

SD新手：入门图文教程

声明：

本文转发自：[腾讯技术工程：Stable Diffusion 新手入门手册](#)，在原文上增加目录，作者：[HkingAuditore](#)

Stable Diffusion 介绍：2022 年发布的深度学习文字到图像生成模型。它主要用于根据文字的描述产生详细图像，能够在几秒钟内创作出令人惊叹的艺术作品，本文是一篇使用入门教程。

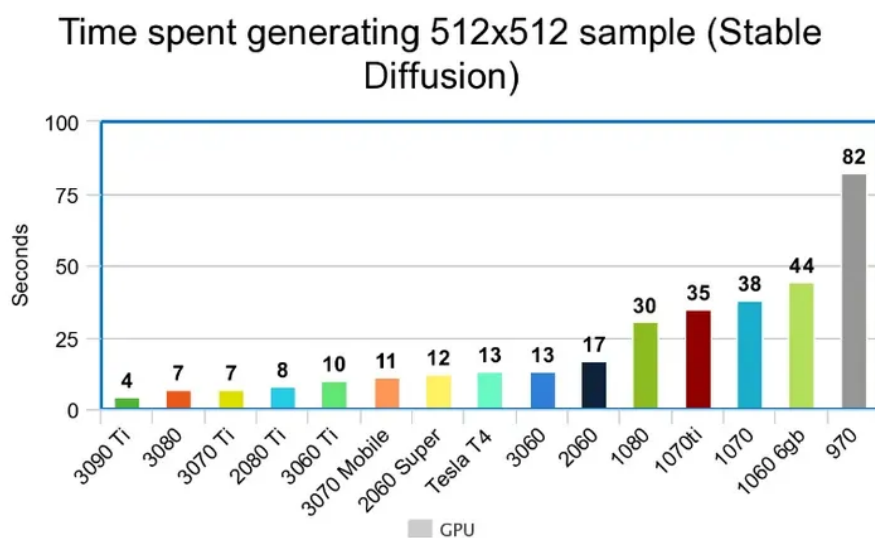
一、硬件要求

建议使用不少于 16 GB 内存，并有 60GB 以上的硬盘空间。需要用到 CUDA 架构，推荐使用 N 卡。（目前已经有了对 A 卡的相关支持，但运算的速度依旧明显慢于 N 卡，参见：

[Install and Run on AMD GPUs · AUTOMATIC1111/stable-diffusion-webui Wiki](#) · [GitHub](#)

过度使用，显卡会有损坏的风险。

进行 512x 图片生成时主流显卡速度对比：



二、环境部署

手动部署

可以参考 webui 的官方 wiki 部署：[Home · AUTOMATIC1111/stable-diffusion-webui Wiki \(github.com\)](#)

stable diffusion webui 的完整环境占用空间极大，能达到几十 G。同时，webui 需要联网下载安装大量的依赖，在境内的网络环境下下载很慢，请自带科学上网工具。

1. 安装 Python 安装 [Python 3.10](#)，安装时须选中 `Add Python to PATH`
2. 安装 Git 在 [Git-scm.com](#) 下载 Git 安装包并安装。
3. 下载 webui 的 github 仓库 按下 `win+r` 输入 cmd，调出命令行窗口。运行：
`cd PATH_TO_CLONE`
`git clone https://github.com/AUTOMATIC1111/stable-diffusion-webui.git`

请把代码中的
`PATH_TO_CLONE`

替换为自己想下载的目录。

4. 装配模型 可在如 [Civitai](#) 上下载标注有 CKPT 的模型，有模型才能作画。下载的模型放入下载后文件路径下的 `models/Stable-diffusion` 目录。
5. 使用 双击运行 `webui-user.bat`。脚本会自动下载依赖，等待一段时间（可能很长），程序会输出一个类似 `http://127.0.0.1:7860/` 的地址，在浏览器中输入这个链接即可。详细可参见 [模型使用](#)。
6. 更新 按下 `win+r` 输入 cmd，调出命令行窗口。运行：
`cd PATH_TO_CLONE`
`git pull`

请把代码中的：
`PATH_TO_CLONE`

替换为自己下载仓库的目录。

整合包

觉得麻烦的同学可以使用整合包，解压即用。比如 [独立研究员](#) 的空间下经常更新整合包。 [秋叶的启动器](#) 也非常好用，将启动器复制到下载仓库的目录下即可，更新管理会更方便。

.git	2023/3/28 10:07	文件夹	
.github	2023/1/17 11:38	文件夹	
.launcher	2023/2/24 15:26	文件夹	
__pycache__	2023/3/13 12:48	文件夹	
configs	2023/2/4 23:20	文件夹	
detected_maps	2023/3/12 12:14	文件夹	
embeddings	2023/2/27 16:29	文件夹	
extensions	2023/3/28 14:36	文件夹	
extensions-builtin	2023/1/23 20:06	文件夹	
git	2022/12/21 10:07	文件夹	
html	2023/3/12 12:10	文件夹	
javascript	2023/3/13 12:46	文件夹	
localizations	2022/12/21 9:09	文件夹	
log	2022/12/21 9:25	文件夹	
models	2023/3/8 15:04	文件夹	
modules	2023/3/13 12:46	文件夹	
outputs	2023/3/27 17:36	文件夹	
python	2022/12/21 9:01	文件夹	
repositories	2022/12/21 9:08	文件夹	
scripts	2023/3/28 14:42	文件夹	
test	2023/3/13 12:46	文件夹	
textual_inversion_templates	2022/12/21 8:40	文件夹	
tmp	2023/3/28 14:36	文件夹	
.gitignore	2023/1/17 11:38	Git Ignore 源文件	1 KB
.pylintrc	2022/12/21 8:40	PYLINTRC 文件	1 KB
A启动器.exe	2023/1/20 14:26	应用程序	1,784 KB
cache.json	2023/3/27 17:36	JSON File	10 KB
CODEOWNERS	2022/12/21 8:40	文件	1 KB
config.json	2023/3/28 14:07	JSON File	9 KB
environment-wsl2.yaml	2022/12/21 8:40	Yaml 源文件	1 KB
launch.py	2023/3/13 12:46	Python 源文件	16 KB
LICENSE.txt	2023/1/17 11:38	文本文件	35 KB
params.txt	2023/3/28 15:44	文本文件	1 KB
README.md	2023/3/12 12:10	Markdown 源文件	11 KB
requirements.txt	2023/2/4 23:20	文本文件	1 KB
requirements_versions.txt	2023/3/13 12:46	文本文件	1 KB
screenshot.png	2023/1/17 11:38	PNG 文件	411 KB
script.js	2023/1/23 20:06	JavaScript 文件	3 KB
SDWebUIStartup.deps.json	2022/12/21 11:00	JSON File	3 KB
SDWebUIStartup.dll	2022/12/21 11:29	应用程序扩展	68 KB
SDWebUIStartup.pdb	2022/12/21 11:29	Program Debug Da...	25 KB
SDWebUIStartup.runtimeconfig.json	2022/12/21 11:00	JSON File	1 KB

