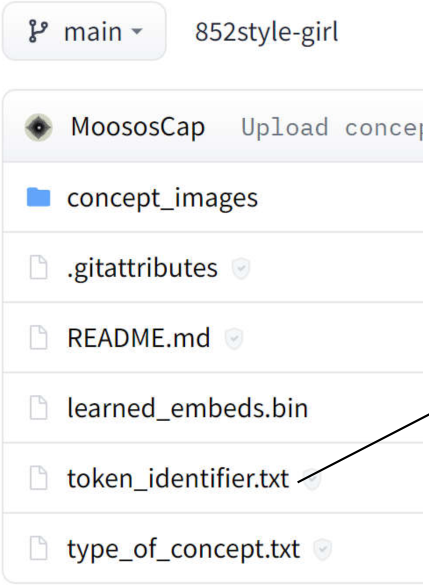






/home/sdo/aidraw/stable_diffusion_openvino/embedding/

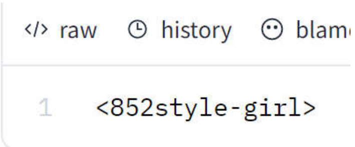
Name	Size (KB)	La
..		
852style-girl.pt	3	20
3d-female-cyborgs.pt	3	20
001glitch-core.pt	3	20



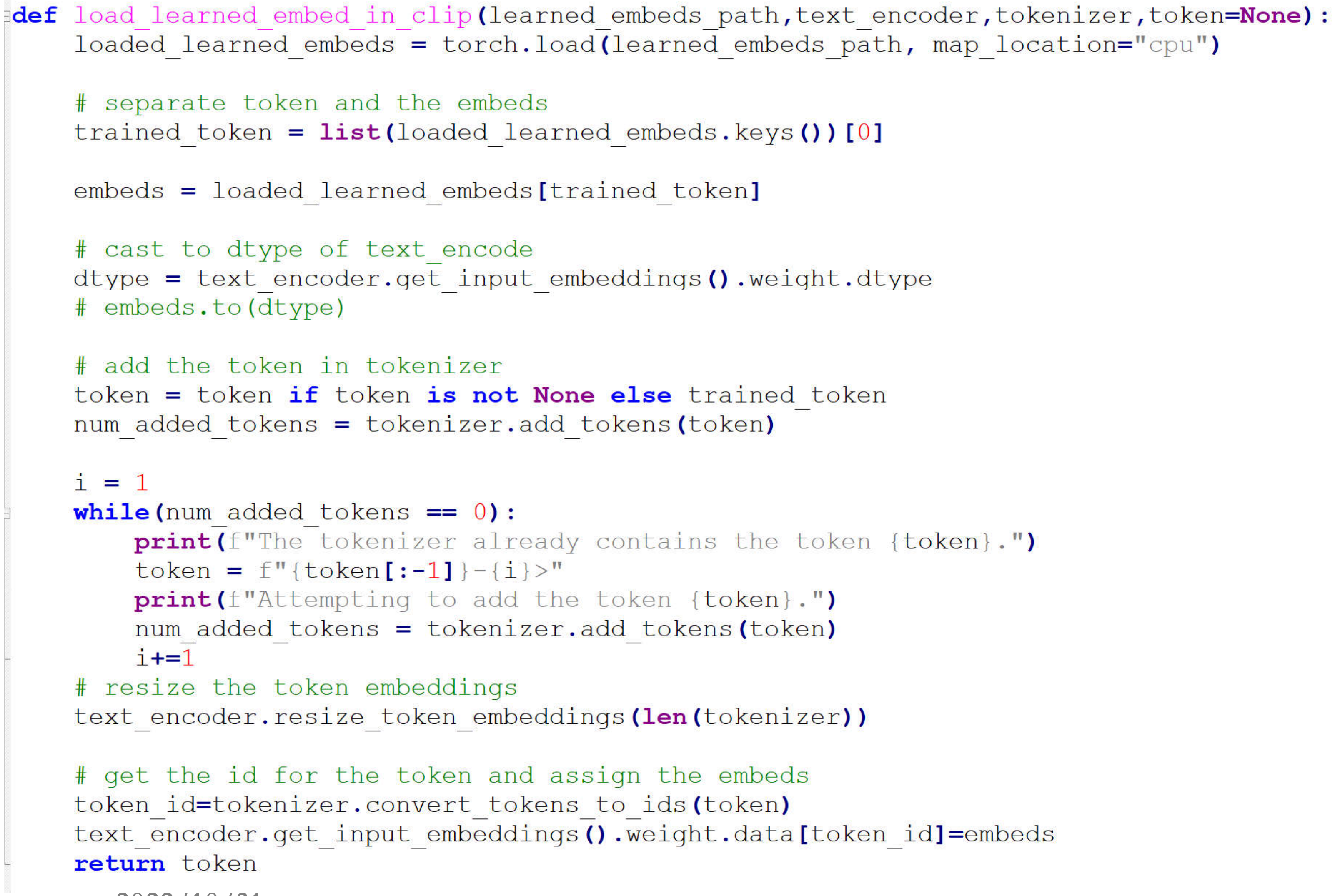
main 852style-girl

MoososCap Upload concep

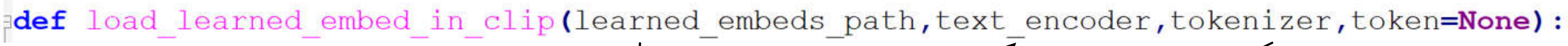
- concept_images
- .gitattributes
- README.md
- learned_embeds.bin
- token_identifier.txt
- type_of_concept.txt



