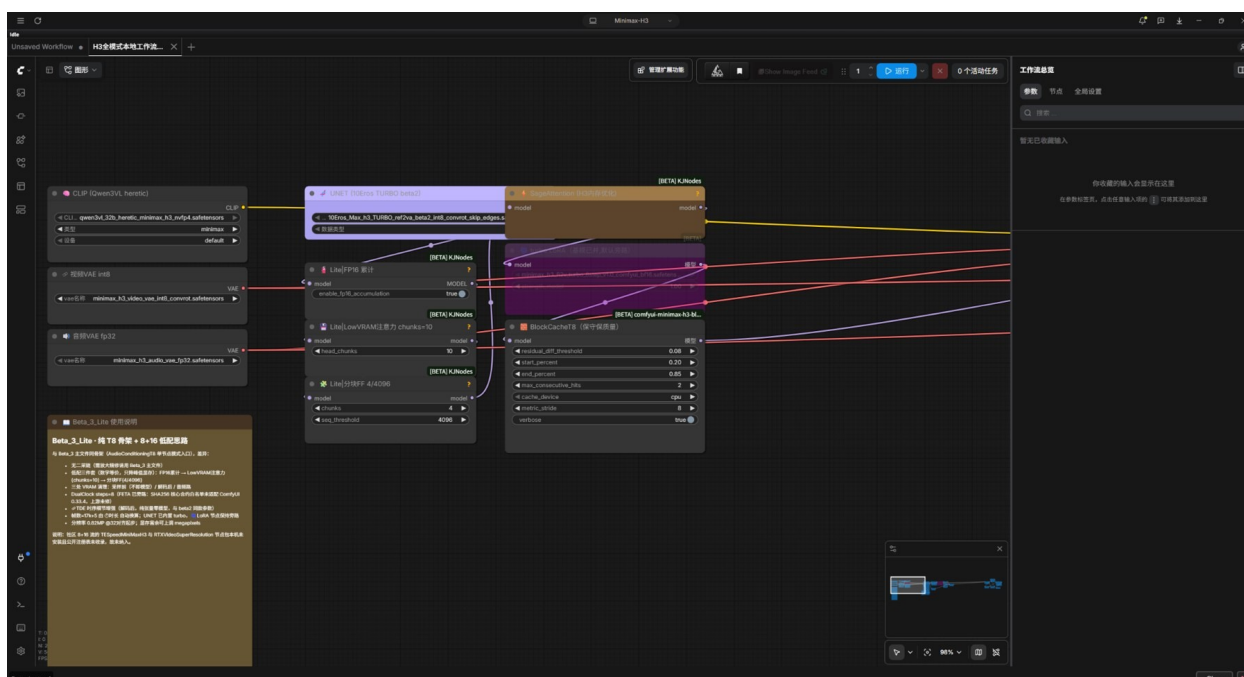


MiniMax H3 × ComfyUI

社区 Beta_3_Lite 低显存 workflows 安装教程

实测定位 Windows + Comfy Desktop 独立实例；面向 8GB NVIDIA 显存。本文按一次真实安装过程整理，优先保证“照着做能起来”。



安装完成后的 Beta 3 Lite 工作流 (实测截图)

版本日期：2026-09-20

适用对象：想在本地 ComfyUI 上跑 MiniMax H3 社区低显存工作流的人

1. 先说结论：这套教程最后会装成什么

最终状态是一套独立的 Minimax-H3 ComfyUI 实例，包含社区 Beta_3_Lite 工作流、低显存节点链、BlockCache，以及约 40GB 的 H3 相关模型文件。

项目	实测/推荐
ComfyUI	Comfy Desktop 独立实例，实测 v0.36.0
Python	实例自带 .venv，实测 3.13.12
GPU	NVIDIA 8GB+；实测 RTX 4070 Laptop 8GB
社区工作流	YixuAn-13/minimax-h3-comfyui-all-in-one → beta3 → Beta_3_Lite
模型体积	实测 38.76 GiB（约 41.6GB 十进制）
目标	先把工作流完整加载并消除 Missing Nodes / Missing Models，再谈参数调优