打开启动器后，可一键启动：



如果有其他需求，可以在高级选项中调整配置。



显存优化根据显卡实际显存选择，不要超过当前显卡显存。不过并不是指定了显存优化量就一定不会超显存，在出图时如果启动了过多的优化项（如高清修复、人脸修复、过大模型）时，依然有超出显存导致出图失败的几率。

xFormers 能极大地改善了内存消耗和速度，建议开启。准备工作完毕后，点击一键启动即可。等待浏览器自动跳出，或是控制台弹出本地 URL 后说明启动成功

```
SD-WebUI API layer loaded
Loading weights [804177259] from G:\stable-diffusion-webui_0220\models\Stable-diffusion\ToyEmpire1130_30000.chkpt
creating model from config: G:\stable-diffusion-webui_0220\configs\vi-inference.yaml
LatentDiffusion: Running in eps-prediction mode
DiffusionWrapper has 859.52 M params.
Couldn't find VAE named vae-ft-mse-840000-ema-pruned.chkpt; using None instead
Applying xformers cross attention optimization.
Textual inversion embeddings loaded(0):
Model loaded in 4.5s (load weights from disk: 1.2s, create model: 0.4s, apply weights to model: 0.7s, apply half(): 0.7s, move model
to device: 0.6s, load textual inversion embeddings: 0.8s).
Running on local URL: http://127.0.0.1:7860

To create a public link, set 'share=True' in 'launch()'.
```

如果报错提示缺少 Pytorch，则需要在启动器中点击配置：

```
prepare_environment()
File "G:\AI\setup\stable-diffusion-webui_0227\launch.py", line 272, in prepare_environment
    run_python("import torch; assert torch.cuda.is_available(), 'Torch is not able to use GPU; add --skip-torch-cuda-test to
COMMANDLINE_ARGS variable to disable this check'")
File "G:\AI\setup\stable-diffusion-webui_0227\launch.py", line 129, in run_python
    return run(f'python' + " " + code)
File "G:\AI\setup\stable-diffusion-webui_0227\launch.py", line 105, in run
    raise RuntimeError(message)
RuntimeError: Error running command.
提示: Python 运行时抛出了一个异常。请检查疑难解答页面。
Command: "G:\AI\setup\stable-diffusion-webui_0227\python\python.exe" -c "import torch; assert torch.cuda.is_available(), 'Torch is
not able to use GPU; add --skip-torch-cuda-test to COMMANDLINE_ARGS variable to disable this check'"
Error code: 1
stdout: <empty>
stderr: G:\AI\setup\stable-diffusion-webui_0227\python\Lib\site-packages\torch\cuda\_init_.py:88: UserWarning: CUDA
Initialization: The NVIDIA driver on your system is too old (found version 10020). Please update your GPU driver by downloading and
installing a new version from the URL: http://www.nvidia.com/Download/index.aspx Alternatively, go to: https://pytorch.org to
install a PyTorch version that has been compiled with your version of the CUDA driver. (Triggered internally at ..\c10\cuda
\CUDAFunctions.cpp:189.)

    return torch._C._cuda_getDeviceCount() > 0

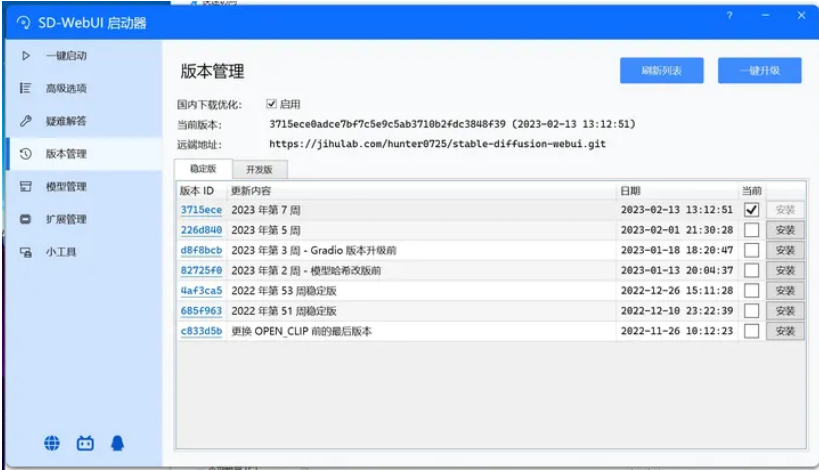
Traceback (most recent call last):
  File "<string>", line 1, in <module>
AssertionError: Torch is not able to use GPU; add --skip-torch-cuda-test to COMMANDLINE_ARGS variable to disable this check
提示: Python 运行时抛出了一个异常。请检查疑难解答页面。

[已退出进程，代码为 1 (0x00000001)]

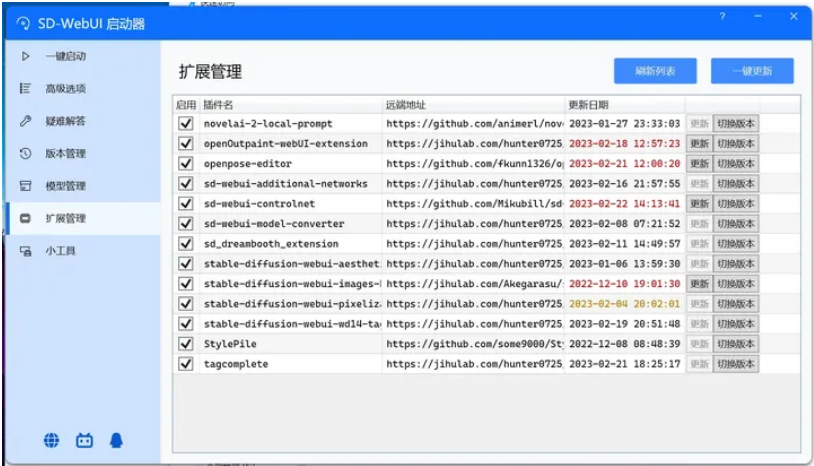
提示: 检测到 SD-WebUI 进程退出状态不正常，建议前往疑难解答页面扫描错误记录或寻求其他帮助。
```



