

Apple Silicon Mac 生成AI活 用ガイド

None

asfdrwe

Copyright (c) 2025 asfdrwe, CC BY 4.0

Table of contents

1. Apple Silicon Mac 生成AI活用ガイド	4
1.1 著作権	4
2. 事前準備	5
2.1 catAIToolsのダウンロード	5
2.2 セキュリティ処理	5
2.3 Xcode Command Line Toolsのインストール	7
3. 画像生成	10
3.1 インストールとモデルのダウンロード	10
3.2 文章に基づく画像作成	27
3.3 画像と文章に基づく画像作成	47
3.4 コントロールネットによる生成画像の制御	54
3.5 AIによる画像加工	72
3.6 その他	85
4. 音楽生成	91
4.1 音楽生成AI	91
4.2 推奨モデルのダウンロード	91
4.3 音楽生成	92
4.4 音楽生成用プロンプトの参考サイト	94
5. 動画生成	95
5.1 動画生成AI	95
5.2 推奨モデルのダウンロード	95
5.3 HunyuanVideoでの文章に基づく動画生成	98
5.4 HunyuanVideo での画像に基づく動画生成	102
5.5 Wan2.1 での文章に基づく動画生成	103
5.6 Wan2.1 での画像に基づく動画生成	104
5.7 Wan2.1 での開始画像と終了画像に基づく動画生成	105
5.8 AI活用動画加工	107
6. 音声生成	115
6.1 音声生成AI	115
6.2 Style Bert VITS2 のインストールと起動	115
6.3 文章を読み上げる音声生成	118
6.4 他のモデル	122
6.5 学習について	122
7. 言語AI	123
7.1 言語 AI	123

7.2 koboldcpp のインストールと起動	123
7.3 言語モデルを読み込み起動	126
7.4 Instruct Mode での AI との対話	129
7.5 他のモード	133
7.6 koboldcpp の終了方法	137
7.7 視覚言語モデルによる画像に対する対話	137
7.8 llama.cpp	141
8. あとがき	152

1. Apple Silicon Mac 生成AI活用ガイド

この文書は生成 AI 初心者向けに Apple Silicon Mac で画像生成 AI・音楽生成 AI・動画生成 AI・音声生成 AI と大規模言語モデルと視覚言語モデルによる言語 AI の活用について解説します。

こちらで作成した [catAITools](#) を活用することで、コマンドに不慣れな方でも問題ないように GUI でのインストールや操作を可能としています。また、高速化 LoRA を活用するなど M4 Mac mini で実用性のある速度で生成可能となるよう調整してあります。

使用ソフトは次の 4 つです。

- ComfyUI (画像生成・音楽生成・動画生成)
- Style Bert VITS2 (音声合成)
- koboldcpp (大規模言語モデル活用)
- llama.cpp (大規模言語モデル活用)

すべてオープンソースソフトです。また、この文書で推奨するモデルもフリーのものなので、無料で活用することができます。

動作確認は 2025/4/18 に最小構成の M4 Mac mini (メモリ 16GB、SSD 256GB) と macOS 15.4 Sequoia で行っています。

おそらくそれ以外の Apple Silicon Mac でもメモリを 16GB 以上搭載している機種なら動くと思いますが、メモリが 8GB では動かないかもしれません。

ComfyUI のワークフローは Windows でも Linux でも動くはずなのでそちらでもご活用ください。

1.1 著作権

Copyright (C) 2025 asfdrwe

この文書は [CC BY 4.0](#) でライセンスします。



2. 事前準備

2.1 catAIToolsのダウンロード

Apple Silicon Mac 向けのユーティリティを作成しました。最初に[こちら](#)からダウンロードしてください。ダウンロードした zip ファイルは自動的に展開されてダウンロードフォルダに catAITools フォルダができているはずです。

2.2 セキュリティ処理

現在 macOS ではセキュリティ強化のため、原則 Apple が問題ないか検証したものだけ 実行できるようになっています。

Apple Store からインストールしたもののみ実行可能にしてそれ以外からの実行を禁止できましす、Apple Store 以外から実行できる設定にした場合も、アプリ開発者が Developer Program に登録し、取得した証明書でアプリに署名し、Apple にアプリを提出して検査を受ける [公証](#)という手続きを経ないと、[GateKeeper\(門番\)](#)がプログラムの実行を防止します。