本教程刻意不做 暂不安装 sol_attn_minimax_v2.py + comfy_kitchen 加速项。先把基础链跑通，之后再叠额外加速，排错更干净。

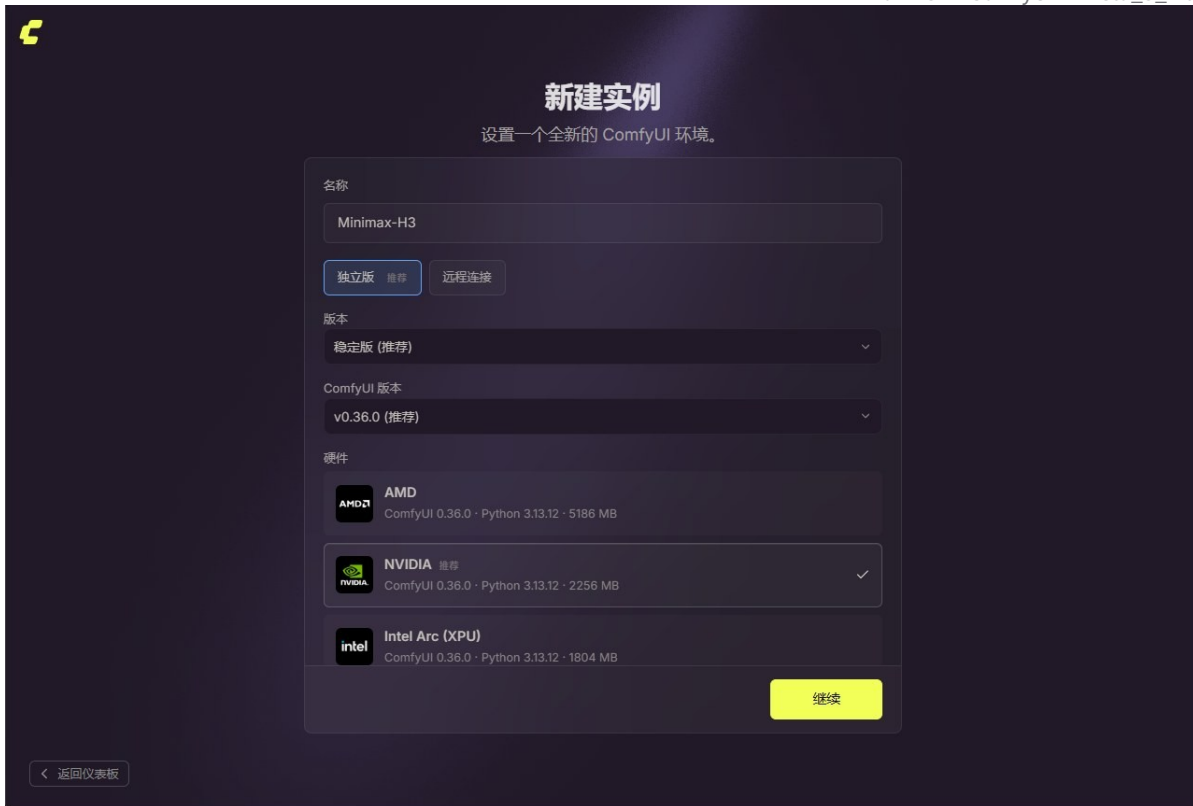
目录结构（本文实测路径）

```
D:\MinimaxH3\minimax-h3-comfyui-all-in-one
D:\Comfy-Desktop\ComfyUI-Installs\Minimax-H3\ComfyUI
├─ custom_nodes
├─ models
├─ user\default\workflows
└─ .venv\Scripts\python.exe
```

