

Part 10

Stable Diffusion 插件

Contents

1. Stable Diffusion 插件安装

2. ControlNet 原理及应用

1. Stable Diffusion 插件安装

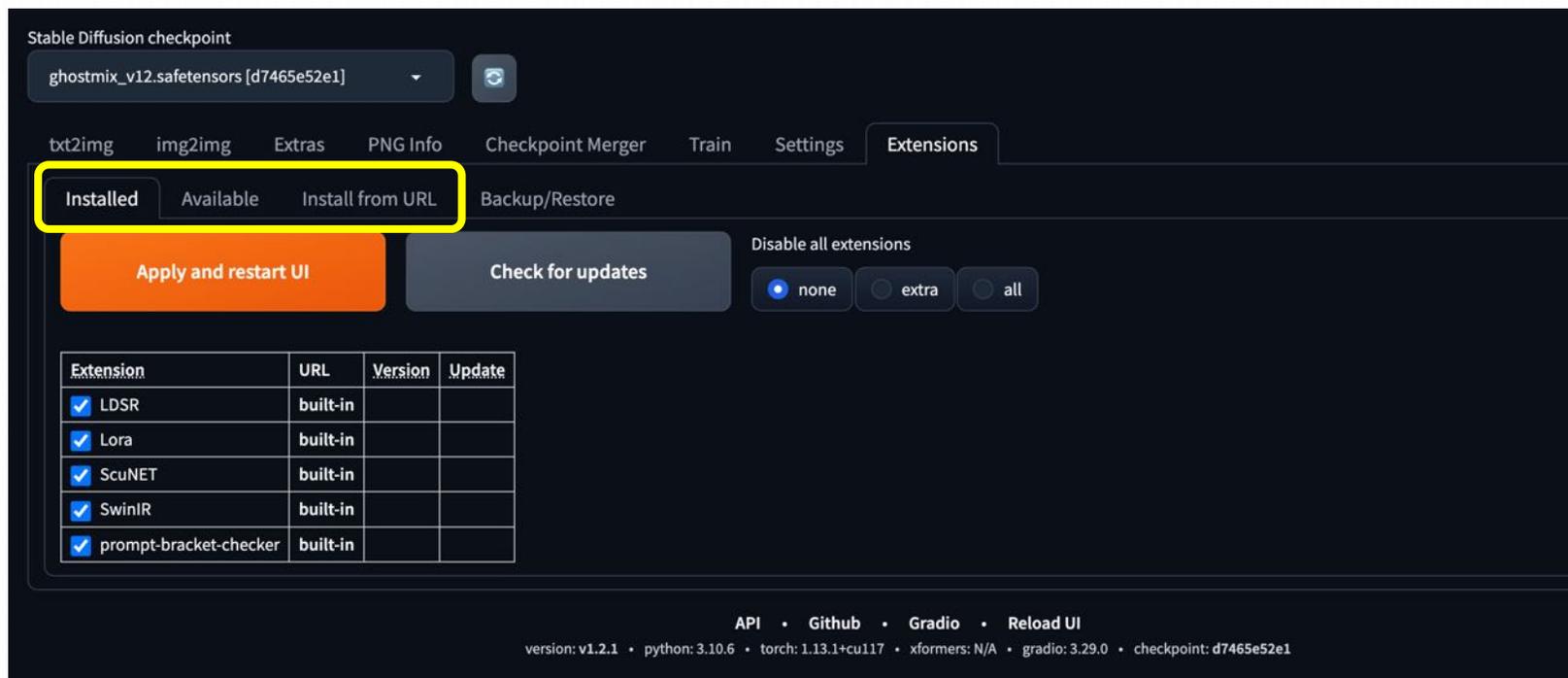
1.1 Stable Diffusion 插件安装方式

1.2 扩展插件推荐

1.1 Stable Diffusion 插件安装方式

了解 Stable Diffusion 里的扩展功能架构，学会通过列表一键安装、仓库链接安装以及拷贝文件安装扩展的方式。

A. 内置列表安装

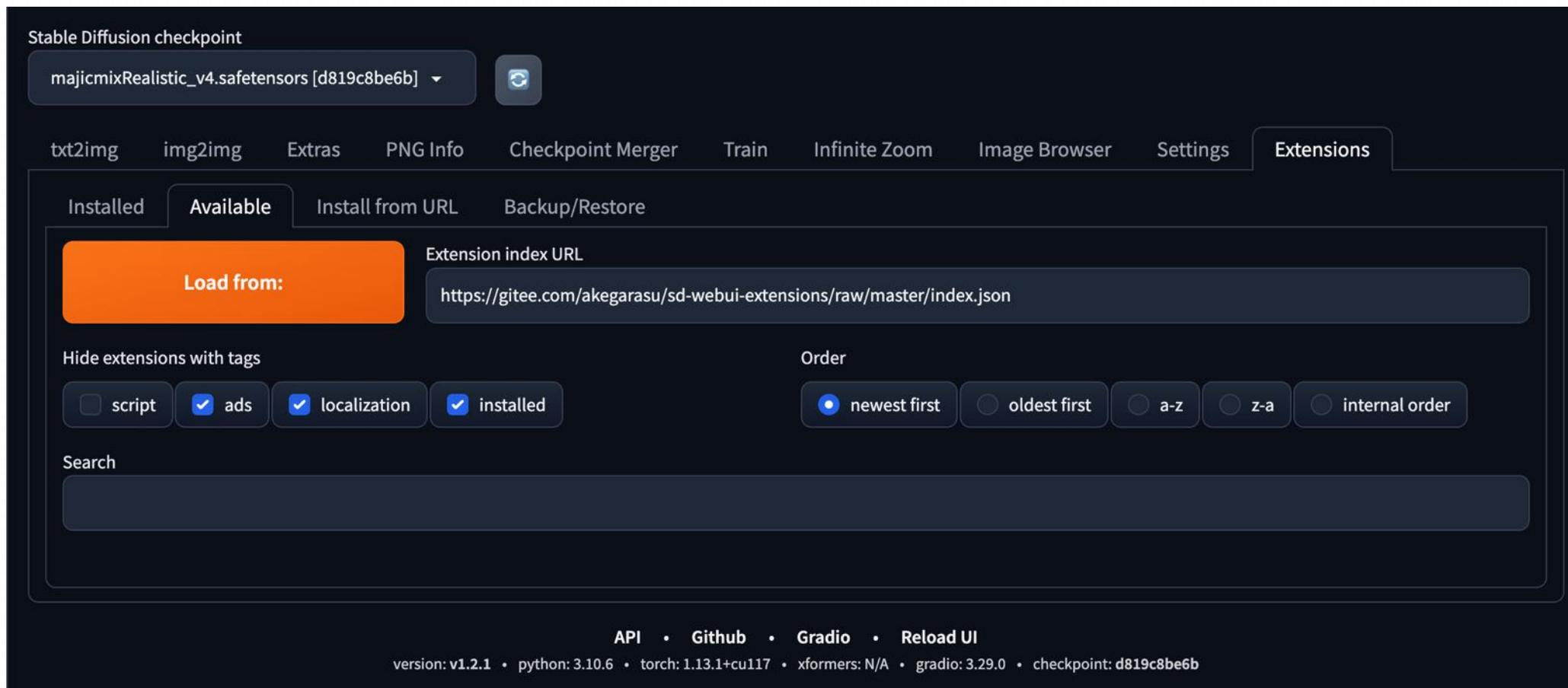


Installed: 已安装
Available: 可用的
Install form URL 通过链接安装

在Stable Diffusion里，所有和扩展有关的功能被集中收纳在了这个扩展标签里进行管理。

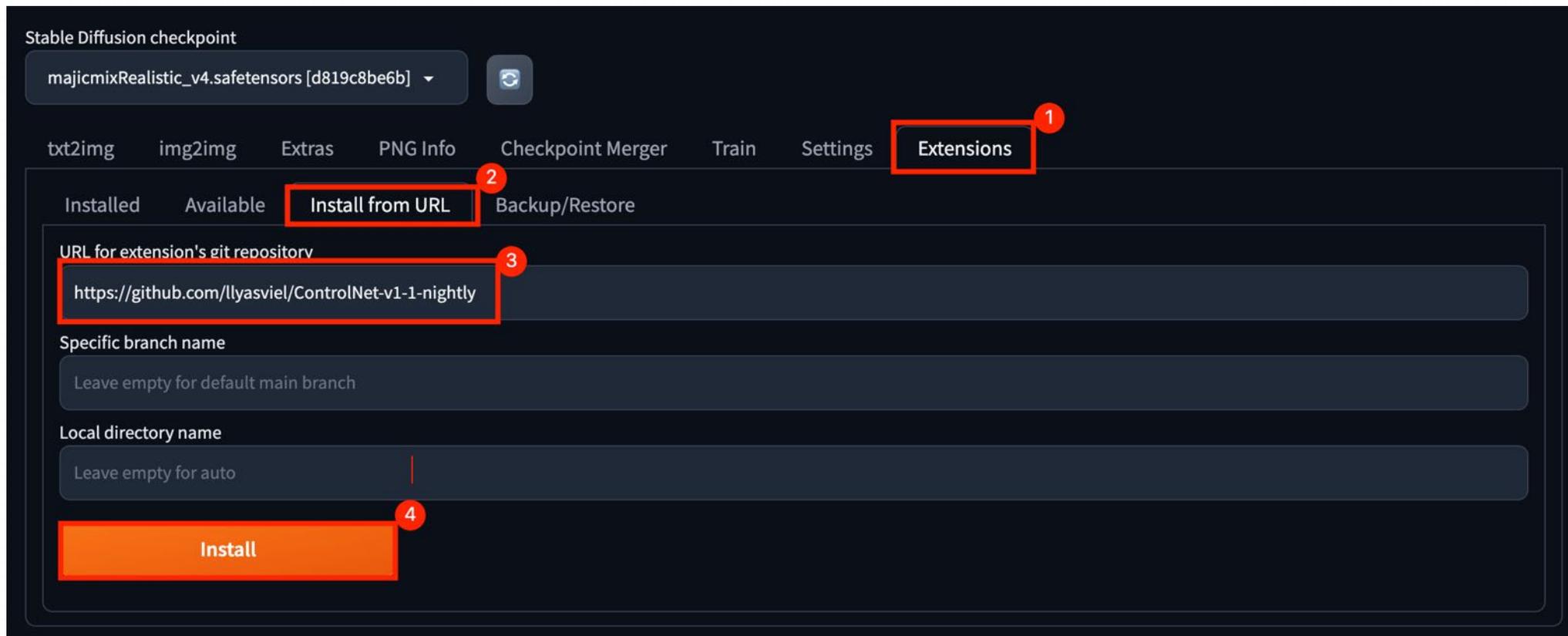
1.1 Stable Diffusion 插件安装方式

A. 内置列表安装



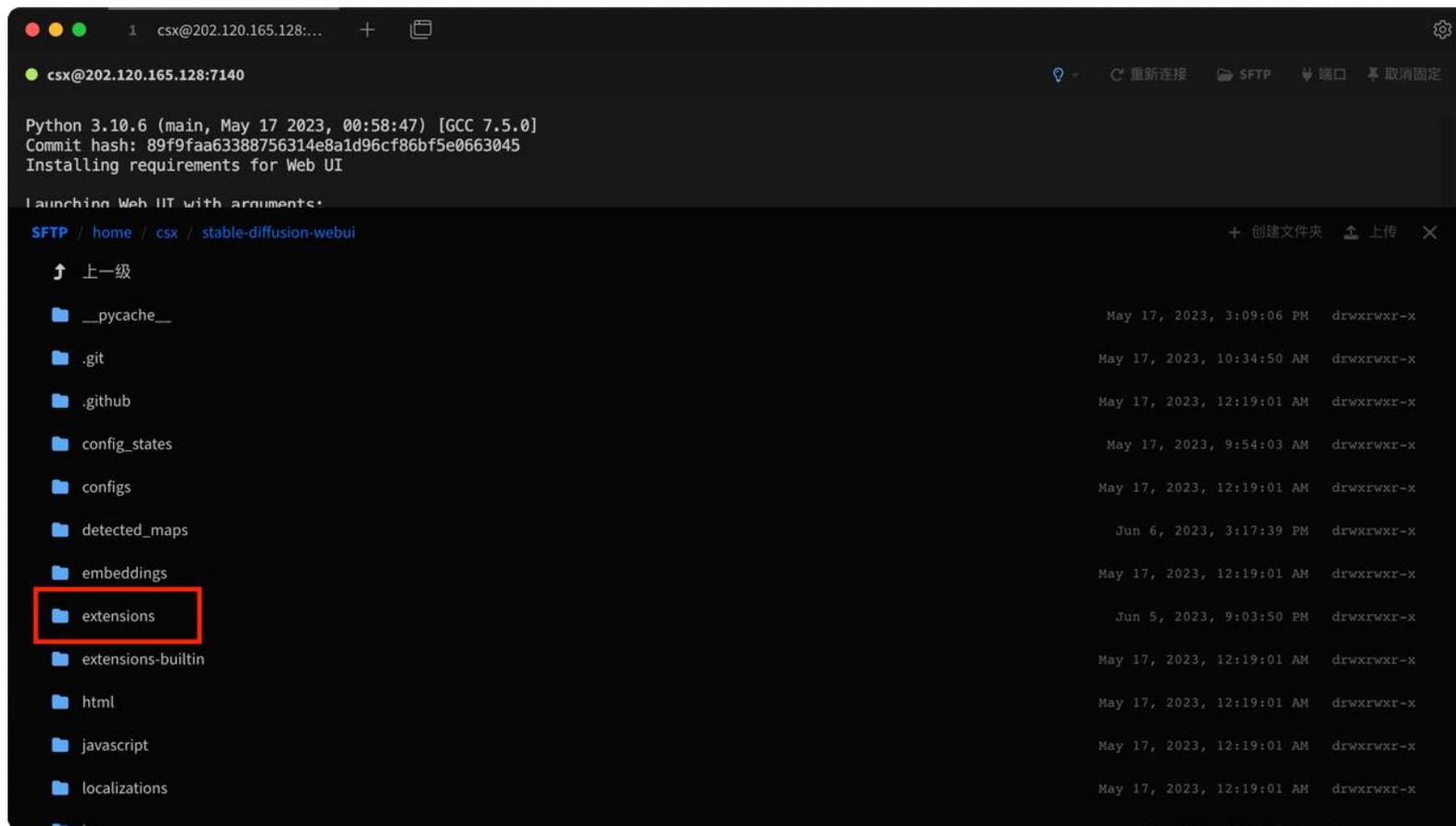
1.1 Stable Diffusion 插件安装方式

B. 链接安装



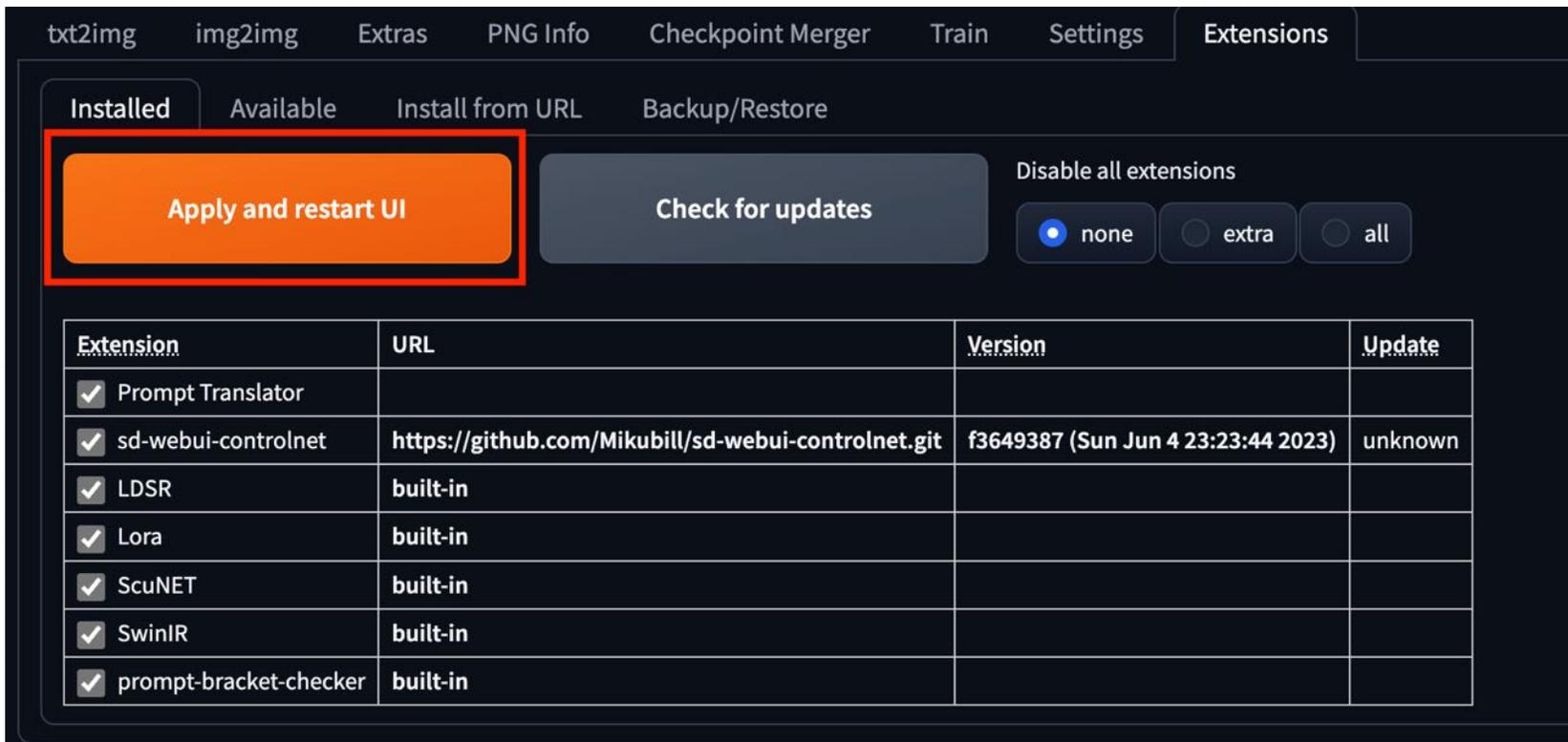
1.1 Stable Diffusion 插件安装方式

C. 本地安装



1.1 Stable Diffusion 插件安装方式

C. 本地安装



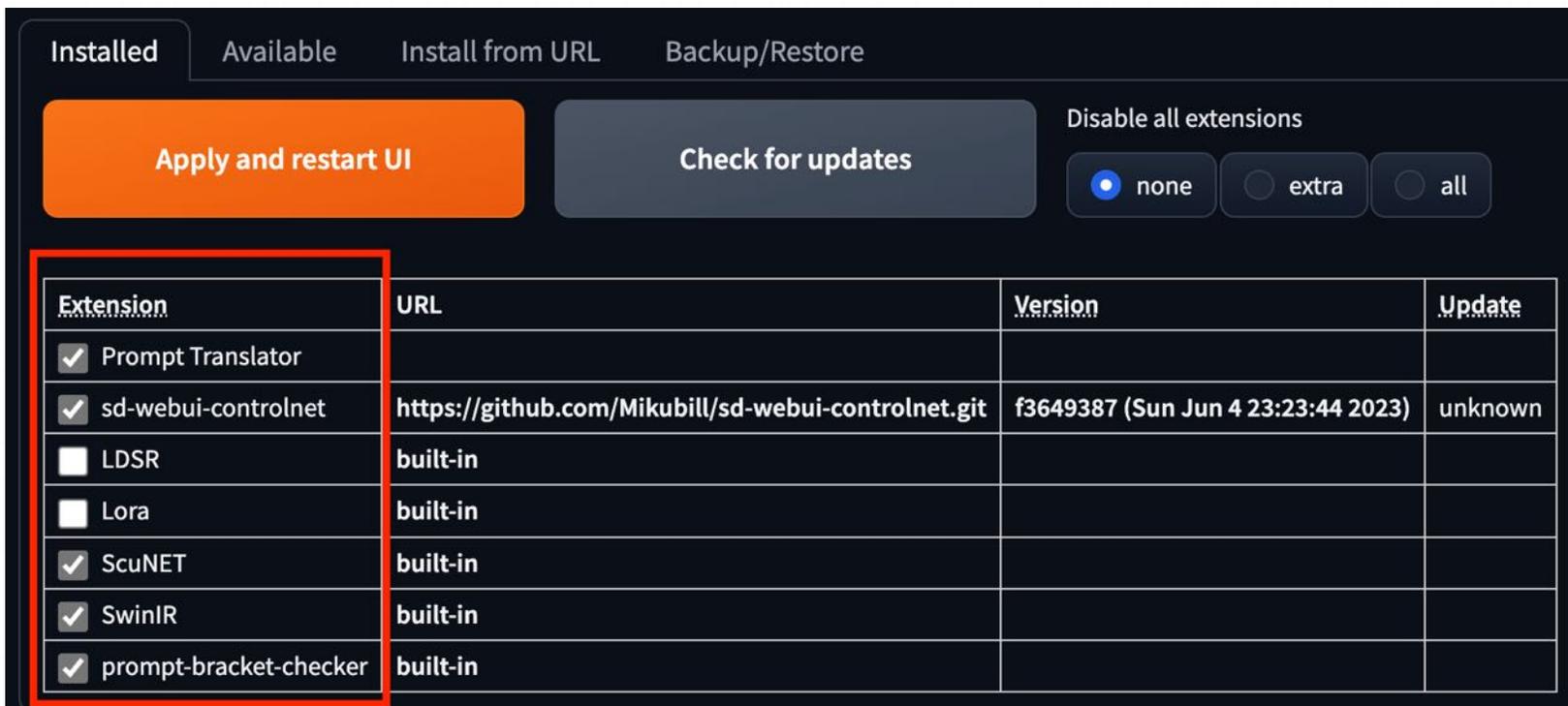
The screenshot shows the 'Extensions' tab in the Stable Diffusion web UI. The 'Installed' sub-tab is active. A red box highlights the 'Apply and restart UI' button. To its right is a 'Check for updates' button and a 'Disable all extensions' section with radio buttons for 'none', 'extra', and 'all'. Below these controls is a table of installed extensions.

Extension	URL	Version	Update
<input checked="" type="checkbox"/> Prompt Translator			
<input checked="" type="checkbox"/> sd-webui-controlnet	https://github.com/Mikubill/sd-webui-controlnet.git	f3649387 (Sun Jun 4 23:23:44 2023)	unknown
<input checked="" type="checkbox"/> LDSR	built-in		
<input checked="" type="checkbox"/> Lora	built-in		
<input checked="" type="checkbox"/> ScuNET	built-in		
<input checked="" type="checkbox"/> SwinIR	built-in		
<input checked="" type="checkbox"/> prompt-bracket-checker	built-in		

大部分扩展安装完以后，会要求你做进一步操作，就是点击上面这个应用并重启web UI的大按钮。它就会为你刷新一遍，再打开扩展就可以用了。

1.1 Stable Diffusion 插件安装方式

C. 本地安装



The screenshot displays the extension management interface in the Stable Diffusion web UI. At the top, there are tabs for 'Installed', 'Available', 'Install from URL', and 'Backup/Restore'. Below the tabs, there are buttons for 'Apply and restart UI' (highlighted in orange), 'Check for updates', and 'Disable all extensions'. The 'Disable all extensions' section has three radio buttons: 'none' (selected), 'extra', and 'all'. Below this is a table of installed extensions.

Extension	URL	Version	Update
<input checked="" type="checkbox"/> Prompt Translator			
<input checked="" type="checkbox"/> sd-webui-controlnet	https://github.com/Mikubill/sd-webui-controlnet.git	f3649387 (Sun Jun 4 23:23:44 2023)	unknown
<input type="checkbox"/> LDSR	built-in		
<input type="checkbox"/> Lora	built-in		
<input checked="" type="checkbox"/> ScuNET	built-in		
<input checked="" type="checkbox"/> SwinIR	built-in		
<input checked="" type="checkbox"/> prompt-bracket-checker	built-in		

如果你的某个插件在web UI更新以后失效了，可以到这里面操作点检查更新对比云端的版本号。一般来说更到最新的就可以有效解决问题。如果有插件带来的问题，你也可以通过前面的复选框取消，它暂时禁用一下，直到问题被排查恢复正常。

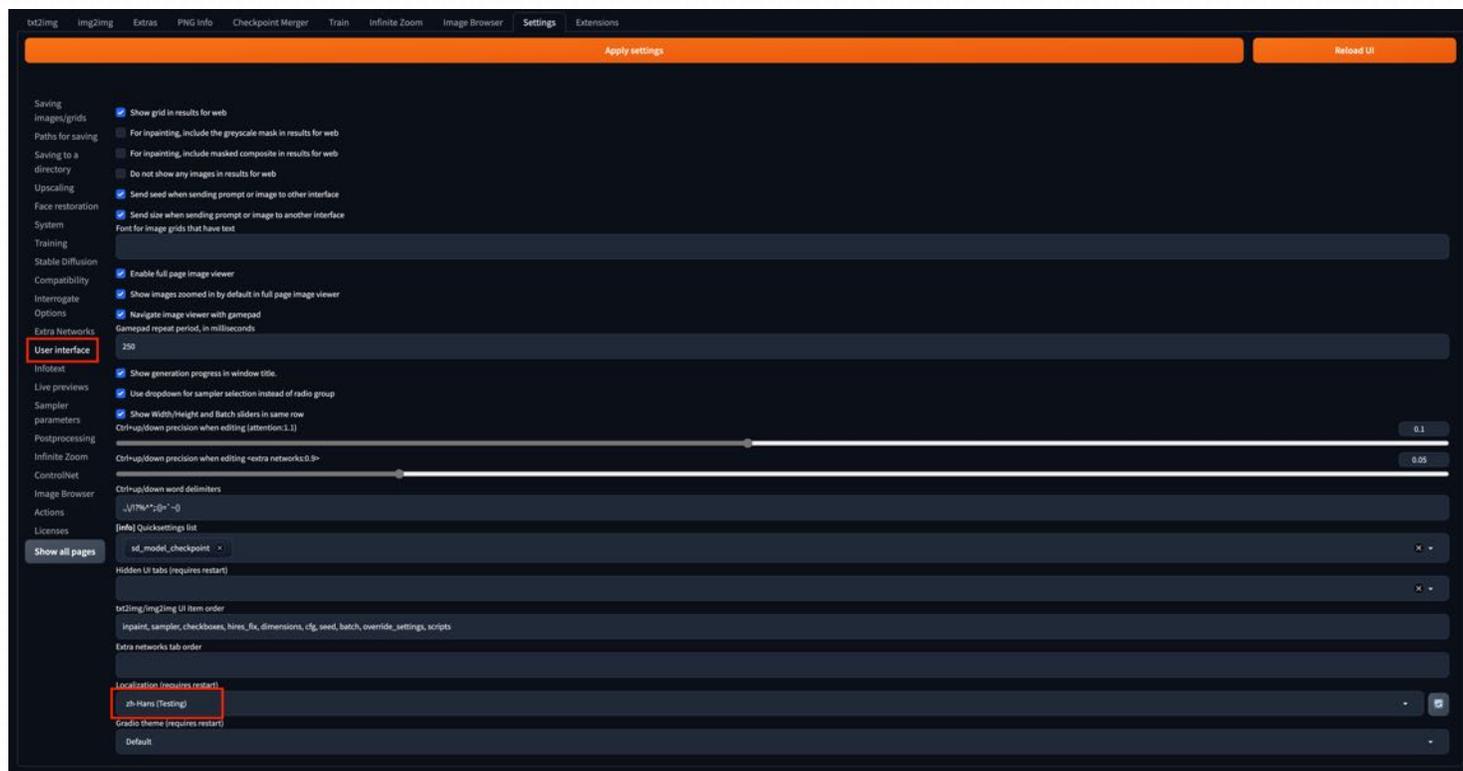
1.2 扩展插件推荐

A. 中文本地化语言包

实现界面汉化，帮助理解操作含义

搜索：zh（取消勾选本地化/Localization的的筛选）

仓库地址：https://github.com/hanamizuki-ai/stable-difusion-webui-localization-zh_Hans.git