Stable Diffusion webui 的更新比较频繁，请根据需求在“版本管理”目录下更新：

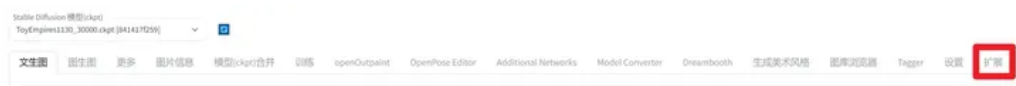


同样地，也请注意插件的更新：

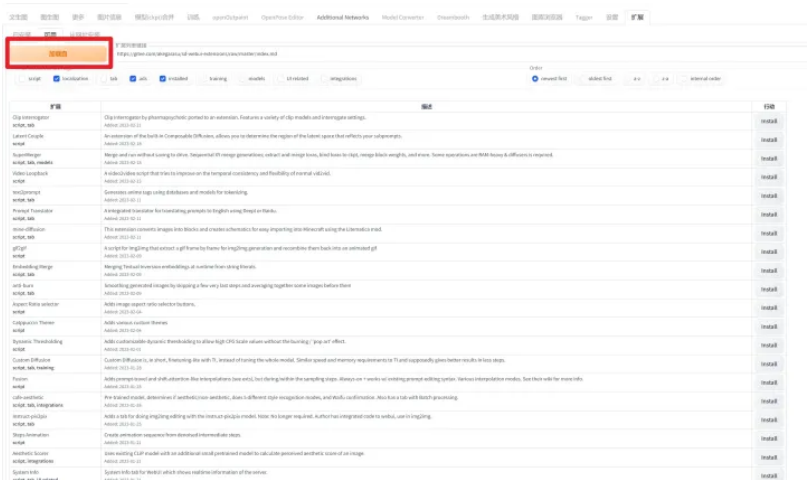


关于插件

Stable Diffusion 可配置大量插件扩展，在 webui 的“扩展”选项卡下，可以安装插件：



点击“加载自”后，目录会刷新，选择需要的插件点击右侧的 install 即可安装。



安装完毕后，需要重新启动用户界面：

文生图 图生图 更多 图片信息 模型(ckpt)合并 训练 openOutpaint OpenPose Editor

已安装 可用 从网址安装

应用并重启用户界面

检查更新

扩展	网址	更新
<input checked="" type="checkbox"/> StylePile	https://github.com/some9000/StylePile.git	未知
<input checked="" type="checkbox"/> novelai-2-local-prompt	https://github.com/animerl/novelai-2-local-prompt.git	未知
<input checked="" type="checkbox"/> openOutpaint-webUI-extension	https://jihulab.com/hunter0725/openOutpaint-webUI-extension.git	未知
<input checked="" type="checkbox"/> openpose-editor	https://github.com/fkunn1326/openpose-editor.git	未知
<input checked="" type="checkbox"/> sd-webui-additional-networks	https://jihulab.com/hunter0725/sd-webui-additional-networks.git	未知
<input checked="" type="checkbox"/> sd-webui-controlnet	https://github.com/Mikubill/sd-webui-controlnet	未知
<input checked="" type="checkbox"/> sd-webui-model-converter	https://jihulab.com/hunter0725/sd-webui-model-converter	未知
<input checked="" type="checkbox"/> sd_dreambooth_extension	https://jihulab.com/hunter0725/sd_dreambooth_extension	未知
<input checked="" type="checkbox"/> 美术风格梯度	https://jihulab.com/hunter0725/stable-diffusion-webui-aesthetic-gradients	未知
<input checked="" type="checkbox"/> 图库浏览器	https://jihulab.com/Akegarasu/sd-webui-images-browser	未知
<input checked="" type="checkbox"/> stable-diffusion-webui-pixelization	https://jihulab.com/hunter0725/stable-diffusion-webui-pixelization.git	未知
<input checked="" type="checkbox"/> stable-diffusion-webui-wd14-tagger	https://jihulab.com/hunter0725/stable-diffusion-webui-wd14-tagger.git	未知
<input checked="" type="checkbox"/> tagcomplete	https://jihulab.com/hunter0725/a1111-sd-webui-tagcomplete.git	未知
<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	

三、功能介绍

文生图最简流程

选择需要使用的模型（底模），这是对生成结果影响最大的因素，主要体现在画面风格上。

Stable Diffusion 模型(ckpt)

dreamshaper_331BakedVae.safetensors [1dceefc07]

abyssorangemix2SFW_abyssorangemix2Sfw.safetensors

Anything-V3.0-pruned-fp32.ckpt

chilloutmix_NiPrunedFp32.safetensors [95afa0d9ea]

CounterfeitV25_25-fp32.ckpt

CounterfeitV25_25.safetensors [a074b8864e]

dalcefoV3Painting_dalcefoV3Painting.safetensors

dreamshaper_331BakedVae.safetensors [1dceefc07]

final-pruned.ckpt [89d59c3dde]

schoolmax25d_11.ckpt [0ce764ebd5]

sd-v1-5-inpainting.ckpt

sd-v1-5.ckpt

ToyEmpires1130_30000.ckpt [841417f259]

模型(ckpt)合并

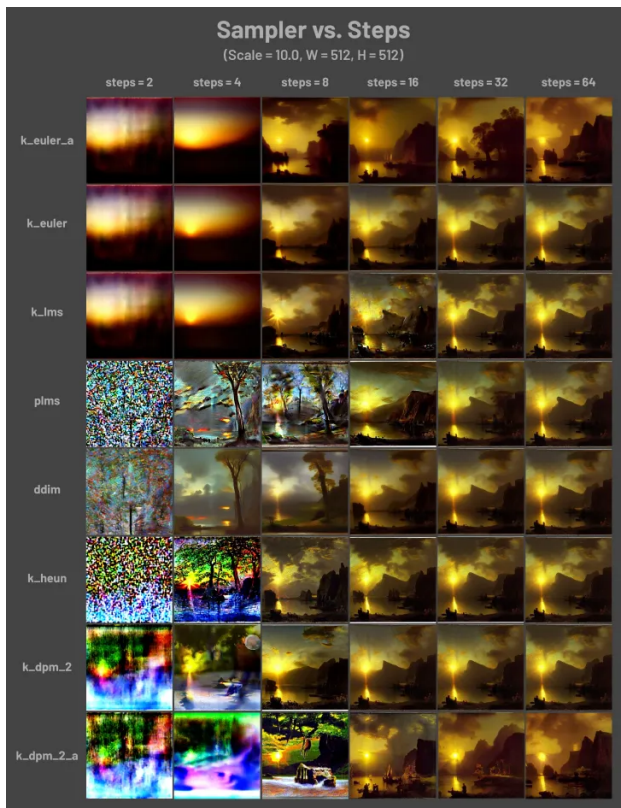
在第一个框中填入提示词（Prompt），对想要生成的东西进行文字描述

Stable Diffusion 模型(ckpt)

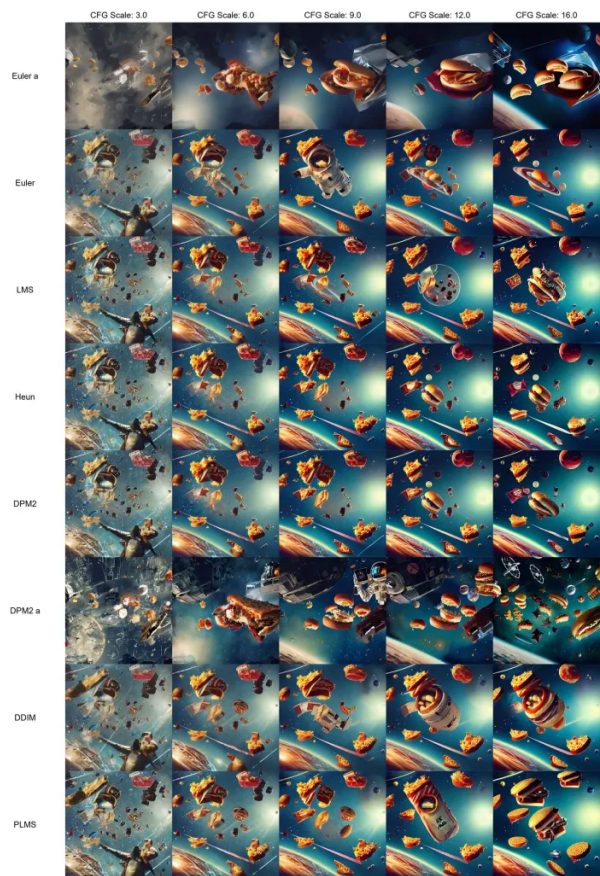
dreamshaper_331BakedVae.safetensors [1dceefc07]