如果你的盘符或目录不同 没关系。后续命令统一通过 \$comfy、\$cn、\$py、\$models 变量控制，先改根路径即可。

2. 安装 Comfy Desktop，并创建独立 Minimax-H3 实例

先完成 Comfy Desktop 自身初始化，再新建一个独立实例。独立实例的好处是 H3 的节点、模型和 Python 依赖都隔离，不污染别的 ComfyUI 环境。



新建实例时：独立版 + NVIDIA；实例名可直接叫 Minimax-H3

- 实例名：Minimax-H3
- 硬件：NVIDIA
- ComfyUI：稳定版即可；本文实测 v0.36.0
- 实例位置：尽量放 D 盘等空间充足的位置

如果第一次初始化被中途打断

Comfy Desktop 可能残留半初始化状态。这时完全退出 Desktop，再使用官方 reset 脚本清理 Desktop 的用户状态，然后重新启动并完整走完初始化。

```
cd D:\ComfyUI
iwr -useb https://raw.githubusercontent.com/Comfy-Org/Comfy-Desktop/main/scripts/reset-windows.ps1 -OutFile reset-windows.ps1
powershell -ExecutionPolicy Bypass -File .\reset-windows.ps1
```

3. Clone 社区整合仓库

```
cd D:\MinimaxH3
git clone https://github.com/Yixuan-13/minimax-h3-comfyui-all-in-one.git
```

这个仓库本身体积很小，因为它主要包含 workflow、魔改节点源码、说明和少量 wheel；真正占空间的是后面单独下载模型权重。

4. 新 Terminal 的第一件事：定义路径变量

```
$comfy = "D:\Comfy-Desktop\ComfyUI-Installs\Minimax-H3\ComfyUI"
$cn = "$comfy\custom_nodes"
$py = "$comfy\.venv\Scripts\python.exe"
$wfDir = "$comfy\user\default\workflows"
$models = "$comfy\models"

cd $comfy
```

PowerShell 小坑 如果提示符变成 “>>”，说明你处在续行状态。命令没按预期结束时，先 Ctrl+C 回到正常的 “PS ...>” 再重来。

5. 安装 Beta_3_Lite 所需社区节点

先停止 Minimax-H3 实例，再进入 custom_nodes 执行下面的 clone。

```
cd $cn

git clone --depth 1 https://github.com/T8mars/comfyui-minimax-h3-audio-T8.git minimax-h3-audio-T8
git clone --depth 1 https://github.com/Kosinkadink/ComfyUI-VideoHelperSuite.git
git clone --depth 1 https://github.com/rgthree/rgthree-comfy.git
git clone --depth 1 https://github.com/kijai/ComfyUI-KJNodes.git
git clone --depth 1 https://github.com/LAOGOU-666/Comfyui-Memory_Cleanup.git
git clone --depth 1 https://github.com/pythongosssss/ComfyUI-Custom-Scripts.git
git clone --depth 1 https://github.com/T8mars/comfyui-minimax-h3-blockcache-T8.git
```

复制整合仓库自带的 tokendance 节点

```
Copy-Item `
"D:\MinimaxH3\minimax-h3-comfyui-all-in-one\beta3\custom_nodes_src\comfyui-tokendance-h3" `
"$cn\comfyui-tokendance-h3" `
-Recurse -Force
```

检查节点目录

```
Get-ChildItem $cn -Directory | Select-Object Name
```

至少应看到：

- minimax-h3-audio-T8
- ComfyUI-VideoHelperSuite
- rgthree-comfy
- ComfyUI-KJNodes

- Comfyui-Memory_Cleanup
- ComfyUI-Custom-Scripts
- comfyui-minimax-h3-blockcache-T8
- comfyui-tokendance-h3

