

第 1 章 ComfyUI 基础入门

Chapter 1

Stable
Diffusion-ComfyUI
AI 绘画 workflow 解析

自从 Stable Diffusion 开源发布以来，已经陆续出现了 WebUI、ForgeUI、ComfyUI 和 SwarmUI 等图形界面。尽管这些界面中运行的都是 Stable Diffusion，但在设计理念、操作方式和适用的用户群体等方面存在显著的差异。本章将首先介绍目前最受欢迎的几款 Stable Diffusion 界面的特点，然后详细讲解 ComfyUI 的基础操作，为后续学习打下基础。

1.1 常用 Stable Diffusion 界面的优缺点

1 WebUI

我们在计算机上使用各种软件时，通常与软件的界面打交道，很少关心这些软件的实现原理以及底层代码如何运作。然而，Stability AI 公司起初只发布了 Stable Diffusion 的代码，甚至没有提供任何交互界面。普通用户下载这些代码后只能望洋兴叹。直到开源社区中的一位 ID 为 Automatic1111 的用户把模型代码打包成图形界面，我们才能通过命令和选项的方式指挥 Stable Diffusion 生成图片。因为这个图形界面在网页浏览器中运行，所以被称作 WebUI。同时，Automatic1111 这个 ID 也成了 WebUI 的代名词。在 ComfyUI 中经常可以看到名称为 Automatic1111 或 1111 的选项，这些选项基本上都代表着兼容 WebUI 的运行机制。

因为没有选择的余地，所以 Stable Diffusion 的早期用户都是从 WebUI 开始接触 AI 绘图的。WebUI 采用了常规软件的界面布局和控件类型，虽然初看略显简陋，但胜在直观明了，而且符合大多数人的操作习惯，如图 1-1 所示。

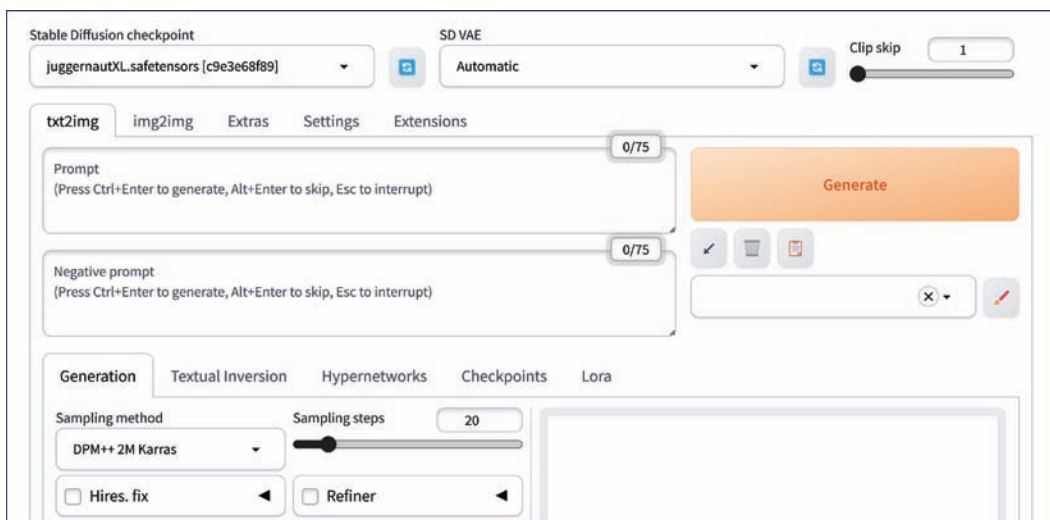


图 1-1

WebUI 的最大优点是操作起来比较简单。当然，这里的“简单”是相对的，很多新用户仍然会被层出不穷的名词和大量参数选项难住。但是，当你使用过 ComfyUI 后，回头再看 WebUI，就会深刻感受到 WebUI 的简洁。

WebUI 的另一个优点是成熟的开源生态。因为具有先发优势，所以 WebUI 的插件数量众多，而且网络上的学习资料丰富，新用户的学习门槛相对较低。然而，WebUI 在显存调度方面存在问题。众所周知，AI 绘图对显存的要求较高，单单载入一个 SDXL 版的大模型就需要 8GB 以上的显存，再加上高清修复和各种插件的使用，低容量显存的用户可能会频繁遭遇显存溢出报错的困扰。尽管 WebUI 和后来的 ForgeUI 在显存调度方面付出过很多努力，但由于底层架构的原因，仍然无法达到 ComfyUI 的使用体验。

2 ForgeUI

ForgeUI 的开发者是张吕敏，他也是著名插件 ControlNet、Layer Diffusion、ICLight 和 AI 绘图工具 Fooocus 的开发者。ForgeUI 在 WebUI 的基础上进行了一系列的底层代码优化，可以有效降低显存的占用率，减少因显存不足导致的报错或性能下降问题，提升了低显存用户的使用体验。

ForgeUI 的界面和 WebUI 完全一致，大部分插件也是通用的，WebUI 用户可以无缝迁移到 ForgeUI。另外，ForgeUI 还集成了一些 WebUI 中没有的功能，很多最前沿的插件，例如生成透明图层的 Layer Diffusion、制作动画的 SVD 等，都能第一时间在 ForgeUI 中使用，如图 1-2 所示。

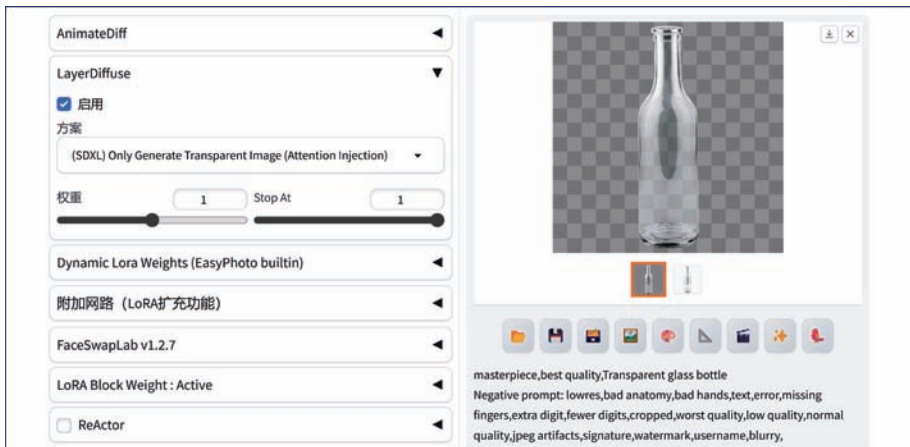


图 1-2

ForgeUI 通过代码优化、分块生成和显存释放等手段，虽然改善了显存溢出的问题，但是对于配备了 6GB 和 8GB 显存的显卡来说，这种程度的改善只是让其“可以”生成图片。一旦涉及高清重绘和超清放大，就只能在降速模式下勉强运行。低显存用户要想获得更高画质或者体验前沿功能和新模型，要么换显卡，要么转移到 ComfyUI。

3 ComfyUI

WebUI 和 ForgeUI 就像是已经出厂的汽车，用户只需要学会如何驾驶即可。而 ComfyUI 采用了节点式的界面设计，相当于只提供了各种各样的零部件，用户不仅要学习驾驶，还要像技师那样掌握一定的装配技能，如图 1-3 所示。



图 1-3

ComfyUI 的优势主要体现在 4 个方面。首先，ComfyUI 的灵活度非常高，用户可以自由组合节点，就像拼装乐高玩具那样，把各种各样的节点组装成自行车、汽车、拖拉机，甚至是火车，从而实现不同的任务目标。更重要的是，每个工作流都可以打造成自动运行的“小程序”，只需输入一段文字或者上传一张图片，就能实现一键换装、卡通头像生成、老照片修复等效果。

其次，ComfyUI 只需在界面中拖入工作流文件或生成的图片，就能完美还原所有节点体系和设置参数，轻松实现工作流的重复使用。这就意味着初学者不仅可以通过“抄作业”的方式学习其他用户分享的工作流，还能随时从网络上调用数以千计的“成品”资源，如图 1-4 所示。

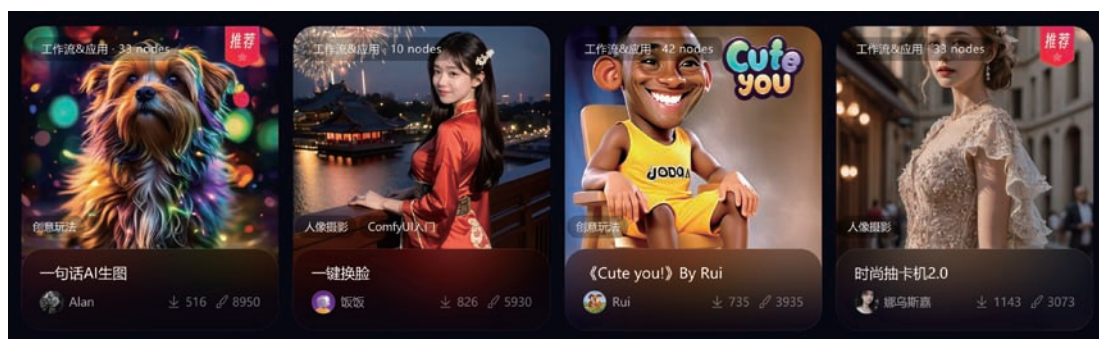


图 1-4

再次，ComfyUI 更节省显存，生成图片的速度也更快。WebUI 相当于一个必须加载所有节点的工作流，而 ComfyUI 可以自由选配组件，不需要的就不添加，即使添加了节点，不运行时也不会载入显存，而且只要不更改设置参数，运行过的节点就不需要重新计算。因此，可以大幅度减少显存溢出报错的问题，同样的显卡可以生成分辨率更高的图片。

