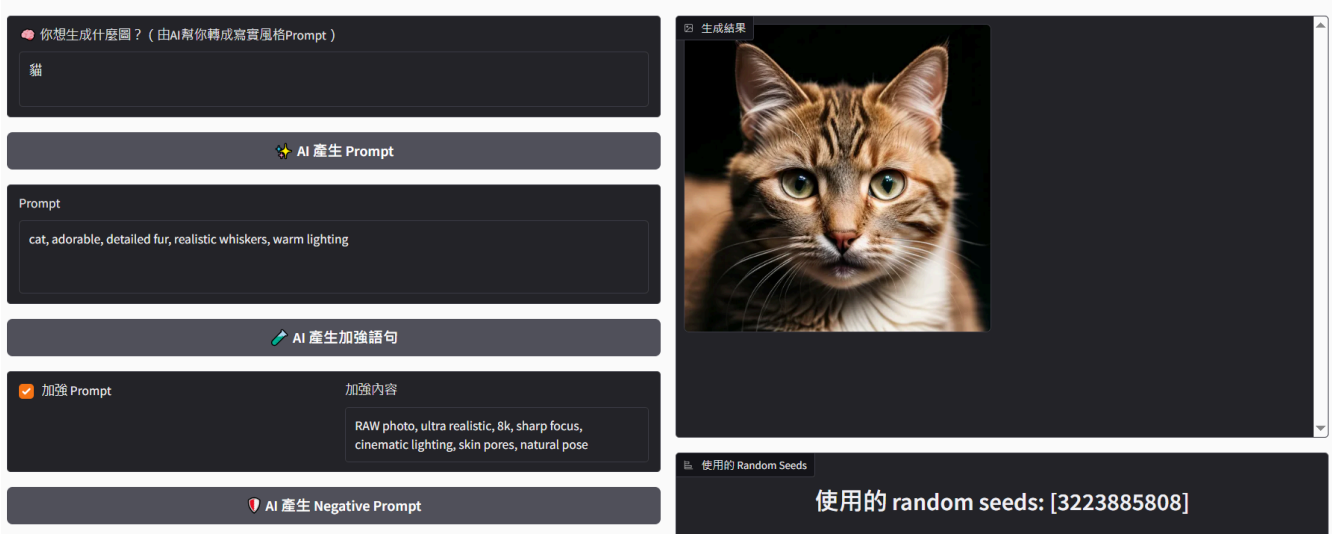
使用Gradio產生寫實風格圖片

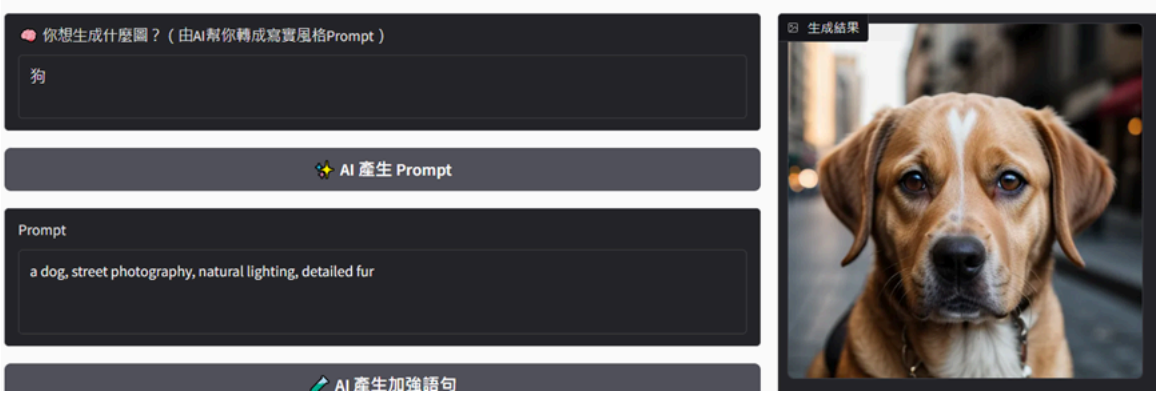
model: Realistic_Vision_V5

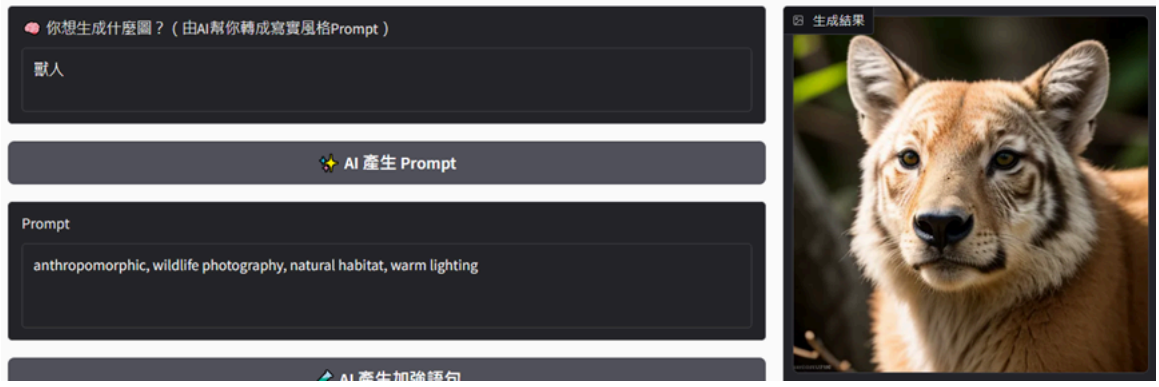
scheduler: EulerAncestralDiscreteScheduler

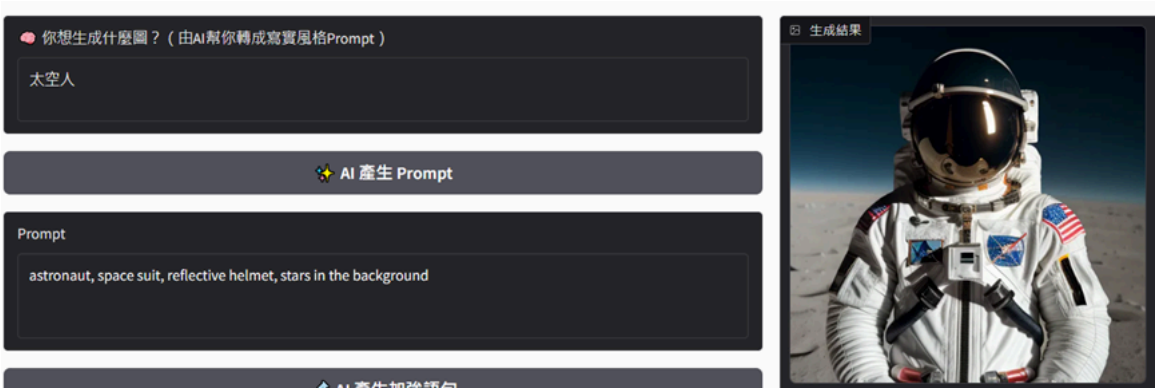
VAE: stabilityai/sd-vae-ft-mse

可以讓AI生成prompt, 加強prompt, negative prompt










```
!pip install diffusers transformers accelerate safetensors huggingface_hub gradio --upgrade
!pip install gradio
```

```
Requirement already satisfied: diffusers in /usr/local/lib/python3.11/dist-packages (0.33.1)
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-packages (4.51.3)
Requirement already satisfied: accelerate in /usr/local/lib/python3.11/dist-packages (1.6.0)
Requirement already satisfied: safetensors in /usr/local/lib/python3.11/dist-packages (0.5.3)
Requirement already satisfied: huggingface_hub in /usr/local/lib/python3.11/dist-packages (0.31.1)
Requirement already satisfied: gradio in /usr/local/lib/python3.11/dist-packages (5.29.0)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.11/dist-packages (from diffusers) (8.7.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from diffusers) (3.18.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from diffusers) (2.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from diffusers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from diffusers) (2.32.3)
Requirement already satisfied: Pillow in /usr/local/lib/python3.11/dist-packages (from diffusers) (11.2.1)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.21.1)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: psutil in /usr/local/lib/python3.11/dist-packages (from accelerate) (5.9.5)
Requirement already satisfied: torch>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from accelerate) (2.6.0+cu124)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface_hub) (2025.3.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface_hub) (4.13.2)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from huggingface_hub) (1.1.0)
Requirement already satisfied: aiofiles<25.0,>=22.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (24.1.0)
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (4.9.0)
Requirement already satisfied: fastapi<1.0,>=0.115.2 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.115.12)
Requirement already satisfied: ffmpy in /usr/local/lib/python3.11/dist-packages (from gradio) (0.5.0)
Requirement already satisfied: gradio-client==1.10.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (1.10.0)
Requirement already satisfied: groovy~=0.1 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.1.2)
Requirement already satisfied: httpx>=0.24.1 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.27.2)
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.1.6)
Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.0.2)
Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.10.18)
Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.2.2)
Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.11.4)
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages (from gradio) (0.25.1)
Requirement already satisfied: python-multipart>=0.0.18 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.0.20)
Requirement already satisfied: ruff>=0.9.3 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.11.9)
Requirement already satisfied: safehttpx<0.2.0,>=0.1.6 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.1.6)
Requirement already satisfied: semantic-version~=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.10.0)
Requirement already satisfied: starlette<1.0,>=0.40.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.46.2)
Requirement already satisfied: tomlkit<0.14.0,>=0.12.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.13.2)
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.15.3)
Requirement already satisfied: uvicorn>=0.14.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.34.2)
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.0->gradio) (15.0.1)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (3.10)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (1.3.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (2025.4.26)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (1.0.9)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx>=0.24.1->gradio) (0.16.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (2.33.2)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (0.4.0)
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages (from torch>=2.0.0->accelerate) (3.4.2)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torch>=2.0.0->accelerate) (12
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torch>=2.0.0->accelerate)
```

按兩下 (或按 Enter 鍵) 即可編輯

```
from diffusers import StableDiffusionPipeline, EulerAncestralDiscreteScheduler, AutoencoderKL
import torch
import gc
import matplotlib.pyplot as plt
```

```
import gradio as gr
import random
import os
from google.colab import userdata
from openai import OpenAI
import re


model_name = "SG161222/Realistic_Vision_V5.1_noVAE"


vae = AutoencoderKL.from_pretrained(
    "stabilityai/sd-vae-ft-mse",
    torch_dtype=torch.float16
).to("cuda")

pipe = StableDiffusionPipeline.from_pretrained(
    model_name,
    torch_dtype=torch.float16,
    use_safetensors=True
).to("cuda")
pipe.vae = vae
pipe.scheduler = EulerAncestralDiscreteScheduler.from_config(pipe.scheduler.config)
```

```
model_index.json: 100%                                                  609/609 [00:00<00:00, 61.1kB/s]