ネットから Safari や Google Chrome でダウンロードすると、ダウンロードした ファイルに自動的に com.apple.quarantine 拡張属性が付与されます。 com.apple.quarantine が付いたファイルやそのファイルに由来するファイルを 実行しようとすると、GateKeeper が自動的に安全性をチェックする仕組みになっています。

署名に必要な Developer Program の参加には費用がかかり、プログラムに対する署名や 公証には色々な作業が必要です。

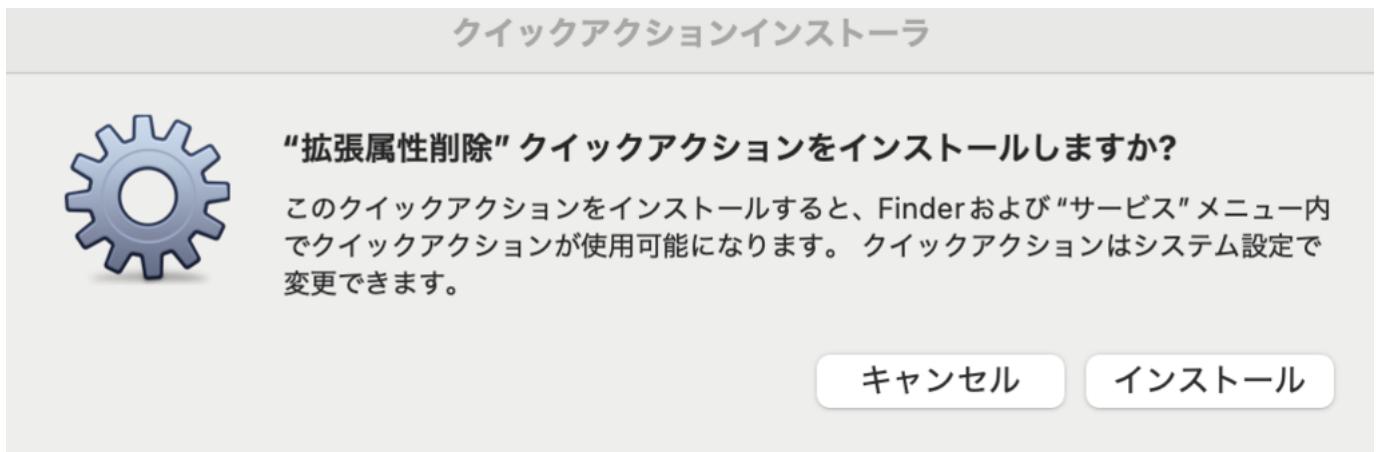
こちらが用意したスクリプトは公証されていないので、そのままでは実行できません。

[こちら](#)を参考に、拡張属性をすべて削除して GateKeeper を回避できるようにするクリックアクションを 用意しました。

Finder でダウンロードフォルダを開き、中にある catAITools フォルダを開いてください。

名前	変更日	サイズ	種類
> ComfyUI用ツール	今日 22:27	--	フォルダ
> data	昨日 14:31	--	フォルダ
> extra	今日 15:45	--	フォルダ
> koboldcpp	今日 17:48	--	フォルダ
LICENSE	2025年4月8日 15:52	1 KB	標準テキスト書類
logo.png	2025年4月9日 19:47	249 KB	PNG画像
> Style-Bert-VITS2	今日 17:48	--	フォルダ
> workflows	今日 22:29	--	フォルダ
XcodeCommandLineTools.install.command	一昨日 16:34	125 バイト	ターミ…クリプト
拡張属性削除.workflow	一昨日 16:20	10 KB	ワークフロー

『拡張属性削除.workflow』をダブルクリックしてください。



インストールを押してインストールしてください。

よく使う項目

- AirDrop
- 最近の項目
- アプリケーション
- デスクトップ
- 書類
- ダウンロード**

場所

- iCloud Drive
- HIDISC SSD
- ネットワーク

タグ

- レッド
- オレンジ

```
ComfyUI-RMGB.install.command
ComfyUI-WanStartEndFramesNative
ComfyUI-ESRGAN.fix.command
ImageModel.download.command
```

名前 サイズ

拡張属性削除.workflow

新規タブで開く

ゴミ箱に入れる

情報を見る

名称変更

“catAITools” を圧縮

複製

エイリアスを作成

クイックルック

コピー

共有...

タグ...