最后，ComfyUI 的架构比较开放，很多最新、最前沿的模型和插件都会首先应用到 ComfyUI 中，而 WebUI 的用户需要等待一段时间后才能体验。

ComfyUI 的缺点是学习门槛较高。以汽车举例，WebUI 的用户只需负责开车，记住常用的参数和操作即可。而 ComfyUI 的用户既要学会开车又要掌握组装技能，没有一定的理论功底很难驾驭复杂的流程。此外，在一些比较简单的应用场景中，由于需要搭建和改造工作流的过程，ComfyUI 在操作方面比 WebUI 烦琐得多。

当然，WebUI 和 ComfyUI 不是非此即彼的关系，因为最占用磁盘空间的模型文件大部分可以共享，所以很多用户会同时安装两套界面。普通的任务可以在自己更熟悉或者实现起来更方便的界面中进行，遇到需要自动运行或者 WebUI 无法实现的任务时，再使用 ComfyUI。

有时，我们还可以把两者的优势结合起来。例如，在 WebUI 中可以非常方便地使用 ControlNet 的 Tile 模型来修复图片，手部的修复效果更好。要想在 ComfyUI 中实现从图片生成到手部修复，再到全图重绘的完整流程，需要创建非常多的节点，且效果可能还比不上 WebUI。如果只从效率和实用性的角度出发，完全可以让 WebUI 接力完成这部分的后期处理工作。

4 SwarmUI

SwarmUI 是 Stability AI 官方发布的界面。第一眼看上去，这个界面和 WebUI 差别不大，都是通过现成的参数选项生成图片，如图 1-5 所示。



图 1-5

切换到“Comfy 工作流编辑器”选项卡后，我们会发现这里原封不动地复刻了 ComfyUI，如图 1-6 所示。由此可知，SwarmUI 的目标是把 WebUI 和 ComfyUI 结合起来，分别执行简单的任务和复杂的流程。



图 1-6

可惜的是，这个界面似乎有些“生不逢时”，未能引起 Stable Diffusion 用户太大的兴趣。原因很简单，从本质上讲，SwarmUI 的核心还是 ComfyUI。许多用户之所以保留 WebUI 和 ComfyUI 两套体系，是因为看重各自的生态，而不仅仅是依赖两套界面的简单结合。此外，更换新界面意味着需要重新下载和配置各种插件和模型，同时还面临不确定性的风险。只是增加一个集成窗口，很难说服用户放弃更为成熟的体系。

1.2 部署和运行 ComfyUI



下面介绍在计算机上部署 ComfyUI 的两种方法。

1 从 GitHub 下载并安装

登录 <https://github.com/comfyanonymous/ComfyUI>，向下拖动页面，找到并单击 Direct link to download 以下载压缩包，如图 1-7 所示。

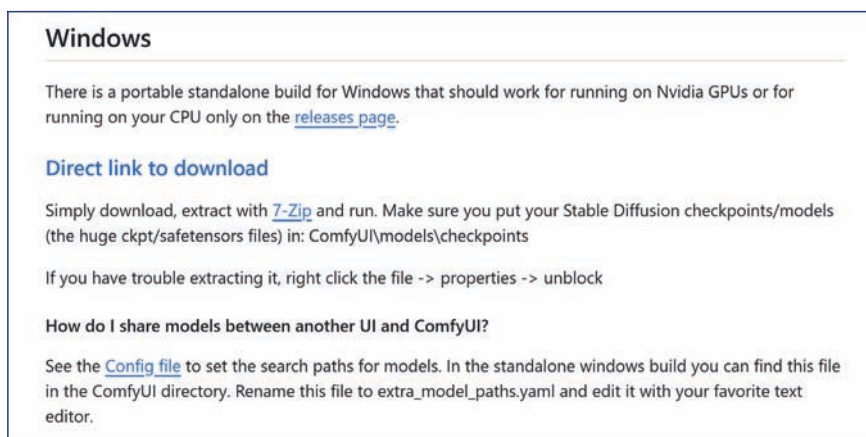


图 1-7

下载完成后，解压缩文件，然后双击运行文件夹中的 run_nvidia_gpu.bat 文件，等命令提示符中的进程加载完成后，将自动打开网页和 ComfyUI 界面。

ComfyUI 界面的右侧有一个设置面板，单击设置面板右上角的  按钮可以将面板最小化。单击设置面板上的  按钮可以打开 Settings 窗口，在 Color palette 菜单中可以选择画布的配色方案，在 Link Render Mode 菜单中可以选择连线的样式，如图 1-8 所示。设置面板以外的区域用于显示节点的画布。滚动鼠标中键可以缩放画布，按住鼠标左键或空格键后移动鼠标指针可以移动画布。

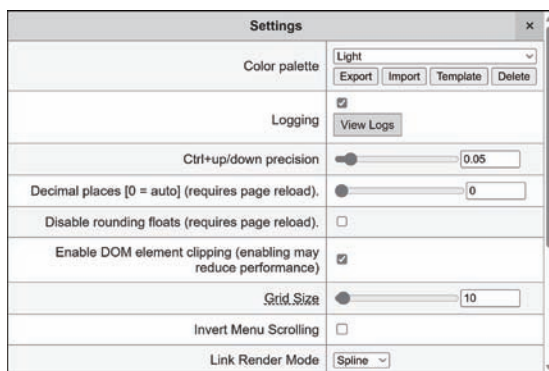


图 1-8

画布上已经创建了默认的文生图工作流。由于尚未下载安装大模型文件，如果现在单击设置面板上的 Queue Prompt 按钮运行工作流，将会弹出如图 1-9 所示的出错提示信息。

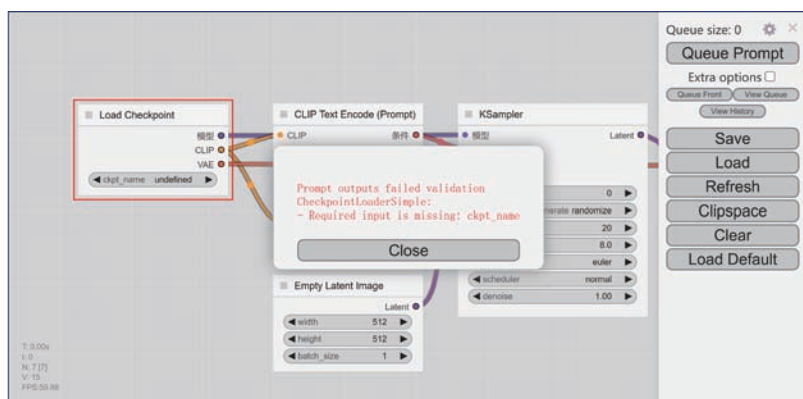


图 1-9

登录一个模型下载网站，这里以 <https://www.liblib.art> 为例。单击首页右上角的“全部类型”按钮，在弹出的菜单中选择 CHECKPOINT，如图 1-10 所示。

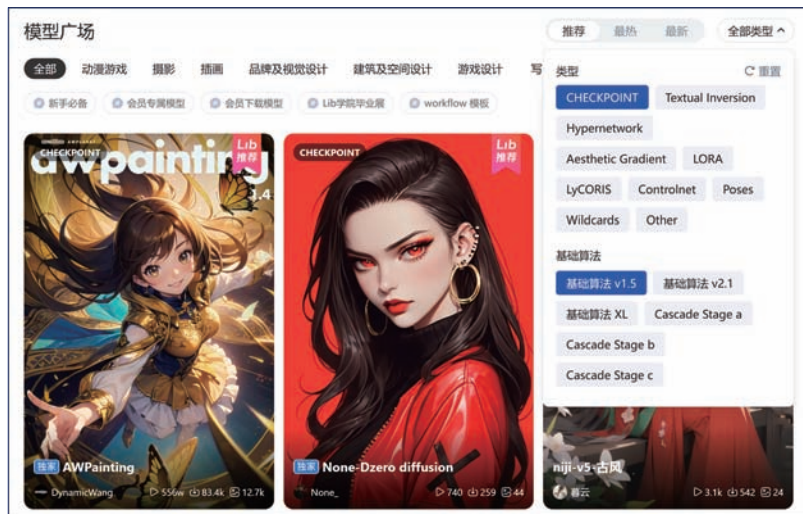


图 1-10

单击一个模型的封面图，然后在详情页的右上方单击链接下载大模型文件，如图 1-11 所示。下载完成后，把模型文件剪贴到 ComfyUI 根目录下的 ComfyUI\models\checkpoints 文件夹中。

单击设置面板上的 Refresh 按钮，或者按 F5 键刷新页面，然后在 Load Checkpoint 节点中选择大模型，如图 1-12 所示。现在单击 Queue Prompt 按钮或者按 Ctrl+Enter 键，等工作流运行完毕，即可生成图片。