Fetching 12 files: 100%                                                 12/12 [01:14<00:00,  8.02s/it]
model.safetensors: 100%                                                 492M/492M [00:24<00:00, 9.03MB/s]
tokenizer_config.json: 100%                                             737/737 [00:00<00:00, 25.2kB/s]
merges.txt: 100%                                                        525k/525k [00:00<00:00, 1.79MB/s]
config.json: 100%                                                       612/612 [00:00<00:00, 8.28kB/s]
special_tokens_map.json: 100%                                           472/472 [00:00<00:00, 11.7kB/s]
vocab.json: 100%                                                        1.06M/1.06M [00:00<00:00, 4.56MB/s]
config.json: 100%                                                       1.55k/1.55k [00:00<00:00, 26.3kB/s]
scheduler_config.json: 100%                                             548/548 [00:00<00:00, 6.27kB/s]
diffusion_pytorch_model.safetensors: 100%                               3.44G/3.44G [01:14<00:00, 85.2MB/s]
config.json: 100%                                                       577/577 [00:00<00:00, 24.5kB/s]
diffusion_pytorch_model.safetensors: 100%                               335M/335M [00:17<00:00, 77.0MB/s]
Loading pipeline components...: 100%                                    5/5 [00:19<00:00,  3.50s/it]
```

```
def generate_images(prompt, use_enhance, enhance_text, use_negative, negative_text,
                                    use_custom_seed, custom_seed, height, width, steps, num_images):

    height = int(height)
    width = int(width)

    if height % 8 != 0 or width % 8 != 0:
        raise ValueError("高度和宽度必须是8的倍数！")

    if use_custom_seed:
        base_seed = int(custom_seed)
    else:
        base_seed = random.randint(0, 2**32 - 1)

    seeds = [base_seed + i for i in range(num_images)]

    prompts = []
    negative_prompts = []
    generators = []

    final_prompt = prompt
    if use_enhance and enhance_text:
        final_prompt = prompt + ", " + enhance_text

    final_negative = negative_text if use_negative else None

    for seed in seeds:
        g = torch.Generator("cuda").manual_seed(seed)
        generators.append(g)
        prompts.append(final_prompt)
        negative_prompts.append(final_negative)

    gc.collect()
    torch.cuda.empty_cache()
```

```
        images = []
        for i in range(num_images):
                with torch.no_grad():
                        image = pipe(
                                prompt=prompts[i],
                                negative_prompt=negative_prompts[i] if final_negative else None,
                                height=height,
                                width=width,
                                num_inference_steps=steps,
                                guidance_scale=7.5,