クイックアクション >

フォルダアクション設定...

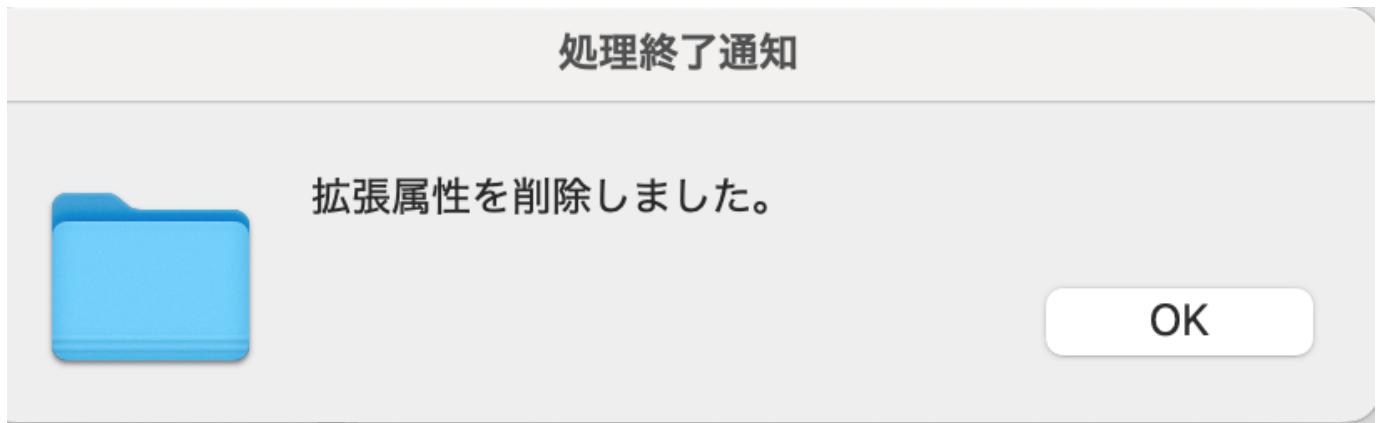
カスタマイズ...

… 拡張属性削除

Finder で適当なファイルやフォルダの上で右クリックするとコンテキストメニューの クイックアクションの中に拡張属性削除というメニューが追加されているはずです。 ダウンロードしたアプリに対してこれを選べば Gatekeeper を無視して実行できるようになります。

ダウンロードフォルダに戻ってください。

catAITools フォルダの上で右クリックしてクリックアクションの 拡張属性削除を選んでください。



これでこちらの用意したユーティリティが動作するようになります。

操作方法がわかっていない場合はこのクリックアクションを使用しなくても 公証されていないものを実行できますが、macOS 15 Sequoia から その操作方法がややこしくなっています。こちらに解説が書かれているのでこれに従って実行可能にしても OK です