图 1-11

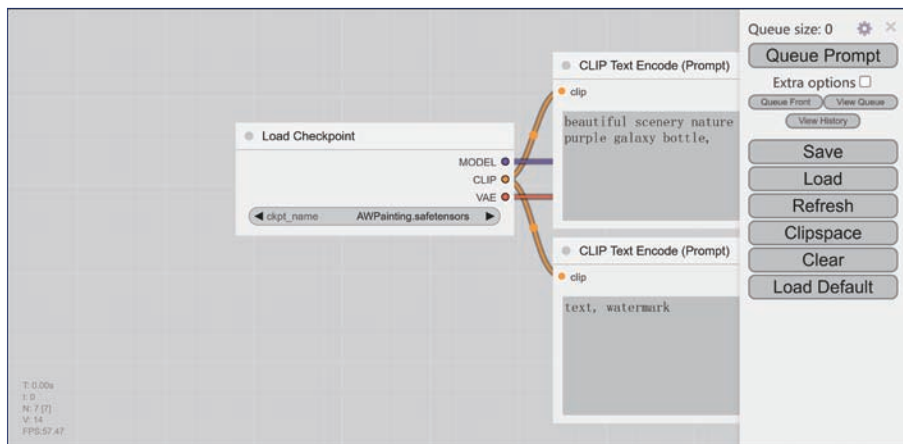


图 1-12

已经安装了 WebUI 的用户可以打开 ComfyUI 根目录中的 ComfyUI 文件夹，把里面的 extra_model_paths.yaml.example 文件更名为 extra_model_paths.yaml。在更名后的文件上右击，执行“在记事本中编辑”命令，把 base_path: 后面的内容修改为 WebUI 的根目录路径后保存文件，如图 1-13 所示。关闭进程窗口，然后重新运行 run_nvidia_gpu.bat 文件，这样就能共享 WebUI 中的模型。

2 通过“绘世”启动器安装

下载 B 站用户“秋葉 aaaki”制作的“绘世”启动器，解压缩后运行启动器，单击右下角的“一键启动”按钮，如图 1-14 所示。等进程加载完成后，启动器将自动打开浏览器和 ComfyUI 界面。

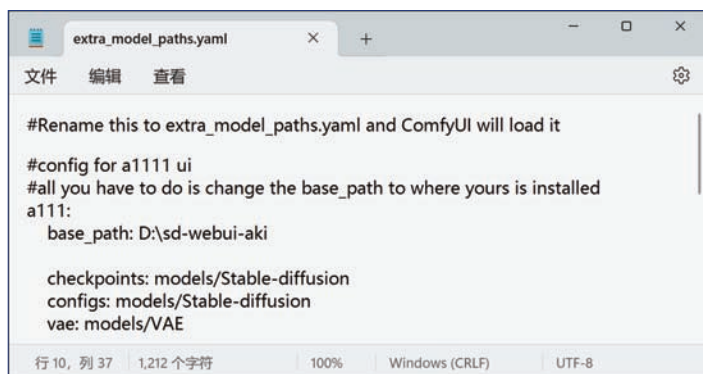


图 1-13



图 1-14

对于大多数用户来说，安装和部署 Stable Diffusion 的系统环境是一件很复杂的事情。虽然现在可以一键运行 ComfyUI，但未来可能还需要安装各种各样的自定义节点，而这些自定义节点往往需要自己的环境依赖，各种依赖的安装配置和相互冲突问题非常令人抓狂。

在这里建议，不喜欢频繁调整系统的用户最好使用“绘世”启动器来运行 ComfyUI。原因是“绘世”启动器把很多底层选项和环境组件集成到了一起，当计算机的系统环境出现问题时，只需依次单击“高级选项”按钮→“环境维护”按钮，就能重新配置相应的组件，如图 1-15 所示。



图 1-15

更重要的是，由于网络限制，使用原生 ComfyUI 时，很多自定义节点所需的模型和依赖都需要手动安装。一些自定义节点的安装配置可能会把普通用户折磨到放弃。相比之下，“绘世”启动器已经集成了很多重要的自定义节点，一部分模型和依赖还可以通过镜像网站实现自动下载，仅这一点就能节省大量时间和精力，如图 1-16 所示。

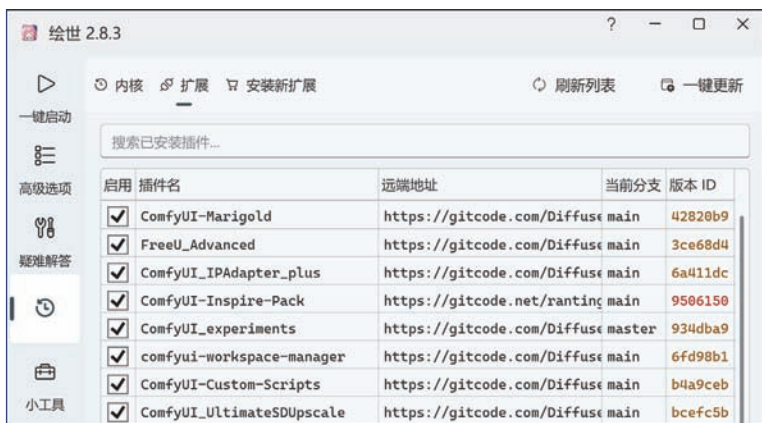


图 1-16

本书使用的是“绘世”启动器 ComfyUI-aki-v1.3 安装包。为了兼顾使用原生 ComfyUI 的读者，已删除所有集成的自定义节点，从最纯净的状态开始学习。

最后，关于安装方面的问题，需要指出的是，ComfyUI 在启动和运行过程中需要频繁调用各种模型文件，因此硬盘速度对 ComfyUI 的整体性能有很大影响。另外，Stable Diffusion 中的模型种类繁多且体积较大，500GB 的存储空间在不知不觉间就会被填满。因此，建议一开始就把启动器解压到磁盘空间最充裕的固态硬盘中。



1.3 安装管理自定义节点

没有安装自定义节点的 ComfyUI 就像是一部新手机，里面只有实现基本功能的应用（App），要想做更多事情，就需要安装各种各样的应用。ComfyUI 中的自定义节点类似于 WebUI 中的插件，有些用于实现特定功能，有些则用于增强界面的外观选项。本节将介绍如何安装两个必备的自定义节点：一个是自定义节点管理器，相当于手机中的应用商店；另一个用于汉化界面。

所需自定义节点	ComfyUI-Manager AIGODLIKE-COMFYUI-TRANSLATION
---------	--

安装自定义节点的方法有很多种，第一种方法是在“绘世”启动器中依次单击“版本管理”按钮→“安装新扩展”按钮，然后单击自定义节点名称右侧的“安装”按钮，如图 1-17 所示。



图 1-17

第二种方法是，原生 ComfyUI 的用户登录 <https://github.com/ltldrdata/ComfyUI-Manager>，单击 Code 按钮后，然后单击 Download ZIP 下载压缩包，如图 1-18 所示。把解压后的文件夹重命名为 ComfyUI-Manager，然后复制到 ComfyUI 根目录下的 ComfyUI\custom_nodes 文件夹中。

第三种方法是在计算机上安装 Git 软件。在 ComfyUI-Manager 的 Git 仓库页面中，单击绿色的 Code 按钮后，再单击 按钮。进入 ComfyUI 根目录下的 ComfyUI\custom_nodes 文件夹，在文件夹的空白处右击，选择菜单中的 OpenGitBashhere 命令。在打开的窗口中，输入 git clone，然后输入一个空格，在窗口中右击，选择 Paste 命令粘贴地址后按 Enter 键，如图 1-19 所示

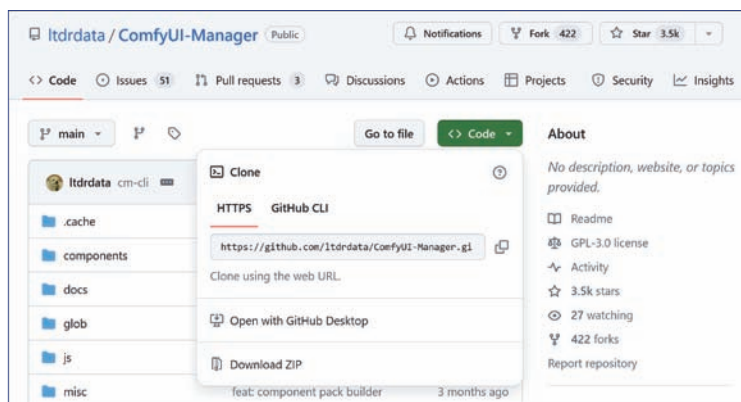


图 1-18



图 1-19

这种方法虽然麻烦，但可以避免很多问题。因为像 ComfyUI-Manager 这样的自定义节点很多，如果不重新命名解压后的文件夹重，即使复制到正确的路径中，也无法正常使用。重新运行 ComfyUI 后，单击设置面板上的 Manager 按钮，在打开的窗口中单击 Update ComfyUI 按钮，即可把 ComfyUI 更新到新版本。单击 Update All 按钮可以更新所有已安装的自定义节点，如图 1-20 所示。