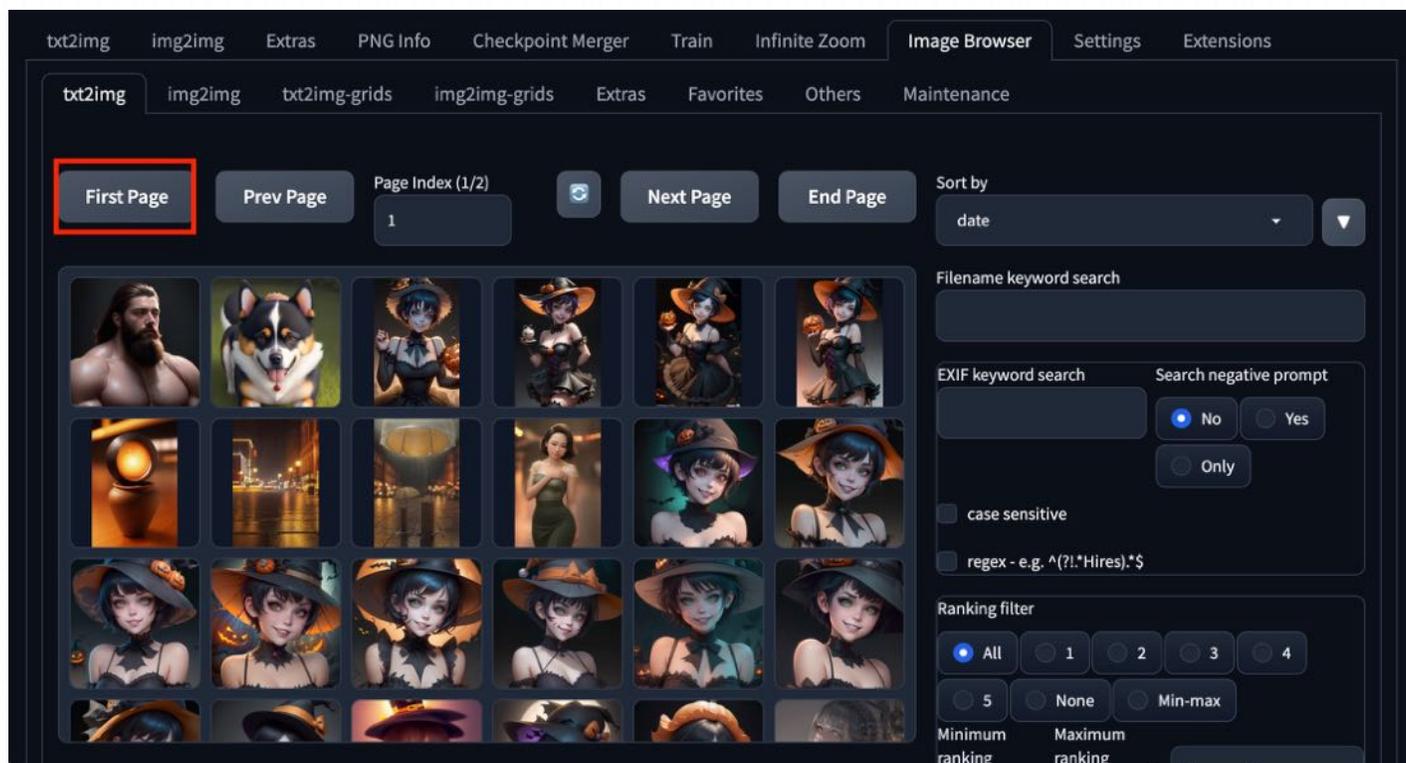
1.2 扩展插件推荐

B. 图库浏览器

方便浏览作品，进行多种查询和管理

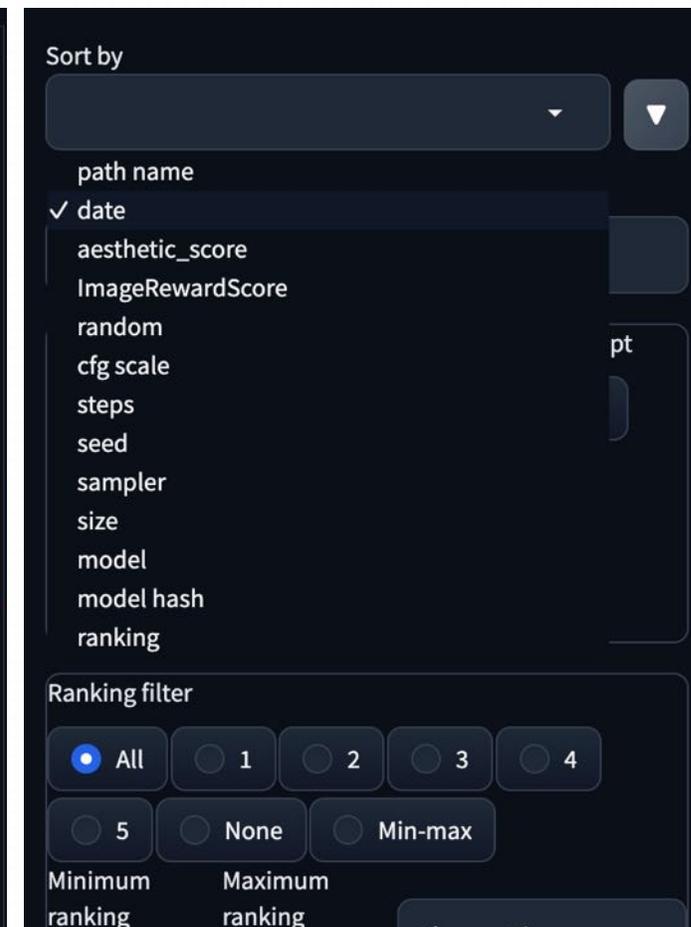
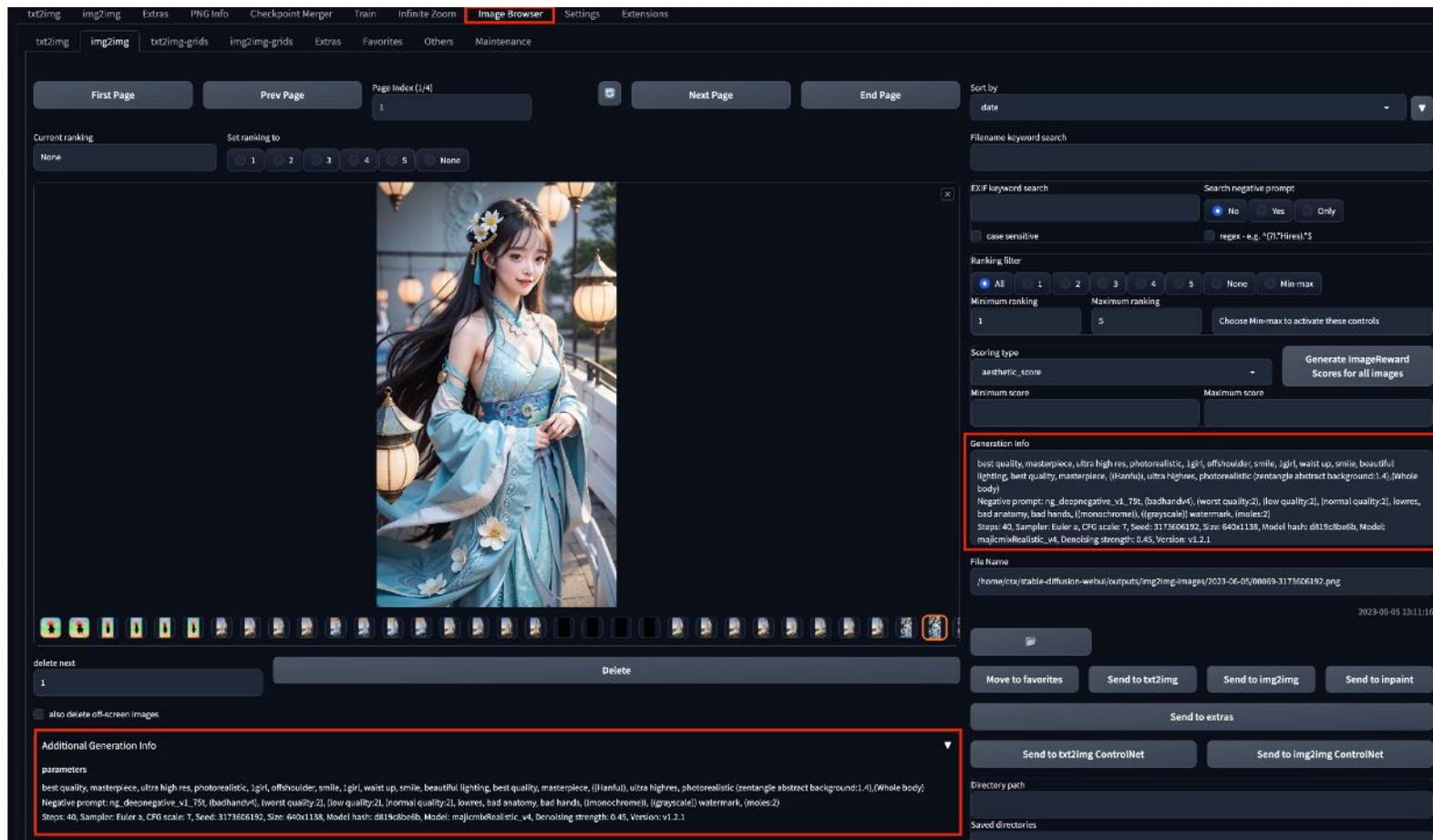
搜索： image browser

仓库地址：<https://github.com/AlUlkesh/stable-diffusion-webui-images-browser>



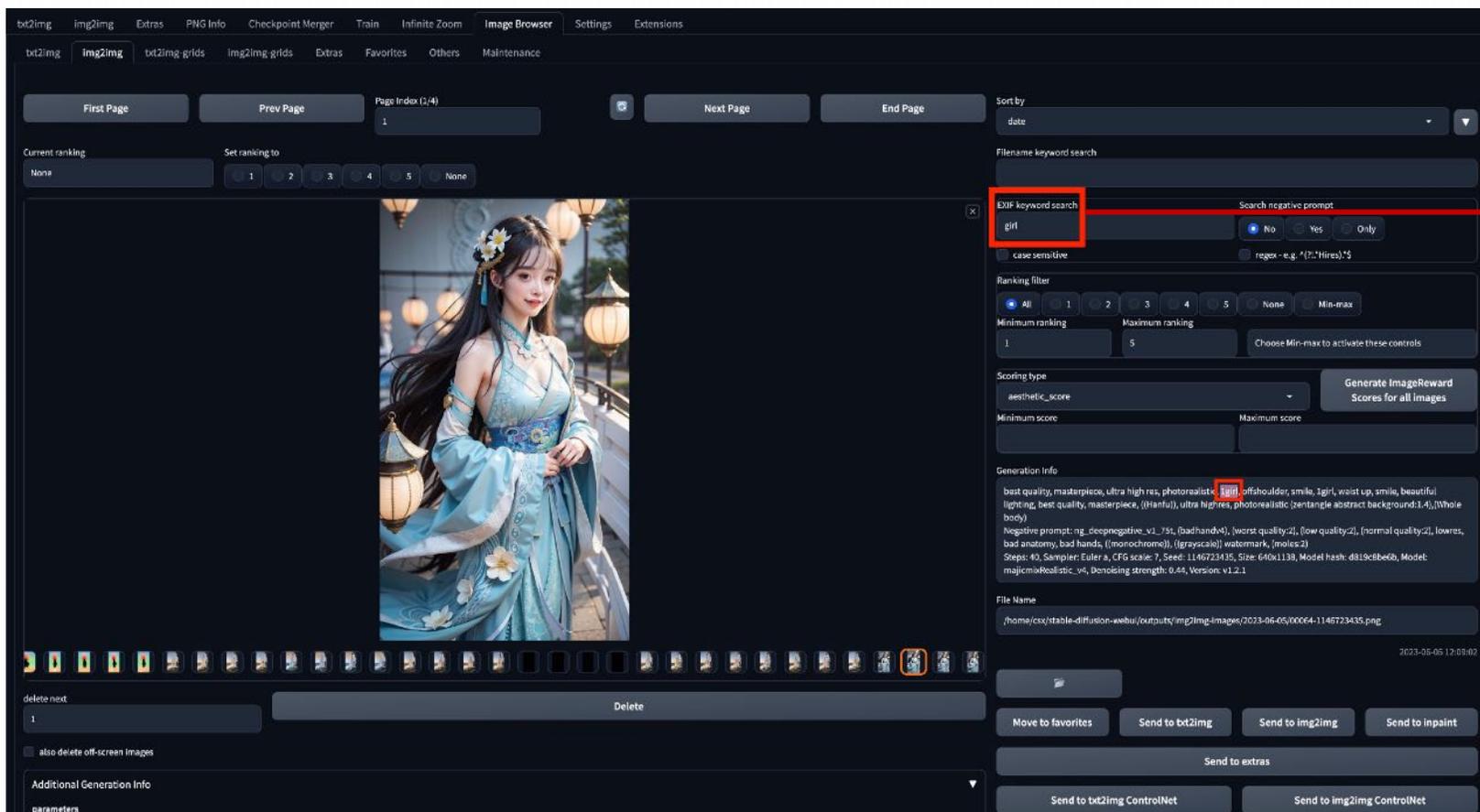
1.2 扩展插件推荐

B. 图库浏览器



1.2 扩展插件推荐

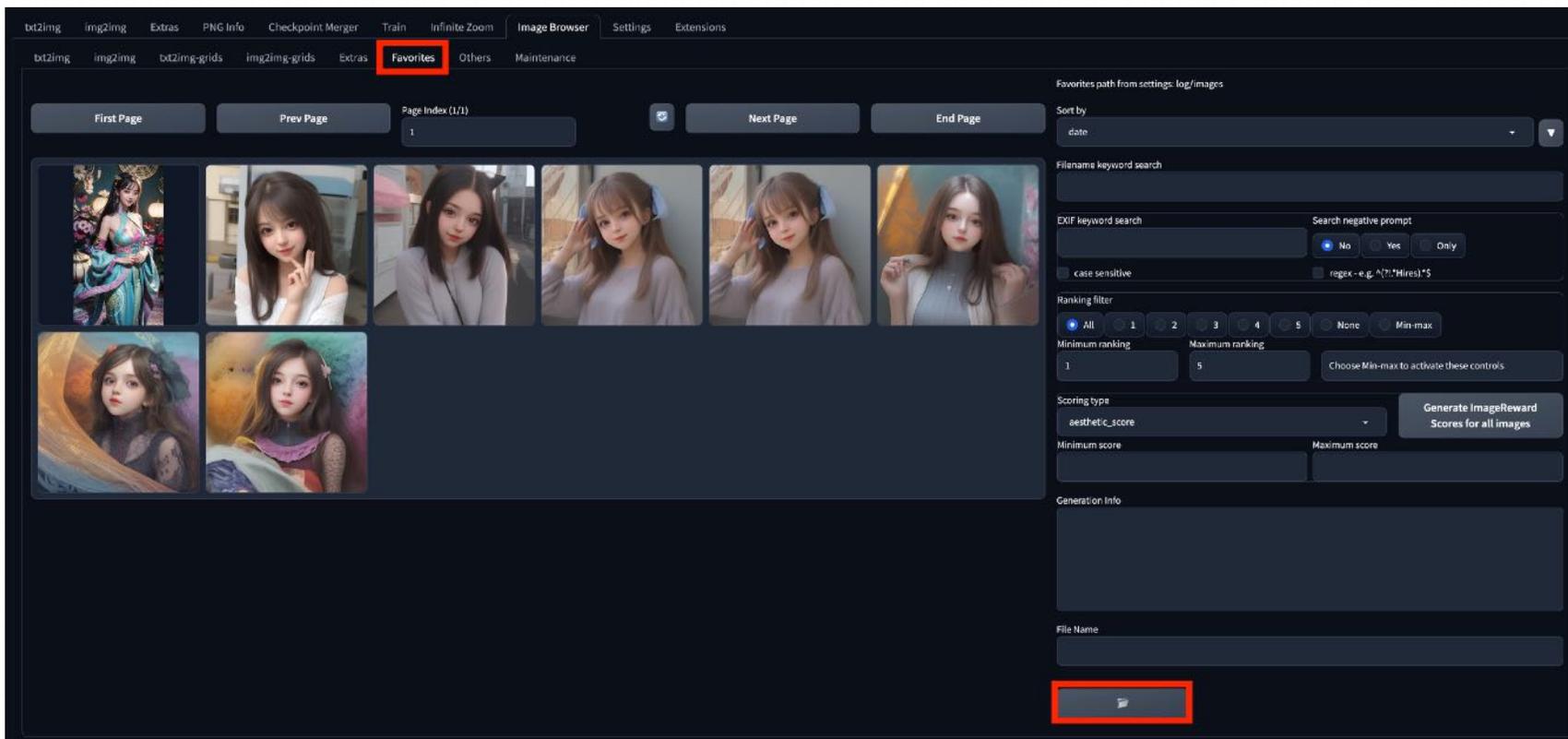
B. 图库浏览器



如果想更精确一点，就用下方的EXIF关键词搜索，它可以帮你精确到每一个输入过的提示词。

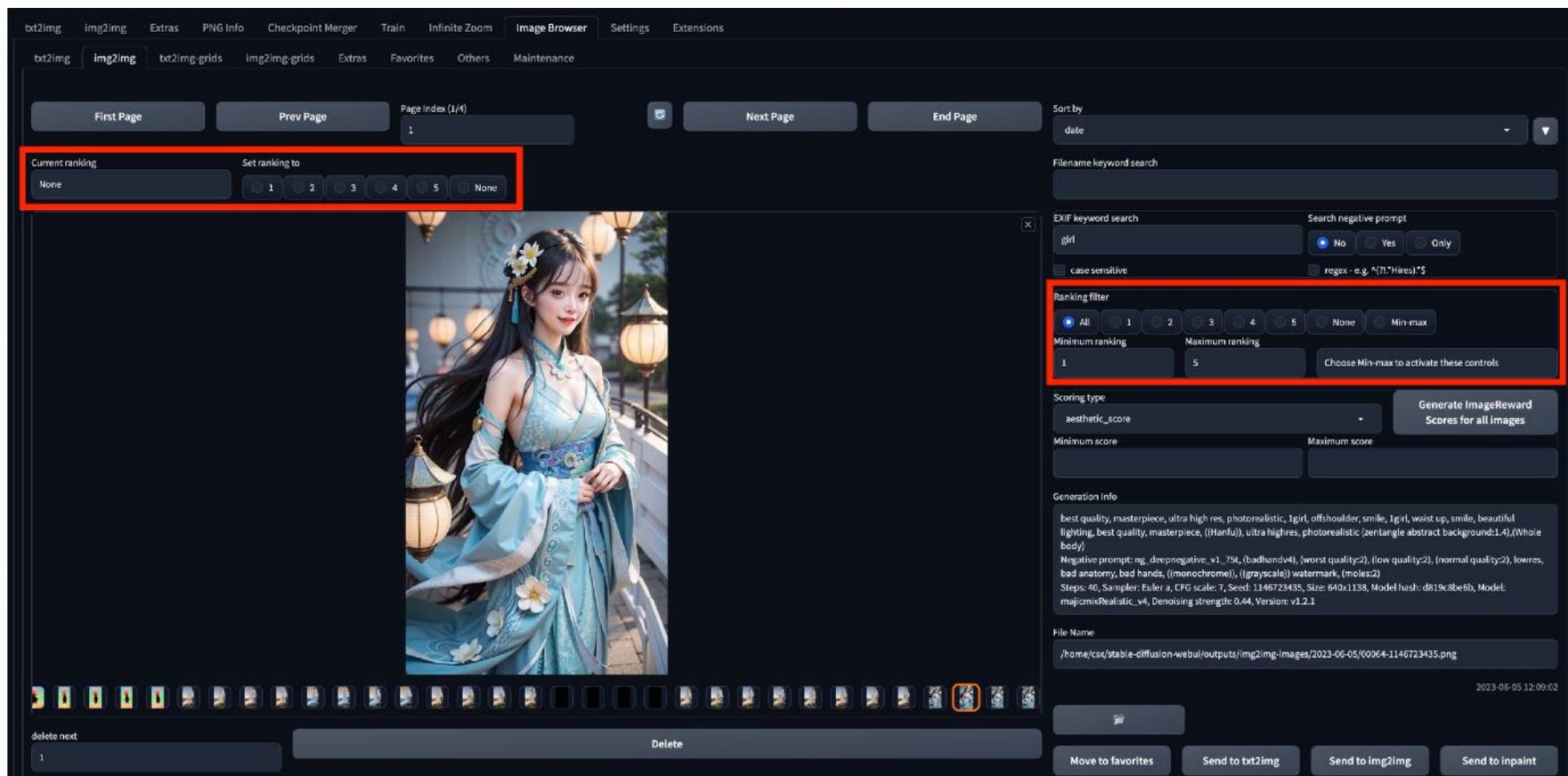
1.2 扩展插件推荐

B. 图库浏览器



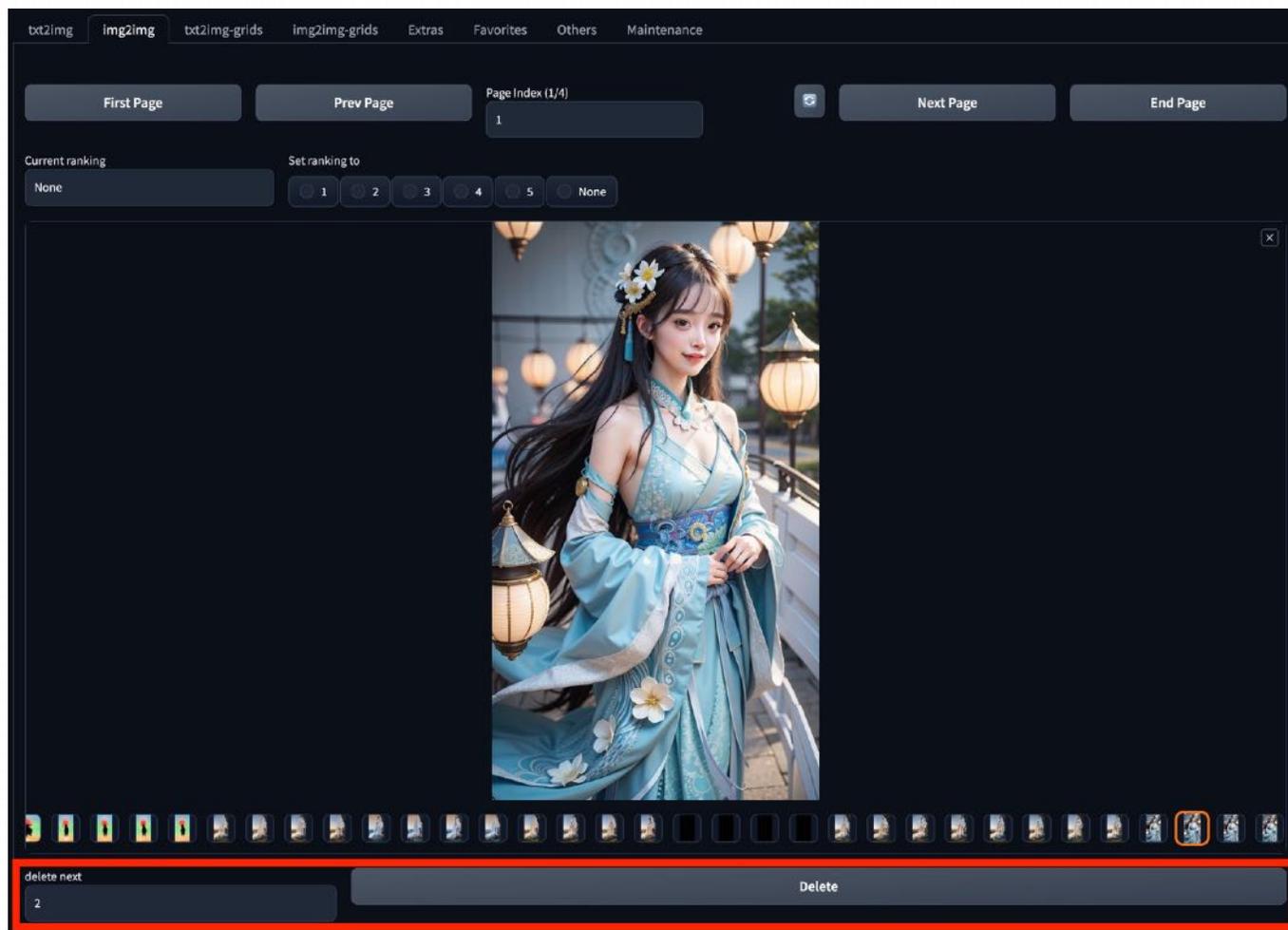
1.2 扩展插件推荐

B. 图库浏览器



1.2 扩展插件推荐

B. 图库浏览器



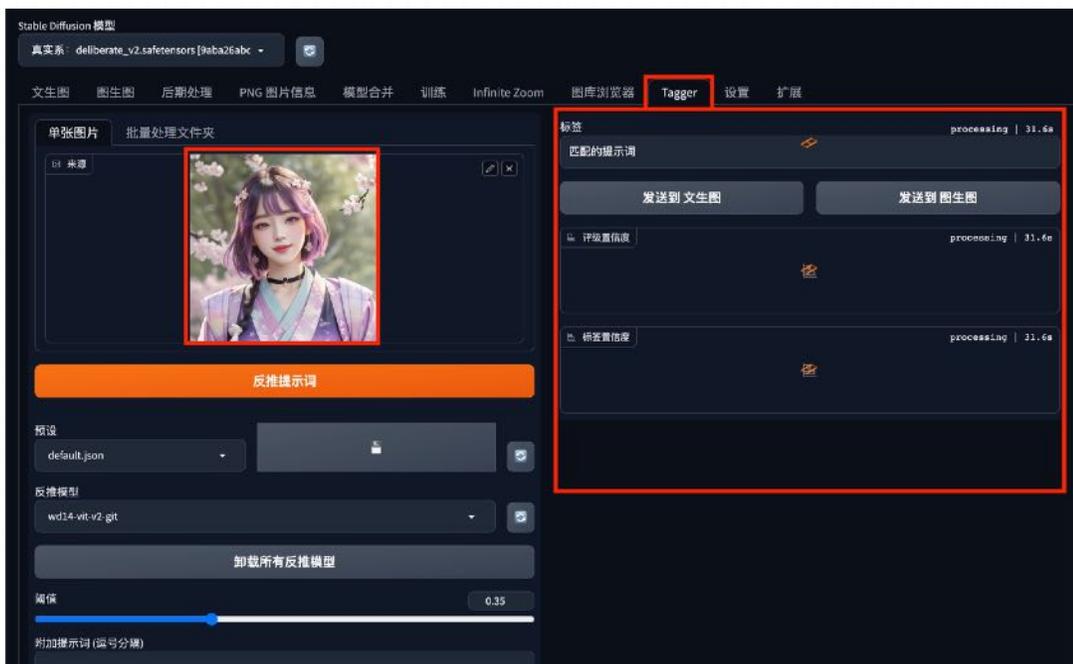
1.2 扩展插件推荐

C. Tagger

基于已有图像，实现提示词高效精确反推

搜索：`tagger`

仓库地址：<https://github.com/toriato/stable-diffusion-webui-wd14-tegger.git>



使用方式：

1. 安装tagger之后，重启命令行，重新加载Stable Diffusion
2. 在窗口中拖入一张照片（第一次使用时，系统会开始下载一个tagger 的模型）

```
This share link expires in 72 hours. For free permanent hosting and GPU upgrades (NEW!), check out Spaces: https://huggingface.co/spaces
Startup time: 10.3s (import torch: 0.9s, import gradio: 1.1s, import ldm: 0.4s, other imports: 0.8s, load scripts: 0.8s, create ui: 1.1s, gradio launch: 5.
2s).
Loading wd14-vit-v2-git model file from SellingWolf/wd-v1-4-vit-tagger-v2
Downloading model.onnx: 14% | ██████████ | 52.4M/373M [00:06:00:33, 9.44MB/s]
```

3. 模型安装完成后，系统会自动开始识别与图片中相关的词汇，相当于Midjourney中的 `/describe` 命令。

1.2 扩展插件推荐

C. Tagger

The screenshot shows the Tagger interface in Stable Diffusion. On the left, an image of a woman with purple hair in a kimono is displayed. Below the image, there are controls for '反推提示词' (Reverse Prompt) and '阈值' (Threshold) set to 0.35. The right panel shows the analysis results:

标签 (Tags):
 1girl, solo, choker, blurry background, looking at viewer, flower, black choker, hair ornament, blurry, braid, upper body, japanese clothes, hair flower, realistic, cherry blossoms, smile, earrings, closed mouth, purple hair, jewelry, outdoors, kimono, bangs, lips, multicolored hair, purple kimono, short hair

发送到 文生图 **发送到 图生图**

评级置信度 (Rating Confidence):

Category	Confidence
general	79%
sensitive	21%
questionable	0%
explicit	0%

标签置信度 (Tag Confidence):

Tag	Confidence
1girl	99%
solo	94%
choker	91%
blurry background	86%
looking_at_viewer	85%
flower	79%