                                generator=generators[i]
                        ).images[0]
                        images.append(image)

        return images, f"使用的 random seeds: {seeds}"


import os
from google.colab import userdata
api_key = userdata.get('Groq')
os.environ["OPENAI_API_KEY"] = api_key
model = "llama3-70b-8192"
base_url="https://api.groq.com/openai/v1"
client = OpenAI(
        base_url=base_url
)


prompt_system_prompt = """
你是 Stable Diffusion 的提示詞專家，專精於寫實風格模型（如 Realistic Vision v5.1）。
請根據使用者的描述，產生適合用於圖像生成的高品質 prompt（不包含 negative prompt）：
- 使用逗號分隔的英文詞語，例如：1girl, street photography, sunset lighting, detailed face
- 圖像風格請以真實人物、自然光、真實環境為主
- 不要加入多餘說明、請只回傳 prompt 本身
請使用繁體中文作答，但 prompt 請維持英文。
"""

enhance_system_prompt = """
你是 Stable Diffusion 的寫實風格優化專家。
請產生一段適合加在 prompt 後方的強化語句，用來提升畫質與細節，例如光影真實、解析度高、材質精緻等。
- 請使用英文，並以逗號分隔，例如：RAW photo, ultra realistic, 8k, sharp focus, cinematic lighting
- 請只回傳這段語句，不要說明文字
"""
negative_system_prompt = """
你是 Stable Diffusion 的 negative prompt 專家。
請產生一組適合寫實風格圖像的 negative prompt，用來避免錯誤或降低品質的畫面。
- 內容包含壞解剖、模糊、多手指、畸形等問題
- 請使用英文，並以逗號分隔，例如：bad anatomy, blurry, extra limbs, low resolution, distorted face
- 請只回傳這段語句，不要解釋
"""



# 共用 LLM 呼叫函式
def ask_llm(system_prompt, user_input):
        response = client.chat.completions.create(
                model=model,
                messages=[
                        {"role": "system", "content": system_prompt},
                        {"role": "user", "content": user_input}
                ],
                temperature=0.8,
                max_tokens=200
        )
        return response.choices[0].message.content.strip()

# 對應功能
def generate_prompt_from_idea(user_idea):
        raw = ask_llm(prompt_system_prompt, user_idea)

        # 優先處理「引號內的部分」
        match = re.search(r'[「“](.*?)[」”]', raw)
        if match:
                return match.group(1).strip()

        # 如果找不到「引號」內容，則嘗試移除開頭英文說明
        lines = raw.strip().split('\n')
        lines = [line for line in lines if not line.lower().startswith("here is")]
        lines = [line for line in lines if not line.strip().startswith("This prompt")]
        return "\n".join(lines).strip()

def generate_enhance_prompt(user_idea):
        return ask_llm(enhance_system_prompt, user_idea)
```

```
def generate_negative_prompt(user_idea):
    return ask_llm(negative_system_prompt, user_idea)


default_enhance = "RAW photo, ultra realistic, 8k, sharp focus, cinematic lighting, skin pores, natural pose"
default_negative = "blurry, low resolution, bad anatomy, distorted face, extra limbs, unrealistic lighting"


with gr.Blocks(css=".gradio-container {background-color: #FAFAFA; padding: 20px;} .gr-button {font-size: 18px; background: linear-gradient(
    gr.Markdown("""
    歡迎使用！輸入提示詞或概念，由AI幫你生成寫實圖片的Prompt，再由模型為你畫出圖像！