文生图 图生图 更多 图片信息 模型(ckpt)合并 训练 openOutpaint OpenPose Editor Additional Networks Model Converter Dreamb

(extremely detailed CG unity 8k wallpaper), (masterpiece), (best quality:1.2), (highres), intricate details, best quality, realistic, ultra-detailed, best illumination, best shadow, ultra-HD, 1man.soldier, fully armed, in battlefield gun, rifle, Modern



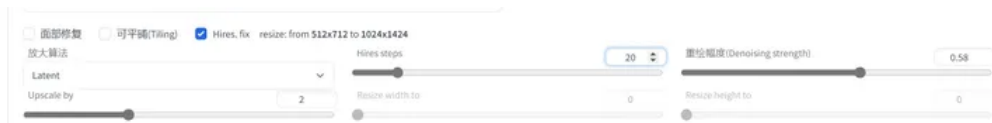
- CFG Scale（提示词相关性） 图像与你的提示的匹配程度。增加这个值将导致图像更接近你的提示，但它也在一定程度上降低了图像质量。 可以用更多的采样步骤来抵消。过高的 CFG Scale 体现为粗犷的线条和过锐化的图像。一般开到 7~11。CFG Scale 与采样器之间的关系：



- 生成批次 每次生成图像的组数。一次运行生成图像的数量为“批次* 批次数量”。
- 每批数量 同时生成多少个图像。增加这个值可以提高性能，但也需要更多的显存。大的 Batch Size 需要消耗巨量显存。若没有超过 12G 的显存，请保持为 1。
- 尺寸 指定图像的长宽。出图尺寸太宽时，图中可能会出现多个主体。1024 之上的尺寸可能会出现不理想的结果，推荐使用小尺寸分辨率+高清修复（Hires fix）。
- 种子 种子决定模型在生成图片时涉及的所有随机性，它初始化了 Diffusion 算法起点的初始值。

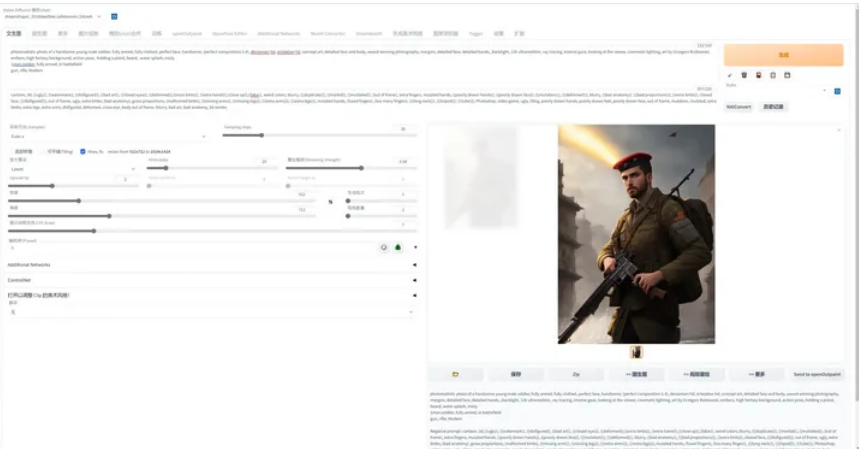
理论上，在应用完全相同参数（如 Step、CFG、Seed、prompts）的情况下，生产的图片应当完全相同。

• 高清修复



通过勾选 "Highres. fix" 来启用。默认情况下，文生图在高分辨率下会生成非常混沌的图像。如果使用高清修复，会型首先按照指定的尺寸生成一张图片，然后通过放大算法将图片分辨率扩大，以实现高清大图效果。最终尺寸为（原分辨率*缩放系数 Upscale by）。

- 放大算法中，Latent 在许多情况下效果不错，但重绘幅度小于 0.5 后就不甚理想。ESRGAN_4x、SwinR 4x 对 0.5 以下的重绘幅度有较好支持。
- Hires step 表示在进行这一步时计算的步数。
- Denoising strength 字面翻译是降噪强度，表现为最后生成图片对原始输入图像内容的变化程度。该值越高，放大后图像就比放大前图像差别越大。低 denoising 意味着修正原图，高 denoising 就和原图就没有大的相关性了。一般来讲阈值是 0.7 左右，超过 0.7 和原图基本上无关，0.3 以下就是稍微改一些。实际执行中，具体的执行步骤为 Denoising strength * Sampling Steps。
- 面部修复 修复画面中人物的面部，但是非写实风格的人物开启面部修复可能导致面部崩坏。
- 点击“生成”



四、提示词

提示词所做的工作是缩小模型出图的解空间，即缩小生成内容时在模型数据里的检索范围，而非直接指定作画结果。提示词的效果也受模型的影响，有些模型对自然语言做特化训练，有些模型对单词标签对特化训练，那么对不同的提示词语言风格的反应就不同。

提示词内容

提示词中可以填写以下内容：

- 自然语言 可以使用描述物体的句子作为提示词。大多数情况下英文有效，也可以使用中文。避免复杂的语法。
- 单词标签 可以使用逗号隔开的单词作为提示词。一般使用普通常见的单词。单词的风格要和图像的整体风格搭配，否则会出现混杂的风格或噪点。避免出现拼写错误。可参考[Tags | Danbooru \(donmai.us\)](https://danbooru.donmai.us/tags)
- Emoji、颜文字 Emoji () 表情符号也是可以使用并且非常准确的。因为 Emoji 只有一个字符，所以在语义准确度上表现良好。关于 emoji 的确切含义，可以参考[Emoji List, v15.0 \(unicode.org\)](https://unicode.org/emoji/charts/)，同时 Emoji 在构图上有影响。



对于使用 Danbooru 数据的模型来说，可以使用西式颜文字在一定程度上控制出图的表情。如：:-) 微笑 :- (不悦 ;-) 使眼色 :-D 开心 :-P 吐舌头 :-C 很悲伤 :-O 惊讶 张大口 :-/ 怀疑

提示词语法

根据自己想画的内容写出提示词，多个提示词之间使用英文半角符号 [,]，如：

masterpiece, best quality, ultra-detailed, illustration, close-up, straight on, face focus, 1girl, white hair, golden eyes, long hair, halo, angel wings, serene expression, looking at viewer

一般而言，概念性的、大范围的、风格化的关键词写在前面，叙述画面内容的关键词其次，最后是描述细节的关键词，大致顺序如：

(画面质量提示词), (画面主题内容)(风格), (相关艺术家), (其他细节)