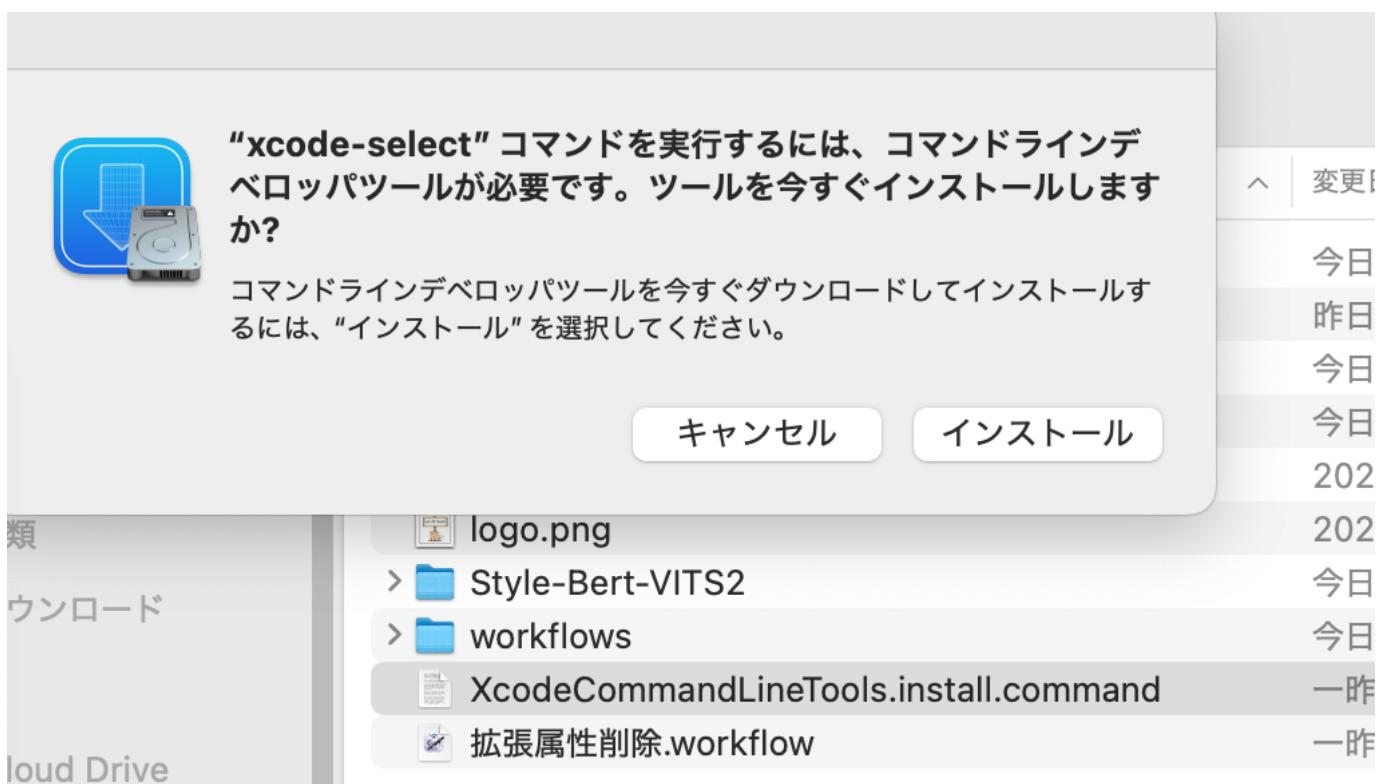
セキュリティ上あまり好ましくないことをやっているので、正しいファイルか 確認の上、気をつけて実行してください。

2.3 Xcode Command Line Toolsのインストール

この文書で解説するプログラムを実行するためには、git や python などさまざまな 開発用ツールが必要となります。Xcode Command Line Tools をインストールすることで 必要なツールがインストールされます。

catAITools フォルダを開き、『XcodeCommandLineTools.install.command』をダブルクリックしてください。

Xcode Command Line Tools がインストールされていない場合、指示に従い利用規約に同意してインストールしてください。



使用許諾契約

Command Line Tools 使用許諾契約

Apple Inc.

macOS SDKおよびXcode契約

Developerソフトウェア（下記で定義されます）を使用される前に、本macOS SDKおよびXcode契約（以下「本契約」といいます）をよくお読みください。お客様は、当該Developerソフトウェアをご使用になることで、本契約の各条項の拘束を受けることに同意されたことになります。お客様が電子的に本Developerソフトウェアにアクセスしている場合、本契約の各条項の拘束を受けることへのお客様の同意は、「同意します」ボタンをクリックすることによって、示されるものとします。本契約の各条項に同意されない場合は、当該Developerソフトウェアを使用せず、「同意しません」ボタンをクリックしてください。

重要な通知：このソフトウェアは、マテリアルを複製することに使用される限りにおいて、著作権の保護を受けないマテリアル、お客様が著作権を有するマテリアル、またはお客様が複製を許諾されたか法的に認められたマテリアルの複製についてのみ、お客様に対してライセンスが付与されるものです。マテリアルの複製権についてご不明な点がありましたら、お客様の法律アドバイザーにご相談ください。

1. 総則

A. ディスク、印刷物または電子文書、読み出し専用メモリ、その他の記録媒体、またはその他あらゆる形式で本契約が添付されているAppleソフトウェア、ツール、ユーティリティ、サンプルまたはエグザンブルコード、文書、インターフェース、コンテンツ、データおよびその他のマテリアル（以下、総称して「Developerソフトウェア」といいます）は、Apple Inc.（以下「Apple」といいます）が、お客様に対して、本契約条件に従う場合に限りライセンスを付与するものであり、販売するものではありません。Appleおよび/またはAppleのライセンサーは、Developerソフトウェア自体の所有権を保有し、お客様に明示的に付与されていない権利のすべてを留保します。本契約の各条項は、オリジナルのDeveloperソフトウェアを置換および/または補完を行うソフトウェアのアップグレードに適用されます。ただし、当該アップグレードが別個のライセンスを伴う場合には、その別個のライセンスが当該アップグレードに適用されます。