    """)

    with gr.Row():
        with gr.Column(scale=6):
            user_idea = gr.Textbox(label="🧠 你想生成什麼圖？（由AI幫你轉成寫實風格Prompt）", placeholder="例如：一位年輕女性站在夕陽
            gen_prompt_btn = gr.Button("✨ AI 產生 Prompt")

            prompt = gr.Textbox(label="Prompt", placeholder="請輸入你的提示詞 (prompt)", lines=3)

            with gr.Row():
                gen_enhance_btn = gr.Button("🧪 AI 產生加強語句")

            with gr.Row():
                use_enhance = gr.Checkbox(label="加強 Prompt", value=True)
                enhance_text = gr.Textbox(label="加強內容", value=default_enhance)

            with gr.Row():
                gen_negative_btn = gr.Button("🛡️ AI 產生 Negative Prompt")

            with gr.Row():
                use_negative = gr.Checkbox(label="使用 Negative Prompt", value=True)
                negative_text = gr.Textbox(label="Negative Prompt 內容", value=default_negative)

            with gr.Row():
                use_custom_seed = gr.Checkbox(label="自訂 Random Seed", value=False)
                custom_seed = gr.Number(label="指定 seed (選填)", value=42)

            with gr.Row():
                height = gr.Dropdown(["512", "768", "1024"], label="高度 Height", value="512")
                width = gr.Dropdown(["512", "768", "1024"], label="寬度 Width", value="512")
            with gr.Row():
                steps = gr.Slider(10, 50, value=20, step=5, label="生成步數 (Steps)")
                num_images = gr.Slider(1, 4, step=1, value=1, label="生成張數")

            generate_btn = gr.Button("🚀 開始生成！")

        with gr.Column(scale=6):
            gallery = gr.Gallery(label="生成結果", columns=2, object_fit="contain", height="auto")
            seed_info = gr.Label(label="使用的 Random Seeds")

    # 👉 註冊按鈕事件
    generate_btn.click(
        fn=generate_images,
        inputs=[prompt, use_enhance, enhance_text, use_negative, negative_text,
                use_custom_seed, custom_seed, height, width, steps, num_images],
        outputs=[gallery, seed_info]
    )

    gen_prompt_btn.click(fn=generate_prompt_from_idea, inputs=[user_idea], outputs=[prompt])
    gen_enhance_btn.click(fn=generate_enhance_prompt, inputs=[user_idea], outputs=[enhance_text])
    gen_negative_btn.click(fn=generate_negative_prompt, inputs=[user_idea], outputs=[negative_text])


demo.launch(share=True, debug=True)
```

Colab notebook detected. This cell will run indefinitely so that you can see errors and logs. To turn off, set debug=False in launch
* Running on public URL: https://3e48d001f289ca715b.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working

100%	20/20	[00:02<00:00,	6.92it/s]
100%	20/20	[00:02<00:00,	6.96it/s]
100%	20/20	[00:02<00:00,	6.99it/s]
100%	35/35	[00:05<00:00,	6.84it/s]
100%	35/35	[00:05<00:00,	6.87it/s]
100%	35/35	[00:05<00:00,	6.86it/s]