不过在模型中，每个词语本身自带的权重可能有所不同，如果模型训练集中较多地出现某种关键词，我们在提示词中只输入一个词就能极大地影响画面，反之如果模型训练集中较少地出现某种关键词，我们在提示词中可能输入很多个相关词汇都对画面的影响效果有限。提示词的顺序很重要，越靠后的权重越低。关键词最好具有特异性，譬如 Anime(动漫)一词就相对泛化，而 Jojo 一词就能清晰地指向 Jojo 动漫的画风。措辞越不抽象越好，尽可能避免留下解释空间的措辞。

可以使用括号人工修改提示词的权重，方法如：

- (word) - 将权重提高 1.1 倍
- ((word)) - 将权重提高 1.21 倍 ($= 1.1 * 1.1$)
- [word] - 将权重降低至原先的 90.91%
- (word:1.5) - 将权重提高 1.5 倍
- (word:0.25) - 将权重减少为原先的 25%
- (word) - 在提示词中使用字面意义上的 () 字符

$(n) = (n:1.1) ((n)) = (n:1.21) (((n))) = (n:1.331) ((((n)))) = (n:1.4641) ((((((n)))))) = (n:1.61051) (((((((n))))))) = (n:1.771561)$

请注意，权重值最好不要超过 1.5。

还可以通过 Prompt Editing 使得 AI 在不同的步数生成不一样的内容，譬如在某阶段后，绘制的主体由男人变成女人。



语法为：

- 1 [to:when] 在指定数量的 step 后，将to处的提示词添加到提示
- 2 [from::when] 在指定数量的 step 后从提示中删除 from处的提示词
- 3 [from:to:when] 在指定数量的 step 后将 from处的提示词替换为 to处的提示词

例如: a [fantasy:cyberpunk:16] landscape 在一开始，读入的提示词为： the model will be drawing a fantasy landscape. 在第 16 步之后,提示词将被替换为： a cyberpunk landscape, 它将继续在之前的图像上计算

又例如，对于提示词为: fantasy landscape with a [mountain:lake:0.25] and [an oak:a christmas tree:0.75][in foreground::0.6][in background:0.25][shoddy:masterful:0.5]，100 步采样，一开始。提示词为： fantasy landscape with a mountain and an oak in foreground shoddy 在第 25 步后，提示词为： fantasy landscape with a lake and an oak in foreground in background shoddy 在第 50 步后，提示词为： fantasy landscape with a lake and an oak in foreground in background masterful 在第 60 步后，提示词为： fantasy landscape with a lake and an oak in background masterful 在第 75 步后，提示词为： fantasy landscape with a lake and a christmas tree in background masterful

提示词还可以轮转，比如

- 1 [cow|horse] in a field

在第一步时，提示词为“cow in a field”；在第二步时，提示词为"horse in a field."；在第三步时，提示词为"cow in a field"，以此类推。



Token

实际上，程序是将输入的关键词以 Token 的形式传入模型进行计算的：

Text input

ID input

(Medieval astronomer using a telescope with a cosmic starry sky in the background),(sketch, hand draw style, icon, uncomplicated background)

Tokenize

Text

Tokens

(medieval astronomer using a telescope with a cosmic starry sky in the background),(sketch , hand draw style , icon , uncomplicated background)

“ *(Medieval astronomer using a telescope with a cosmic starry sky in the background.sketch, hand draw style, con, uncomplicated background)*” 转换为 Token ID 即： 263, 10789, 40036, 1996, 320, 19037, 593, 320, 18304, 30963, 2390, 530, 518, 5994, 8, 11, 263, 5269, 267, 2463, 4001, 1844, 267, 5646, 267, 569, 16621, 5994, 264。

一个单词可能对应一个或多个 Token，多个单词也可能对应同一个 Token。

提示词模板

可参考[Civitai | Stable Diffusion models, embeddings, hypernetworks and more](#)中优秀作品的提示词作为模板。

类似的网站还有：

- Majinai: [MajinAI | Home](#)
- 词图: [词图 PromptTool - AI 绘画资料管理网站](#)
- Black Lily: [black_lily](#)
- Danbooru 标签超市: [Danbooru 标签超市](#)

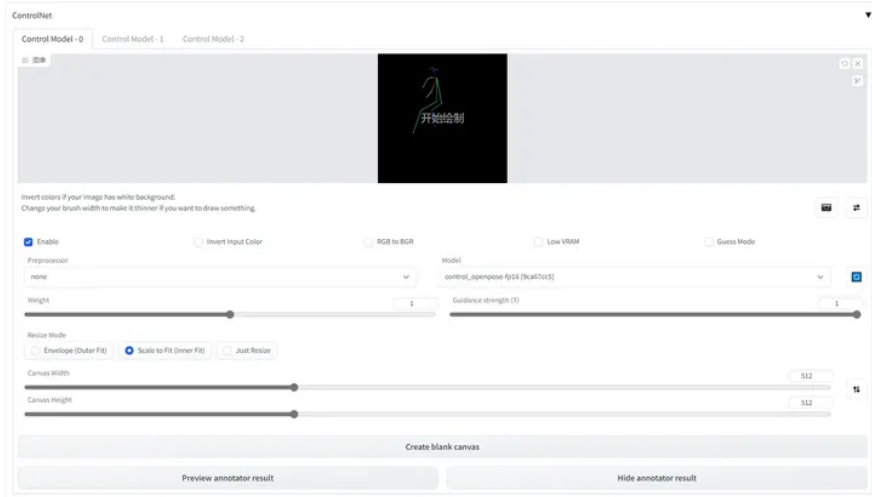
- 魔咒百科词典：[魔咒百科词典](#)
- AI 词汇加速器：[AI 词汇加速器 AcceleratorI Prompt](#)
NovelAI 魔导书：[NovelAI 魔导书](#)
- 螫哲法典：[螫哲法典](#)
- Danbooru tag：[Tag Groups Wiki | Danbooru \(donmai.us\)](#)
- AIBooru：[AIBooru: Anime Image Board](#)

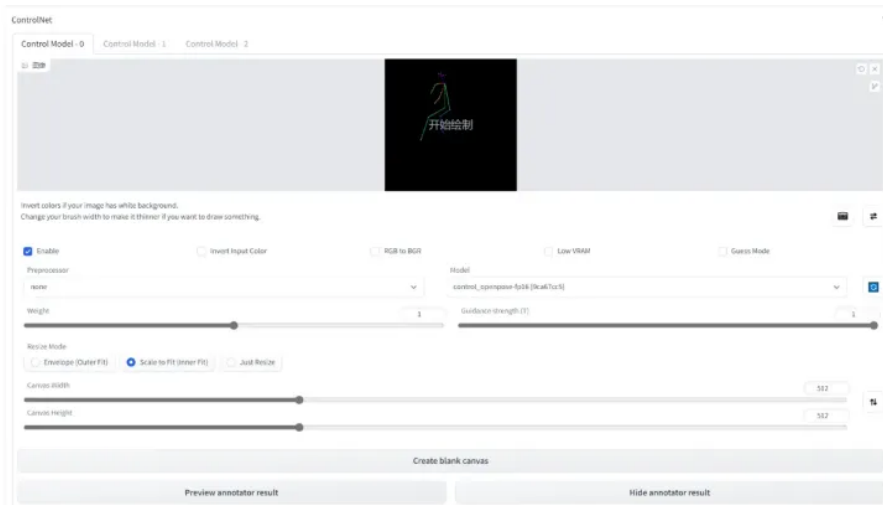
五、Controlnet

Controlnet 允许通过线稿、动作识别、深度信息等对生成的图像进行控制。
请注意，在使用前请确保 ControlNet 设置下的路径与本地 Stable Diffusion 的路径同步