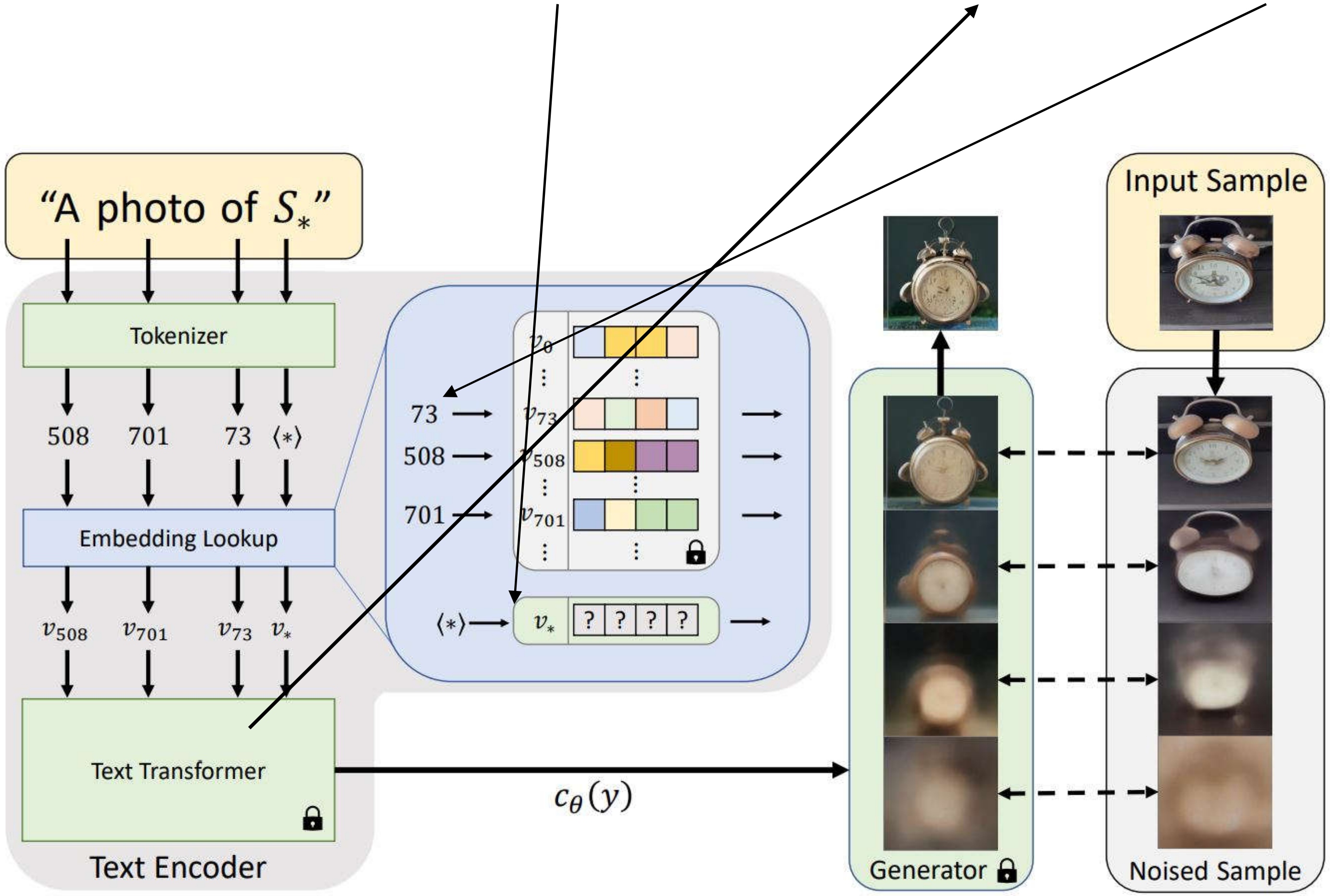
```
<852style-girl>
```



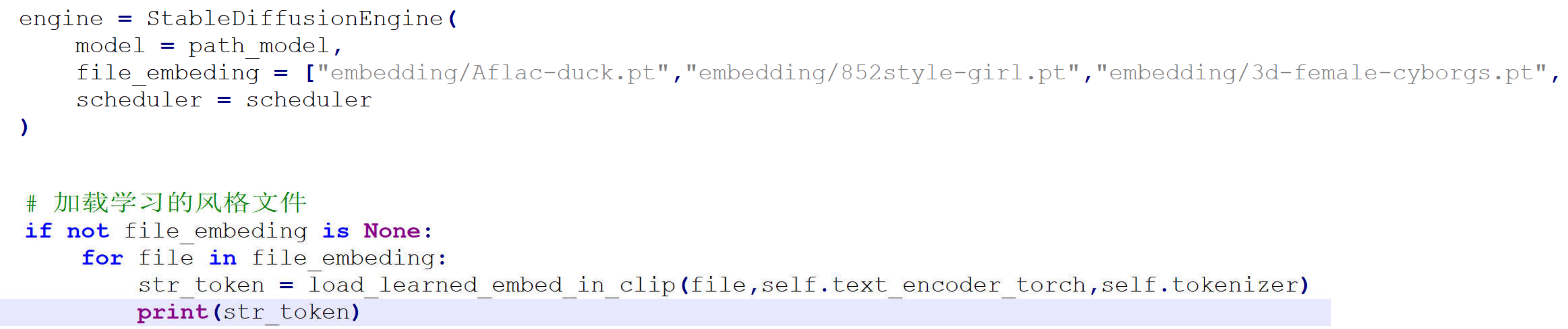
```
def load_learned_embed_in_clip(learned_embeds_path,text_encoder,tokenizer,token=None):
    loaded_learned_embeds = torch.load(learned_embeds_path, map_location="cpu")

    # separate token and the embeds
    trained_token = list(loaded_learned_embeds.keys())[0]

    embeds = loaded_learned_embeds[trained_token]

    # cast to dtype of text_encode
    dtype = text_encoder.get_input_embeddings().weight.dtype
    # embeds.to(dtype)

    # add the token in tokenizer
    token = token if token is not None else trained_token
    num_added_tokens = tokenizer.add_tokens(token)

    i = 1
    while(num_added_tokens == 0):
        print(f"The tokenizer already contains the token {token}.")
        token = f"{token[:-1]}-{i}>"
        print(f"Attempting to add the token {token}.")
        num_added_tokens = tokenizer.add_tokens(token)
        i+=1
    # resize the token embeddings
    text_encoder.resize_token_embeddings(len(tokenizer))

    # get the id for the token and assign the embeds
    token_id=tokenizer.convert_tokens_to_ids(token)
    text_encoder.get_input_embeddings().weight.data[token_id]=embeds
    return token
```



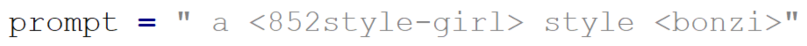
```
def load_learned_embed_in_clip(learned_embeds_path,text_encoder,tokenizer,token=None):
```