SD会自动识别图片的信息，除了会提供该图像的Prompt，还有提供这些提示词的优先度。越靠上的是SD认为的与图片越接近、越有价值的提示词。

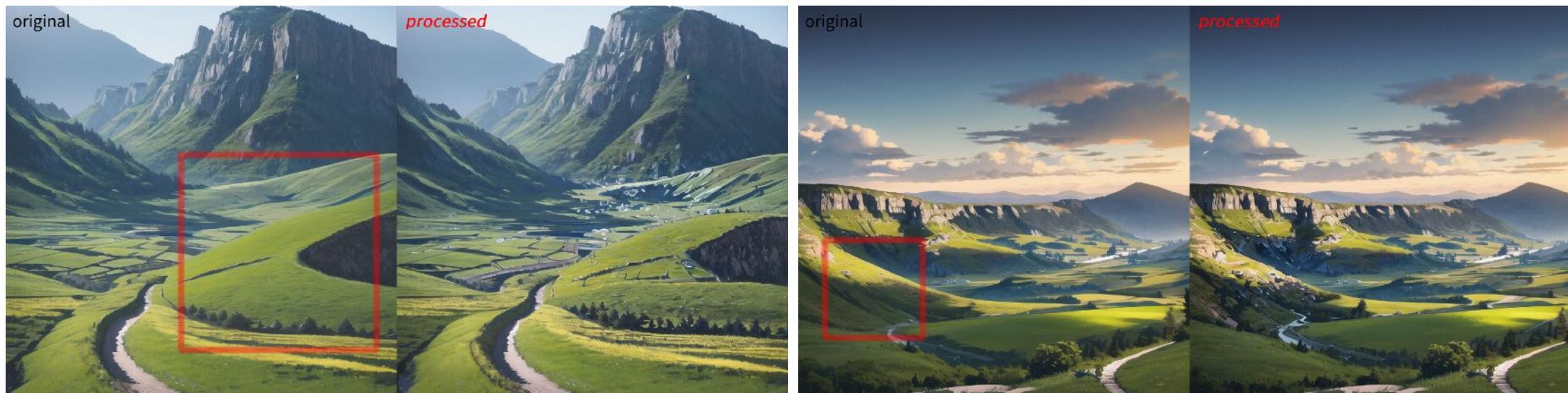
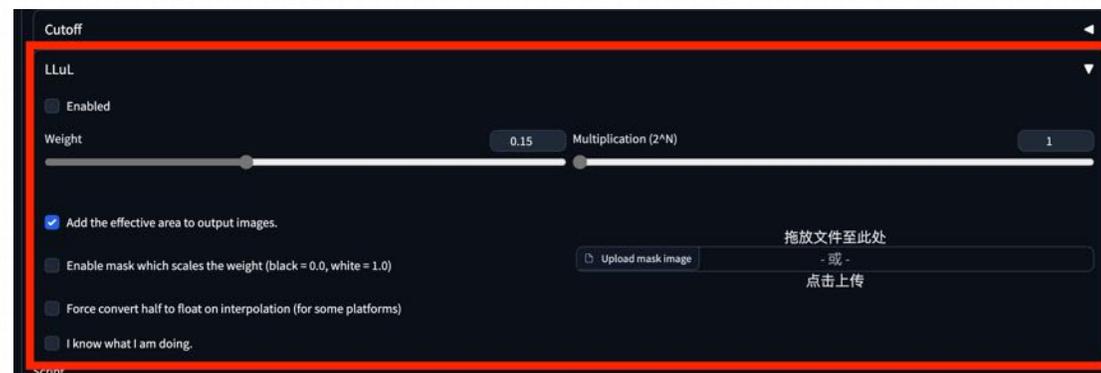
1.2 扩展插件推荐

D. Local Latent Couple (LLuL)

借助潜空间放大，局部添加细节

搜索：LLuL

仓库地址：<https://github.com/hnmr293/sd-webui-llul>



1.2 扩展插件推荐

D. Local Latent Couple (LLu1)

借助潜空间放大，局部添加细节

搜索：LLu1

仓库地址：<https://github.com/hnmr293/sd-webui-llul>



1.2 扩展插件推荐

E. Cutoff

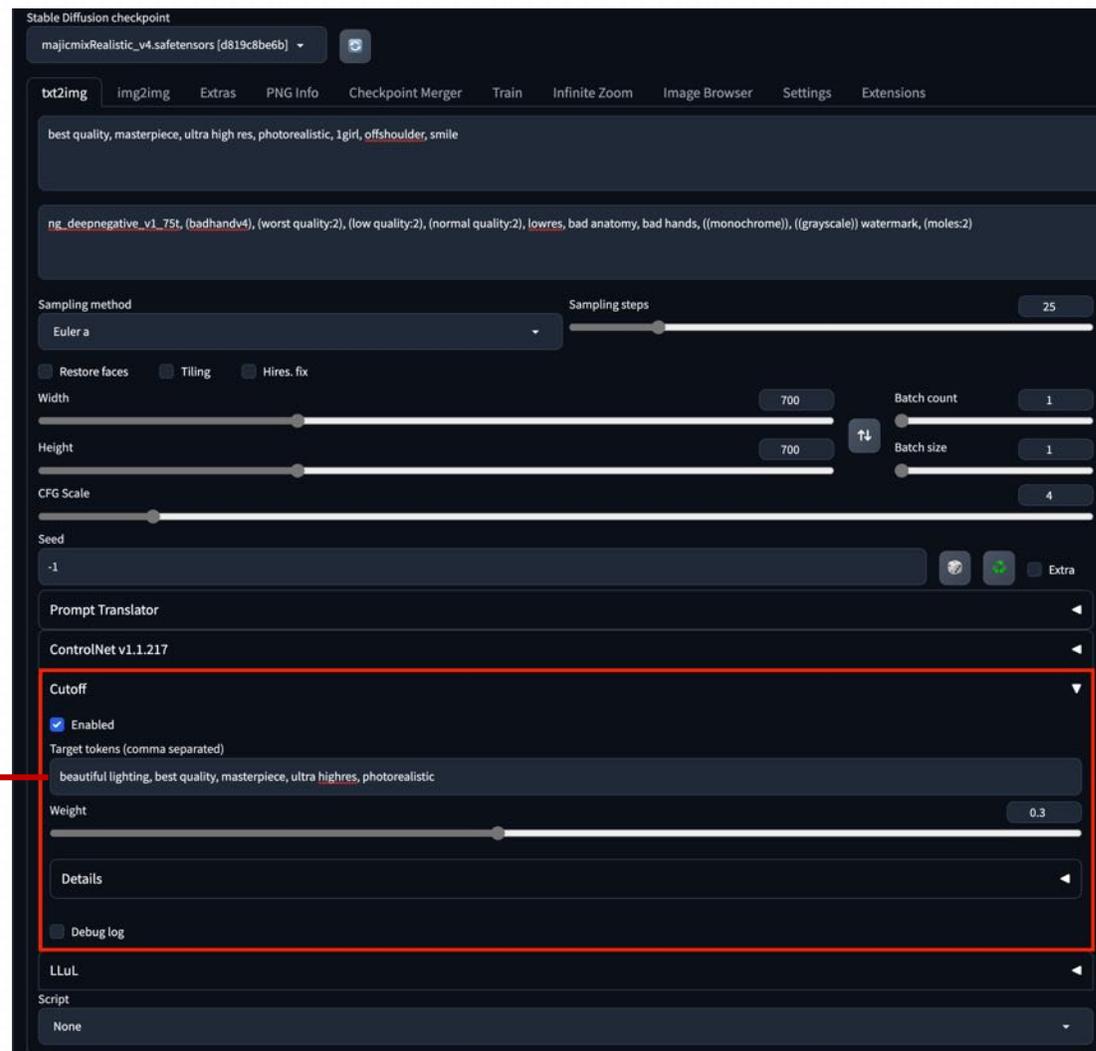
提示词的描述成分分离控制

搜索: **Cutoff**

仓库地址: <https://github.com/hnmr293/sd-webui-cutoff.git>

这个插件主要用于帮我们解决一些提示词之间的互相干预

将一段容易被AI搞混的关键词复制进来, 放在这个分隔目标提示词里。权重可以选择一个0.3到0.5之间的数值, 代表这种分隔作用的强度勾选启用。



1.2 扩展插件推荐

E. Cutoff

示例:

prompt

```
((masterpiece, best quality, ultra-detailed)),  
illustration, anime,  
lgirl, blue hair, (orange hat), white jacket, black  
tank top, pink skirt,  
upper body, city background,
```

Negative prompt

```
EasyNegative, verybadimagenegative_v1.3,  
ng_deepnegative_v1_75t, nsfw, out of frame, lowres,  
text, error, cropped, worst quality, low quality,  
mutation, deformed, blurry, dehydrated, bad anatomy,  
bad proportions, extra limbs, cloned face, disfigured,  
gross proportions, malformed limbs,
```



1.2 扩展插件推荐

E. Cutoff

Cutoff ▾

启用

分隔目标提示词(逗号分隔)

blue hair, (orange hat), white jacket, black tank top, pink skirt,

权重

详细设置 ▾

调试日志



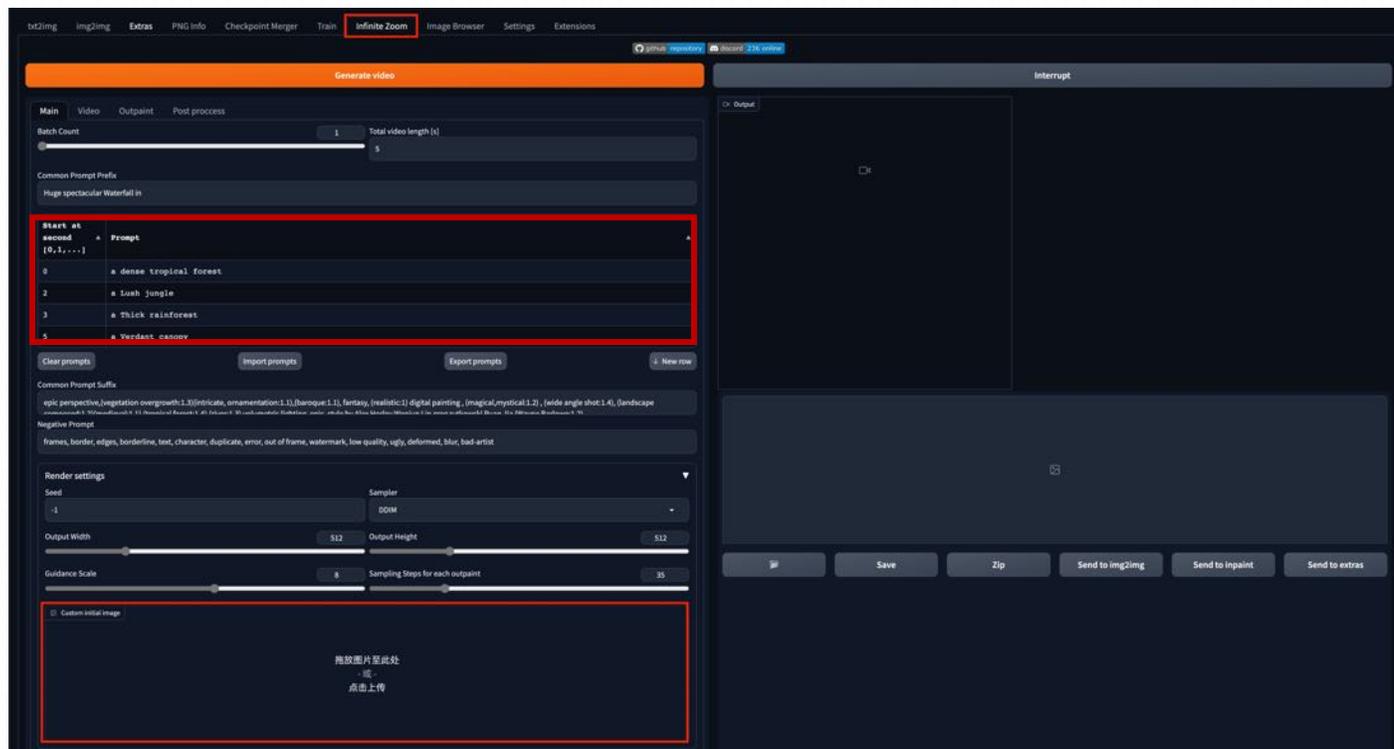
1.2 扩展插件推荐

F. Infinite Zoom

生成趣味无限缩放视频

搜索: `Infnite Zoom`

仓库地址: <https://github.com/vBhid/infnite-zoom-automatic1111-webui.git>



2. ControlNet 原理及应用

2.1 ControlNet 基本原理

2.2 ControlNet 安装

2.3 ControlNet 使用方式

2.4 ControlNet Model 详解

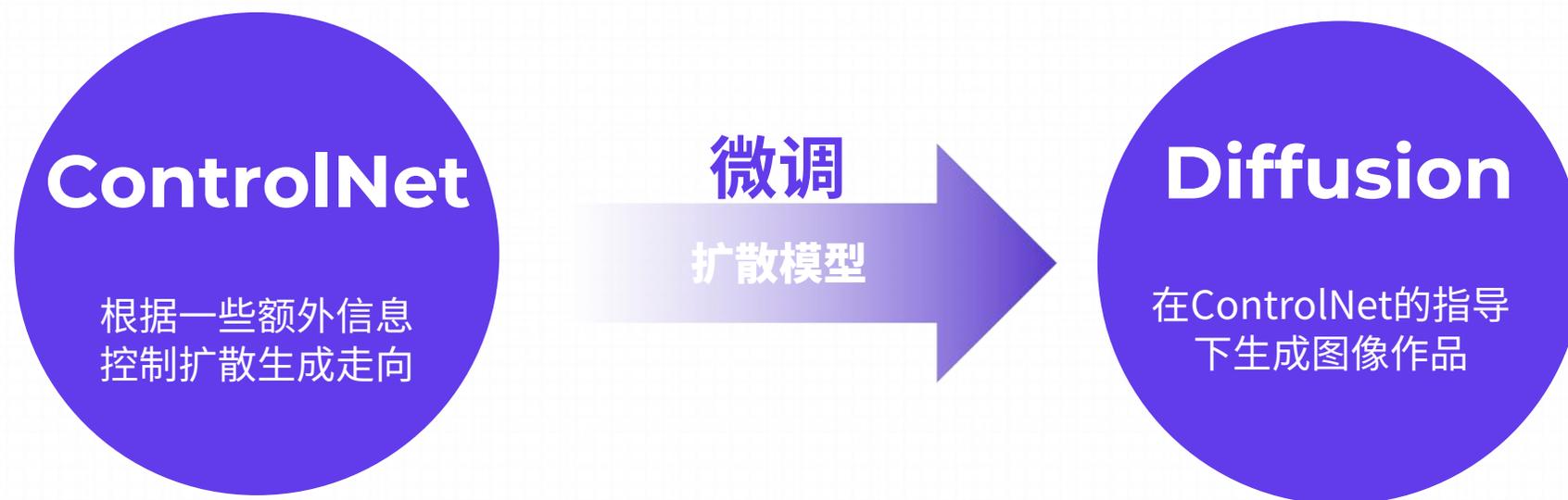
2.5 多重ControlNet 应用

2.1 ControlNet 基本原理

ControlNet, 直译过来是“控制网”,它本质上是一个StableDiffusion的扩展插件。Controlnet 是当前备受瞩目的AI 绘图算法之一。它是一种神经网络结构,通过添加额外的条件来控制基础扩散模型,从而实现了对图像构图或人物姿势的精细控制。结合文生图的操作,它还能实现线稿转全彩图的功能。

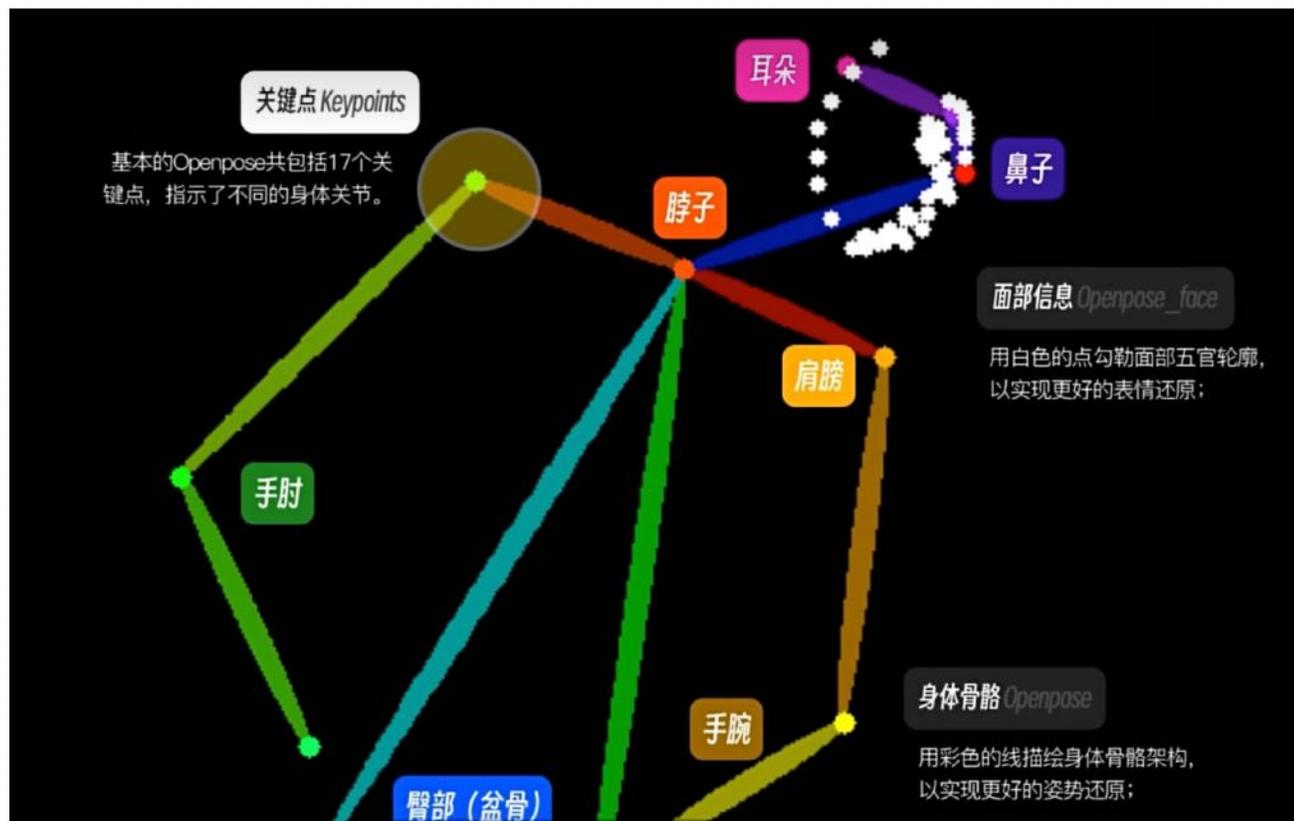
ControlNet 的意义在于它不再需要通过大量的关键词来堆砌构图效果。即使使用大量关键词,生成的效果也难以令人满意。借助 Controlnet 可以在最开始就引导它往你需要的构图方向上走,从而实现更准确的图像生成。

2.1 ControlNet 基本原理



ControlNet的核心作用是**基于一些额外输入给它的信息**，来给扩散模型的生成提供明确的指引

2.1 ControlNet 基本原理



这张图片上各种不同颜色的点线，代表的就是人物的五官，四肢关节。而ControlNet搭载了许多和checkpoint lower一样，用大量图片数据训练出来的记录了对应信息ControlNet控制模型。

ControlNet 控制模型和图生图有点像，它们本质上都是在通过一些方式给AI提供额外的信息。但ControlNet记录的信息比图像图里的图片更为纯粹，排除了图片本身元素——比如图片上面已有的颜色、线条的影响，只是单纯的输入姿势这一个点的信息，那就不会对其他你想要通过一些提示词，lora等去输入的信息构成太多影响，是“精准控制”里非常重要的一点。

2.1 ControlNet 基本原理

Anntator

从图片中提取对
ControlNet有用的
额外信息

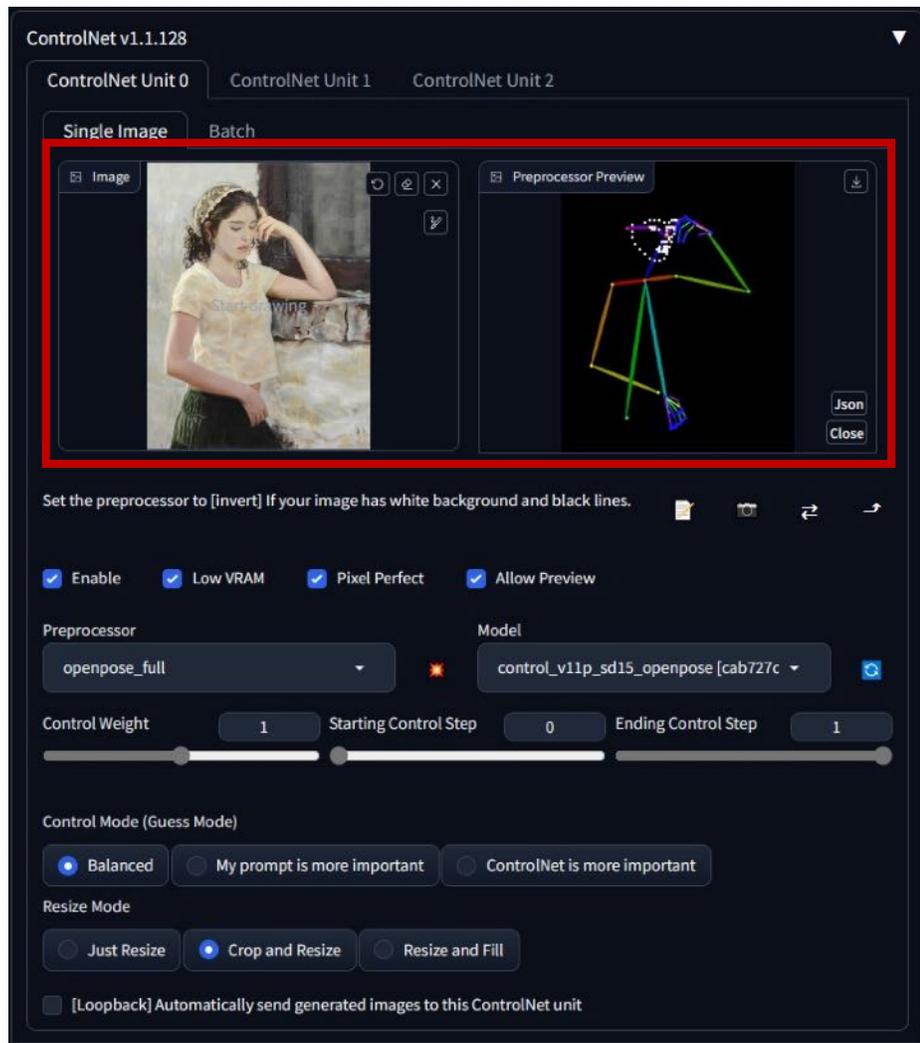
ControlNet

根据一些额外信息
控制扩散生成走向

Diffusion

在ControlNet的指导
下生成图像作品

2.1 ControlNet 基本原理



如果你在出图过程中使用了ControlNet，那出完图以后，它还会给你产出一张利用预处理器从图片里提取出来的信息图。

例如在openpose里就是一个大黑底，上面一堆五颜六色的线条了。我们一般把它叫做open pose的骨骼图，把它的刚刚产出的图片放在一起看，可以看到 AI 通过这些点线，构成了骨骼，确认了人物的五官，四肢到底在什么位置，从而实现了精准的姿势控制。

2.2 ControlNet 安装

A. 从扩展中安装

隐藏含有以下标签的扩展

脚本 localization tab dropdown 含广告 已安装 训练相关

模型相关 UI 界面相关 提示词相关 editing manipulations 线上服务

动画 query science 后期处理

排序

按发布日期倒序 按发布日期正序 a-z z-a 内部排

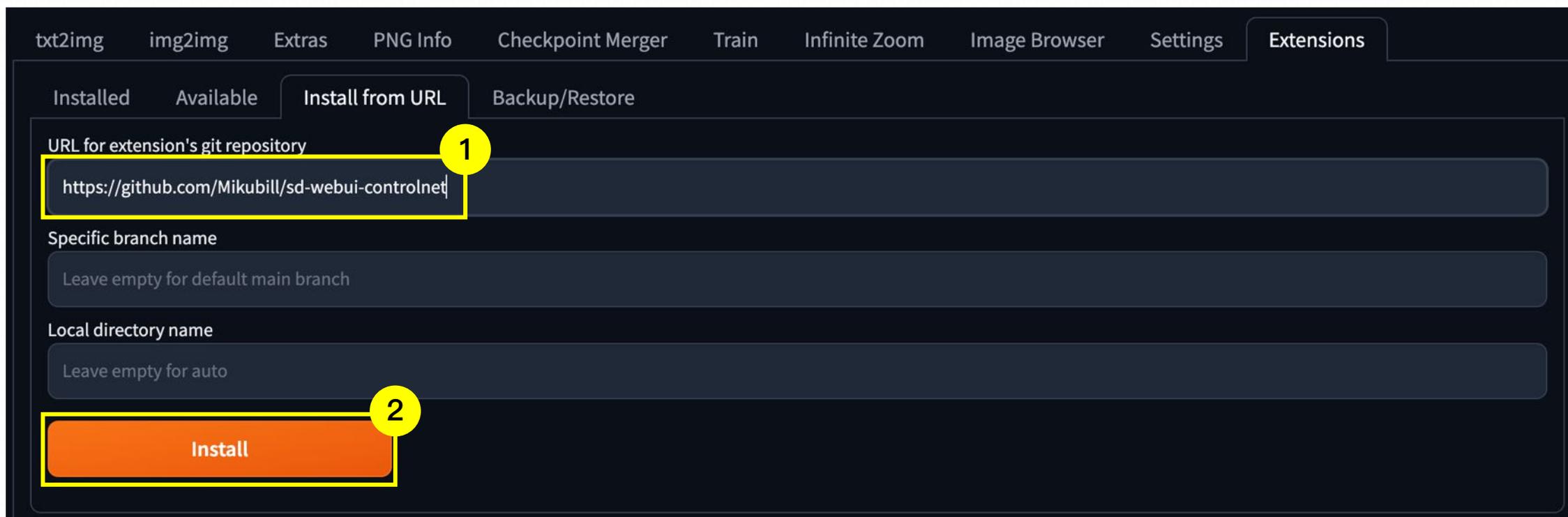
搜索

Controlnet

扩展	描述	操作
SD-CN-Animation 选项卡, 动画	Allows users to create video2video and text2video animations using any SD models as a backbone. Please, make sure that 'sd-webui-controlnet' extension is also installed. Added: 2023-05-06	安装
3D Openpose Editor 选项卡	Edit the pose of 3D models in the WebUI, and generate Openpose/Depth/Normal/Canny maps for ControlNet. Added: 2023-03-16	安装
3D Model&Pose Loader 选项卡	Load your 3D model/animation inside webui, or edit model pose as well, then send screenshot to txt2img or img2img to ControlNet. Added: 2023-03-11	安装
ebsynth_utility 选项卡, 动画	Extension for creating videos using img2img and ebsynth. Output edited videos using ebsynth. Works with ControlNet extension. Added: 2023-03-04	安装
sd-webui-controlnet manipulations	WebUI extension for ControlNet. Note: (WIP), so don't expect seed reproducibility - as updates may change things. Added: 2023-02-18	安装