基本流程





- 点击 Enable 启用该项 ControlNet
- Preprocessor 指预处理器，它将对输入的图像进行预处理。如果图像已经符合预处理后的结果，请选择 None。譬如，图中导入的图像已经是 OpenPose 需要的骨架图，那么 preprocessor 选择 none 即可。
- 在 Weight 下，可以调整该项 ControlNet 的在合成中的影响权重，与在 prompt 中调整的权重类似。Guidance strength 用来控制图像生成的前百分之多少步由 Controlnet 主导生成，这点与[:]语法类似。
- Invert Input Color 表示启动反色模式，如果输入的图片是白色背景，开启它。
- RGB to BGR 表示将输入的色彩通道信息反转，即 RGB 信息当做 BGR 信息解析，只是因为 OpenCV 中使用的是 BGR 格式。如果输入的图是法线贴图，开启它。
- Low VRAM 表示开启低显存优化，需要配合启动参数 “--lowvram”。
- Guess Mode 表示无提示词模式，需要在设置中启用基于 CFG 的引导。
- Model 中请选择想要使用解析模型，应该与输入的图像或者预处理器对应。请注意，预处理器可以为空，但模型不能为空。

可用预处理/模型

- canny 用于识别输入图像的边缘信息。

原图	预处理结果	生成图像结果
		

- depth 用于识别输入图像的深度信息。

原图	预处理结果	生成图像结果
		

- hed 用于识别输入图像的边缘信息，但边缘更柔和。

原图	预处理结果	生成图像结果
		


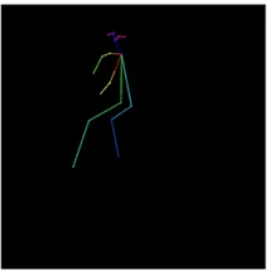

- mlsd 用于识别输入图像的边缘信息，一种轻量级的边缘检测。它对横平竖直的线条非常敏感，因此更适用于室内图的生成。

原图	预处理结果	生成图像结果
		

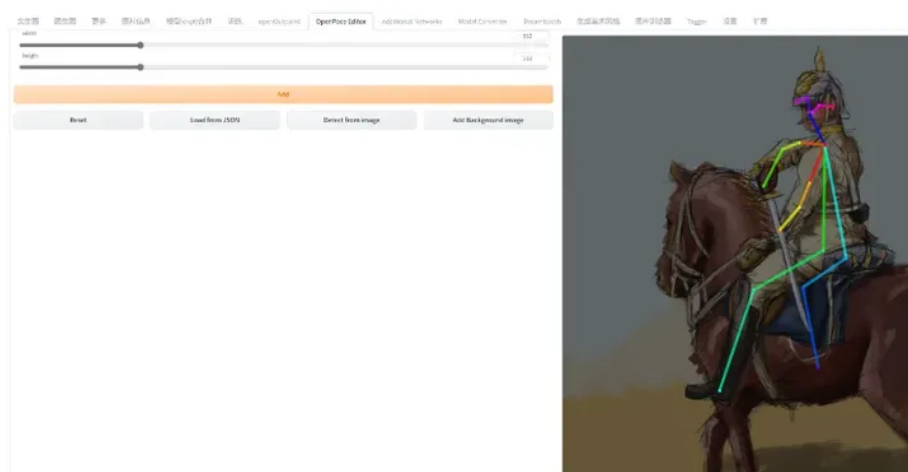
- normal 用于识别输入图像的法线信息。

原图	预处理结果	生成图像结果
		

- openpose 用于识别输入图像的动作信息。

原图	预处理结果	生成图像结果
		

OpenPose Editor 插件可以自行修改姿势，导出到文生图或图生图。



- scribble 将输入图像作为线稿识别。如果线稿是白色背景，务必勾选 “Invert Input Color”

原图	预处理结果	生成图像结果
		

- fake_scribble 识别输入图像的线稿，然后再将它作为线稿生成图像。

原图	预处理结果	生成图像结果
		

- segmentation 识别输入图像各区域分别是什么类型的物品，再用此构图信息生成图像。

原图	预处理结果	生成图像结果
		

如果想绘制一张符合 segementation 规范的图像，可以使用以下色表绘制。

[color_coding_semantic_segmentation_classes - Google 表格](#)