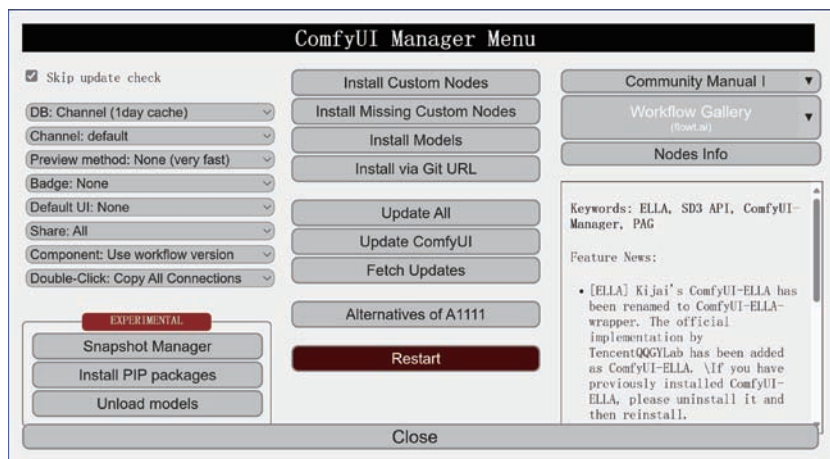


图 1-20

单击 Install Custom Nodes 按钮，在窗口的右上方搜索 ComfyUI-Translation。单击 Search 按钮后，再单击 Install 按钮安装自定义节点。等待安装完成后，单击 RESTART 按钮，如图 1-21 所示。如果“绘世”启动器的控制台中出现退出提示，则需要单击右上角的“一键启动”按钮重新运行 ComfyUI。

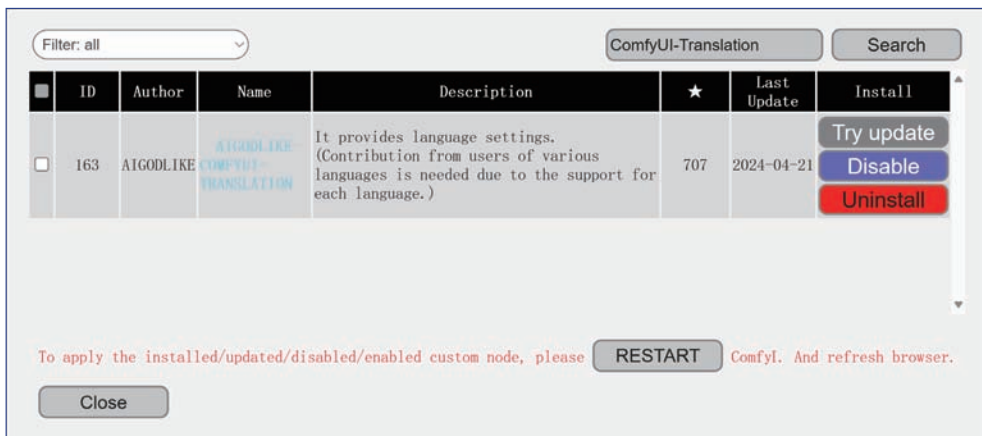


图 1-21

在窗口左上角的下拉菜单中选择“已安装”，即可更新或卸载已安装的自定义节点。如果安装的自定义节点与其他自定义节点存在冲突，会在描述中用黄色底纹标注出来，如图 1-22 所示。当需要查看某个自定义节点的介绍和使用说明时，单击蓝色的自定义节点名称即可直接访问 Git 页面。



图 1-22

导入别人分享的工作流时，如果全部显示为红色节点，则说明 ComfyUI 中缺少自定义

节点，如图 1-23 所示。在设置面板上打开管理器窗口，单击“安装缺失节点”按钮，就能看到需要安装的自定义节点。

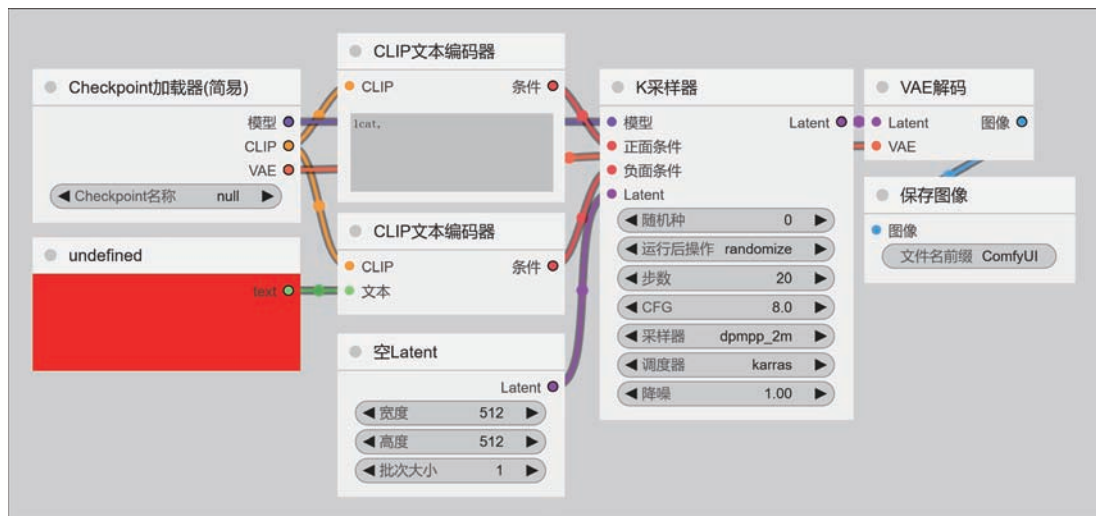


图 1-23

管理器窗口中还提供了两个比较有用的设置选项。在窗口左侧的“标签”下拉菜单中选择“ID+ 名称”，就能在节点的右上角显示所有节点的创建顺序。如果画布上有缺失的自定义节点，还能看到缺失节点的名称。节点右上角有狐狸图标的是 ComfyUI 的自带节点，如图 1-24 所示。

当 workflow 运行到“K 采样器”节点时，节点下方会动态显示采样过程，如图 1-25 所示。在管理器窗口的“预览方法”菜单中，可以选择是否显示采样过程。



图 1-24



图 1-25



1.4 理解 Stable Diffusion 的工作原理

互联网上有无以计数的图片资源，如果把所有图片都下载下来，并为每张图片添加文字注释，标明图片上的内容，就会形成一个庞大的数据库。输入关键词后，就能检索到符合条件的图片。

然而，对于 AI 绘画来说，这种方式不能解决所有问题。假设我们想画一只在太空舱里穿着宇航服的狐狸，但发现数据库里没有这样的图片，该怎么办？解决办法是分别调出太空舱、宇航服和狐狸的图片，把这三张图片的特征混合到一起，然后重新画一张，如图 1-26 所示。

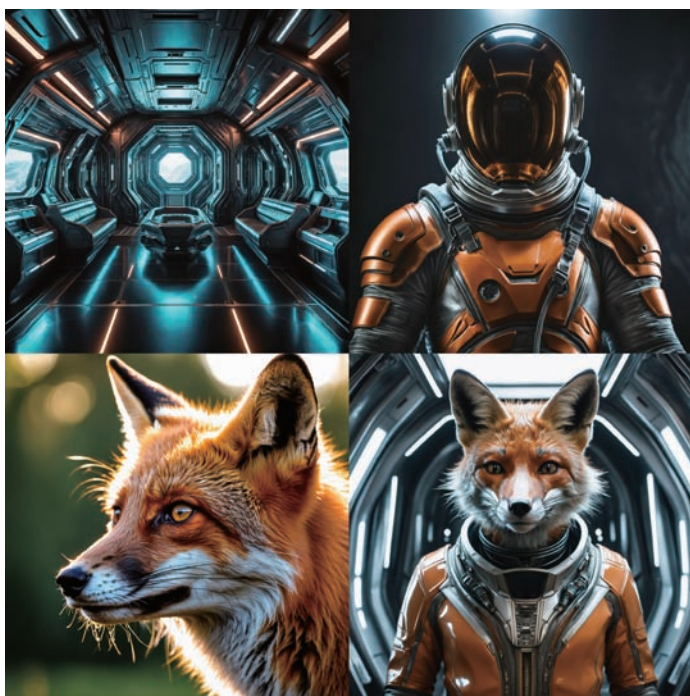


图 1-26

要想用文字生成图片，需要解决三个问题：第一个问题是如何在家用级别的计算机中装下几十亿甚至上百亿张图片；第二个问题是图片由大量像素点组成，每个像素又有红绿蓝三个分量，如果逐个像素进行计算，仅一张 512×512 像素的图片就需要计算 78 万次，这样庞大的算力资源从何而来；第三个问题是如何让程序理解文字，并生成符合语义描述的画面内容。

为了解决以上问题，Stable Diffusion 提供了三个核心组件。第一个组件被称为自动编解码器，这个组件分成编码器和解码器两个部分。编码器的作用是压缩图片的尺寸和颜色信息，同时在图片上逐步添加噪声，直到将图片处理成肉眼无法识别，仅程序才能读懂的向量特征。经过 48 倍的压缩后，图片的体积变得非常小，从而解决存储问题，如图 1-27 所示。



图 1-27

如果在压缩图片的同时为每张图片添加描述画面内容的语义标签，这个过程就是训练模型。把训练得到的所有语义标签和图像特征代码整合到一个文件包中，这个文件包就是大模型。大模型是预先训练好的，其中包含了能生成“世间万物”的图像特征。在 ComfyUI 的默认工作流中，只需在“Checkpoint 加载器”节点中选择一个大模型，就能从中调取生成图片所需的图像向量特征，如图 1-28 所示。