2.2 ControlNet 安装

B. 从 github 网址安装

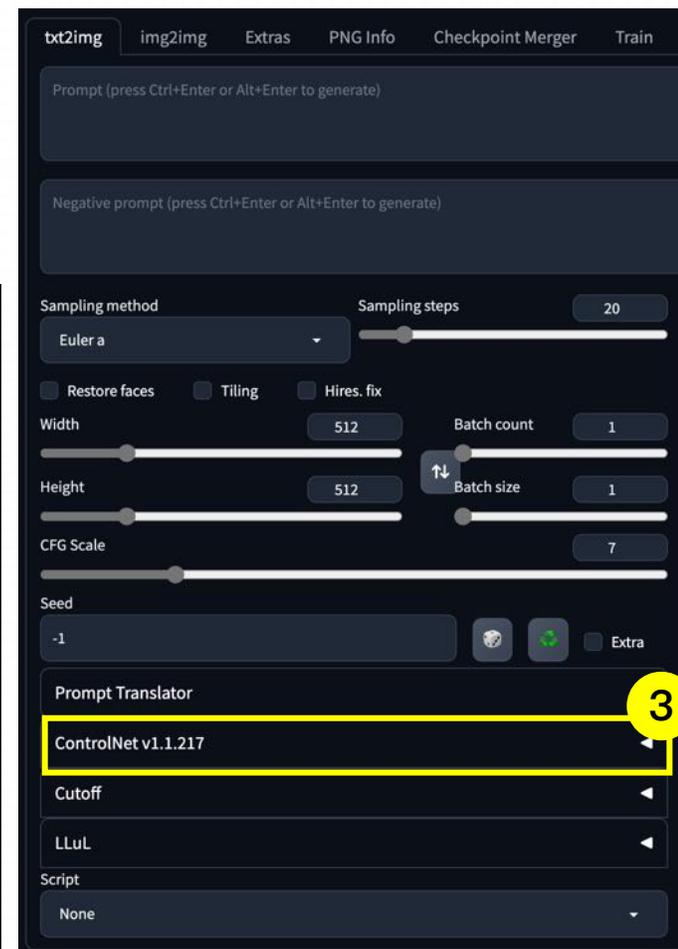


1. 点击扩展，然后再点击输入框

2. 复制网址后，点击安装

2.2 ControlNet 安装

B. 从 github 网址安装



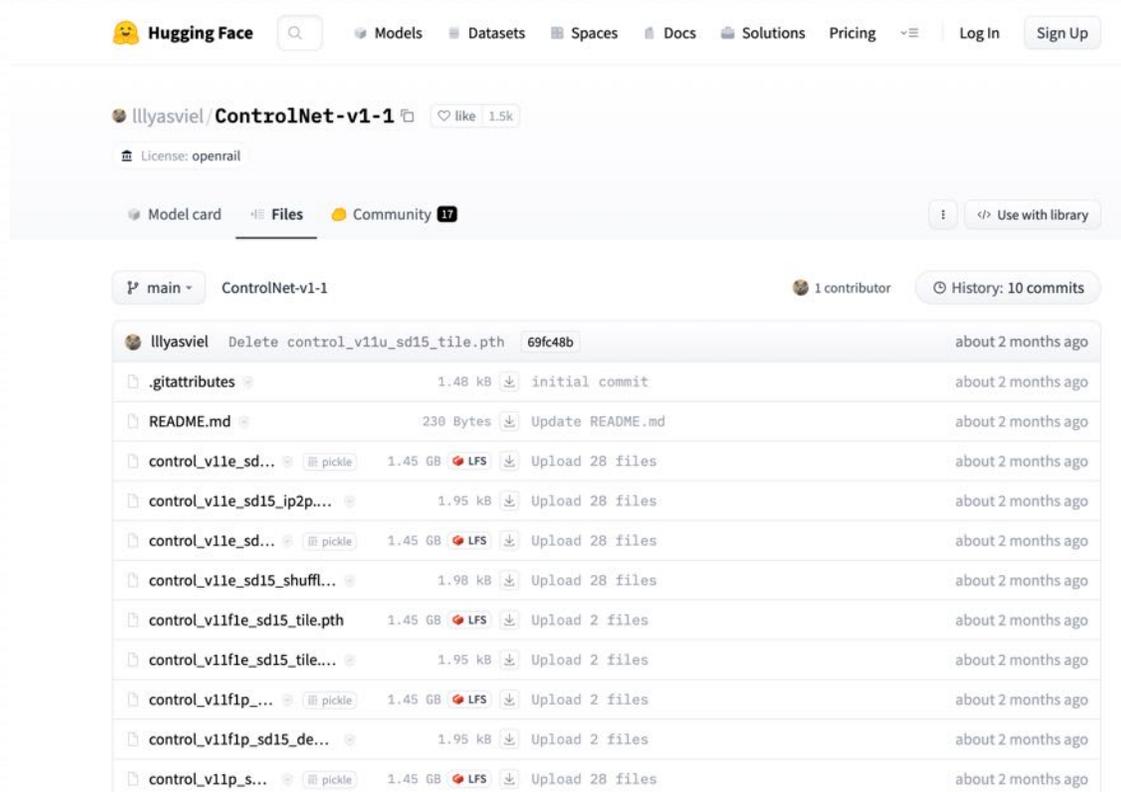
1. 勾选要启用的插件

2. 点击应用并重启

3. 回到首页确认有相对应插件即可

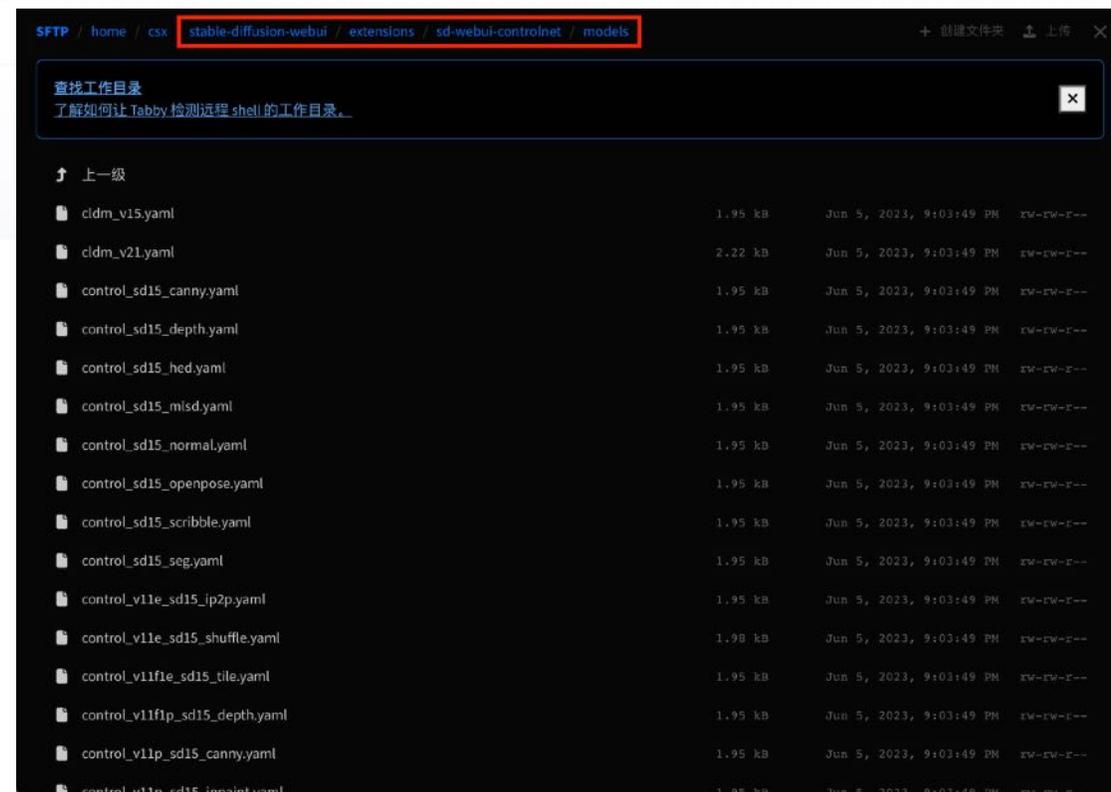
2.2 ControlNet 安装

安装完 ControlNet 插件后，还无法立刻使用它，ControlNet 需要控制模型才能生成图像



下载界面

<https://huggingface.co/lllyasviel/ControlNet-v1-1/tree/main>

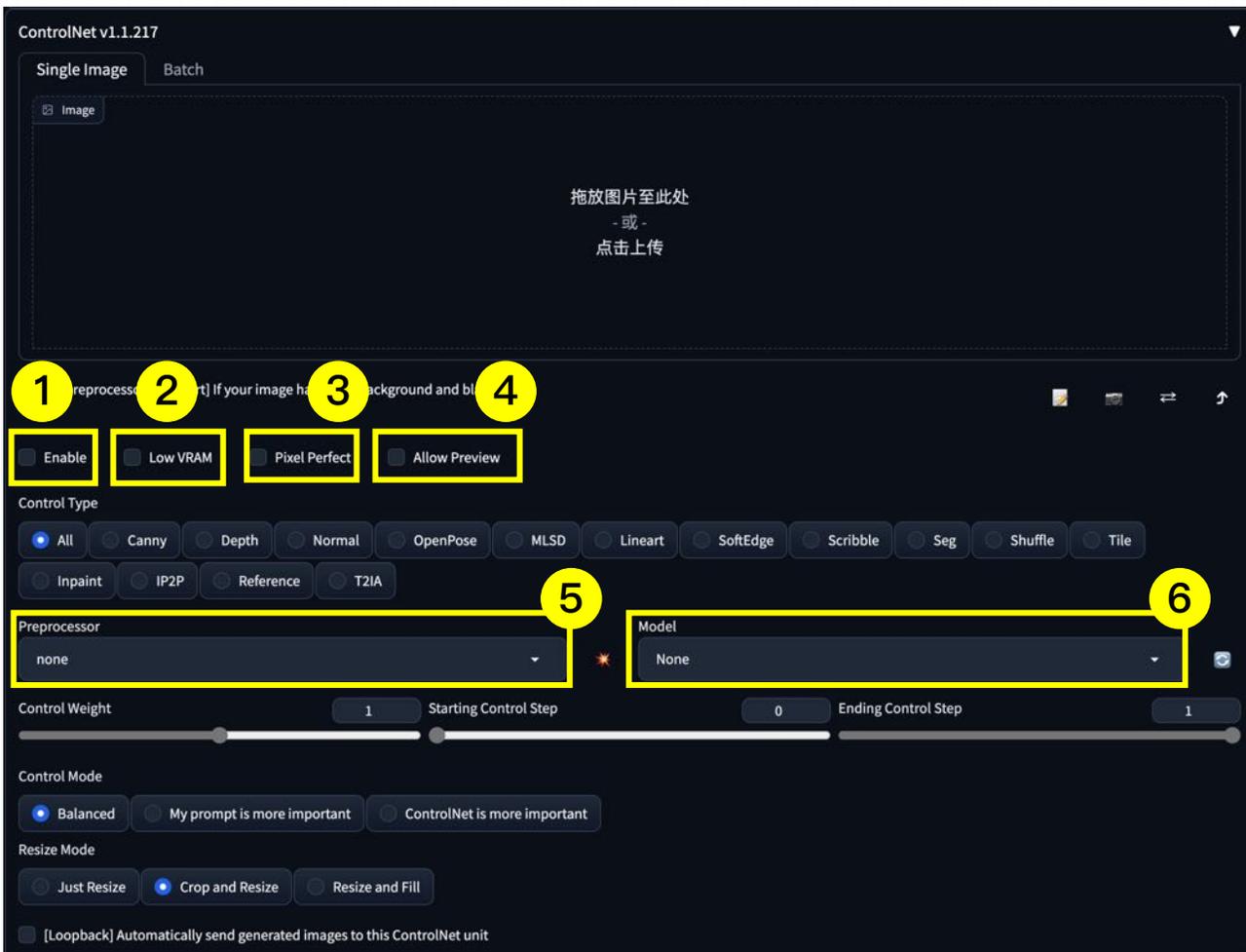


安装路径

[Stable-diffusion-webui/extensions/sd-webui-controlnet/models](https://github.com/lllyasviel/stable-diffusion-webui/tree/main/extensions/sd-webui-controlnet/models)

2.3 ControlNet 使用方式

A. ControlNet 界面介绍



1. **启用 (Enable)** : 勾选此选项后, 点击 “生成” 按钮时, ControlNet 才会生效。

2. **低显存优化 (Low VRAM)** : 低显存模式, 如果你的显卡内存小于等于4GB, 建议勾选此选项。

3. **完美像素 (Pixel Perfect)** : 它可以自动计算预处理器产出图像的最适合分辨率, 避免因尺寸不合导致的图像模糊变形。

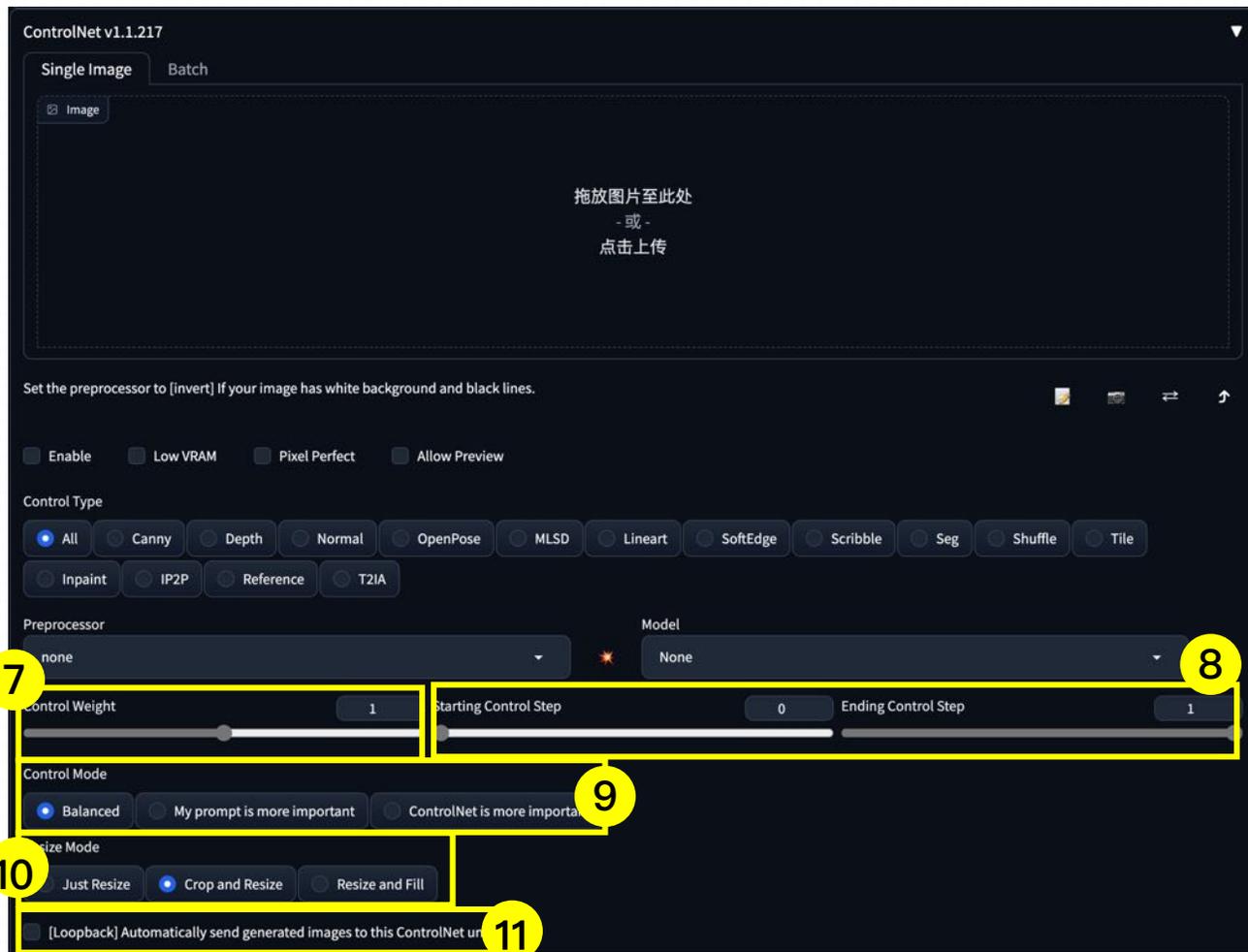
4. **预览 (Allow Preview)** : 会打开另外一个小的浏览器窗口, 是专门为预处理图像准备的。

5. **预处理器 (Preprocessor)** : 在此列表我们可选择需要的预处理器, 每个 ControlNet 的预处理器都有不同的功能。

6. **模型 (Model)** : 配套各预处理器需要的专属模型。该列表内的模型必须与预处理选项框内的名称选择一致, 才能保证正确生成预期结果。如果预处理与模型不致其实也可以出图, 但效果无法预料, 且一般效果并不理想。

2.3 ControlNet 使用方式

A. ControlNet 界面介绍



7. 控制权重 (Control Weight) : 决定这个控制效应在图片中呈现出来的强度。

8. Starting/Ending Control Step: 在图像不断扩散的过程中, 什么时候加入control net的影响。

- 介入时机: 代表第几步介入对画面的影响
- 退出时机: 代表第几步退出对画面的影响。

9. 控制模式 (Control Mode) :控制在提示词的影响与control net的信息之间更倾向于哪一个

- balanced mode(平衡模式)-代表二者兼顾

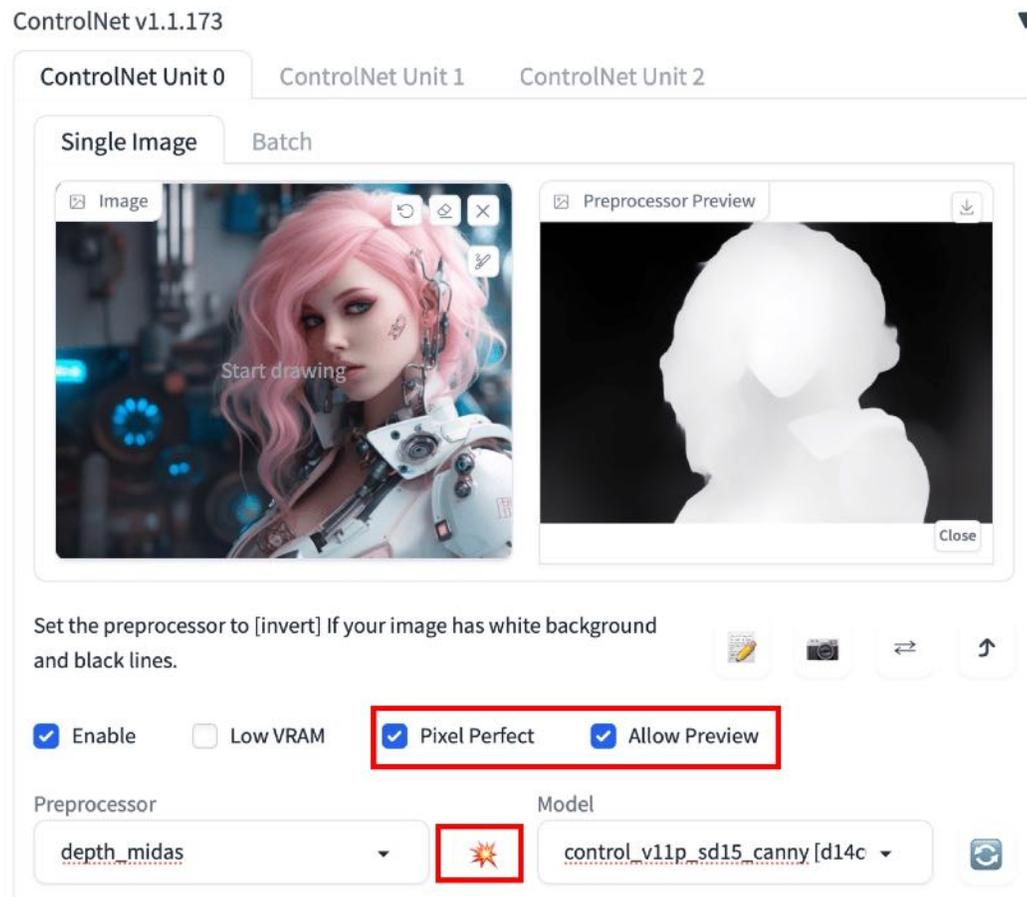
10. 缩放模式 (Resize Mode) :用于选择调整图像大小的模式。

- Just Resize(调整大小)
- Crop and Resize (裁剪和调整大小)
- Resize and Fill (调整大小并填充)

11. [loopback]: 一般只在一些反复迭代的连续生成场景下会用到。是一个比较特殊的选项, 默认关闭即可。

2.3 ControlNet 使用方式

B. ControlNet设置预处理器和模型



预处理器与模型的关系：

- 使用预处理器时要与模型一一对应起来,如与处理器选择 canny, 则相对应的模型也要选择 canny
- 预处理器和模型在 Stable Diffusion 中是相辅相成的, 预处理器通过对输入图像进行预处理和数据增强, 为模型提供了更好的输入, 而模型则通过对输入进行特征提取和生成高质量图像, 为预处理器提供了更好的反馈

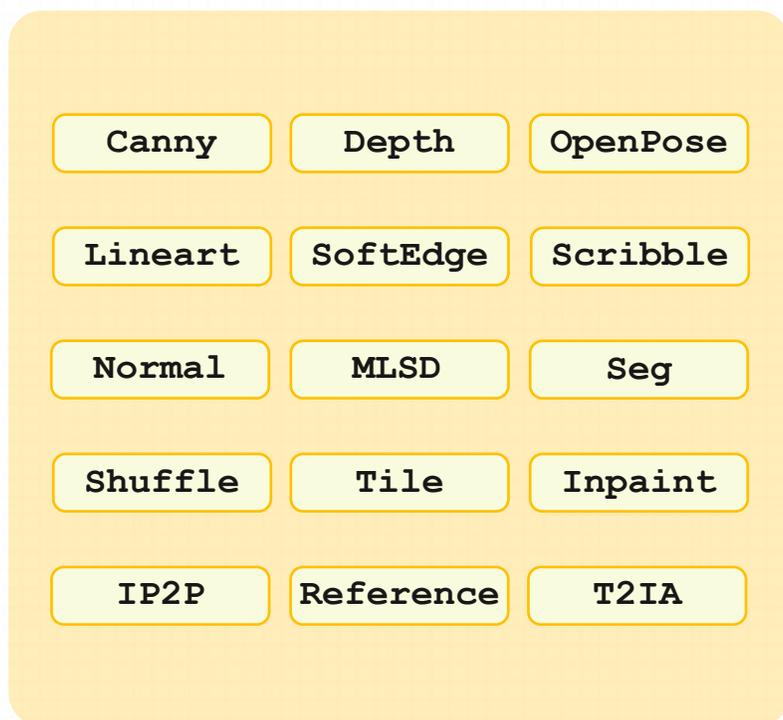
使用ControlNet的第一步是选择一个预处理器。打开预览功能会很有帮助, 这样您就知道预处理器在做什么。一旦预处理完成, 原始图像就会被丢弃, 只使用预处理后的图像进行ControlNet。

要打开预览功能：

1. 选择“Allow Preview”。
2. 可选择“Pixel Perfect”。ControlNet将使用您在文本到图像中指定的图像高度和宽度生成预处理后的图像。
3. 点击预处理器下拉菜单旁边的🔥图标。

2.3 ControlNet 使用方式

C. ControlNet 预处理器

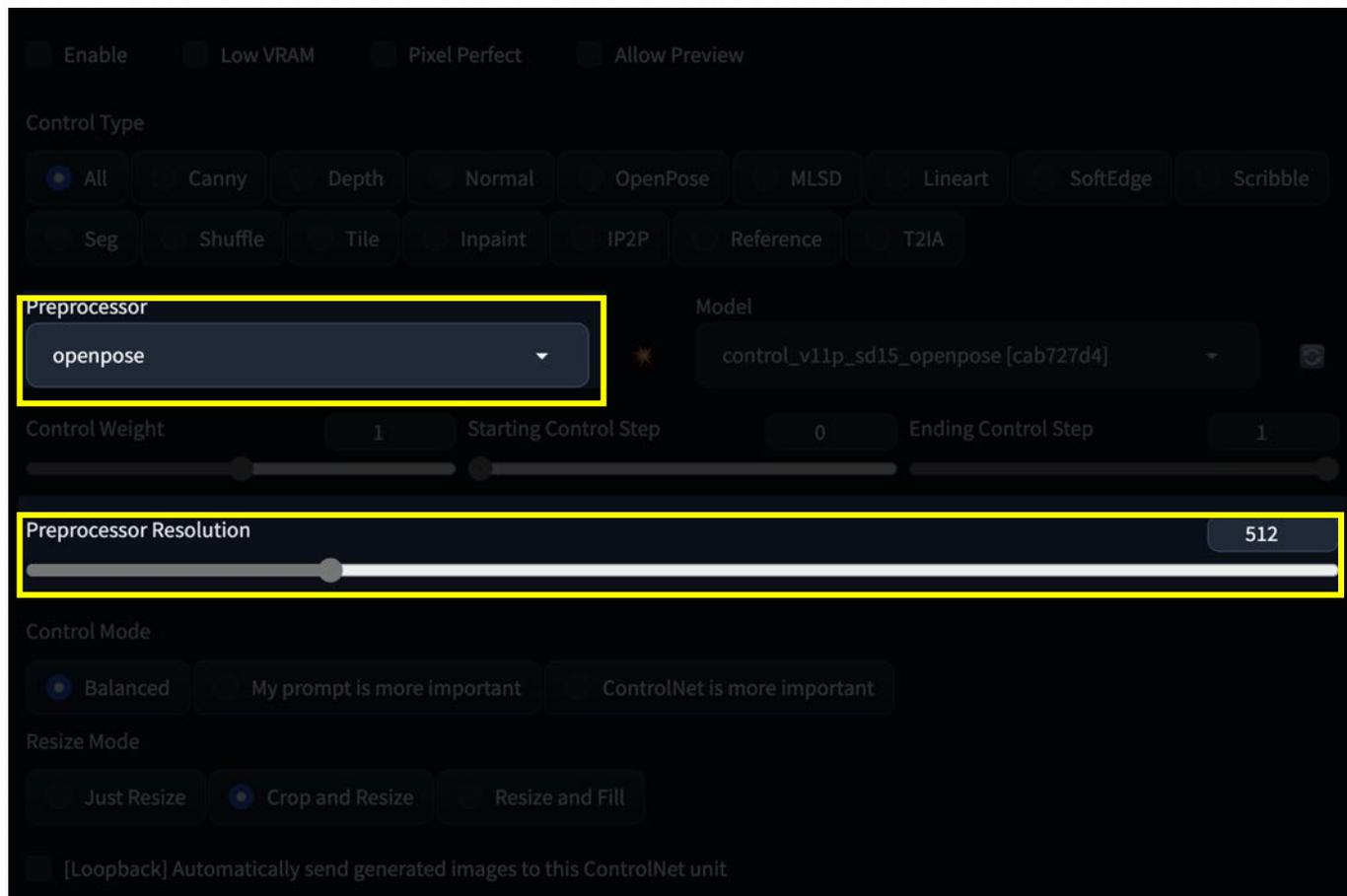


1.0 版本预处理器	1.1 版本预处理器	1.1 版本对应模型	模型说明
canny 边缘检测	canny 边缘检测	control_v11p_sd15_canny.pth	边缘检测, 提取线稿
mlsd 直线检测	mlsd 直线检测	control_v11p_sd15_mlsd.pth	直线检测, 适用于建筑设计
depth 深度检测	depth_midas	control_v11p_sd15_depth.pth	深度检测, 提取深度结构
depth_leres	depth_leres		
	depth_zoe		
normal_map 法线贴图	normal_bae	control_v11p_sd15_normalbae.pth	法线贴图, 提取法线信息
hed 边缘检测	softedge_hed	control_v11p_sd15_softedge.pth	软边缘检测, 保留更多边缘细节
	softedge_hed_safe		
	softedge_pidinet		
	softedge_pidinet_safe		
openpose 姿势检测	openpose 身体	control_v11p_sd15_openpose.pth	提取人物姿势
openpose_hand 手	openpose_face 身体+脸		
	openpose_faceonly 脸		
	openpose_hand 手		
	openpose_full 身体+手+脸		
scribble 涂鸦	scribble_hed	control_v11p_sd15_scribble.pth	涂鸦生图
fake_scribble	scribble_pidinet		
	scribble_xdog		
seg 语义分割	Seg_OFADE20K	control_v11p_sd15_seg.pth	语义分割, 不同颜色语义对应不同对象类型
	Seg_UFADE20K		
	Seg_OFCOCO		
	lineart_coarse	control_v11p_sd15_lineart.pth	提取精细线稿
	lineart_realistic		
	lineart_standard		
	lineart_anime	control_v11p_sd15_lineart_anime.pth	动漫线条输入
	inpaint	control_v11p_sd15_inpaint.pth	图像修复
	shuffle	control_v11e_sd15_shuffle.pth	风格迁移
		control_v11e_sd15_ip2p.pth	Pix2Pix 图片指令

```
invert (from white bg & black line)
canny
depth_leres
depth_leres++
depth_midas
depth_zoe
inpaint_global_harmonious
inpaint_only
lineart_anime
lineart_anime_denoise
lineart_coarse
lineart_realistic
lineart_standard (from white bg & black line)
mediapipe_face
mlsd
normal_bae
normal_midas
openpose
openpose_face
openpose_faceonly
openpose_full
openpose_hand
reference_adain
reference_adain+attn
reference_only
scribble_hed
scribble_pidinet
scribble_xdog
seg_ofade20k
seg_ofcoco
seg_ufade20k
shuffle
softedge_hed
softedge_hedsafe
softedge_pidinet
softedge_pidisafe
t2ia_color_grid
t2ia_sketch_pidinet
t2ia_style_clipvision
threshold
tile_colorfix
tile_colorfix+sharp
tile_resample
```

2.3 ControlNet 使用方式

C. ControlNet 预处理器



选好预处理器以后，下面也会多出一系列的参数，之前主要决定预处理器如何从图片里提取信息并生成信息图。每一种预处理器可能会有差别，初始状态下的参数可以应付绝大多数场景，你可以不用去更改它。

Processor Resolution代表的是预处理图像的分辨率，一般它都会比原图小很多来节约资源，因为标注这些信息也不需要很高清的图片。

如果性能富余，又是在使用Canny、Lineart、Depth等对精度有一定要求的模型时，可以尝试将它开大，或者设置成和图片一样的分辨率来获取更好的效果。

2.3 ControlNet 使用方式

D. ControlNet 模型

项目名称

官方的 controlnet 总是使用 [Control] 。任何第三方模型可以使用其他名称，以区别于官方模型。比如 T2-Adapter 的文件是以 [t2iadapter] 开头的

质量标志

[p] 表示可以正式使用
[e] 表示实验版本，可能不稳定
[u] 表示未完成，功能不可用。

控制方式

我们强烈建议使用的模型必须与预处理器(注释器)的名称一致或部分一致。

↓

control_v11p_sd15_canny.pth



版本标志

V11表示版本1.1，第三方模型可以使用任意标志，对 Controlnet 官方来说：[f] 表示 bug已修复，V11f1 表示版本1.1的第一版修复。

大模型/底模型型号：

sd1.5表示 stable diffusion 1.5，sd2.1v表示 stable diffusion 2.1v-768 预处理器和.pth模型的版本号要一致，比如 t2iadapter 的 sd1.4 color 预处理器就必须配 sd1.4 的.pth模型,如果是 sd1.5 的.pth模型就会不起作用。

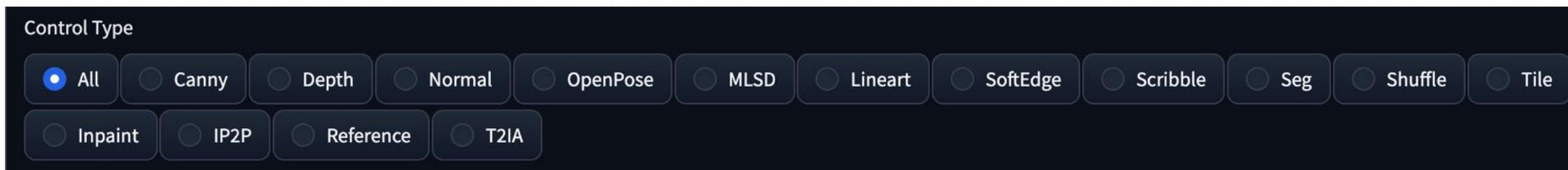
文件扩展名

[pth] 表示模型文件
[yaml] 表示配置文件
从 Controlnet V1.1 开始所有模型都必须有一个文件名完全相同的配置文件。

2.3 ControlNet 使用方式

D. ControlNet 模型

预处理器需要与相对应的模型一起使用，才可以起到作用。在v1.1中，很容易确定要使用哪个模型，只需要选择与预处理器具有相同开头关键词的模型即可。



Preprocessor	Model
depth_xxxx	control_xxxx_depth
lineart_xxxx	control_xxxx_lineart
openpose_xxxx	control_xxxx_openpose