A	B	C	D	E	F	G	H	I
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		wall
2	0.1072	6046	612	1	(180, 120, 120)	#B47878		building,edifice
3	0.0878	8265	796	1	(6, 230, 230)	#06E6E6		sky
4	0.0621	9336	917	1	(80, 50, 50)	#503232		floor,flooring
5	0.048	6678	641	0	(4, 205, 3)	#04C803		tree
6	0.045	6604	643	1	(120, 120, 80)	#787850		ceiling
7	0.0398	4023	408	1	(140, 140, 140)	#8C8C8C		road,route
8	0.0231	1906	199	0	(204, 5, 255)	#CC05FF		bed
9	0.0198	4688	460	0	(230, 230, 230)	#E6E6E6		windowpane>window
10	0.0183	2423	225	1	(4, 250, 7)	#04FA07		grass
11	0.0181	2874	294	0	(224, 5, 255)	#E005FF		cabinet
12	0.0166	3068	310	1	(235, 255, 7)	#E8FF07		sidewalk>pavement
13	0.016	5075	526	0	(150, 5, 61)	#90053D		person,individual;someone,somebody;mortal;soul
14	0.0151	1804	190	0	(120, 120, 70)	#787846		earth;ground
15	0.0118	6666	796	0	(8, 255, 51)	#08FF33		door,double;door
16	0.011	4269	411	0	(255, 6, 82)	#FF0652		table
17	0.0109	1691	160	1	(143, 255, 140)	#8FFF8C		mountain;mount
18	0.0104	3999	441	0	(204, 255, 4)	#CCFF04		plant;flora;plant;life
19	0.0104	2149	217	0	(255, 51, 7)	#FF3307		curtain;drape;drapery;mantle;pall
20	0.0103	3261	318	0	(204, 70, 3)	#CC4603		chair
21	0.0098	3164	306	0	(0, 102, 200)	#0066C8		car;auto;automobile;machine;motorcar
22	0.0074	709	75	1	(61, 230, 250)	#3DE6FA		water
23	0.0067	3296	315	0	(255, 6, 51)	#FF0633		painting;picture
24	0.0065	1191	106	0	(11, 102, 255)	#0B66FF		sofa;couch;lounge
25	0.0061	1516	162	0	(255, 7, 71)	#FF0747		shelf
26	0.006	667	69	1	(255, 9, 224)	#FF09E0		house
27	0.0053	651	57	1	(9, 7, 230)	#0907E8		sea
28	0.0052	1847	224	0	(220, 220, 220)	#DCDCDC		mirror
29	0.0046	1158	128	1	(255, 9, 92)	#FF095C		rug;carpet;carpeting
30	0.0044	480	44	1	(112, 9, 255)	#7009FF		field
31	0.0044	1172	98	0	(8, 255, 214)	#08FFD6		armchair
32	0.0044	1292	184	0	(7, 255, 224)	#07FFE0		seat
33	0.0033	1386	138	0	(255, 184, 6)	#FFB806		fence;fencing
34	0.0031	698	61	0	(10, 255, 71)	#0AFF47		desk
35	0.003	781	73	0	(255, 41, 10)	#FF290A		rock;stone
36	0.0027	380	43	0	(7, 255, 255)	#07FFFF		wardrobe;closet;press
37	0.0026	3089	302	0	(224, 255, 8)	#E0FFF8		lamp
38	0.0024	404	37	0	(102, 8, 255)	#6608FF		bath;ub;bathing;tub;bath;tub
39	0.0024	804	99	0	(255, 61, 6)	#FF3D06		railing;rail
40	0.0023	1453	153	0	(255, 194, 7)	#FFC207		cushion
41	0.0023	411	37	0	(255, 122, 8)	#FF7A08		base;pedestal;stand
42	0.0022	1440	162	0	(0, 255, 20)	#00FF14		box
43	0.0022	800	77	0	(255, 8, 41)	#FF0829		column;pillar
44	0.002	2650	298	0	(255, 5, 153)	#FF0599		signboard;sign
45	0.0019	549	46	0	(6, 51, 255)	#0633FF		chest;of;drawers;chest;bureau;dresser
46	0.0019	367	36	0	(235, 12, 255)	#E80CFF		counter
47	0.0018	311	30	1	(160, 150, 20)	#A09614		sand
48	0.0018	1181	122	0	(0, 163, 255)	#00A3FF		sink
49	0.0018	287	23	1	(140, 140, 140)	#8C8C8C		skyscraper
50	0.0018	468	38	0	(0250, 10, 15)	#FA0A0F		fireplace;hearth;open;fireplace
51	0.0018	402	43	0	(20, 255, 0)	#14FF00		refrigerator;icebox
52	0.0018	130	12	1	(31, 255, 0)	#1FFF00		grandstand;covered;stand
53	0.0018	561	64	1	(255, 31, 0)	#FF1F00		path
54	0.0017	880	102	0	(255, 224, 0)	#FFE000		stairs;steps
55	0.0017	86	12	1	(153, 255, 0)	#99FF00		runway
56	0.0017	172	11	0	(0, 0, 255)	#0000FF		case;display;case;showcase;vitrine
57	0.0017	198	18	0	(255, 71, 0)	#FF4700		pool;table;billard;table;snooker;table
58	0.0017	930	109	0	(0, 235, 255)	#00E8FF		pillow
59	0.0015	139	18	0	(0, 173, 255)	#00ADFF		screen;door;screen
60	0.0015	564	52	1	(31, 0, 255)	#1F00FF		stairway;staircase
61	0.0015	320	26	1	(11, 200, 200)	#0BC8C8		river
62	0.0015	261	29	1	(255, 82, 0)	#FF5200		bridge;span
63	0.0014	275	22	0	(0, 255, 245)	#00FFF5		bookcase

多 ControlNet 合成

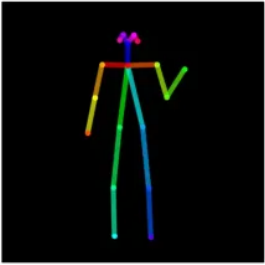
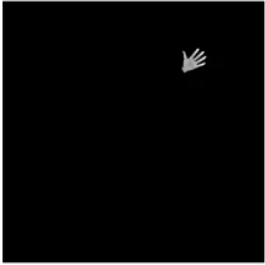

在 ControlNet 的设置下，可以调整可用 ControlNet 的数量。

Multi ControlNet: Max models amount (requires restart)

Model cache size (requires restart)

在多个 ControlNet 模式下，结果会将输入的信息合并生成图像：



输入A	输入B	结果
		

六、更多模型

模型下载

模型能够有效地控制生成的画风和内容。 常用的模型网站有：

[Civitai | Stable Diffusion models, embeddings, hypernetworks and more](#) > [Models - Hugging Face](#) > [SD - WebUI 资源站](#) > [元素法典 AI 模型收集站](#) - [AI 绘图指南 wiki \(aiguidebook.top\)](#) > [AI 绘画模型博物馆 \(subrecovery.top\)](#)

模型安装

下载模型后需要将之放置在指定的目录下，请注意，不同类型的模型应该拖放到不同的目录下。 模型的类型可以通过[Stable Diffusion 法术解析](#)检测。

Stable Diffusion 法术解析

从 Stable Diffusion 生成的图片读取 prompt / Stable Diffusion 模型解析



拖动文件到这里或者点击选择文件

模型信息

文件名	dreamshaper_331BakedVae.safetensors
文件大小	5.57 GB
模型种类	Stable Diffusion 模型
模型用法	大模型。放入 models/Stable-diffusion 文件夹后，进入 webui 在左上角点击刷新后选择模型。

[图文详解！最全模型用法](#)

*运算完全在你的电脑上运行不会上传到云端

如果您觉得本项目对您有帮助 请在 [GitHub](#) 上点个star

Made with ❤ by [@Akegarasu](#) | [秋葉aaaki](#) | [NovelAI.Dev](#)

- 大模型（Ckpt）：放入 models\Stable-diffusion

电脑 > 新加卷 (G:) > stable-diffusion-webui_0220 > models > Stable-diffusion

在 Stable-diffusion 中搜索

名称	修改日期	类型	大小
abyssorangemix2SFW_abyssorangemix2Sfw...	2023/2/15 22:51	SAFETENSORS 文件	5,440,239 KB
Anything-V3.0.vae.pt	2023/1/29 12:41	PT 文件	803,519 KB
Anything-V3.0-pruned-fp32.ckpt	2023/1/29 12:40	CKPT 文件	4,165,360 KB
chilloutmix_NiPrunedFp32.safetensors	2023/2/13 11:32	SAFETENSORS 文件	4,165,134 KB
CounterfeitV25_25.safetensors	2023/2/16 13:51	SAFETENSORS 文件	4,165,134 KB
CounterfeitV25_25-fp32.ckpt	2023/2/17 14:04	CKPT 文件	4,165,360 KB
dalcefoV3Painting_dalcefoV3Painting.safet...	2023/2/15 22:44	SAFETENSORS 文件	3,761,487 KB
dreamshaper_331BakedVae.safetensors	2023/2/15 22:50	SAFETENSORS 文件	5,843,978 KB
final-pruned.ckpt	2022/10/8 3:19	CKPT 文件	4,165,408 KB
final-pruned.vae.pt	2022/10/8 3:17	PT 文件	803,519 KB
schoolmax25d_11.ckpt	2023/2/16 13:54	CKPT 文件	5,440,609 KB
sd-v1-5.ckpt	2023/1/29 13:29	CKPT 文件	4,165,411 KB
sd-v1-5-inpainting.ckpt	2023/1/29 13:19	CKPT 文件	4,165,467 KB
ToyEmpires1130_30000.ckpt	2022/11/30 23:55	CKPT 文件	4,001,974 KB