6. 用实例自己的 Python 安装节点依赖

关键原则 不要用 PowerShell 里 Conda/base 的 Python。一定使用 Minimax-H3 实例的 .venv\Scripts\python.exe。

```
$nodes = @(
    "minimax-h3-audio-T8",
    "ComfyUI-VideoHelperSuite",
    "rgthree-comfy",
    "ComfyUI-KJNodes",
    "Comfyui-Memory_Cleanup",
    "ComfyUI-Custom-Scripts",
    "comfyui-minimax-h3-blockcache-T8",
    "comfyui-tokendance-h3"
)

foreach ($node in $nodes) {
    $req = "$cn\$node\requirements.txt"
    if (Test-Path $req) {
        Write-Host "===== $node ====="
        & $py -m pip install -r $req
        if ($LASTEXITCODE -ne 0) { throw "依赖安装失败: $node" }
    } else {
        Write-Host "[无 requirements] $node"
    }
}
```

实测中，ComfyUI-Custom-Scripts 与 comfyui-tokendance-h3 没有 requirements.txt 时属于正常情况；脚本会自动跳过。

7. 复制并加载 Beta_3_Lite 工作流

```
New-Item -ItemType Directory -Force -Path $wfDir | Out-Null

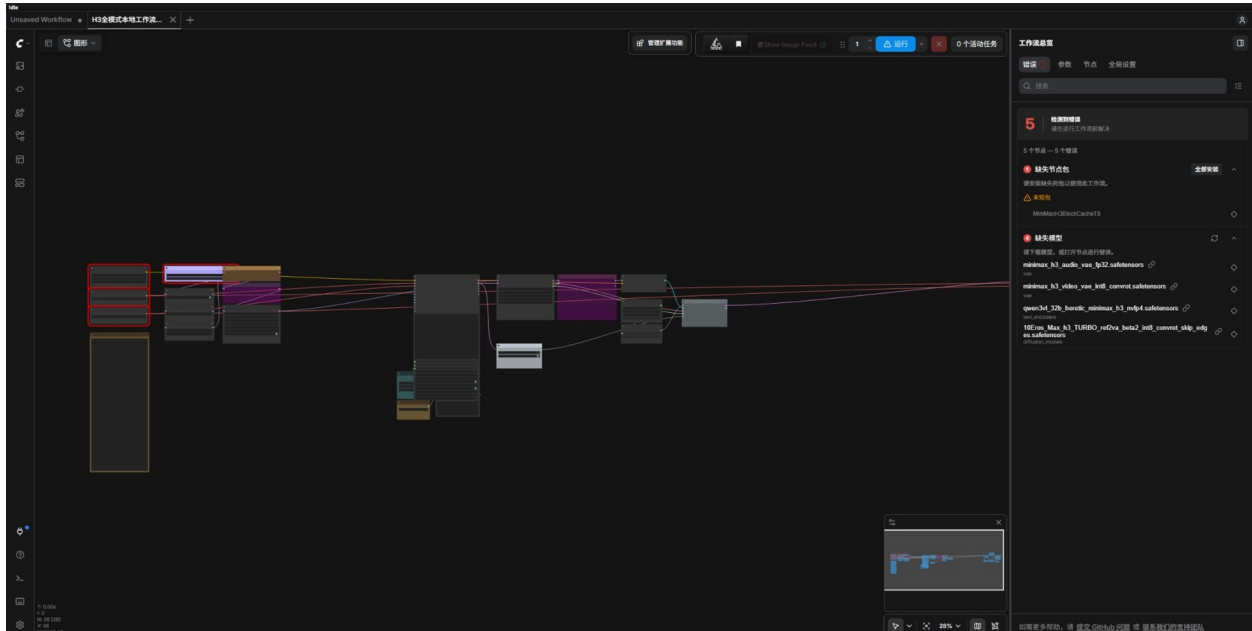
Copy-Item `
"D:\MinimaxH3\minimax-h3-comfyui-all-in-one\beta3\workflows\H3 全模式本地工作流_Beta_3_Lite.json" `
"$wfDir\H3 全模式本地工作流_Beta_3_Lite.json" `
-Force

Get-ChildItem $wfDir
```