2.4 ControlNet Model

ControlNet是一种稳定扩散模型，可以从参考图像中复制构图或人体姿势。

经验丰富的稳定扩散用户知道，生成他们想要的精确构图是多么困难。生成的图像有些随机。你能做的只是进行尝试：生成大量图像，然后选择你喜欢的一张。

有了ControlNet，稳定扩散用户终于有一种方式来精确控制主题的位置和外观！

2.4.1 OPENPOSE (姿态检测)

Openpose 模型有5个预处理器

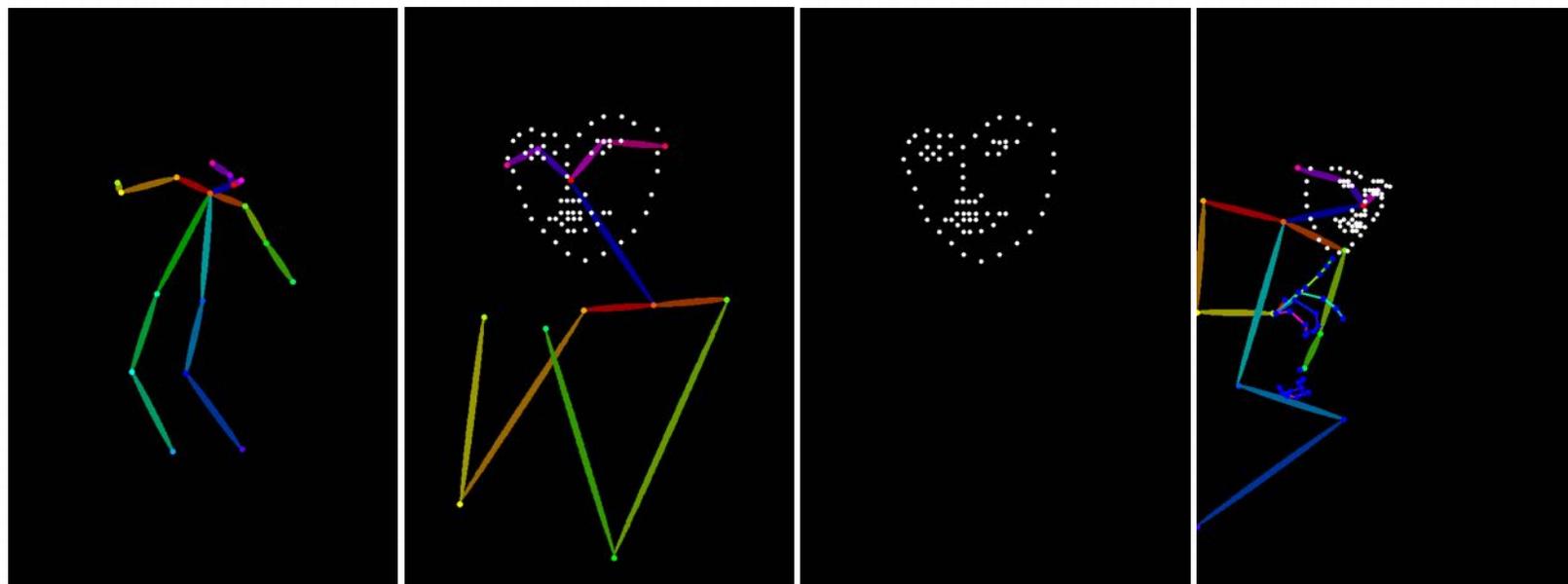
openpose

openpose_face

openPose_faceonly

openpose_full

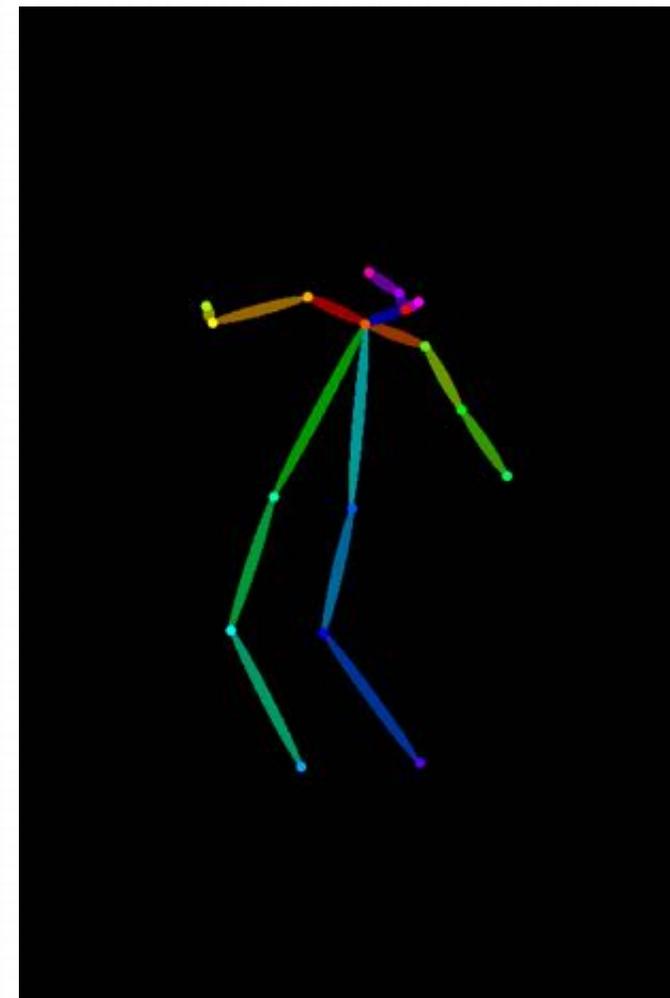
openpose_hand



2.4.1 OPENPOSE (姿态检测)

A. openpose

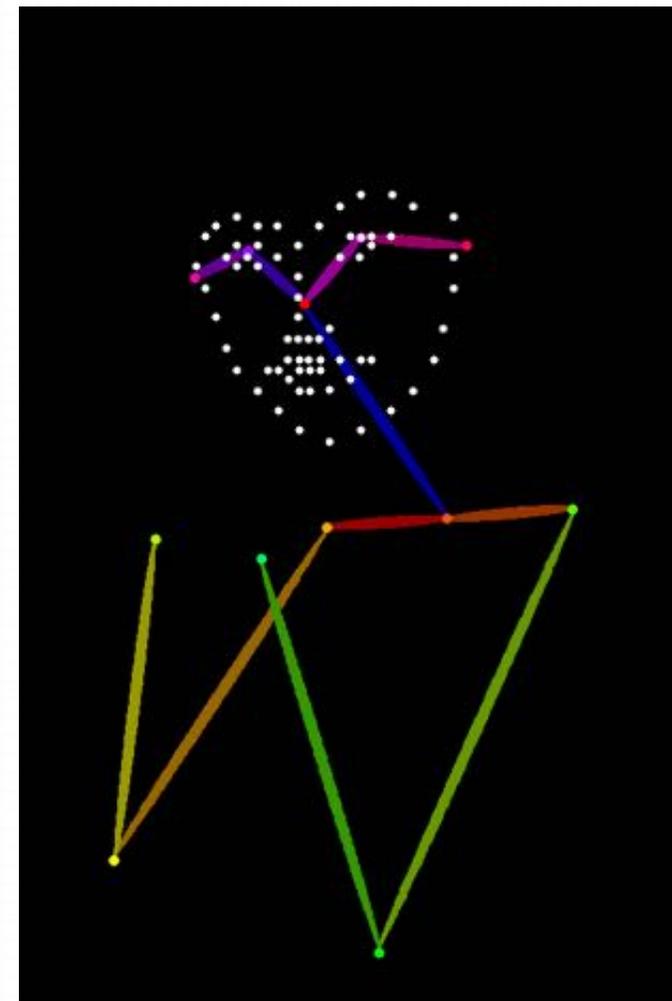
OpenPose 是基本的 OpenPose 预处理器，可检测眼睛，鼻子，眼睛，颈部，肩部，肘部，手腕，膝盖和脚踝的位置。



2.4.1 OPENPOSE (姿态检测)

B. openpose_face

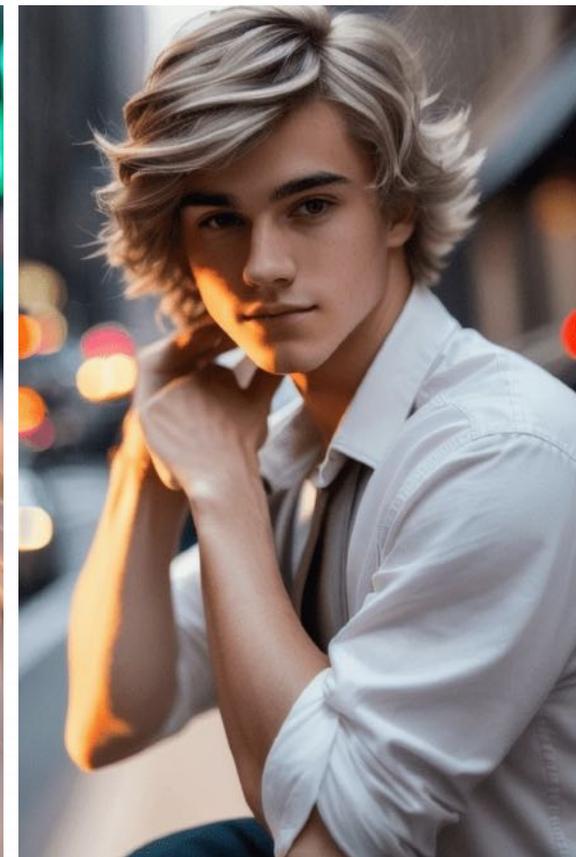
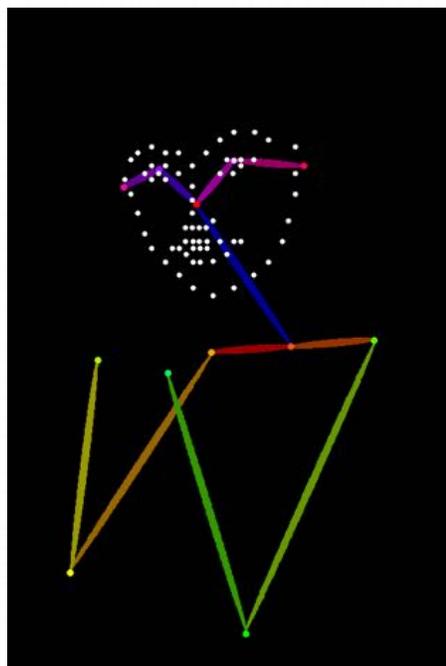
选择openpose_face的，会锁定人物的面部表情特征。ControlNet 会通过具体的小点把眼睛、鼻子、嘴巴的轮廓勾勒出来了。这就意味着它可以控制人物的具体面部表情了。



2.4.1 OPENPOSE (姿态检测)

B. openpose_face

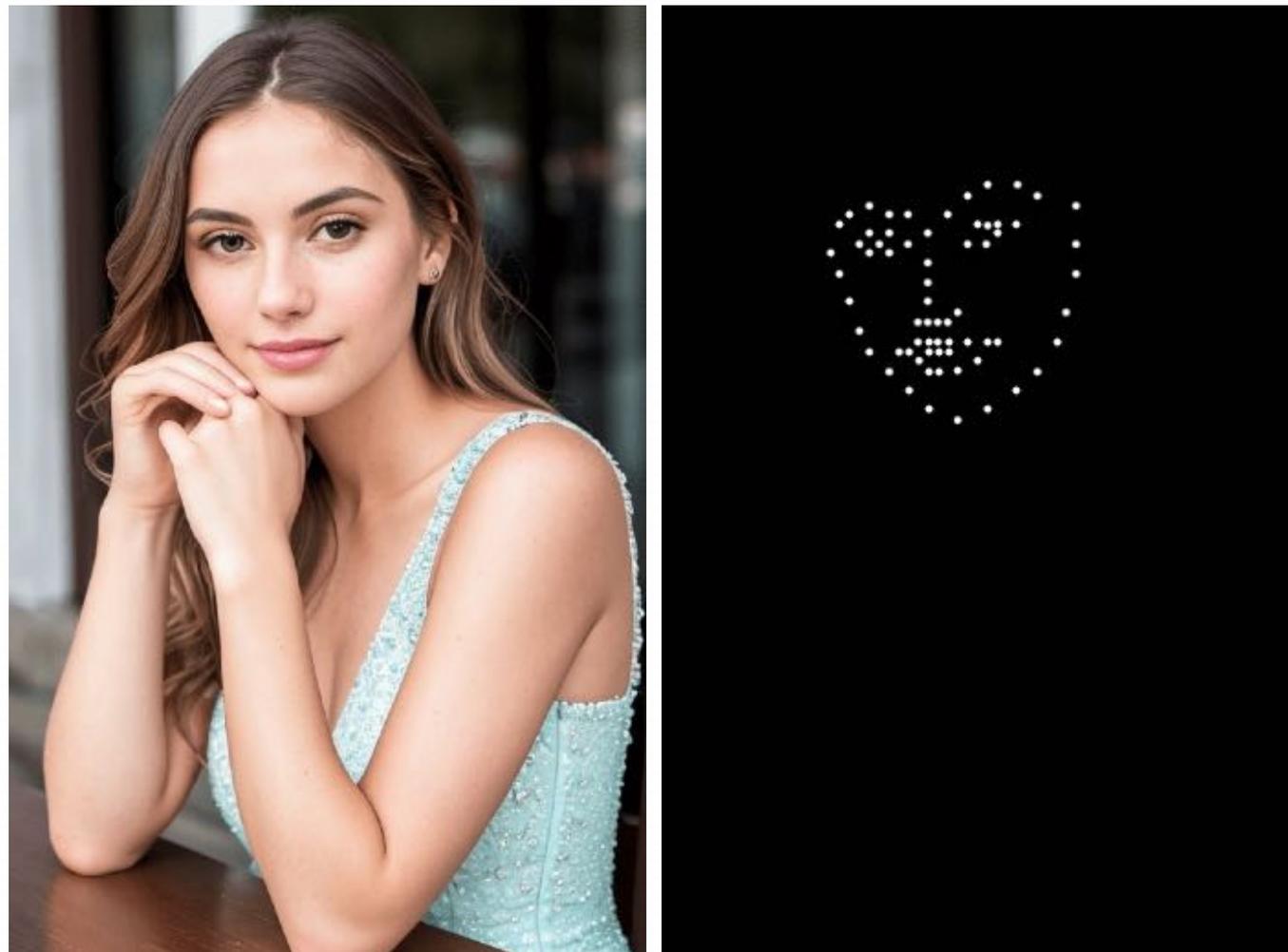
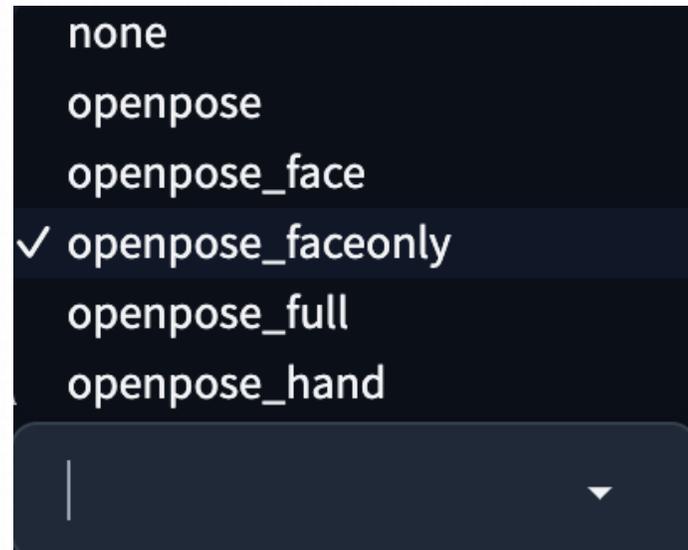
示例图像:



2.4.1 OPENPOSE (姿态检测)

C. openpose_faceonly

如果画面中没有四肢，主要用于刻画面部，那也可以考虑faceonly，它只会预处理面部的信息，比较适合大头照。

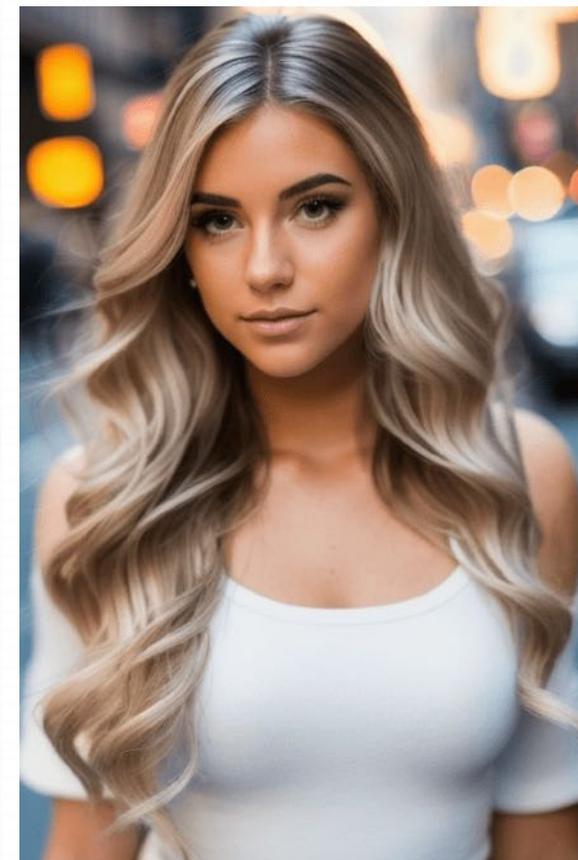
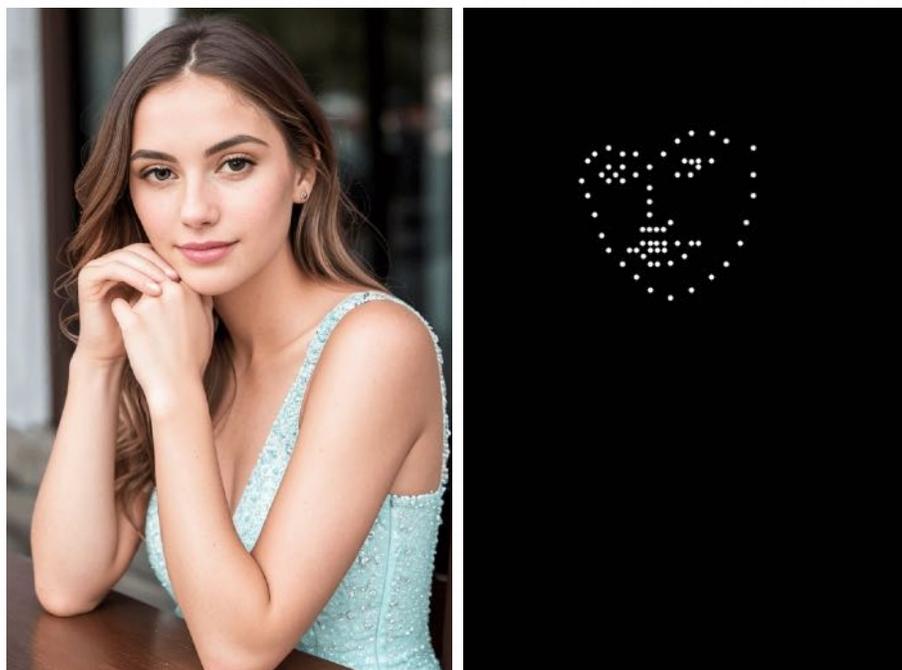


2.4.1 OPENPOSE (姿态检测)

C. openpose_faceonly

示例图像:

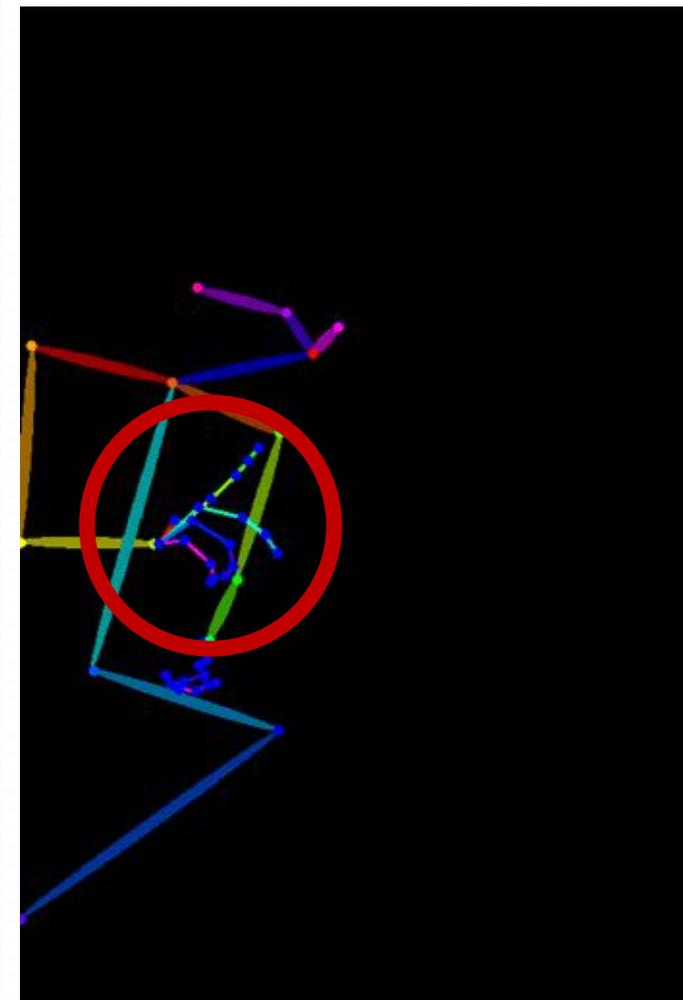
如右图所示，固定了人物的面部，但身体不受约束。



2.4.1 OPENPOSE (姿态检测)

D. openpose_hand

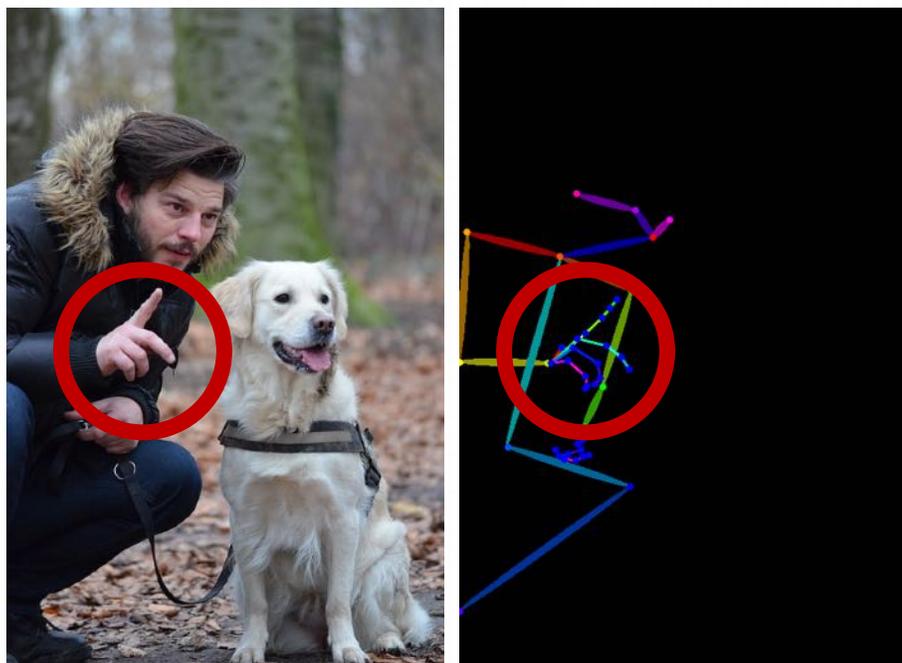
Openpose_hand 解决了绘制手的问题。如下图所示，基本的骨骼之外添加对手部骨骼的精细刻画。这些节点是精确到每个手指关节上的。利用带手部骨骼的control net生成图片人物的手指效果非常显著。



2.4.1 OPENPOSE (姿态检测)

D. openpose_hand

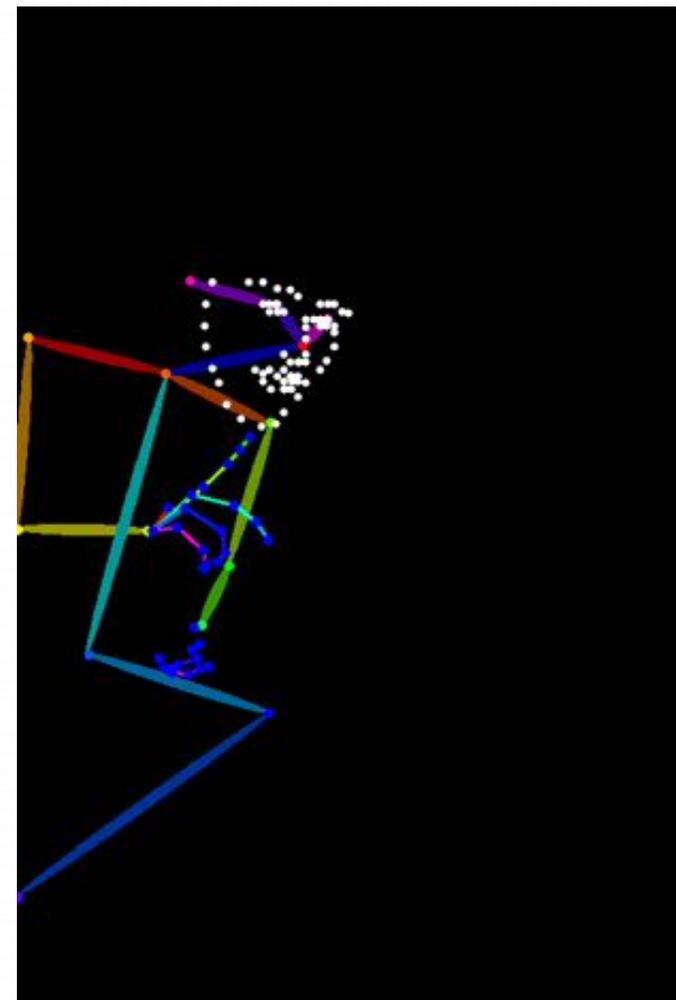
示例图像:



2.4.1 OPENPOSE (姿态检测)

E. openpose_full

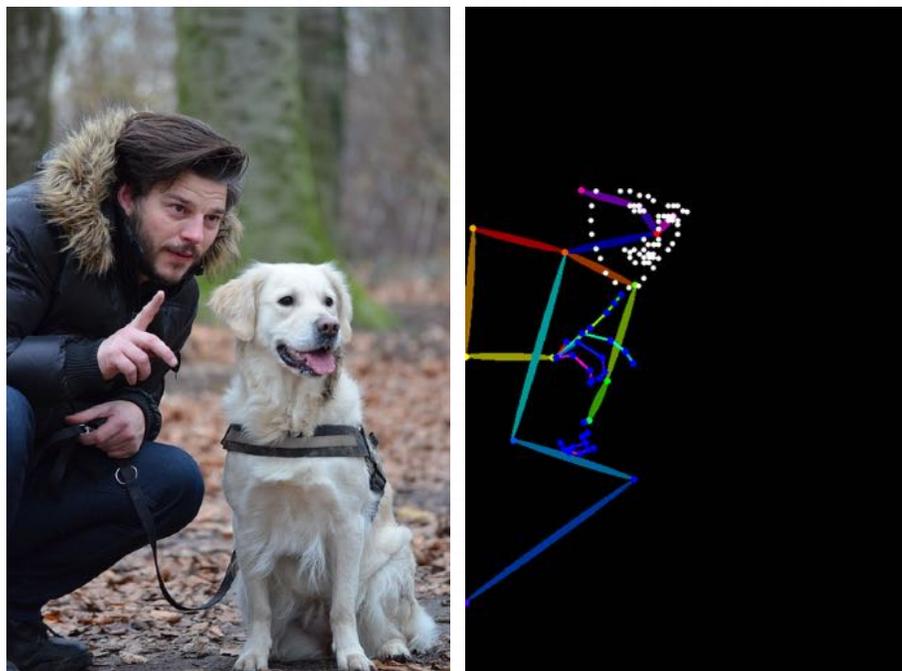
把上述所有强大的功能加在一起，就是这个 openpose_full 了。又有脸又有手。