B. Developerソフトウェアにより表示されるか、アクセスされるコンテンツの一切の所有権および知的財産権は、それぞれの

[同意しない](#)

[同意する](#)

ソフトウェアをインストール中



残り約4分 49秒

[中止](#)



asfdrwe — XcodeCommandLineTools.install.command — 80x24

```
Last login: Sun Apr 13 23:10:41 on ttys006
asfdrwe@    -M4 ~ % /Users/asfdrwe/Downloads/catAITools/XcodeCommandLineTools
.install.command ; exit;
xcode-select: note: install requested for command line developer tools

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[プロセスが完了しました]
```

インストールが完了したらそのままウィンドウを閉じてください（インストール済みの場合は何も起こらずプログラムが終了するので、そのままウィンドウを閉じてください）。

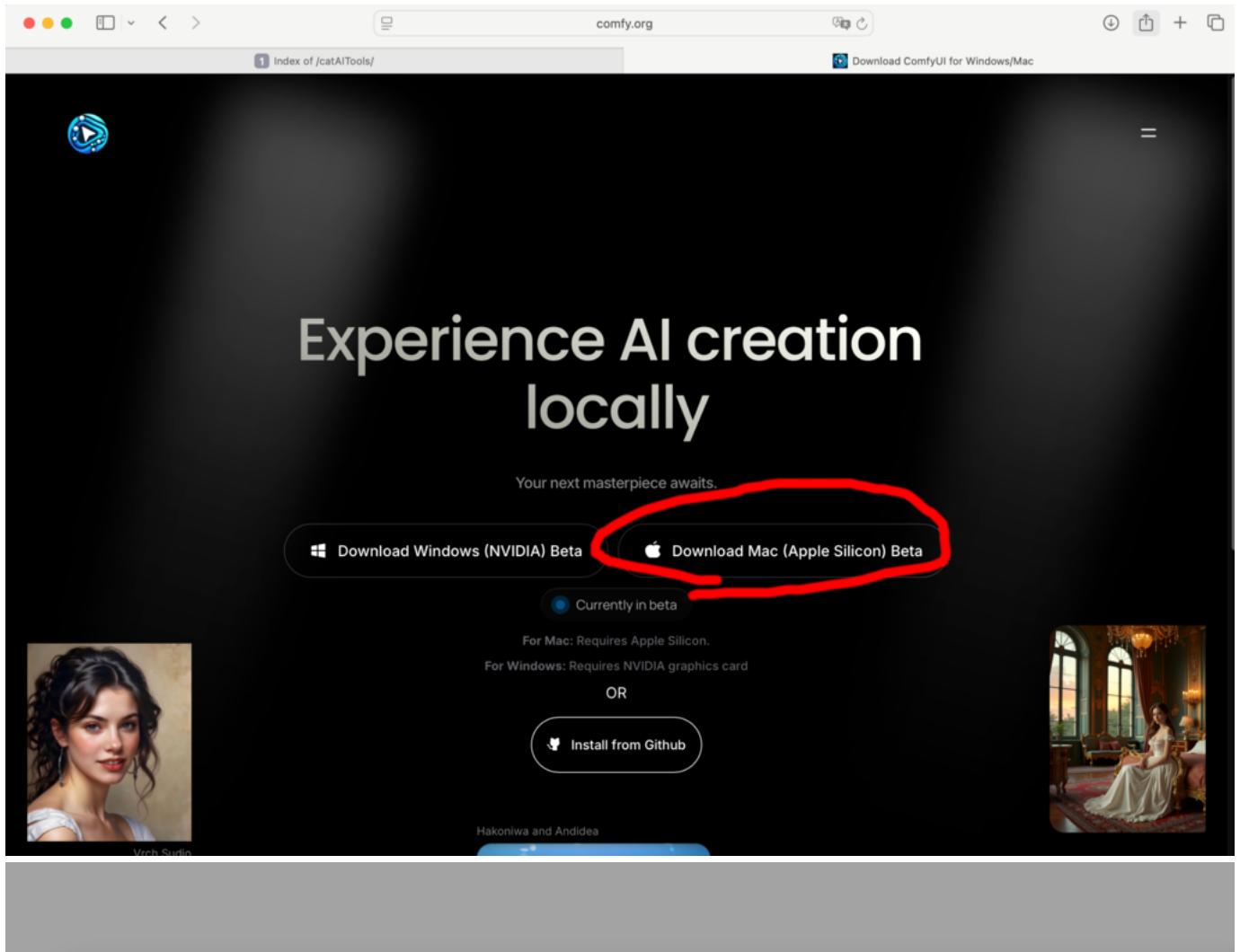
3. 画像生成

3.1 インストールとモデルのダウンロード

ComfyUI のインストール

ComfyUI Desktop を使用して画像生成を行います。

[ComfyUI Desktop](#) からダウンロードしてください。



“comfy.org” と “dl.todesktop.com” からのダウンロードを許可しますか？

Safari 設定の “Web サイト” セクションで、ファイルをダウンロードできる Web サイトを変更できます。