提示：默认的 Unsaved Workflow 不是 H3；请从工作流列表打开，或直接拖入 Beta_3_Lite.json。

8. 第一次打开工作流：先解决 Missing Nodes

正确加载 Beta_3_Lite 后，画布会变成一整条 H3 音视频流水线。此时右侧错误面板最有价值：缺什么，就补什么。



实测第一次加载：1 个缺失节点 + 4 个缺失模型

如果提示缺少 MiniMaxH3BlockCacheT8，确认已经执行：

```
git clone --depth 1 https://github.com/T8mars/comfyui-minimax-h3-blockcache-T8.git
```

BlockCache 节点没有额外 Python 依赖，clone 后完全重启实例即可。

9. 下载 4 个模型文件（国内镜像 + 断点续传）

为什么不用 hf CLI 走镜像 实测新版 huggingface_hub + hf-mirror 会遇到 X-Repo-Commit 头不兼容。对大文件最省事的方案是直接利用 hf-mirror URL + curl -C - 断点续传。

先确保目录存在

```
New-Item -ItemType Directory -Force -Path "$models\diffusion_models" | Out-Null
New-Item -ItemType Directory -Force -Path "$models\text_encoders" | Out-Null
New-Item -ItemType Directory -Force -Path "$models\vae" | Out-Null
```

9.1 主扩散模型 - 20.97 GiB

```
curl.exe -L --fail `
--retry 50 --retry-delay 3 --retry-all-errors --connect-timeout 20 `
-C - `
"https://hf-mirror.net/cicalooo/10Eros-Max-h3-int8-convrot/resolve/main/
```

```
10Eros_Max_h3_TURBO_ref2va_beta2_int8_convrot_skip_edges.safetensors" `
-o "$models\diffusion_models\10Eros_Max_h3_TURBO_ref2va_beta2_int8_convrot_skip_edges.safetensors"
```

9.2 Qwen3-VL Heretic 文本编码器 - 14.61 GiB

```
curl.exe -L --fail `
--retry 50 --retry-delay 3 --retry-all-errors --connect-timeout 20 `
-C - `
"https://hf-mirror.net/Momoking/Qwen3-VL-32B-Heretic-MiniMax-H3-NVFP4/resolve/main/
qwen3vl_32b_heretic_minimax_h3_nvfp4.safetensors" `
-o "$models\text_encoders\qwen3vl_32b_heretic_minimax_h3_nvfp4.safetensors"
```

9.3 视频 VAE - 2.62 GiB

```
curl.exe -L --fail `
--retry 50 --retry-delay 3 --retry-all-errors --connect-timeout 20 `
-C - `
"https://hf-mirror.net/Comfy-0rg/MiniMax-H3/resolve/main/vae/minimax_h3_video_vae_int8_convrot.safetensors" `
-o "$models\vae\minimax_h3_video_vae_int8_convrot.safetensors"
```

9.4 音频 VAE - 0.56 GiB

```
curl.exe -L --fail `
--retry 50 --retry-delay 3 --retry-all-errors --connect-timeout 20 `
-C - `
"https://hf-mirror.net/Comfy-0rg/MiniMax-H3/resolve/main/vae/minimax_h3_audio_vae_fp32.safetensors" `
-o "$models\vae\minimax_h3_audio_vae_fp32.safetensors"
```