2.4.1 OPENPOSE (姿态检测)

E. openpose_full

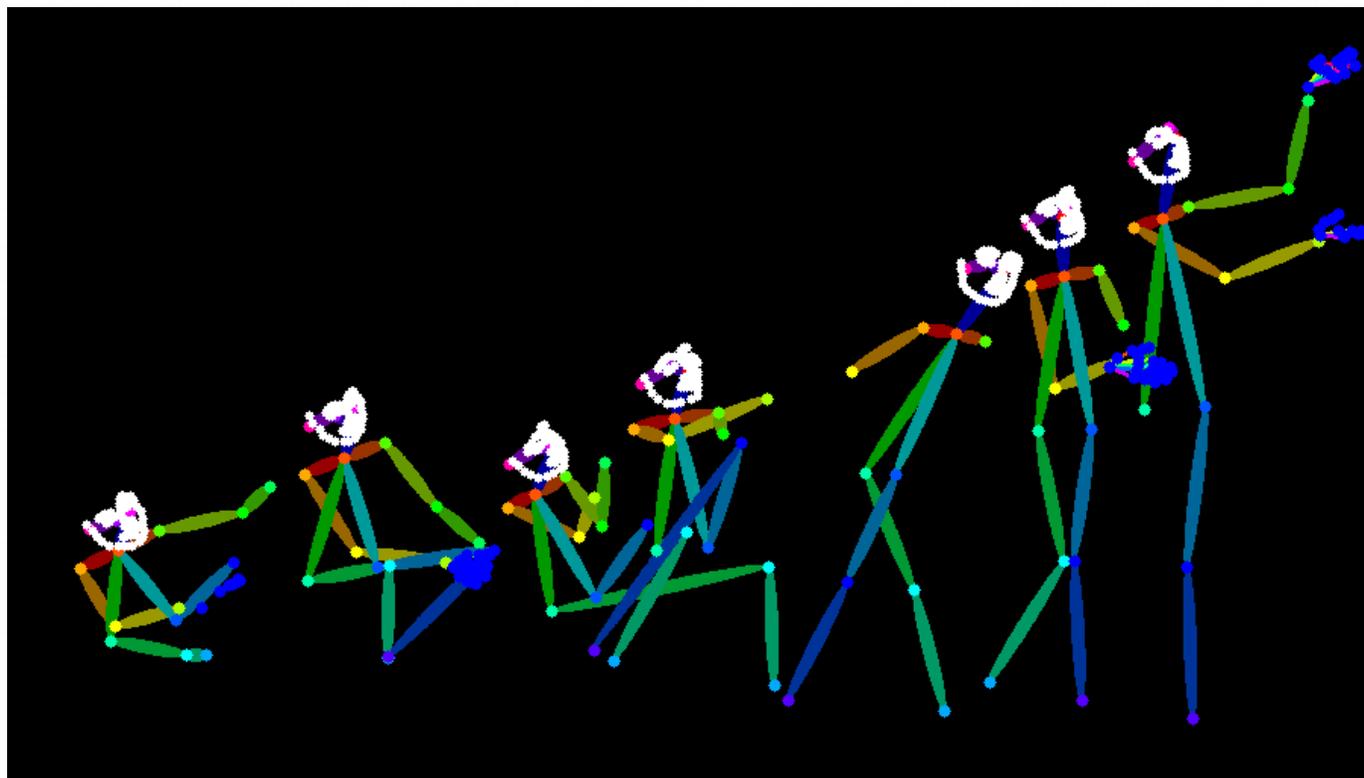
示例图像:



2.4.1 OPENPOSE (姿态检测)

E. openpose_full

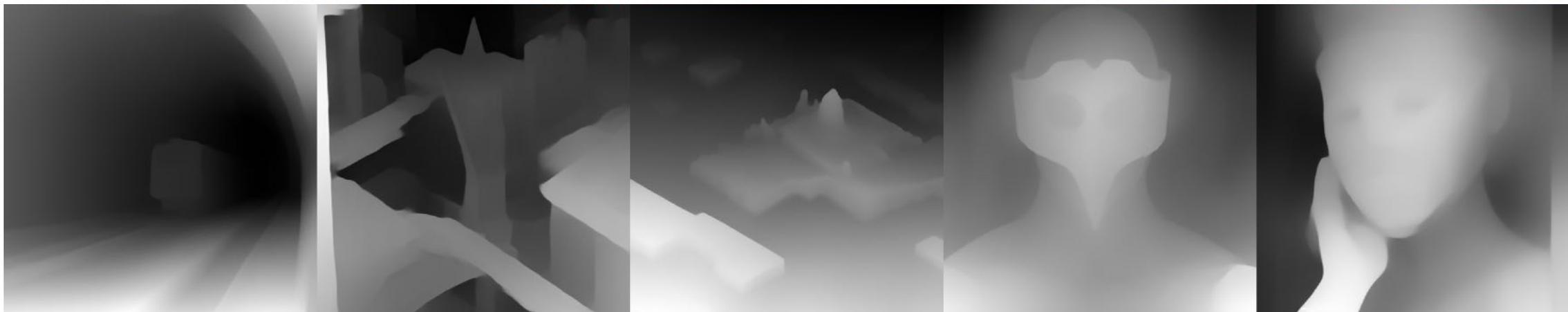
除了生成单人的姿势， openpose甚至可以生成多人的姿势，这点非常关键，在此之前AI生成的画面里多个人物的特定动作是几乎无法靠提示词来实现的。



2.4.2 Depth (深度)

我们上传一张图片，Depth 会识别出一张纯黑白的图片，在这张黑白图里颜色代表的就是一种空间层面的深度。越黑的地方离你越远，而越白的地方离你越近。我们就可以通过这样一张非常简单的图片来向AI完整的传递这幅图片空间信息。

Depth 的应用重点是对场景的描绘还原，尤其是富有空间感的多层次场景。



2.4.2 Depth (深度)

目前depth有4种预处理器，其中 **Leres++** 这个得到的效果是最为精细的。但它的处理时间会很长。所以如果你不精细度要求没有这么高，可以采用其他预处理器。

Depth Midas

经典的深度估计器。也被用于官方v2深度到图像模型中。

Depth Leres

更多细节，但也倾向于渲染背景。

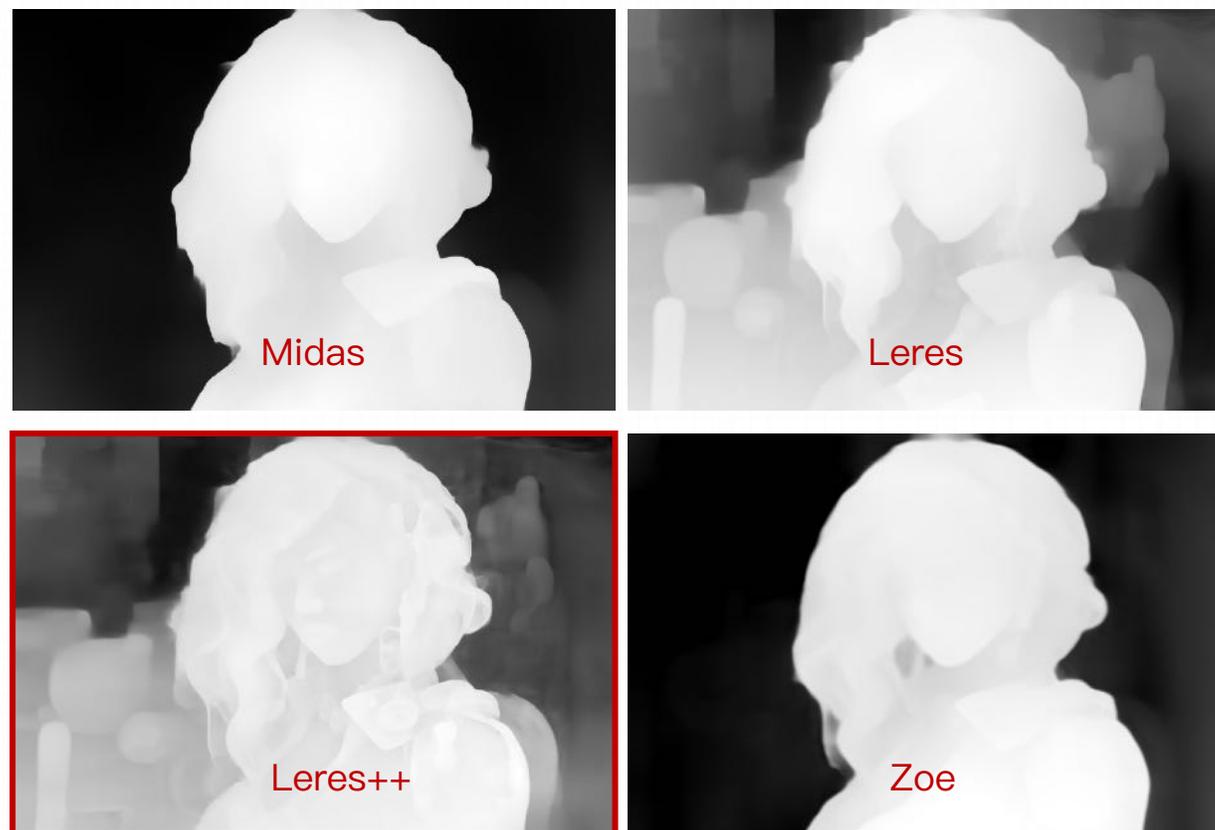
Depth Leres++

更多细节。

Zoe

细节水平介于Midas和Leres之间。

2.4.2 Depth (深度)



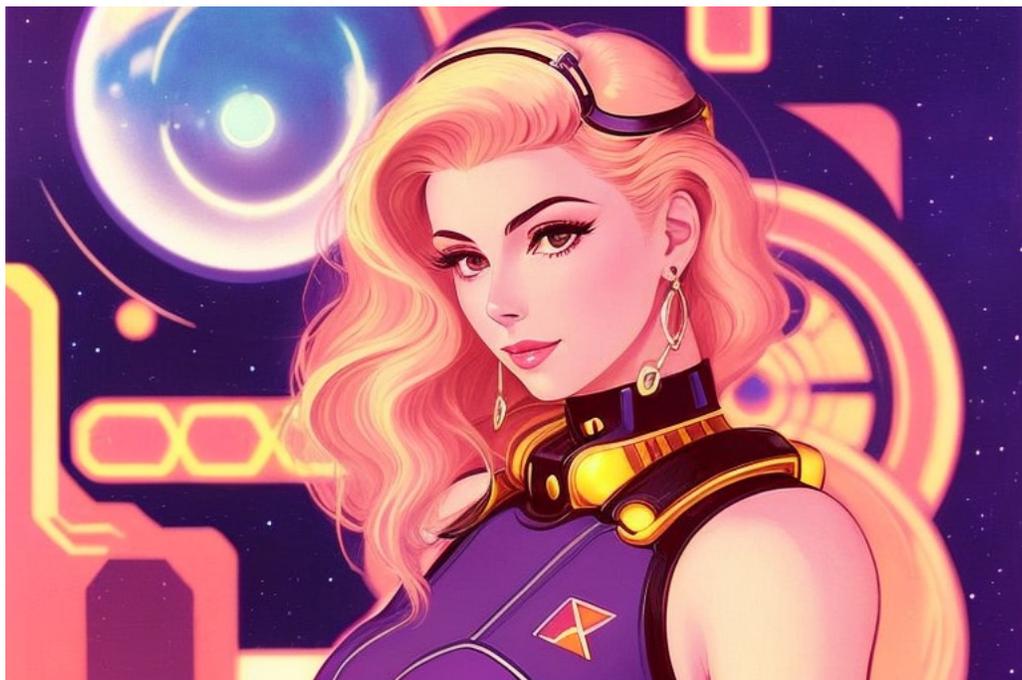
2.4.2 Depth (深度)

Prompt

a woman retrofuturism

negative prompt

disfigured, ugly, bad, immature



Depth Zoe



Leres++

2.4.2 Depth (深度)

Depth 其实也可以用于刻画人物，尤其是那些涉及肢体交叠、人体透视的场景。

以这张图片为例，这是一位在做拉伸动作的男性。我们如果希望 AI 还原他的动作，一般思维就是使用openpose，但你会发现open pose 长出来的字是奇奇怪怪的，和原图根本一点关联都没有。看一眼骨骼图，会发现 openpose 认不得这个手臂交叉的动作。



2.4.2 Depth (深度)

Depth 其实也可以用于刻画人物，尤其是那些涉及肢体交叠、人体透视的场景。

但如果你选用的是depth来处理，可以很清晰的看到这个一只手臂在前，一只手臂在后的结构。这样AI就能更好的理解这个动作了。



2.4.2 Depth（深度）

在后面我们会提到的Multi ControlNet这里，他们还可以一起联合共同发挥作用。



但有一点要注意，使用depth，描摹人像会带来一个副作用，人物和背景间的深度差异，会限制人物形象的发挥空间，就是个人体形状会非常固定，不利于AI的二创。如果它影响了你想要实现的效果，就适当降低权重，或者选用提示词更重要的模式。

2.4.3 Canny（边缘检测）

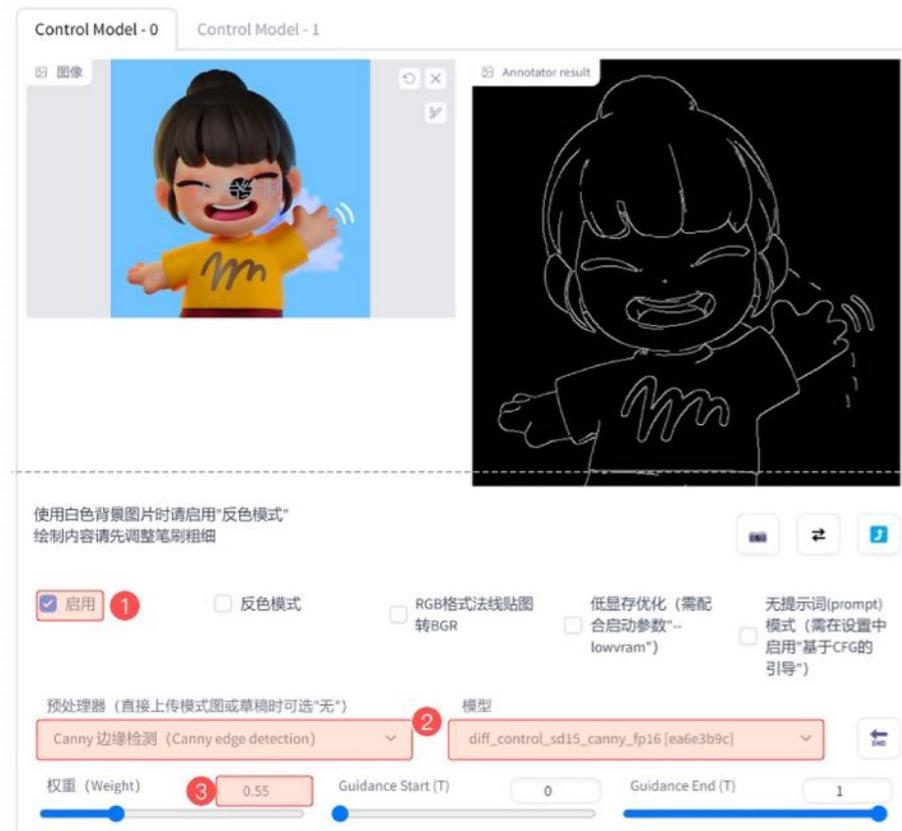
Canny 概念

- Canny 是一种经典的边缘检测算法，能够很好地识别出图像中的边缘信息，对于图像生成任务来说，可以帮助模型更好地生成具有清晰边缘的图像。
- Canny 从上传的图片中生成线稿，然后根据关键词来生成与上传图片同样构图的画面。
- Canny 处理图片的方式是将图片转为线稿，对于画面细节不复杂的图像来说很好；但处理复杂图像时，主体会容易被除主题元素以外的元素影响。

2.4.3 Canny (边缘检测)

Canny 使用说明

扩散控制网络(ControlNet)



①
勾选启用
否则不生效

②
处理器
对应模型

③
勿忘调节
权重

使用说明 (以下其它模型同理) :

1. 展开 ControlNet 面板, 上传参考图, 勾选 Enable 启用 (如果显存小于等于 4G, 勾选低显存模式)。
2. 预处理器选择 Canny (注意: 如果上传的是已经经过预处理的线稿图片, 则预处理器选择 none, 不进行预处理), 模型选择对应的 `control_v11p_sd15_canny` 模型。
3. 勾选 Allow Preview 允许预览, 点击预处理器旁的  按钮生成预览。

<https://www.uisdc.com/controlnet>

2.4.3 Canny (边缘检测)

Canny

Canny只有一种处理手段，运行后处理出来的线条图是上一幅带有草稿味道的黑底白线线条图。



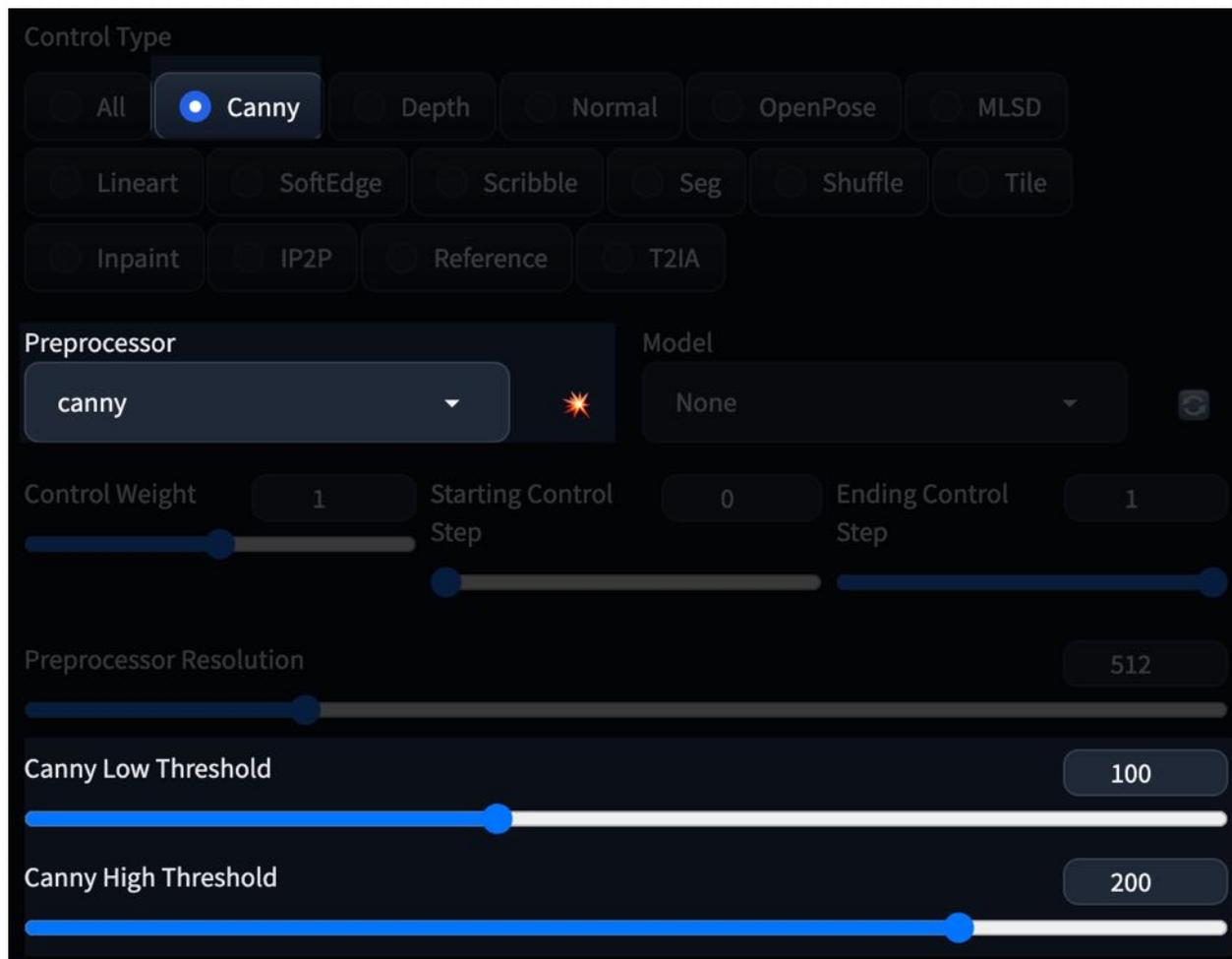
2.4.3 Canny (边缘检测)

Canny 可以被应用在几乎任何需要还原图像外形特征的场景里。比如画一只狗狗，一辆车子。这些复杂形象通过它们的外形得到了，那Canny就会把这种外形特征使用线条勾勒出来，在生成的时候控制图像还原。另外如果你的图像里包含了一些需要精确表达的内容，例如文字标识，那使用Candy可以有效确保它不变形。



生成的图像将遵循轮廓

2.4.3 Canny (边缘检测)



利用Canny识别的时候，如果有些线条无法被识别出来，那你可以考虑适当降低 **Canny Low / High threshold** 的数值，它能将更多明暗差异不是那么明显的部分的线条识别主要让形体更加准确，但注意线条也不能太过密集了。不然生成出来的图像上可能会有很多无用的小细节，显得有点脏。

- Canny 低阈值 (识别暗部)
- Canny 高阈值 (识别亮部)

2.4.4 Lineart

自从 ControlNet 更新到1.1版本后，Canny也有了一个更为高级的实现，就是Lineart。Lineart 提取的线稿更加精细，细节更加丰富。

Lineart 的四种预处理器模式：

Line art anime

动漫风格的线稿提取

Line art anime denoise

粗略线稿提取

Line art realistic

写实线稿提取

Line art coarse

线条更粗重的写实风格

2.4.4 Lineart

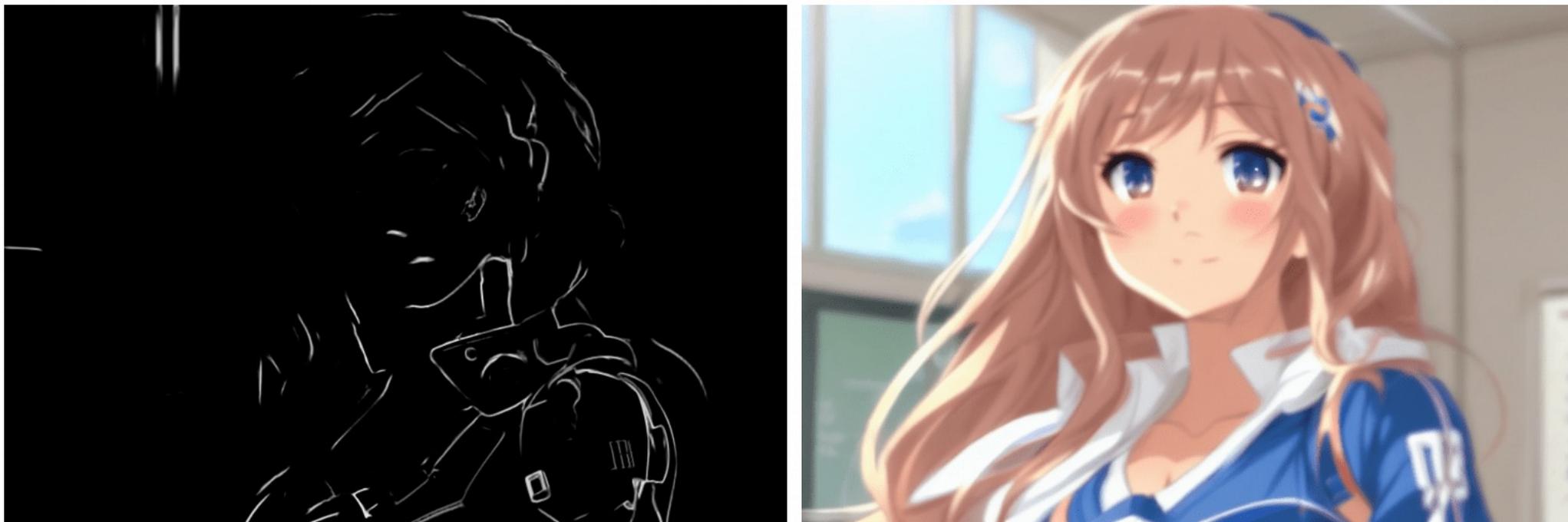
A. Line Art Anime 动漫风格的线条



上方的图像是使用控制权重设置为0.7生成的。

2.4.4 Lineart

B. Line Art Anime Denoise 细节更少的动漫风格线条



上方的图像是使用控制权重设置为0.7生成的。

2.4.4 Lineart

C. Line Art Realistic 写实风格的线条



上方的图像是使用控制权重设置为0.7生成的。
的。

2.4.4 Lineart

D. Line Art Coarse 线条更粗重的写实风格



上方的图像是使用控制权重设置为0.7生成的。

2.4.5 SoftEdge（柔和边缘）

SoftEdge 在之前的版本中，被称为 Hed（Holistically-Nested Edge Detection），全称是整体边缘线条检测，在最新的版本里，被一个更好理解的名字 Soft Edge（柔和边缘）取代了。

SoftEdge可以在物体周围创建清晰和精细的边界，输出类似于 Canny，它的有效性在于能够捕捉复杂的细节和轮廓，同时保留细节特征(面部表情、头发、手指等)。SoftEdge 在建立边缘信息时，线条会更为模糊，这种模糊会让 ControlNet 稍微放松一点。SoftEdge可保留更多柔和的边缘细节，类似手绘效果。

SoftEdge的意义：过度的控制会引发一种强制性的感觉，然而，有时候给予AI更多的自由度可能会带来更好的创作结果。

2.4.5 SoftEdge (柔和边缘)

在你用Canny觉得太过于受到约束了的时候，可以切换到soft edge。



Canny预处理

SoftEdge 预处理



Canny生成效果

SoftEdge 生成效果

你可以很直观的看到上面这组图片的对比，Canny 会更关注人物内部的细节。但 SoftEdge 这一类处理只会把大的轮廓保留下来，内部其实是有更大的发挥空间。另外背景里有一些对比不那么强烈的大线条，Canny 检测不到就不会控制，但 SoftEdge 就能做到。

2.4.5 SoftEdge (柔和边缘)

SoftEdge 预处理器

Softedge_hed

Softedge_hedsafe

Softedge_pidinet

Softedge_pidsafe

SoftEdge 的预处理器有四个，但归根结底是两个hed，两个pidinet。两个hed 的相对质量要比pidinet高。两个带safe的是他们的精简版，也可以防止生成的图像带有不良内容。

四个预处理器按结果质量排序：

SoftEdge_HED > SoftEdge_PIDI > SoftEdge_HED_safe > SoftEdge_PIDI_safe

一般来说，越精细的 ControlNet 需要的时间和显存也越多。如果你的显卡有负担，可以尝试使用更低级一点的处理器。这一原则在任何一个控制模型上都适用。

2.4.6 Scribble（涂鸦乱画）

Scribble指的是涂鸦，使用方法是通过 ControlNet 的 Scribble 模型提取涂鸦图（可提取参考图涂鸦，或者手绘涂鸦图），再根据提示词和风格模型对图像进行着色和风格化。

Scribble 比 Canny、SoftEdge 和 Lineart 的自由发挥度要更高，也可以用于对手绘稿进行着色和风格处理。

Scribble 的预处理器有三种模式：

Scribble HED

Holistically-Nested Edge Detection (HED) 是一种边缘检测器，能够产生类似于真实人类轮廓的轮廓线条。根据ControlNet的作者，HED适用于图像重新上色和重新风格化。

Scribble Pidinet

Pixel Difference network (Pidinet) 用于检测曲线和直线边缘。其结果类似于HED，但通常会产生更清晰、细节更少的线条。

Scribble xdog

EXtended Difference of Gaussian (XDoG) 是一种边缘检测方法。重要的是要调整xDoG的阈值并观察预处理器的输出

2.4.6 Scribble (涂鸦乱画)

A. Scribble HED

涂鸦的目的是从简单的黑白线条画和草图生成图像。用户也可以使用“Canvas”选项创建特定大小的空白画布，用于手动素描（也可以直接上传图像）。如果草图和绘图由白色背景上的黑线组成，则需要选中“Invert Input Color”复选框。用于这个预处理器的最佳模型是 control_sd15_openpose。



2.4.6 Scribble (涂鸦乱画)

B. Scribble Pidinnet

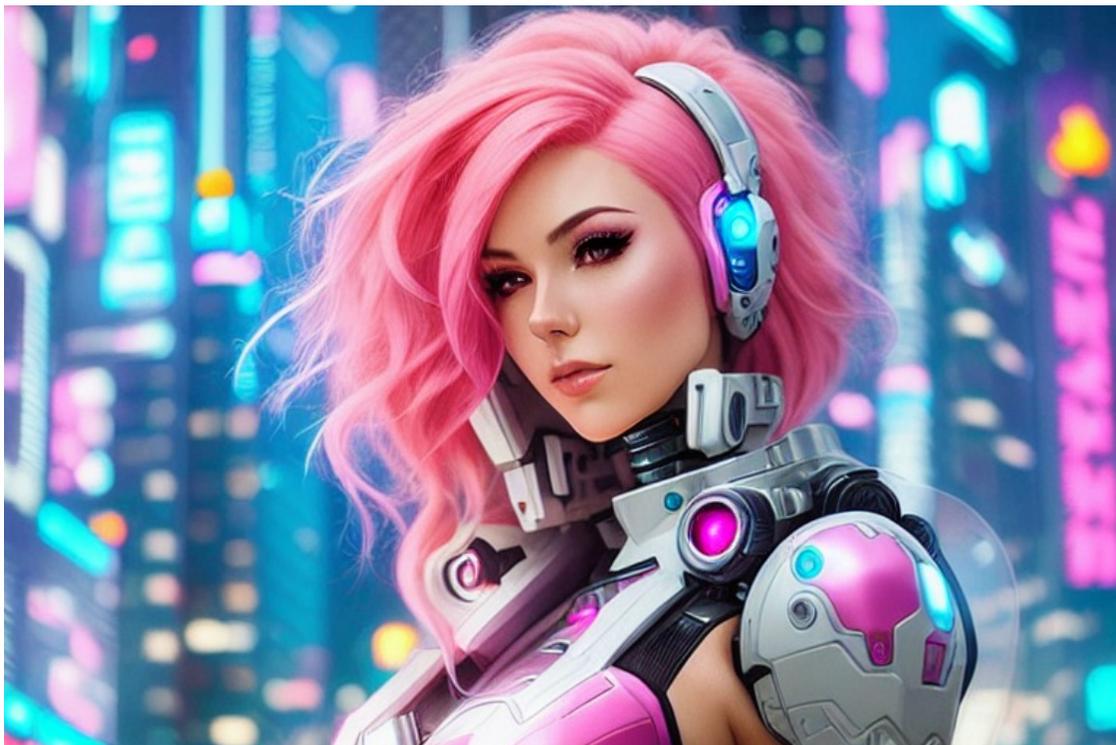
Pidinet倾向于产生细节较少的粗糙线条。它适用于复制板块的轮廓而无需精细细节。