第二个组件为 CLIP 文本编码器。这个编码器从互联网上抓取了 4 亿张图片以及描述图片内容的文本，分别提取文本和图像的特征后进行对比学习。文本和图像特征匹配的作为正样本，不匹配的作为负样本。经过不断地重复训练，文本和图像之间的对应关系就建立起来了。



图 1-28

ComfyUI 中有两个“CLIP 文本编码器”节点：一个用于输入正向提示词，也就是希望画面中出现的内容；另一个用于输入反向提示词，也就是不希望画面中出现的内容，如图 1-29 所示。用户在这里输入提示词后，文本编码器会把文字转换成由数字组成的词元向量。这些词元向量一端发送给“Checkpoint 加载器”节点，从大模型中筛选符合条件的图像向量特征，另一端发送给“K 采样器”节点。

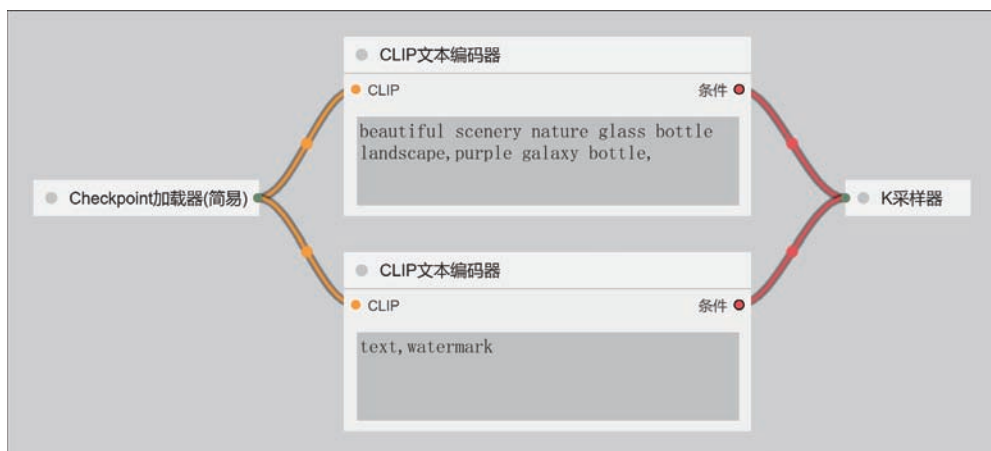


图 1-29

第三个组件是扩散模型，也就是 ComfyUI 中的“K 采样器”节点。扩散模型接收到文本编码器发来的词元向量和模型发来的图像特征向量后，从“空 Latent”节点中调取一张随机生成的噪声图（或称为噪波图），然后在词元向量和图像特征向量的指导下，逐步去除噪声，如图 1-30 所示。因为去除噪声的过程是在图片经过高度压缩后的“潜空间”中进行的，需要处理的像素数量很少，所以家用级显卡也能满足这种算力需求。

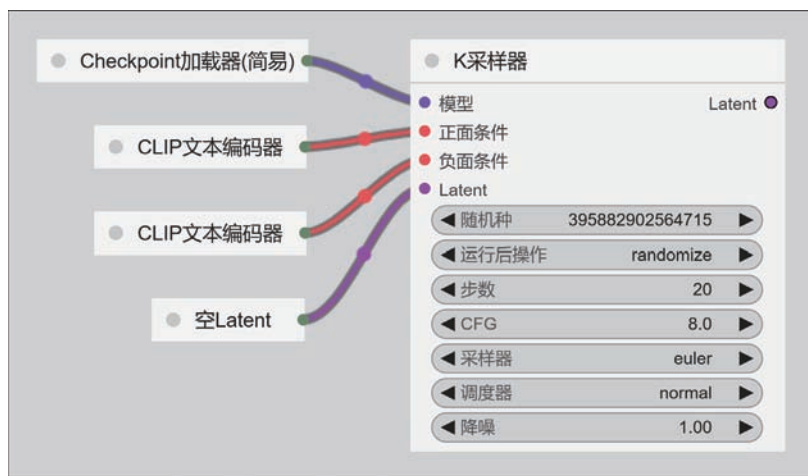


图 1-30

接下来，“VAE 解码”节点进行一次反向扩散处理，把潜空间中的生成结果升维放大到像素空间，转换为我们可以识别的图片。最后，利用“保存图像”节点把图片保存到硬盘上，如图 1-31 所示。

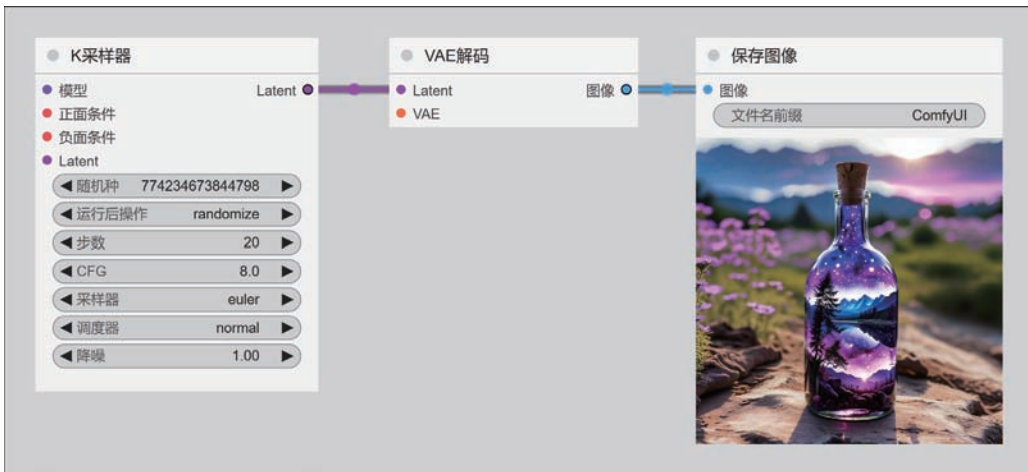


图 1-31

1.5 ComfyUI 的基本操作



在大致了解了 Stable Diffusion 的工作原理和几个主要节点的作用后，本节将从零开始还原一次默认 workflow，让读者熟悉 ComfyUI 中的基本操作。

单击设置面板上的“清除”按钮，然后在弹出的窗口中单击“确定”按钮删除所有节点。接着，在画布上右击，在弹出的菜单中依次选择“新建节点”→“加载器”→“Checkpoint 加载器（简易）”命令。继续在画布的空白处双击，打开搜索窗口，通过搜索名称的方式添加“K 采样器”节点，如图 1-32 所示。



图 1-32

节点上有颜色的圆点称为端口，左侧的输入端口用来接收上一个节点发来的信息，右侧的输出端口负责把处理后的信息发送给下一个节点。不同颜色的端口接收和发送的信息类型也不同，如图 1-33 所示。

在任意一个节点上右击，在弹出的菜单中可以修改节点的标题名称、颜色和形状。单击标题名称前面圆点，可以折叠或展开节点。把光标移到节点的右下角，按住鼠标拖动可以调整节点的大小，如图 1-34 所示。



图 1-33



图 1-34

通过鼠标左键按住一个输出端口，把光标移动到另一个颜色相同的输入端口上，然后松开鼠标即可在两个端口之间建立连线。连线的中间位置有一个圆点，单击这个圆点后再单击“删除”命令就能断开连接，如图 1-35 所示。当连线中间的圆点被节点遮挡时，可以用鼠标右击输出端口，在弹出的菜单中选择 Disconnect Links 命令来断开连接。



图 1-35

通过鼠标左键按住“K 采样器”节点上的 Latent 端口，拖出连线后松开鼠标，在弹出的菜单中单击“VAE 解码”，就能创建与之连接的节点，如图 1-36 所示。

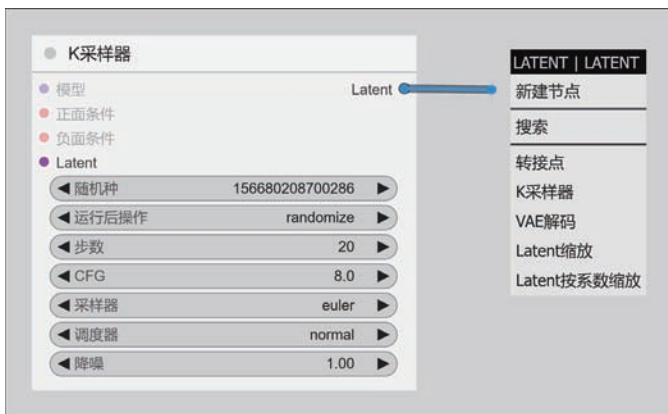


图 1-36

在“Checkpoint 加载器”节点的 CLIP 端口上拖出连线，松开鼠标后选择“CLIP 文本编码器”命令。选中新建的节点后按 Ctrl+C 键进行复制，然后按 Ctrl+Shift+V 键粘贴带有连线的节点。按住 Alt 键后拖动节点，可以复制不带连线的节点。接下来，把两个“CLIP 文本编码器”节点的“条件”端口分别与“K 采样器”节点的“正面条件”和“负面条件”端口连接起来，如图 1-37 所示。