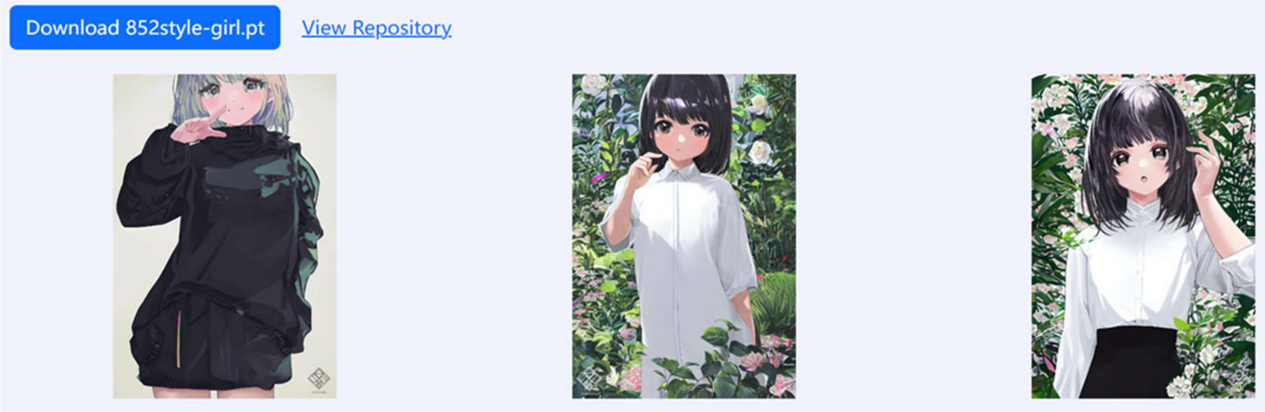
多风格组合

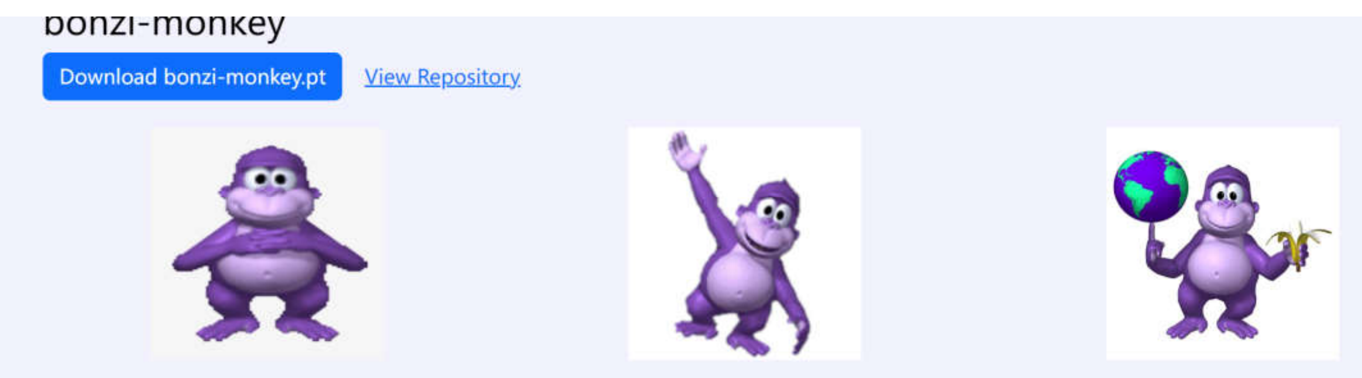


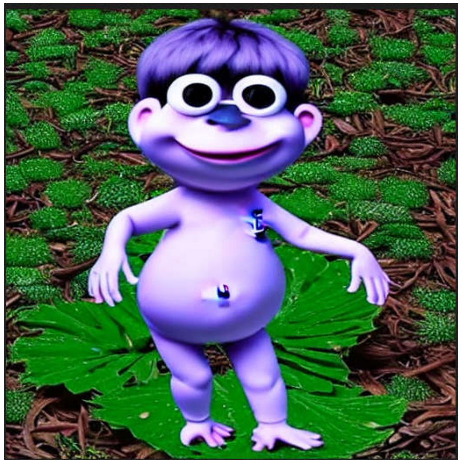
```
engine = StableDiffusionEngine(
    model = path_model,
    file_embeding = ["embedding/Aflac-duck.pt","embedding/852style-girl.pt","embedding/3d-female-cyborgs.pt",
    scheduler = scheduler
)


 # 加载学习的风格文件
 if not file_embeding is None:
     for file in file_embeding:
         str_token = load_learned_embed_in_clip(file,self.text_encoder_torch,self.tokenizer)
         print(str_token)
```