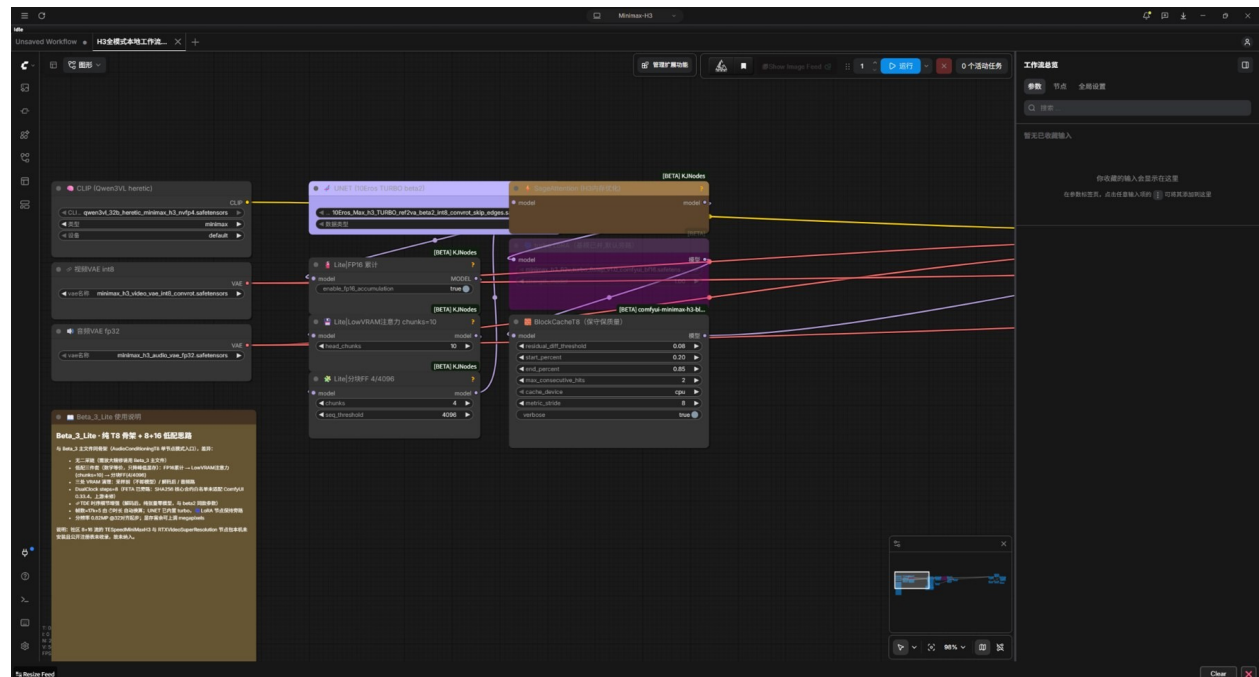
下载后做大小总检查

```
Get-Item `
"$models\diffusion_models\10Eros_Max_h3_TURBO_ref2va_beta2_int8_convrot_skip_edges.safetensors",
"$models\text_encoders\qwen3vl_32b_heretic_minimax_h3_nvfp4.safetensors",
"$models\vae\minimax_h3_video_vae_int8_convrot.safetensors",
"$models\vae\minimax_h3_audio_vae_fp32.safetensors" |
Select-Object Name,@{N="Size_GiB";E={[math]::Round($_.Length/1GB,2)}},Length
```

项目	实测/推荐
10Eros_Max_h3_TURBO_ref2va_beta2_int8_convrot_skip_edges.safetensors	20.97 GiB • 22,513,527,856 bytes
qwen3vl_32b_heretic_minimax_h3_nvfp4.safetensors	14.61 GiB • 15,683,129,587 bytes
minimax_h3_video_vae_int8_convrot.safetensors	2.62 GiB • 2,811,065,184 bytes
minimax_h3_audio_vae_fp32.safetensors	0.56 GiB • 605,254,808 bytes

10. 最终验收：什么叫“装成了”

- 1. 完全停止 Minimax-H3 实例。
- 2. 重新启动实例，让 ComfyUI 重新扫描 custom_nodes 和 models。
- 3. 打开 H3 全模式本地工作流_Beta_3_Lite。
- 4. 确认右侧不再出现 Missing Nodes / Missing Models。
- 5. 确认模型下拉框中已选中四个对应文件。