2.4.6 Scribble (涂鸦乱画)

C. Scribble xDoG

通过调整XDoG的阈值，可以控制轮廓的细节，保留大致结构，再进行着色和风格化。xDoG成为创建涂鸦的多功能预处理器。



2.4.6 Scribble（涂鸦乱画）

Example

① 切换预处理器和模型



② 点击“创建空白画布”。



2.4.6 Scribble (涂鸦乱画)

Example

③绘制线稿，这里绘制了一盏煤油灯。



2.4.6 Scribble (涂鸦乱画)

Example

④ 输入关键词:

old electronic kerosene lamp in anthracite blue metal, warm orange metal reflections, Intricate, Highly detailed, Warm lighting, Sharp focus, Digital painting, Artstation, Concept art, trending, inspired by art by Zdenek Burian and frederick catherwood

旧电子煤油灯在无烟煤蓝色金属，温暖的橙色金属反射，错综复杂，高度详细，温暖的照明，锐利的焦点，数字绘画，艺术站，概念艺术，趋势，灵感来自艺术 zdenek burian 和 frederick catherwood

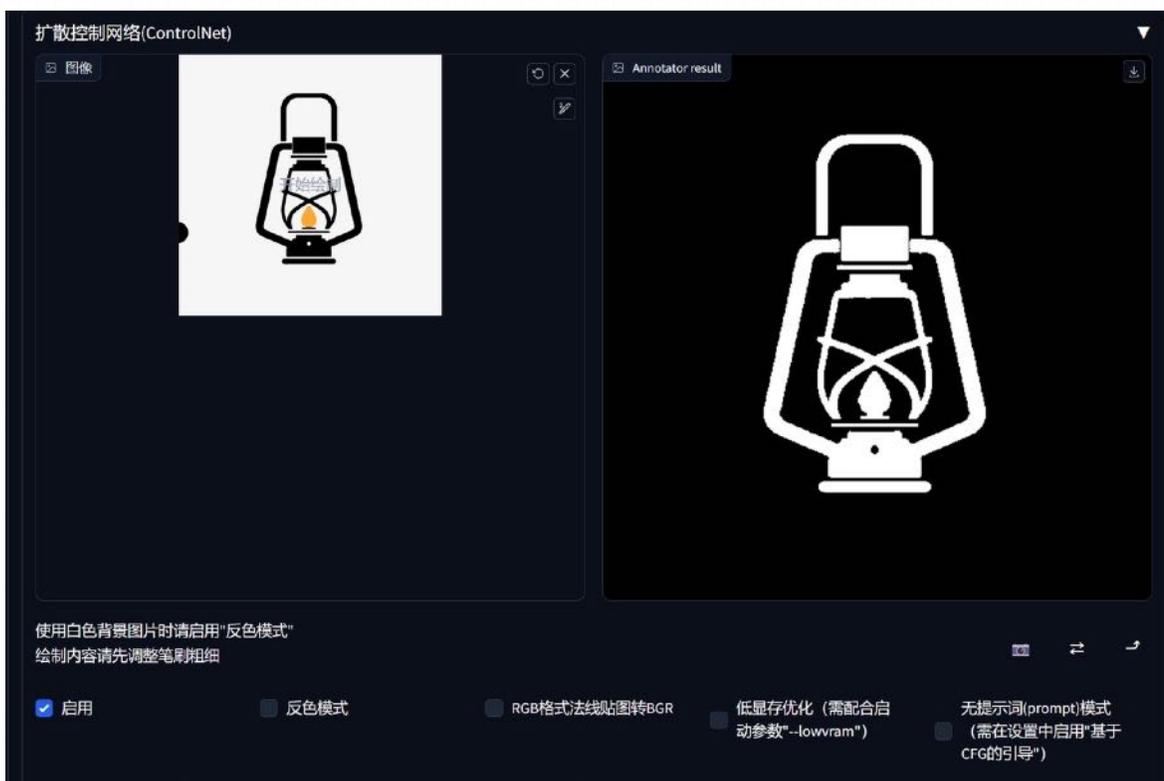


这个模型是根据你绘画的线稿精准控制物体的构图、构造。细节根据关键词来控制。

2.4.6 Scribble (涂鸦乱画)

我们也可以通过其他软件绘制之后上传线稿生成图像

1) 上传线稿图, 点击“预览预处理器结果”



2.4.6 Scribble (涂鸦乱画)

我们也可以通过其他软件绘制之后上传线稿生成图像

2) 输入上面的关键词, 然后生成图片



2.4.7 Normal map

法线图使用了三种主要颜色(红、绿、蓝)，通过不同的角度来精确定位物体的粗糙度和光滑程度。使用 Normal map 生成法线图可以保留相当多的细节，但可能会产生意想不到的结果，因为法线图完全来自图像，而不是在 3D 建模软件中构建的。

法线图有利于突出复杂的细节和轮廓，并且在定位对象方面也很有效，特别是在接近度和距离方面，设置一个更高的阈值可以移除背景的远处部分，降低阈值将命令 AI 保留甚至显示额外的背景元素。用于此预处理器的最佳模型是 control_sd15_normal。

法线贴图指定曲面的方向。对于 ControlNet，它是一个图像，用于指定每个像素所在的表面的方向。图像像素表示表面朝向的方向，而不是颜色值。法线贴图的使用与深度贴图类似。它们用于传输参考图像的 3D 构图。

从原图中提取 3D 物体的法线向量，绘制的新图与原图的光影效果完全相同。

法线贴图预处理器：

Normal Midas

从Midas深度图估计法线贴图

Normal Bae

使用Bae等人提出的法线不确定性方法估计法线贴图

2.4.7 Normal map

A. Normal Midas

与Midas深度图一样，Midas法线贴图也适用于将主题与背景分离。



2.4.7 Normal map

B. Normal Bae

Bae法线贴图倾向于在背景和前景中都渲染细节



2.4.7 Normal map

Example

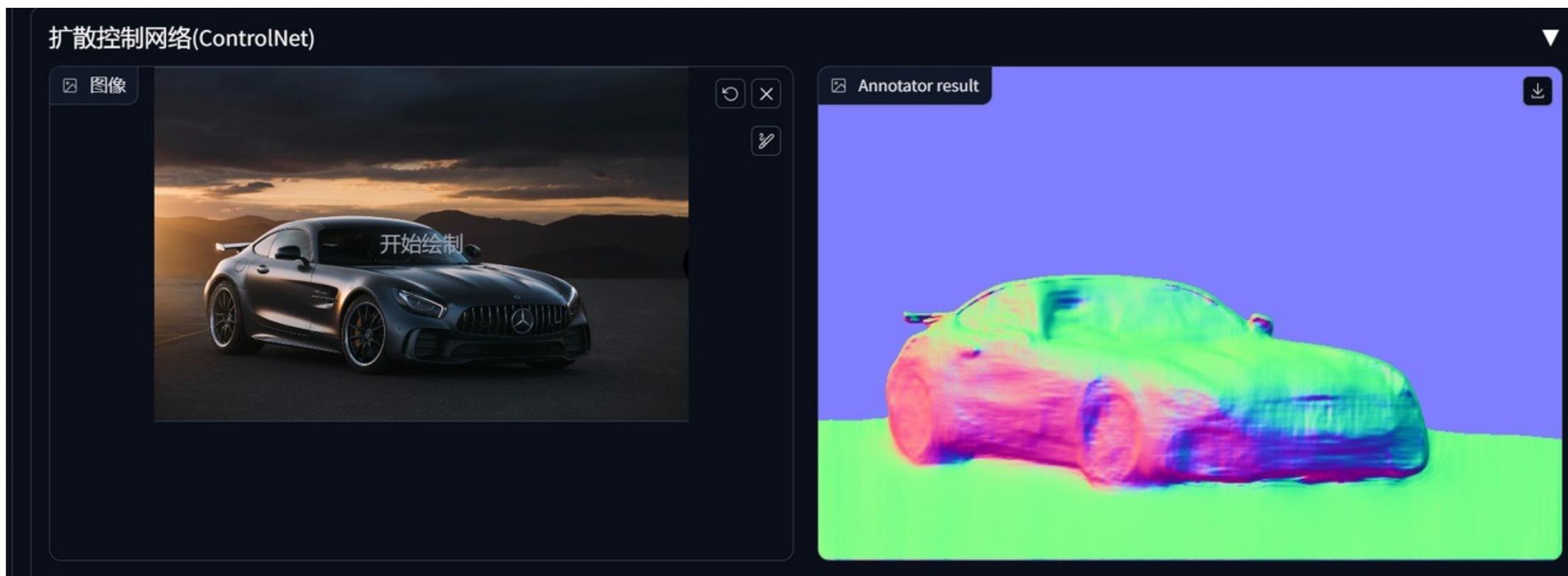
① 切换预处理器和模型



2.4.7 Normal map

Example

② 上传一张图片，然后点击“预览预处理结果”，然后就会给你生成一张法线图



2.4.7 Normal map

Example

③ 输入关键词

Mercedes-Benz AMG GT, staring out of the window with regal air, Moody lighting, High detail, realistic fur texture, Digital painting, trending on artstation

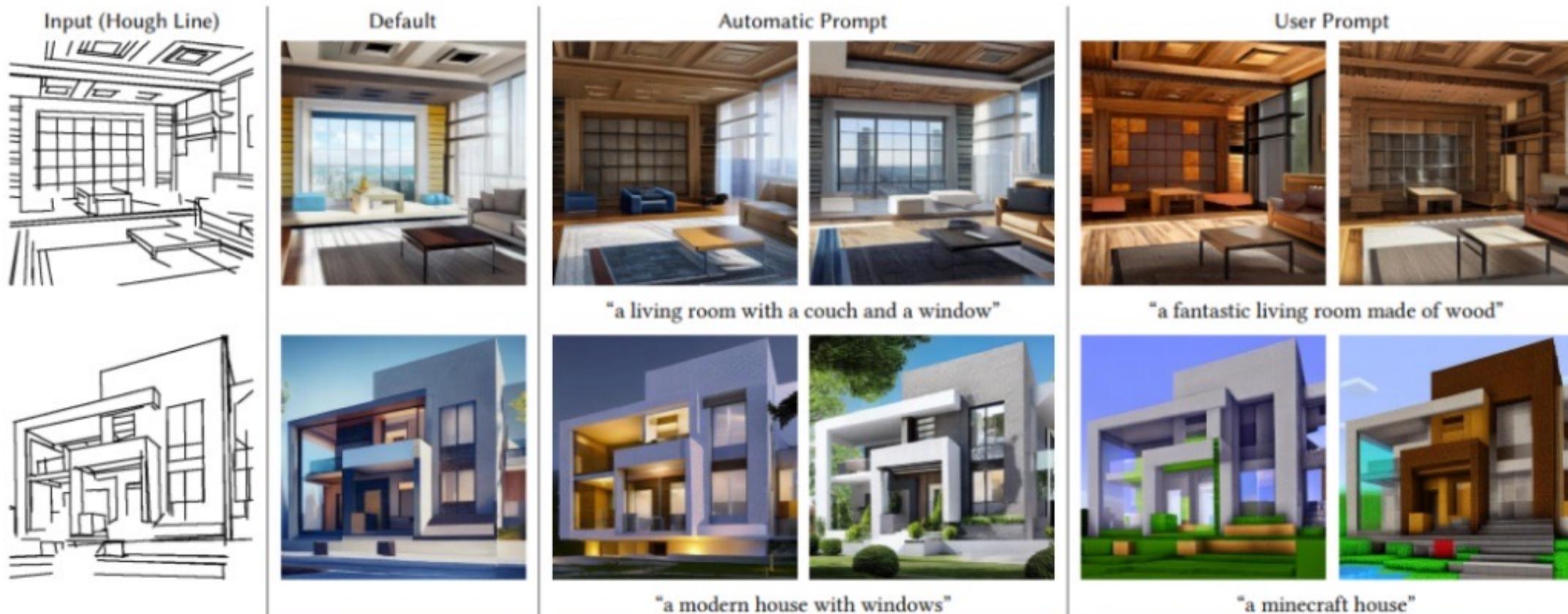
梅赛德斯-奔驰 AMG GT, 凝视窗外的皇家空气, 穆迪照明, 高细节, 逼真的皮毛纹理, 数字绘画, 艺术站的趋势



这个模型主要是通过 RGB 颜色通道来标记表示凹凸, 生成的图片立体感很强。

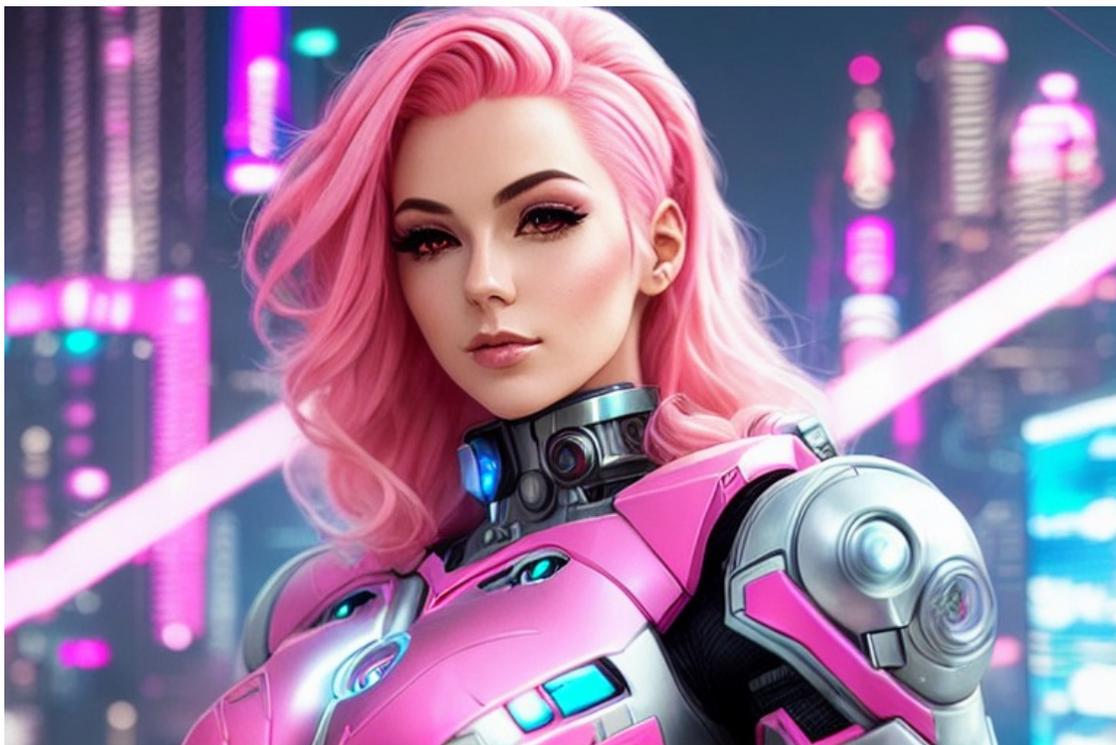
2.4.8 MLSD

MLSD Preprocessor可以进行直线检测，这些线条能够检测出需要独特和刚性轮廓的建筑和其他人造作品。但是它不适用于处理非刚性或弯曲的物体。MLSD 适用于生成室内布局或建筑结构，因为它可以突出直线和边缘。用于此预处理器的最佳模型是 control_sd15_mlsl。



2.4.8 MLSD

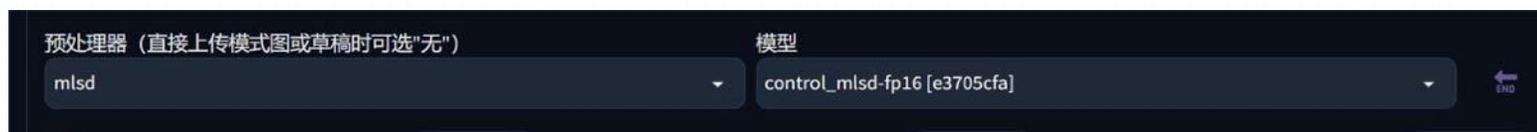
它可用于提取具有直边的轮廓，如室内设计、建筑物、街景、相框和纸边。但画面中的曲线将被忽略。



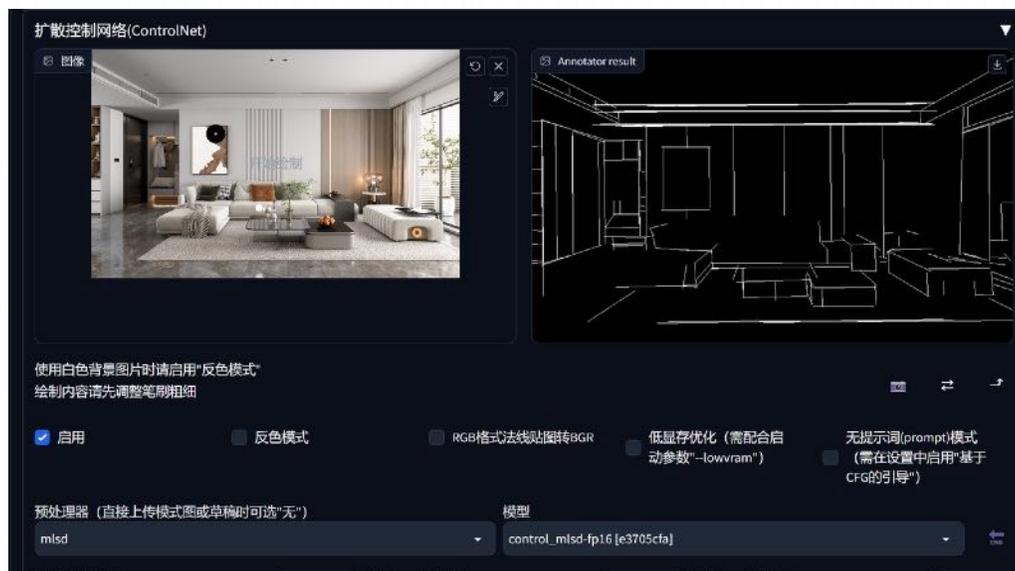
2.4.8 MLSD

Example

① 切换预处理器和相应模型



② 上传一张室内设计的图片，然后点击“预处理结果”



2.4.8 MLSD

Example

③输入关键词

pearl white walls, windsor gray floors, bedroom, Natural lighting, earthly tones, simple and clean, trending on interior design, digital artwork by Audrey kawasaki, Smooth, Detailed, Artstation, Neutral colors, elegant

珍珠白的墙壁, 温莎灰色的地板, 卧室, 自然光, 朴实的色调, 简单干净, 室内设计的趋势, 由 Audrey kawasaki 设计的数字艺术品, 光滑, 细节, 艺术站, 中性色, 优雅

④ 调整参数



2.4.8 MLSD

Example

5. 生成图片



2.4.9 Segmentation

分割预处理器检测会将上传的图像分割为同一图像内的段或区域。该模型在生成一组新的图像时，将 detectmap 图像应用于文本提示。用于此预处理器的最佳模型是 control_sd15_seg。

建筑物、天空、树木、人物和人行道使用不同且预定义的颜色进行标记。



<https://docs.qq.com/sheet/DYmtkWG5taWxhVlx2?tab=BB08J2>

2.4.9 Segmentation

分割预处理器用于标记参考图像中的对象类型。Segmentation 语义分割模型，标注画面中的不同区块颜色和结构（不同颜色代表不同类型对象），从而控制画面的构图和内容。

Seg 预处理器：

ufade20k 基于ADE20K数据集训练的UniFormer (uf) 分割

ofade20k 基于ADE20k数据集训练的OneFormer (of) 分割

ofcoco 基于COCO数据集训练的OnFormer分割

Idx	Ratio	Train	Val	Stuff	Color_Code (R,G,B)	Color_Code(hex)	Color	Name
1	0.1576	11664	1172	1	(120, 120, 120)	#787878	墙 wall	
2	0.1072	6046	612	1	(180, 120, 120)	#B47878	建筑 building;edifice	
3	0.0878	8265	796	1	(8, 230, 230)	#06E6E6	天空 sky	
4	0.0621	9336	917	1	(80, 50, 50)	#503232	地板 floor;flooring	
5	0.048	6678	641	0	(4, 200, 3)	#04C803	树 tree	
6	0.045	6604	643	1	(120, 120, 80)	#787850	天花板 ceiling	
7	0.0398	4023	408	1	(140, 140, 140)	#8C8C8C	路 road;route	
8	0.0231	1906	199	0	(204, 5, 255)	#CC05FF	床 bed	
9	0.0198	4688	460	0	(230, 230, 230)	#E6E6E6	窗 windowpane>window	
10	0.0183	2423	225	1	(4, 250, 7)	#04FA07	草 grass	
11	0.0181	2874	294	0	(224, 5, 255)	#E005FF	柜子 cabinet	
12	0.0166	3068	310	1	(235, 255, 7)	#EBFF07	人行道 sidewalk;pavement	
13	0.016	5075	526	0	(150, 5, 61)	#96053D	人 person;individual;someone;somebody;mortal;soul	
14	0.0151	1804	190	1	(120, 120, 70)	#787846	地面 earth;ground	
15	0.0118	6666	796	0	(8, 255, 51)	#08FF33	门 door;double;door	
16	0.011	4269	411	0	(255, 6, 82)	#FF0652	桌子 table	
17	0.0109	1691	160	1	(143, 255, 140)	#8FFF8C	山 mountain;mount	
18	0.0104	3999	441	0	(204, 255, 4)	#CCFF04	植物 plant;flora;plant;life	
19	0.0104	2149	217	0	(255, 51, 7)	#FF3307	布料 curtain;drape;drapery;mantle;pall	
20	0.0103	3261	318	0	(204, 70, 3)	#CC4603	椅子 chair	
21	0.0098	3164	306	0	(0, 102, 200)	#0066C8	车 car;auto;automobile;machine;motorcar	
22	0.0074	709	75	1	(61, 230, 250)	#3DE6FA	水 water	
23	0.0067	3296	315	0	(255, 6, 51)	#FF0633	画/照片 painting;picture	
24	0.0065	1191	106	0	(11, 102, 255)	#0B66FF	沙发 sofa;couch;lounge	
25	0.0061	1516	162	0	(255, 7, 71)	#FF0747	架子 shelf	
26	0.006	667	69	1	(255, 9, 224)	#FF09E0	房子 house	
27	0.0053	651	57	1	(9, 7, 230)	#0907E6	海 sea	
28	0.0052	1847	224	0	(220, 220, 220)	#DCDCDC	镜子 mirror	
29	0.0046	1158	128	1	(255, 9, 92)	#FF095C	地毯 rug;carpet;carpeting	
30	0.0044	480	44	1	(112, 9, 255)	#7009FF	田地 field	
31	0.0044	1172	98	0	(8, 255, 214)	#08FFD6	扶手椅 armchair	
32	0.0044	1292	184	0	(7, 255, 224)	#07FFE0	座位 seat	
33	0.0033	1386	138	0	(255, 184, 6)	#FFB806	围栏 fence;fencing	
34	0.0031	698	61	0	(10, 255, 71)	#0AFF47	桌子 desk	
35	0.003	781	73	0	(255, 41, 10)	#FF290A	岩石 rock;stone	
36	0.0027	380	43	0	(7, 255, 255)	#07FFFF	橱柜 wardrobe;closet;press	

ADE20k和COCO分割的颜色映射是不同的。我们可以使用分割预处理器来转移对象的位置和形状。

语义分割词表：

<https://docs.qq.com/sheet/DQVNHhWhvbEF6cGxo?tab=2gegib>

2.4.9 Segmentation

A. UniFormer ADE20k (ufade20k)

UniFormer在此示例中准确地对所有对象进行了标记。

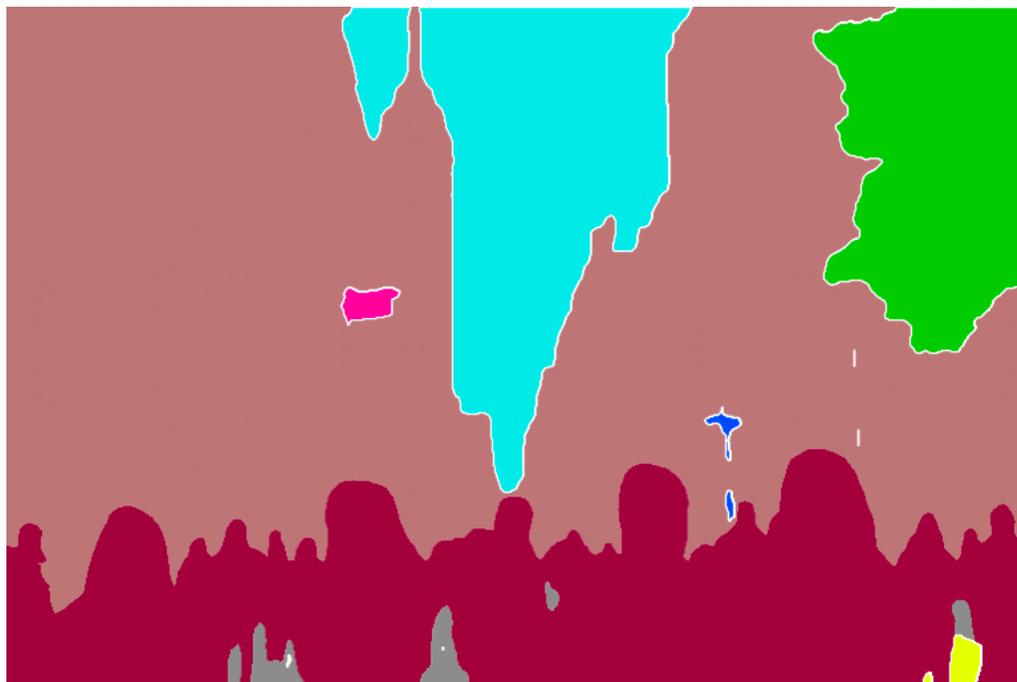
以下是在相同提示和种子下使用这些预处理器的示例。未来城市、树木、建筑物、赛博朋克



2.4.9 Segmentation

B. OneFormer ADE20k (ofade20k)

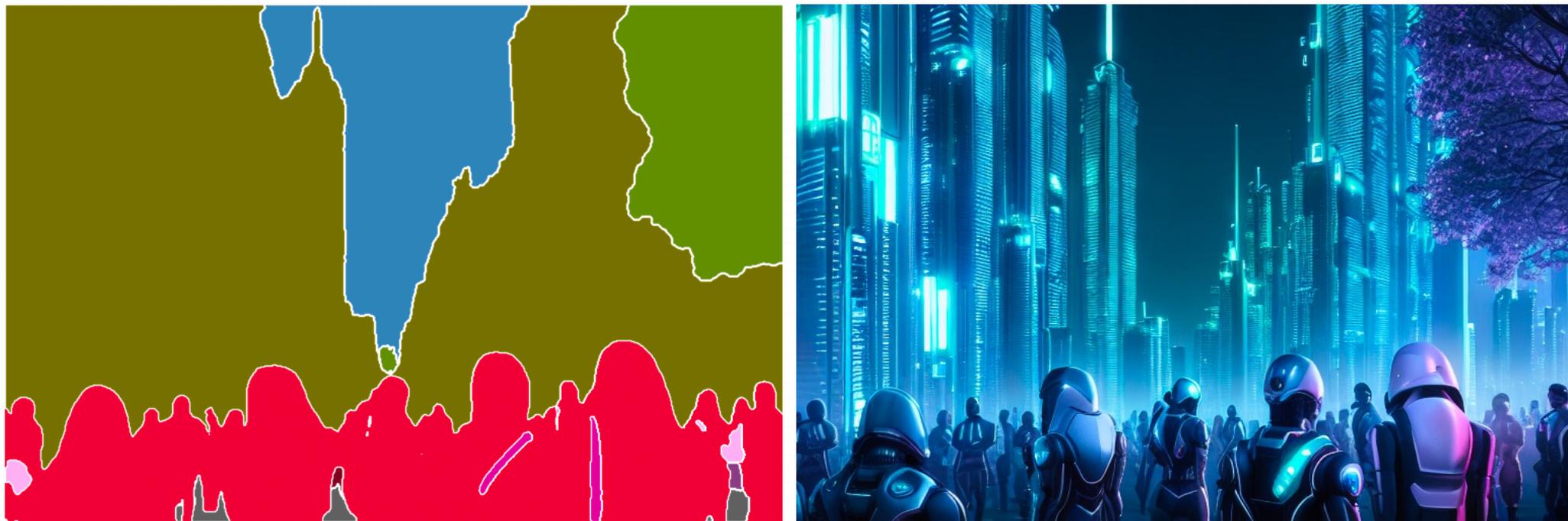
在这种情况下，OneFormer有些噪点，但不会影响最终图像



2.4.9 Segmentation

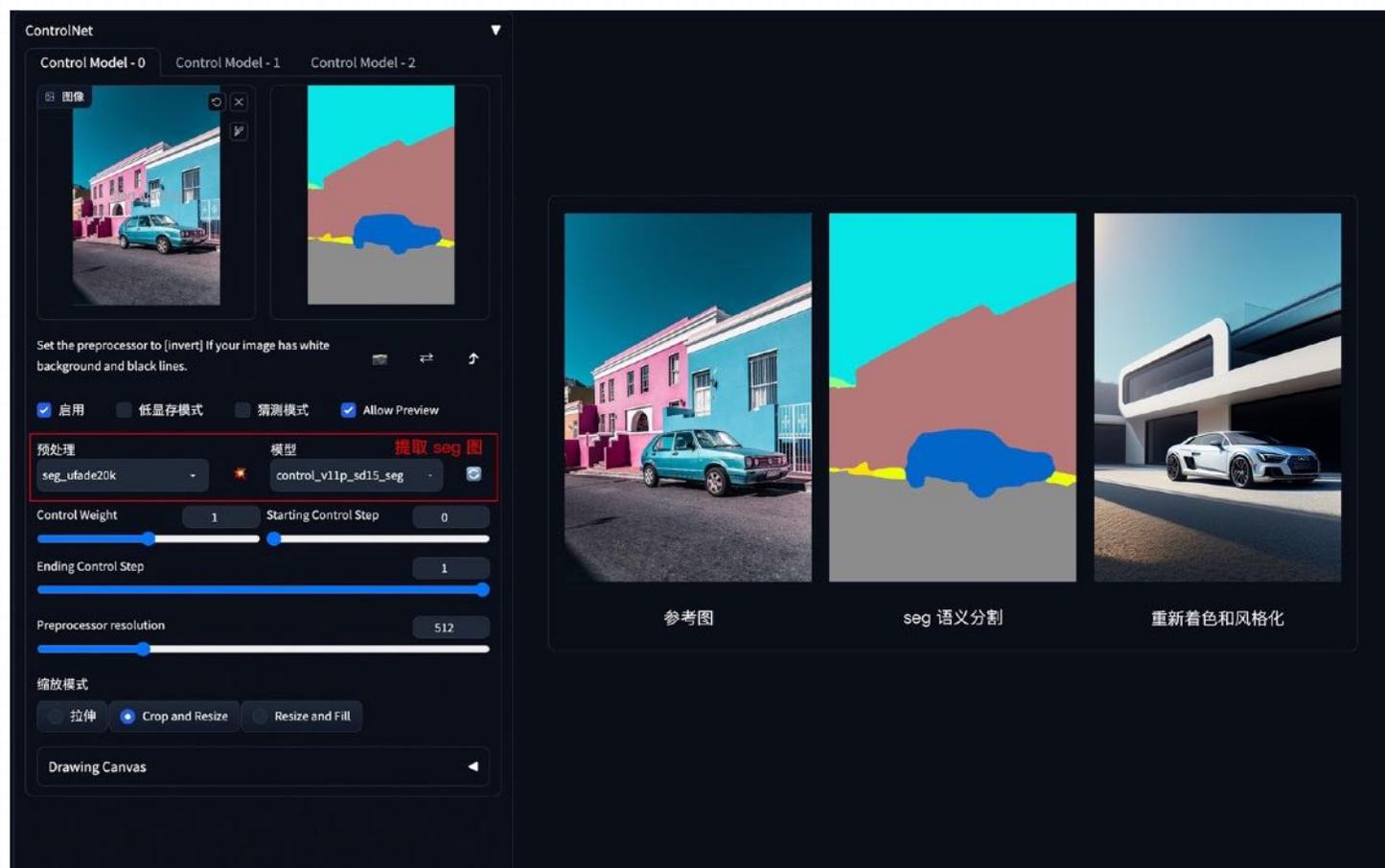
C. OneFormer COCO (ofcoco)

OneFormer COCO表现类似，但存在一些错误标记。



2.4.9 Segmentation

Seg 示例：（提取参考图内容和结构，再进行着色和风格化）



2.4.9 Segmentation

Seg 示例：如果还想在车前面加一个人，只需在 Seg 预处理图上对应人物色值，添加人物色块再生成图像即可。

ControlNet

Control Model - 0 Control Model - 1 Control Model - 2

图像

Start drawing 添加人物 seg 图

Set the preprocessor to [invert] if your image has white background and black lines.