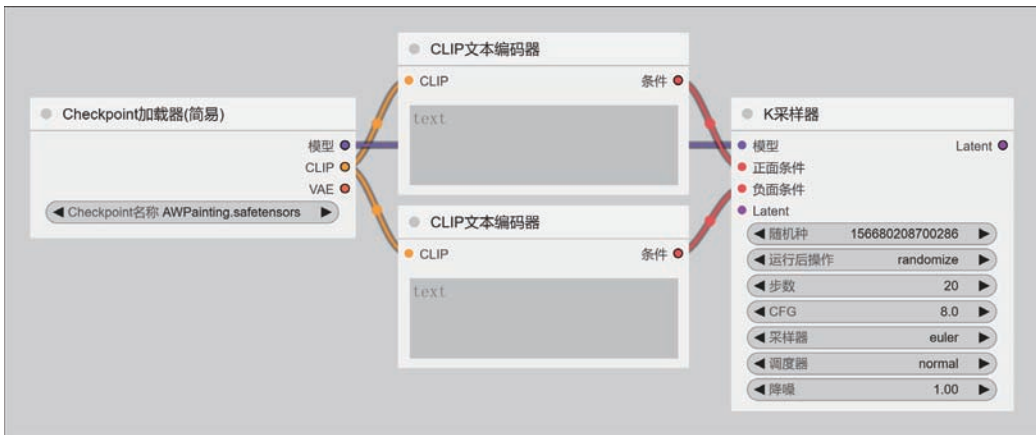


图 1-37

继续通过“K 采样器”节点的 Latent 端口创建“空 Latent”节点，通过“VEA 解码”节点的“图像”端口创建“保存图像”节点。单击设置面板上的“添加提示词队列”按钮，如果 workflow 上有尚未连接的端口，系统会用红色边框标注通路中断的节点，用红色圆圈标注未连接的端口，如图 1-38 所示。



图 1-38

把“Checkpoint 加载器（简易）”节点的 VAE 端口连接到“VEA 解码”节点，默认的文生图工作流就搭建完成了，如图 1-39 所示。按住 Ctrl 键后，可以用框选的方式同时选中多个节点，按住 Shift 键后，可以同时移动选中的节点。



图 1-39

在画布的空白处右击，在弹出的菜单中选择“新建分组”命令。拖动组节点的右下角调整大小，继续拖动组节点的标题栏，与之相交的所有节点就能一起移动，如图 1-40 所示。

单击设置面板上的“添加提示词队列”按钮，节点边框会从左至右依次变成绿色，表示整个工作流的运行顺序和进度。

在“空 Latent”节点上，“批次大小”参数用来设置同时生成的图片数量。这个参数对显卡的要求比较高，需要生成多张图片时，我们可以多次单击“添加提示词队列”按钮，或者勾选“更多选项”复选框后，设置“批次数”参数，以便一张接一张地生成图片。单击设置面板上的“显示队列”按钮，可以查看或者取消正在排队运行的流程，如图 1-41 所示。



图 1-40

生成多张图片后，按快捷键 H，或者单击设置面板上的“显示历史”按钮，在展开的列表中可以查看生成过的图片，如图 1-42 所示。所有生成结果都被保存在“绘世”启动器根目录下的 output 文件夹中。



图 1-41



图 1-42

1.6 模型的分类和调用



模型是一个非常宽泛的概念。Stable Diffusion 本身就是一种模型，其中包括多个子模型，例如：生成图片的大模型、生成特殊画风和形象的 Lora 模型、捕捉和嵌入图像特征的 Embedding 模型、修补色彩的 VAE 模型等。为了实现引导画面、高清放大、提取遮罩等功能，各种自定义节点也需要使用专门的模型算法。本节将介绍 Stable Diffusion 中一些常用的模型以及调用这些模型的方法。

所需自定义节点

EmbeddingPicker

1 大模型

我们常说的大模型分为官方大模型和 Checkpoint 模型两种。官方大模型使用海量图片和算力资源进行训练，就像百科全书一样，集合了 AI 绘图所需的所有信息。官方大模型包括 SD1.1~1.5、SD2、SDXL 以及最新的 Stable Diffusion 3，目前使用最广泛的是 SD 1.5 和 SDXL 版的大模型。

尽管官方大模型可以生成各种图像，但它们无法呈现所绘对象的所有细节特征。于是，在官方模型的基础上进行微调和扩展，产生了擅长生成特定画风或特殊形象的 Checkpoint 模型。接触过 AI 绘画的读者都知道，大模型是决定画面风格和质量的关键因素。在模型网站上，有成百上千的 Checkpoint 模型，它们各有所长，有的擅长表现写实人物，有的擅长生成建筑景观，还有的擅长生成卡通形象。使用相同的提示词和设置参数，不同的大模型会产生截然不同的生成结果，如图 1-43 所示。

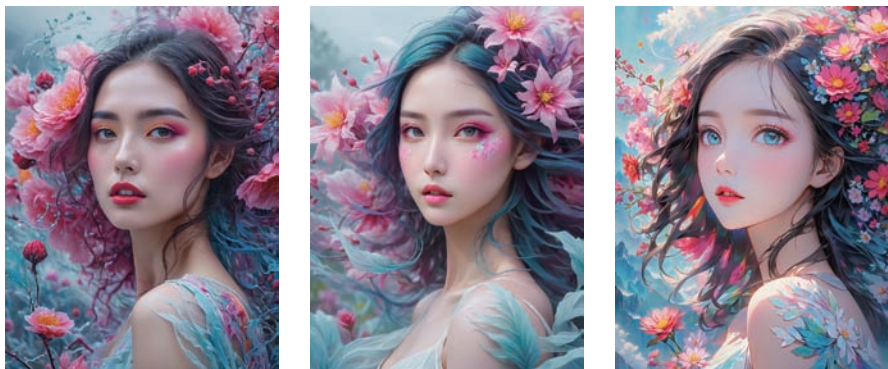


图 1-43

2 Lora 模型

Lora 模型能够把训练参数插入大模型的神经网络中，以较少的训练参数和算力就能把某种人物特征或风格固定下来。大模型的体积较大，SD 1.5 版大模型的体积大致为 2GB，SDXL 版大模型的体积在 6GB 左右。相对而言，Lora 模型的体积只有大模型的十分之一，这不仅能让硬盘装下更多风格的模型，还可以用多个 Lora 模型共同控制生成结果的风格。

由于训练难度低、效果优异且模型体积小，因此 Lora 模型成为目前最热门且广泛应用的模型。使用 Lora 模型生成图的效果如图 1-44 所示。Lora 模型的安装路径是“绘世”启动器根目录下的 models\loras 文件夹。

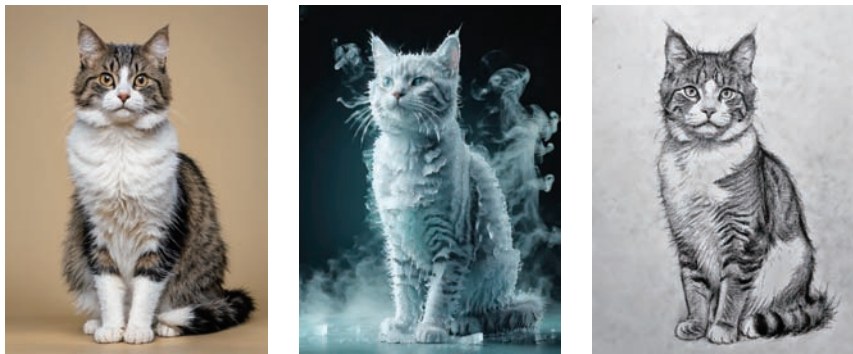


图 1-44

Lora 模型不能单独使用，必须挂载到大模型上。调用 Lora 模型的方法是在画布的空白处右击，然后依次选择“新建节点”→“加载器”→“LoRA 加载器”命令来创建节点。把“LoRA 加载器”节点的“模型”端口连接到“K 采样器”节点，把 CLIP 端口连接到两个“CLIP 文本编码器”节点，如图 1-45 所示。

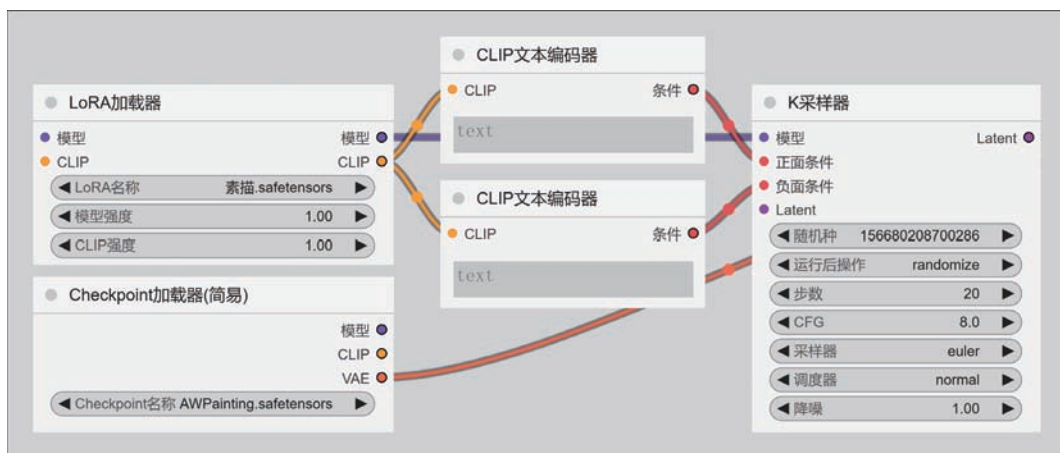


图 1-45

接下来，把“Checkpoint 加载器（简易）”节点的“模型”端口和 CLIP 端口连接到“LoRA 加载器”节点，这样 Lora 模型的文生图 workflow 就搭建完成了，如图 1-46 所示。