最终状态：Beta_3_Lite 工作流完整加载，模型和低显存节点链全部就位

到这一步就可以收工 “安装” 阶段完成。提示词、T2VA/I2VA/FL2VA、时长、分辨率、8GB 稳定参数，属于后续 “会用” 阶段，不必在安装时一起解决。

11. 实测踩坑速查

A. curl: (28) Failed to connect / timeout

huggingface.co 直连不稳定时，改用 hf-mirror.net。大文件继续保留 -C -，这样重新执行同一条命令会从已有字节位置续传。

B. curl: (18) end of response with ... bytes missing

镜像服务器提前断流。不要删半截文件，原样重跑带 -C - 的命令即可。实测 22GB 主模型就是这样断点续下来的。

C. 文本编码器镜像返回 401

某些 Hugging Face 仓库通过镜像访问会返回 401。本文最终使用 Momoking/Qwen3-VL-32B-Heretic-MiniMax-H3-NVFP4 的同名文件成功下载。

D. hf.exe 提示缺 X-Repo-Commit

这是新版 huggingface_hub 对 Hub 响应头的兼容性检查，不代表文件不存在。对于本教程的镜像下载，直接改用 curl 即可。

E. Custom Scripts 没有 requirements.txt

正常。clone 到 custom_nodes 后重启即可，不需要额外 pip install。

F. 顶部写着 Minimax-H3，但画布还是普通 UNet / KSampler

Minimax-H3 是实例名，不等于当前打开的 workflow。看到 Unsaved Workflow 时，拖入或打开 Beta_3_Lite.json。

G. 只剩 MiniMaxH3BlockCacheT8 缺失

补 clone T8mars/comfyui-minimax-h3-blockcache-T8，然后完全重启实例。

12. 一页验收清单

- ☐ Comfy Desktop 完整初始化完成
- ☐ Minimax-H3 独立实例可正常启动
- ☐ 社区 all-in-one 仓库已 clone
- ☐ 7 个公开 custom node 仓库已 clone
- ☐ comfyui-tokendance-h3 已复制到 custom_nodes
- ☐ requirements 已用实例 .venv Python 安装
- ☐ Beta_3_Lite.json 已复制/加载
- ☐ MiniMaxH3BlockCacheT8 不再 Missing
- ☐ 4 个模型文件大小与本文实测一致
- ☐ 重启后右侧 Missing Nodes / Models 清空

13. 参考仓库与文件来源

社区整合工作流: <https://github.com/YixuAn-13/minimax-h3-comfyui-all-in-one>

MiniMax H3 Audio T8: <https://github.com/T8mars/comfyui-minimax-h3-audio-T8>

MiniMax H3 Block Cache T8: <https://github.com/T8mars/comfyui-minimax-h3-blockcache-T8>

ComfyUI-VideoHelperSuite: <https://github.com/Kosinkadink/ComfyUI-VideoHelperSuite>

rgthree-comfy: <https://github.com/rgthree/rgthree-comfy>

ComfyUI-KJNodes: <https://github.com/kijai/ComfyUI-KJNodes>

Comfyui-Memory_Cleanup: https://github.com/LAOGOU-666/Comfyui-Memory_Cleanup

ComfyUI-Custom-Scripts: <https://github.com/pythongosssss/ComfyUI-Custom-Scripts>

H3 主模型来源: <https://huggingface.co/cicalooo/10Eros-Max-h3-int8-convrot>

Qwen3-VL Heretic NVFP4: <https://huggingface.co/Momoking/Qwen3-VL-32B-Heretic-MiniMax-H3-NVFP4>

Comfy-Org MiniMax-H3 VAE: <https://huggingface.co/Comfy-Org/MiniMax-H3>

Hugging Face 国内镜像: <https://hf-mirror.net>

最后提醒 社区节点、模型仓库和 ComfyUI 版本会继续更新。若未来节点名或文件名变化，以整合仓库的 beta3/README.md 和工作流实际 Missing 提示为准。