- VAE 模型：一些大模型需要配合 vae 使用，对应的 vae 同样放置在 models\Stable-diffusion 或 models\VAE 目录，然后在 webui 的设置栏目选择。

电脑 > 新加卷 (G:) > stable-diffusion-webui_0220 > models > VAE

名称	修改日期
Put VAE here.txt	2022/12/21 8:40



缓存在内存(RAM)中的模型(ckpt)

VAE Checkpoints to cache in RAM

模型的 VAE (SD VAE)

Automatic

无

Anything-V3.0.vae.pt

final-pruned.vae.pt

Noise multiplier for img2img

- Lora/LoHA/LoCon 模型：放入 extensions\sd-webui-additional-networks\models\lora，也可以在 models/Lora 目录

tensions > sd-webui-additional-networks > models > lora

名称	修改日期
.keep	2023/2/20 18:58
highrisemix_v25.safetensors	2023/2/16 13:45
koreanDollLikeness_v10.safetensors	2023/2/13 12:41
ToyEmpires.safetensors	2023/2/3 15:05
女性单手修复 handfix.safetensors	2023/2/6 14:22

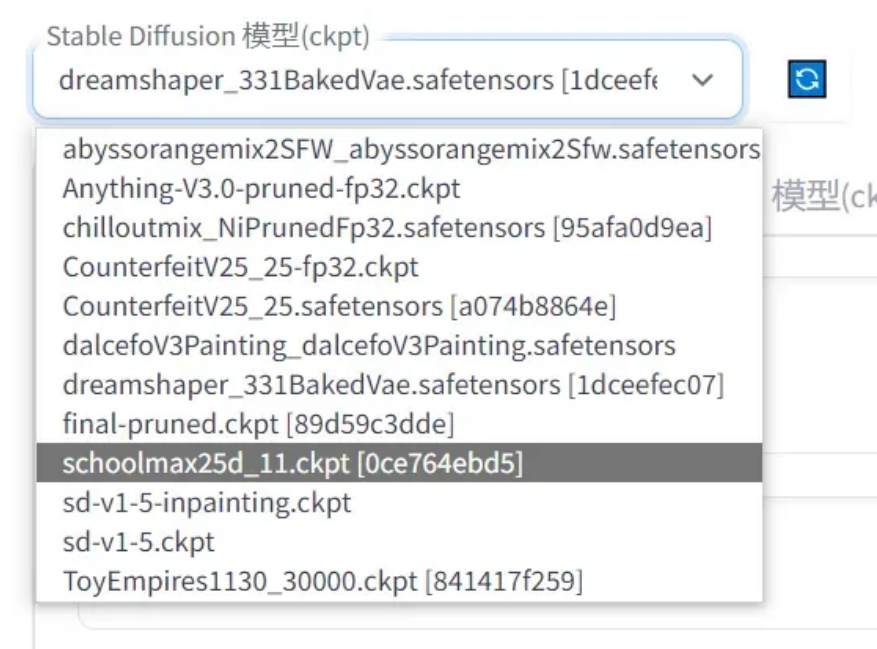
脑 > 新加卷 (G:) > stable-diffusion-webui_0220 > models > lora				在 lora 中搜索
名称	修改日期	类型	大小	
highrisemix_v25.safetensors	2023/2/16 13:45	SAFETENSORS 文件	2,082,643 KB	
koreanDollLikeness_v10.safetensors	2023/2/13 12:41	SAFETENSORS 文件	147,575 KB	
ToyEmpires.safetensors	2023/2/3 15:05	SAFETENSORS 文件	36,976 KB	
女性单手修复 handfix.safetensors	2023/2/6 14:22	SAFETENSORS 文件	147,569 KB	

- Embedding 模型：放入 embeddings 目录

名称	修改日期
 Place Textual Inversion embeddings here.txt	2022/12/21 8:40

模型使用

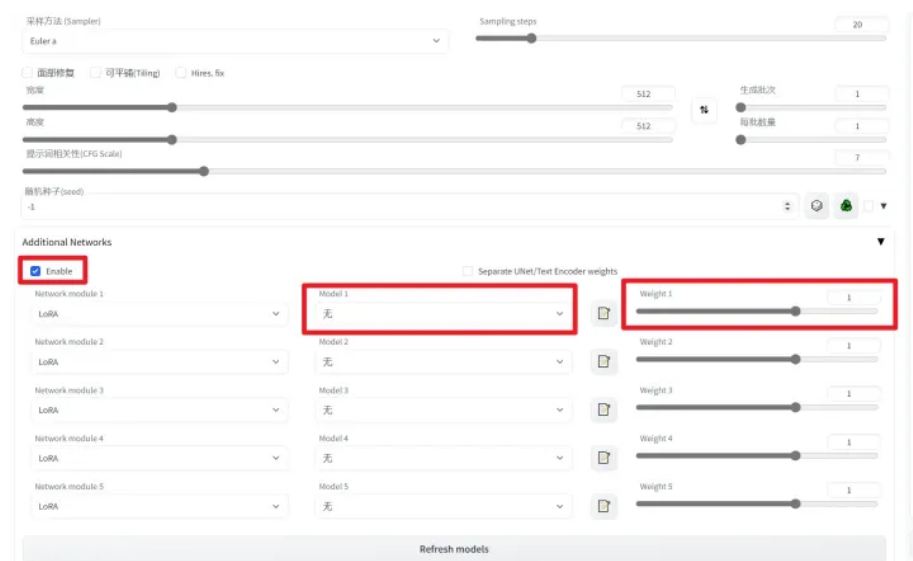
- Checkpoint(ckpt)模型 对效果影响最大的模型。在 webui 界面的左上角选择使用。



一些模型会有触发词，即在提示词内输入相应的单词才会生效。

- Lora 模型 / LoHA 模型 / LoCon 模型

对人物、姿势、物体表现较好的模型，在 ckpt 模型上附加使用。在 webui 界面的 Additional Networks 下勾选 Enable 启用，然后在 Model 下选择模型，并可用 Weight 调整权重。权重越大，该 Lora 的影响也越大。不建议权重过大（超过 1.2），否则很容易出现扭曲的结果。



多个 lora 模型混合使用可以起到叠加效果，譬如一个控制面部的 lora 配合一个控制画风的 lora 就可以生成具有特定画风的特定人物。因此可以使用多个专注于不同方面优化的 Lora，分别调整权重，结合出自己想要实现的效果。

LoHA 模型是一种 LORA 模型的改进。

LoCon 模型也是一种 LORA 模型的改进，泛化能力更强。

- Embedding

对人物、画风都有调整效果的模型。在提示词中加入对应的关键词即可。大部分 Embedding 模型的关键词与文件名相同，譬如一个名为为 “SomeCharacter.pt” 的模型，触发它的关键词检索 “SomeCharacter”。