图 1-46

我们还可以在“Checkpoint 加载器”和“LoRA 加载器”节点之间继续串联 LoRA 加载器，以使用多个 Lora 模型共同微调大模型，如图 1-47 所示。



图 1-47

在“LoRA 加载器”节点中，“模型强度”参数用于控制 Lora 模型的作用程度，数值越大，模型的特征或画风就越强烈，如图 1-48 所示。“CLIP 强度”一般和“模型强度”设置为相同的数值。至于这两个参数应取多大值，最理想的情况是使用模型下载页面中作者给出的建议参数，或者先把数值设置成常用的 0.8，然后根据生成结果的表现进行调整。

使用 Lora 模型时还要注意三个问题。首先，Lora 模型和大模型需要使用相同的版本，要么都使用 SDXL，要么都使用 Stable Diffusion 1.5。其次，Lora 模型是基于某些特定的大模型训练出来的，虽然有一定的适配性，但使用配套的大模型能产生更好的效果。再次，有些 Lora 模型提供了触发词。这些触发词需要在正向提示词中填写，虽然不填写也能让 Lora 模型生效，但无法得到最佳效果。填写提示词前后的对比效果如图 1-49 所示。



模型强度 = 0.6

模型强度 = 0.9

图 1-48



图 1-49

3 Embedding 模型

Embedding 模型，也称为 Textual Inversion 模型，是一种在训练时只处理语义标签的模型，使 AI 更精确地理解某些词组的含义。可以把 Embedding 模型理解为一组封装好的提示词：用作正向提示词时，可以生成指定角色的特征或画风；用作反向提示词时，可以避免色彩、肢体等画面元素出现错误。由于这种模型只起到标记作用，本身不包含信息，因此体积只有几十千字节。WebUI 的安装路径是根目录下的 embeddings 文件夹，而 ComfyUI 的安装路径是根目录下的 models\embeddings 文件夹。

调用 Embedding 模型的方法是在提示词中输入模型的文件名。例如，如果我们下载了一个可以生成游戏人物形象的 Embedding 模型，可以把文件重新命名为 D.va 并安装到正确的路径中。因为我们希望生成这个人物，所以在正向提示词中输入 embedding:D.va。为了避免出现肢体错误和不良构图，我们可以使用修复类模型，如 EasyNegativeV2、badhandv4、ng_deepnegative_v1_75t 等，并在反向提示词中输入，如图 1-50 所示。

这种添加模型的方式很不友好，我们可以安装自定义节点 Embedding Picker。需要添加 Embedding 模型时，只需在“CLIP 文本编码器”节点上右击，选择 Prepend Embedding Picker 命令，然后在新建的节点中选择模型。提示词也应在新建的节点中输入，如图 1-51 所示。



图 1-50

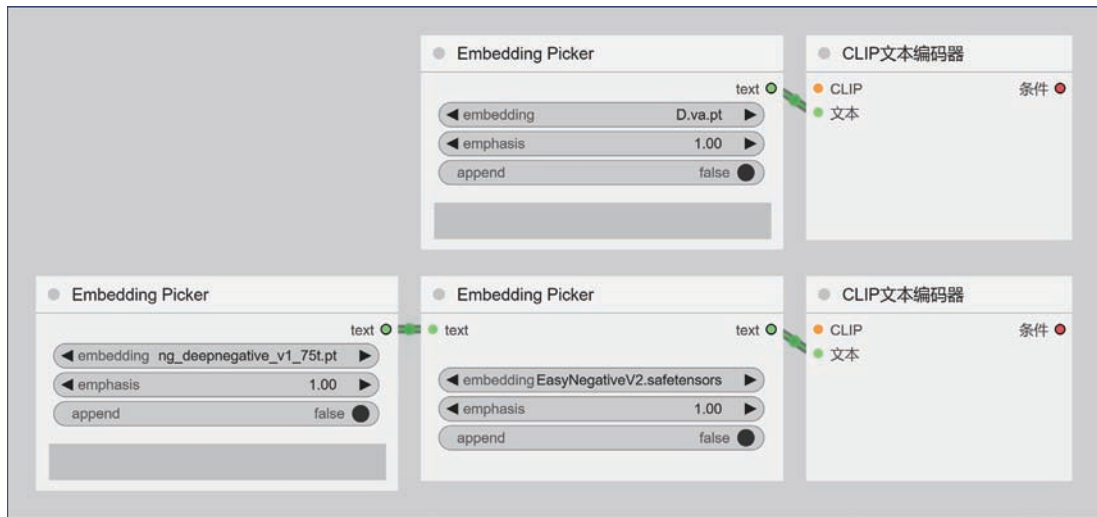


图 1-51

现在，无须输入任何提示词，只需按 Ctrl+Enter 键运行 workflow，即可生成 D.va 形象的角色，如图 1-52 所示。

4 VAE 模型

VAE 模型就像滤镜一样，起到修复色彩和细节微调的作用。VAE 模型通常集成在大模型中，当 workflow 进行到最后一步时，通过 VAE 端口发送给“VAE 解码”节点。有些大模型由于 VAE 文件损坏，生成的图片会出现灰暗、颜色不鲜明的现象，这时需要使用外挂的 VAE 模型进行修复。



图 1-52

使用外挂 VAE 模型的方法是在画布的空白处右击，依次选择“新建节点”→“加载器”→“VAE 加载器”命令创建节点，然后把 VAE 端口与“VAE 解码”节点连接起来，并在新建的节点中加载模型文件，如图 1-53 所示。

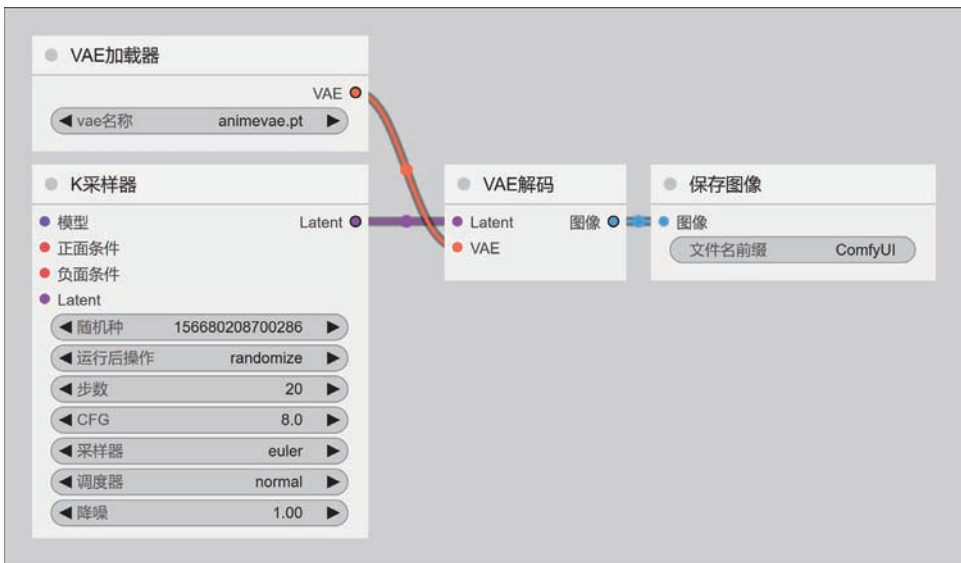


图 1-53

1.7 打造更舒适的界面



ComfyUI 属于第一眼看上去想放弃，用过一段时间后又离不开的界面。对于初学者来说，最大的挑战是入门阶段陡峭的学习曲线。好在很多开发者已经对 ComfyUI 的设置选项和常用操作进行了各种优化，特别是本节将介绍的三个自定义节点，已经成为大多数 ComfyUI 用户的必备插件。

所需自定义节点	failfast-comfyui-extensions、Crystools和ComfyUI-Custom-Scripts
---------	--

开源工具的一大魅力在于它们能够在全球开发者的共同努力下不断完善和优化，ComfyUI 中的许多不便之处都可以通过自定义节点来解决。例如，提示词节点中的字体较小，安装自定义节点 failfast-comfyui-extensions 后，就能在设置窗口中调整文字大小。此外，还可以设置连线宽度、是否显示节点阴影等外观细节的选项，如图 1-54 所示。