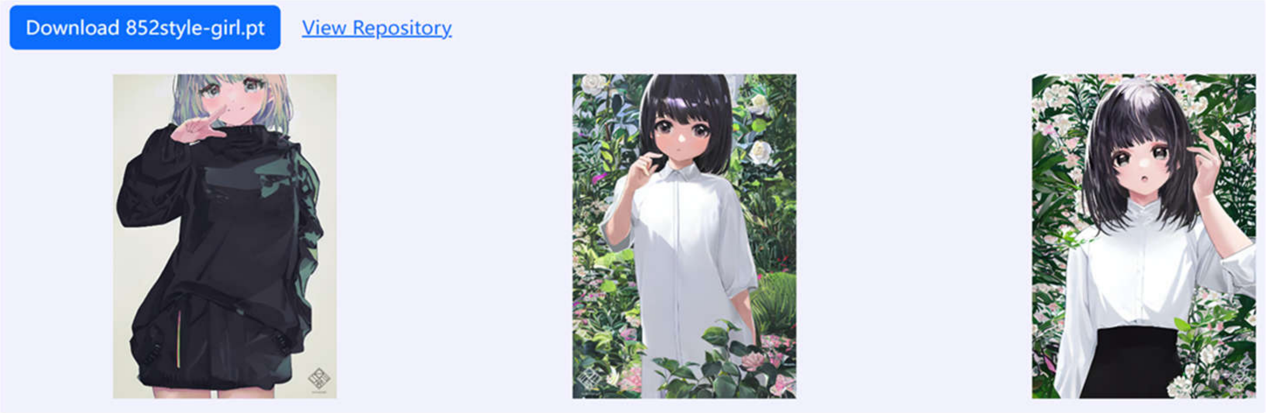
```
prompt = " a <852style-girl> style <bonzi>"
```

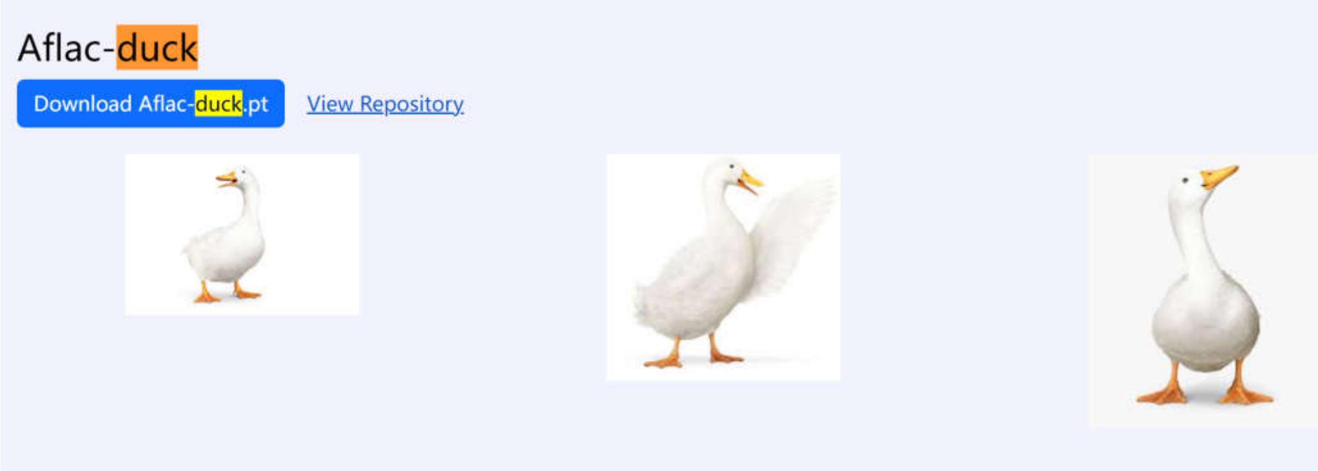



bonzi-monkey



私炉特征组合





"a <852style-girl> style <aflac duck>"