启用 低显存模式 猜测模式 Allow Preview

预处理 模型

none control_v11p_sd15_seg [e1]

Control Weight 1 Starting Control Step 0

Ending Control Step 1

缩放模式

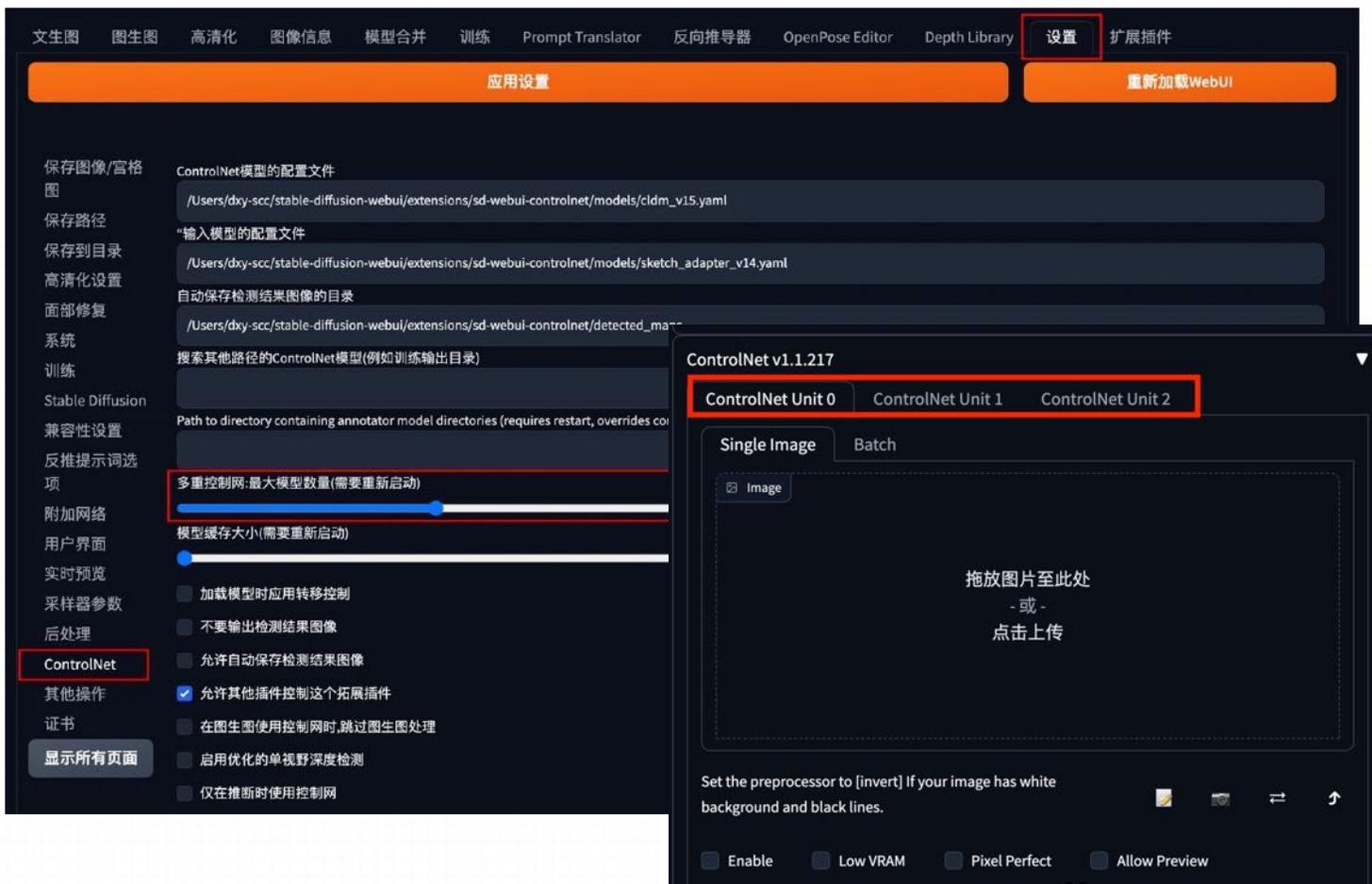
拉伸 Crop and Resize Resize and Fill

Drawing Canvas

seg 语义分割 重新着色和风格化

2.5 多重 ControlNet 应用

ControlNet 还支持多个模型的组合使用，从而对图像进行多条件控制。ControlNet 的多模型控制可以在设置面板中的 ControlNet 模块中开启：



1. 在设置选项卡中找到ControlNet
2. 修改多重控制网的数值，意味着我们可以选择几个模型搭配使用。

可以在设置选单里找到ControlNet的设置选项，在里面有一个多重control net的最大模型数量默认会维持一。但你可以将它调大，我们设置为三保存设置，然后通过右侧按钮重新选单。你的controlnet就变得稍微不一样了。原来我们只有一组controlnet选项，但现在上面冒出来了两个新的unit，可以让你切换过去配置完全不同的预处理器和模型组，也可以使用完全不一样的图片作为输入信息。

2.5 多重 ControlNet 应用

了解借助MultiControlNet（多重控制网），调用多个不同模型，对图像实施更严谨的控制与调节的方式，参透ControlNet的组合逻辑。

目前系统中14种不同的控制模型，他们仅仅两两组合就会有超过90种实际应用。但其实并不是随便两个凑一起就一定行了。这里面的组合关键就在于你需要让他们存在一定程度的互补，让一个网络帮另一个网络实现那些他做不到的事情。而如果两个控制网本来的性质就相似，比如Canny 加hat 对边缘的检测是高度重复的，那这种组合明显不具备特别大的实用价值，和只开一个没什么区别。

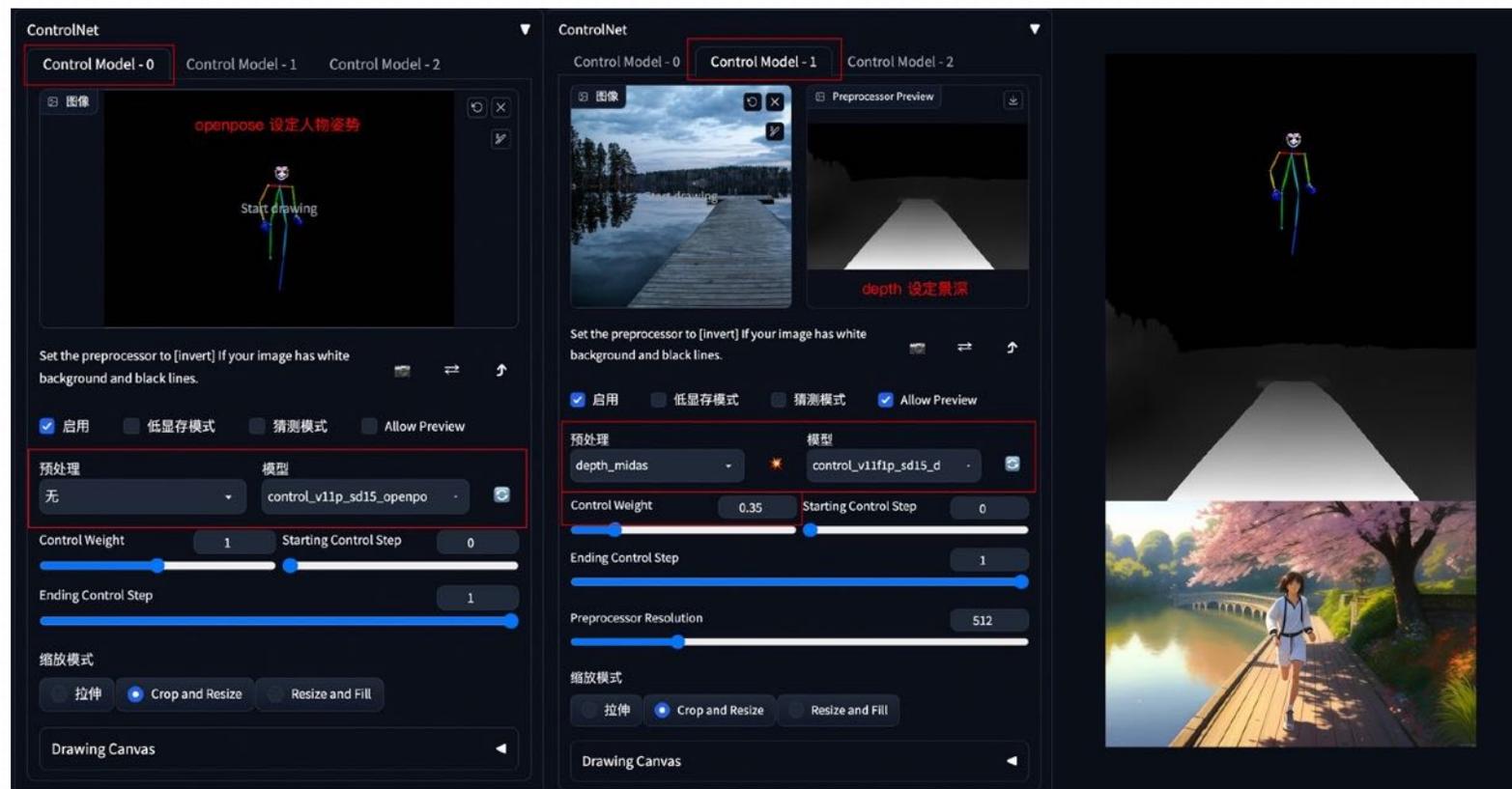
注意：多重控制网对显存的要求会更高。当你撒了两张，三张网下去，显存就是两倍，三倍往上翻的。如果你的显存顶不住，一定要选用稍微低级一点的预处理，或者将低显存模式打开。

2.5 多重 ControlNet 应用

A. 人物和背景分别控制

方法：设置 2 个 ControlNet，第一个 ControlNet 通过 OpenPose 控制人物姿态，第二个 ControlNet 通过 Seg 或 Depth 控制背景构成。调整 ControlNet 权重，如 OpenPose 权重高于 Depth 权重，以确保人物姿态被正确识别，再通过提示词和风格模型进行内容和风格控制。

应用模型：OpenPose、Seg（自定义背景内容和结构）、Depth

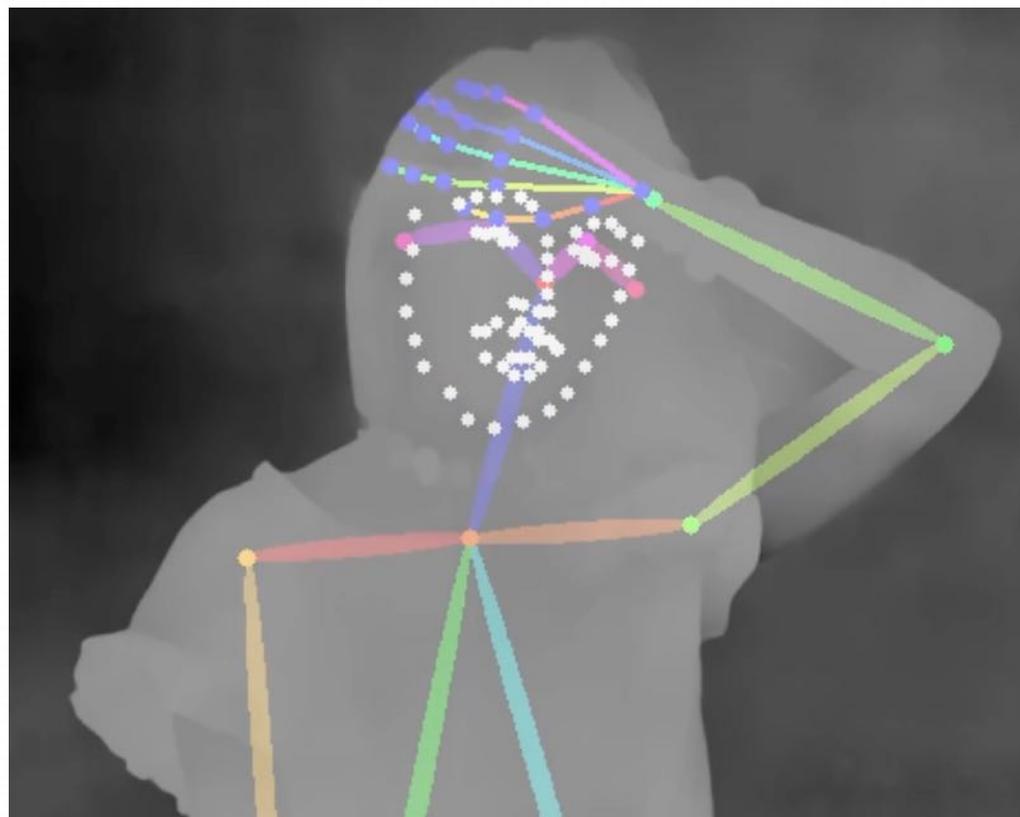


2.5 多重 ControlNet 应用

A. 人物和背景分别控制

方法： open pose可以识别手，甚至精确到手指，但因为不带有空间信息，所以可能无法判断手到底是在前在后。但depth是可以识别出来的，如果手在头前面，他就一定会有这样的深度差异表现。而如果手在脑袋后面，他就识别不到类似的信息。所以我们让open pose组织手的基本走向，然后再让depth把这个空间关系梳理好，就可以识别到完美的图像信息

应用模型： OpenPose、Depth

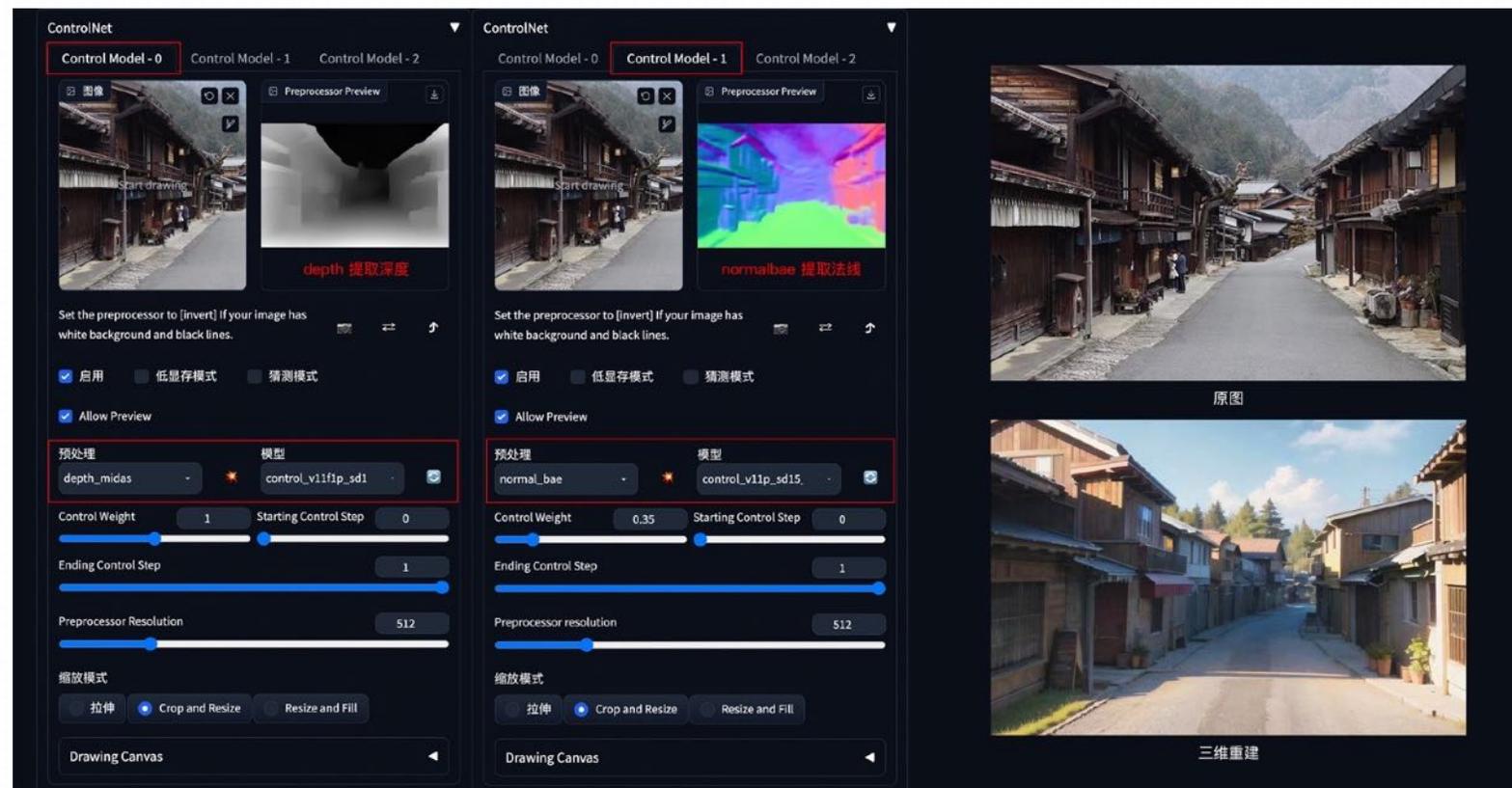


2.5 多重 ControlNet 应用

B. 三维重建

方法：通过 Depth 深度检测和 Normalbae 法线贴图模型，识别三维目标。再配合提示词和风格模型，重新构建出三维物体和场景。

应用模型：Depth、Normalbae

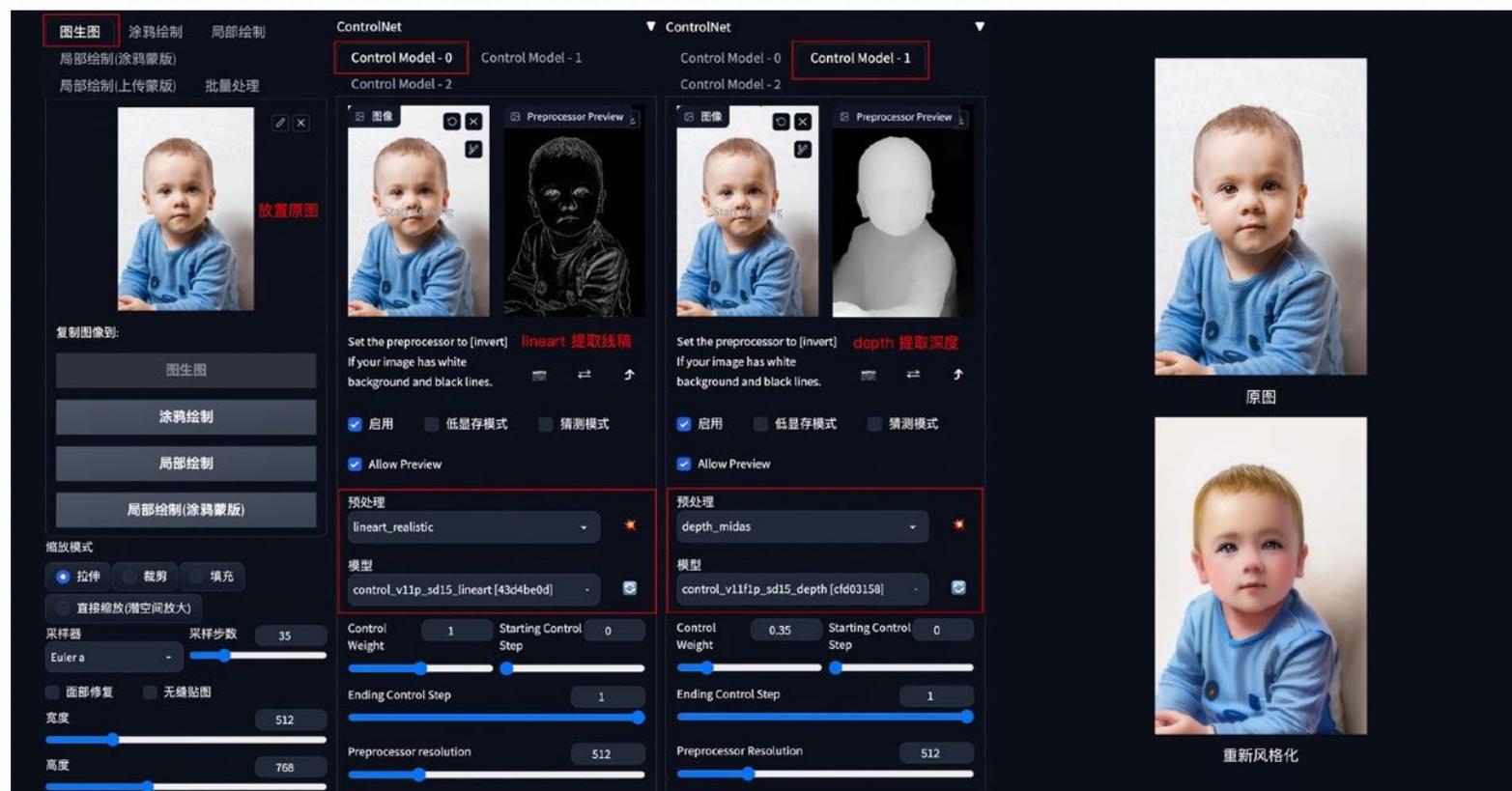


2.5 多重 ControlNet 应用

C. 更精准的图片风格化

方法：在 img2img 图生图中，通过叠加 Lineart 和 Depth 模型，可以更加精准的提取图像结构，最大程度保留原图细节，再配合提示词和风格模型重新生成图像。

应用模型：Lineart、Depth

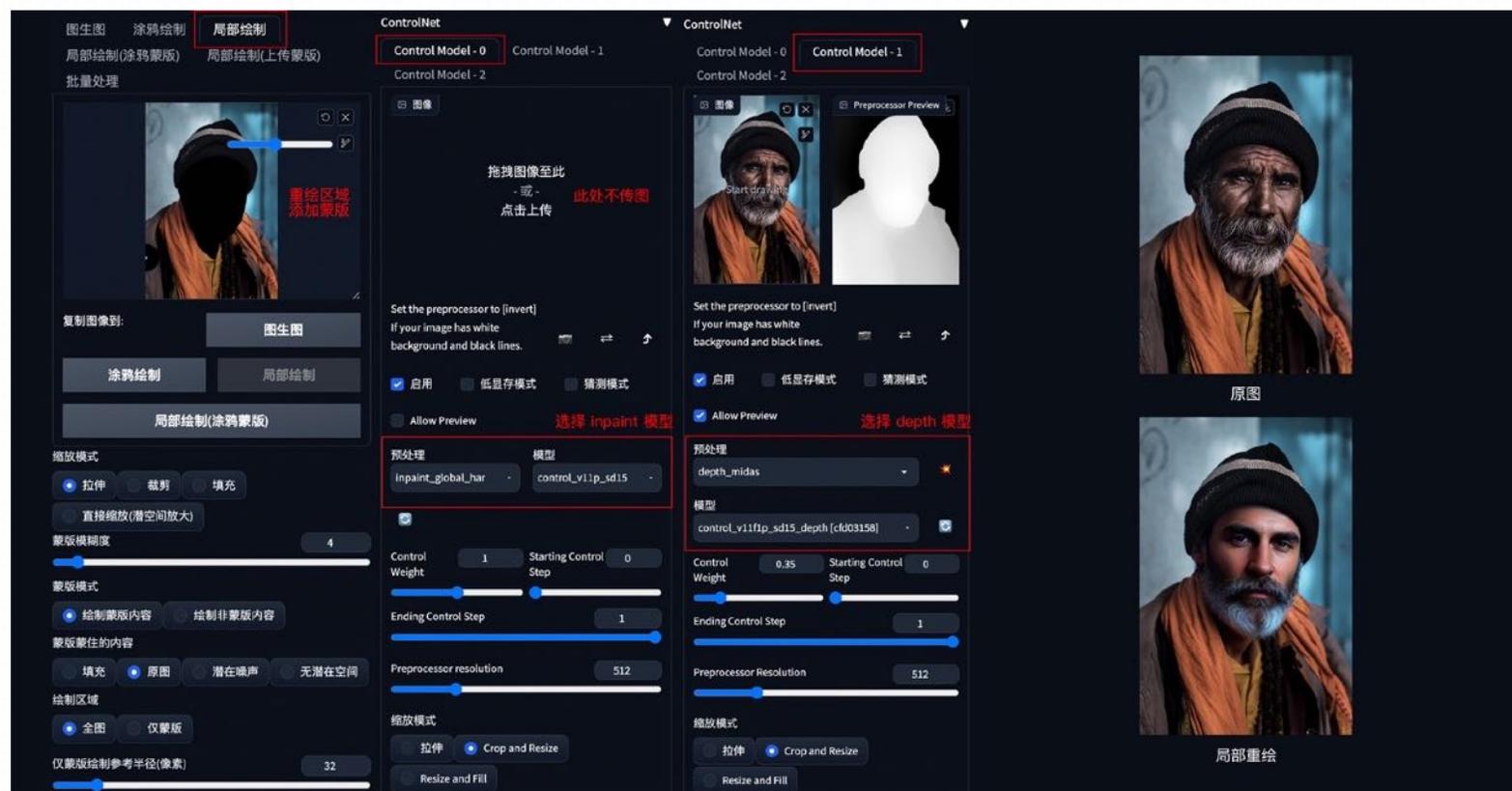


2.5 多重 ControlNet 应用

D. 更精准的图片局部重绘

方法：在 img2img 图生图的局部重绘中，通过叠加 Canny 和 Inpaint 模型，可以更加精准的对图像进行局部重绘。

应用模型：Canny、Inpaint





智能大数据可视化实验室

INTELLIGENT BIG DATA VISUALIZATION LAB