对于显存不充裕的用户，可以安装自定义节点 Crystools，这样可以在设置面板上直接

查看 CPU、GPU、显存等系统资源的占用情况。同时，还能看到图片的生成进度和生成图片花费的时间，如图 1-55 所示。

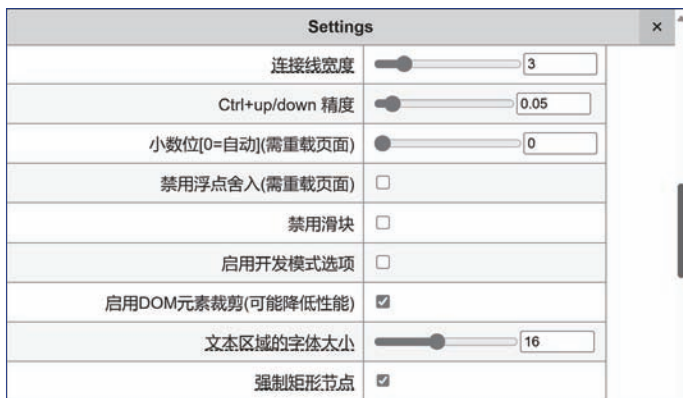


图 1-54

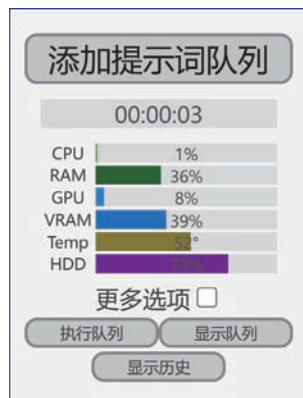


图 1-55

ComfyUI-Custom-Scripts 是一个堪称瑞士军刀的自定义节点，安装完成后，最显著的变化是界面下方多出一个工具条。以前查看生成结果时，需要按 H 键显示历史列表，在加载历史记录前无法看到生成结果的预览图。现在可以在工具条中以缩略图的方式查看所有生成结果，并通过拖动工具条左下角的两个滑块调整缩略图的大小和显示数量，如图 1-56 所示。将缩略图拖到画布的空白处，即可调出生成这张图片的设置参数。

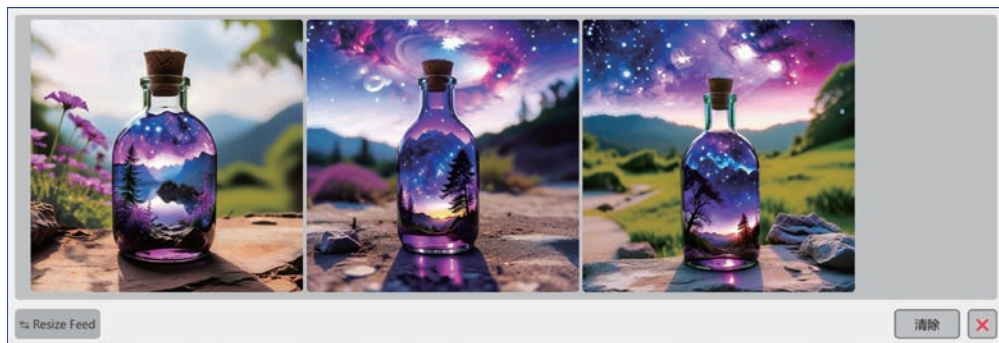


图 1-56

不想显示工具条时，可以先将设置面板最小化，然后单击工具条右下角的 X 按钮。要重新显示工具条，单击设置面板右上方的 按钮。

单击设置面板上的设置按钮，在“图像面板位置”下拉菜单中可以选择工具条的停靠位置。取消“在菜单显示图像”复选框的勾选，可以关闭设置面板上显示的生成结果，如图 1-57 所示。

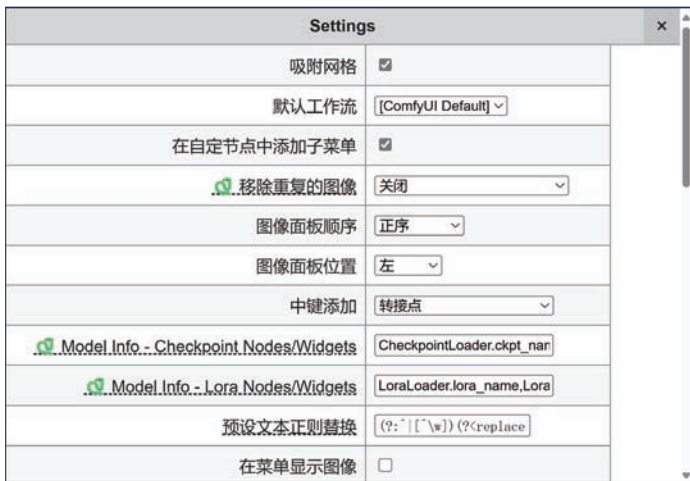


图 1-57

以前添加 Lora 模型时，需要先创建节点，再连接端口。现在，只需在“Checkpoint 加载器”节点上右击，选择“添加 LoRA”命令，即可在创建节点的同时自动连接所有端口。

在不同的节点上右击，还可以创建与之相关的节点和流程。例如，在“K 采样器”节点上右击，选择“添加高清修复”命令，只需重新连接一个端口，就能得到具有高清修复功能的文生图工作流，如图 1-58 所示。

在实际应用中，经常需要添加 Lora 模型和 Embedding 模型。我们在“Checkpoint 加载器”节点上右击，选择“添加 LoRA”命令，然后在反向提示词的“CLIP 文本编码器”节点上右击，再选择 Prepend Embedding Picker 命令。



图 1-58

在“LoRA 加载器”节点上将“模型强度”和“CLIP 强度”参数设置为 0，在 Embedding Picker 节点上将 emphasis 参数设置为 0。在“K 采样器”节点的“采样器”菜单中选择 dpmp2m，在“调度器”菜单中选择 karras，一个更实用的文生图流程就创建完成了，如图 1-59 所示。

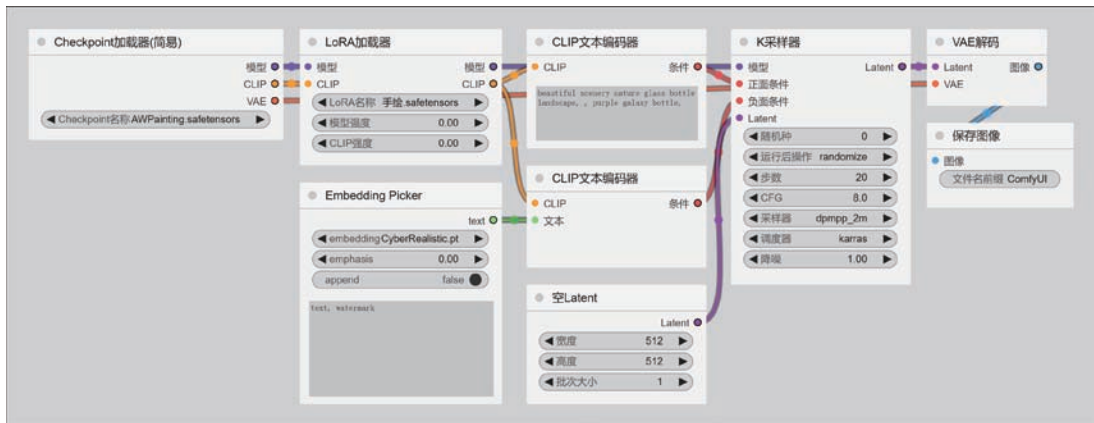


图 1-59

在设置面板上单击“保存”按钮右侧的下拉按钮，选择菜单中的“保存到工作流”命令。在弹出的窗口中输入“默认增强”后，单击“确定”按钮。然后，单击“加载”右侧的下拉按钮，就能看到保存过的工作流列表，如图 1-60 所示。

运行工作流时，ComfyUI 需要把大模型载入显存中，只要不更换大模型，始终会保持载入状态。单击设置面板上的 Unload Models 按钮，可以从显存中释放载入的大模型。

单击设置面板上的“设置”按钮，在“默认工作流”菜单中选择刚刚保存的工作流，如图 1-61 所示。现在单击设置面板上的 Clear 按钮，然后单击“加载默认”按钮，就能载入自定义的工作流。

在设置窗口中单击“文本补全”中的“自定义语句”按钮，在打开的窗口中单击“加载”按钮后，再单击“保存”按钮，如图 1-62 所示。



图 1-60



图 1-61



图 1-62

在“CLIP 文本编码器”节点中输入任意字母，文本补全功能就会弹出包含这个字母的单词列表，如图 1-63 所示。现在，我们又多了一种添加 Embedding 模型的便捷方式：在提示词节点中输入 embed，就会弹出已安装的 Embedding 模型列表，单击带下划线的文本，或者用上下方向键选择模型后按 Enter 键，就能应用模型，如图 1-63 所示。

这个插件还提供了几个实用的节点。在画布的空白处右击，执行“新建节点”→“实用工具”→“播放声音”命令，然后把新建的节点连接到“VAE 解码”节点。这样，图片生成后就会播放提示音，如图 1-64 所示。



图 1-63

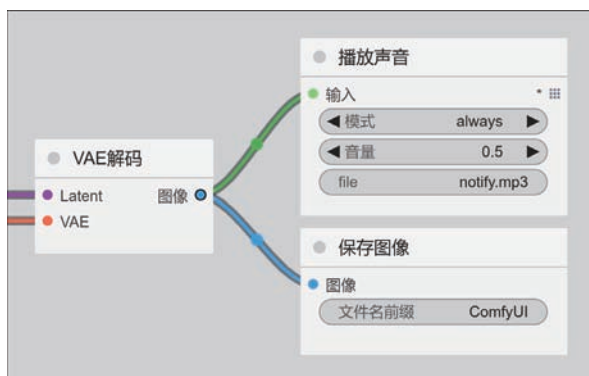


图 1-64