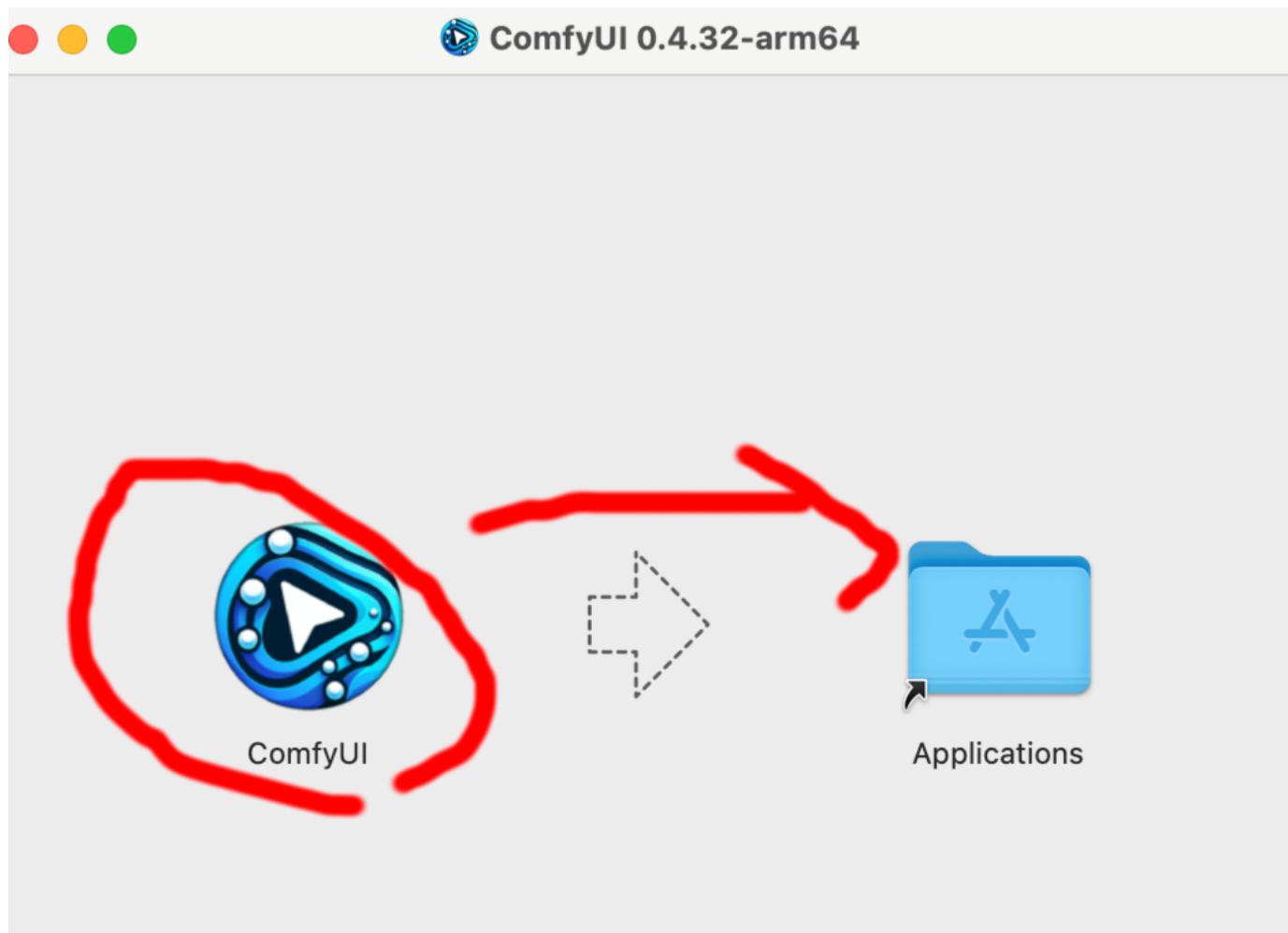
キャンセル

許可

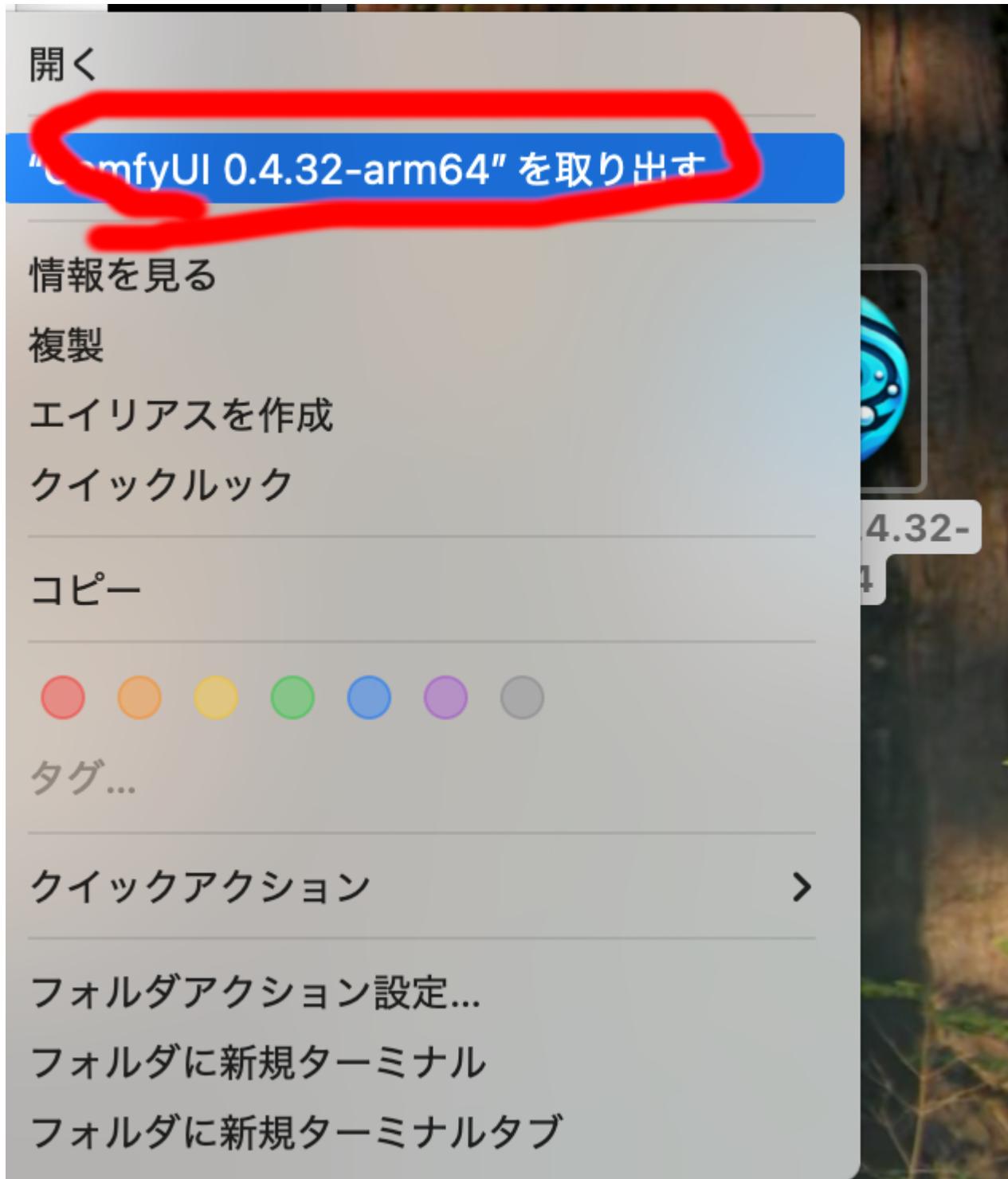
Finder でダウンロードを開いてください。



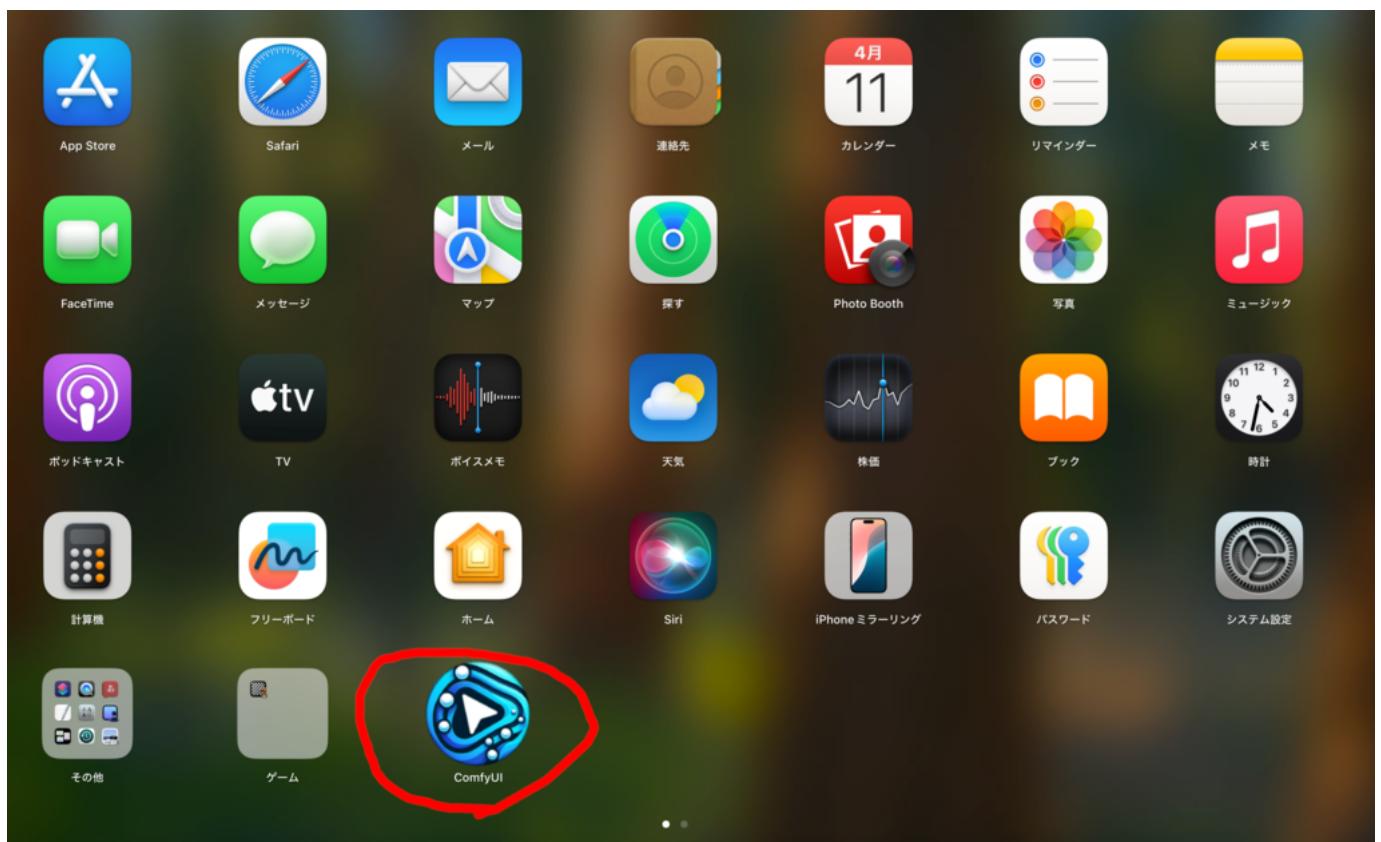
『ComfyUI 0.4.34 - arm64.dmg』(バージョンは異なる場合があります)をダブルクリックしてください。



ComfyUI のアイコンを Applications にドラッグしてインストールしてください。



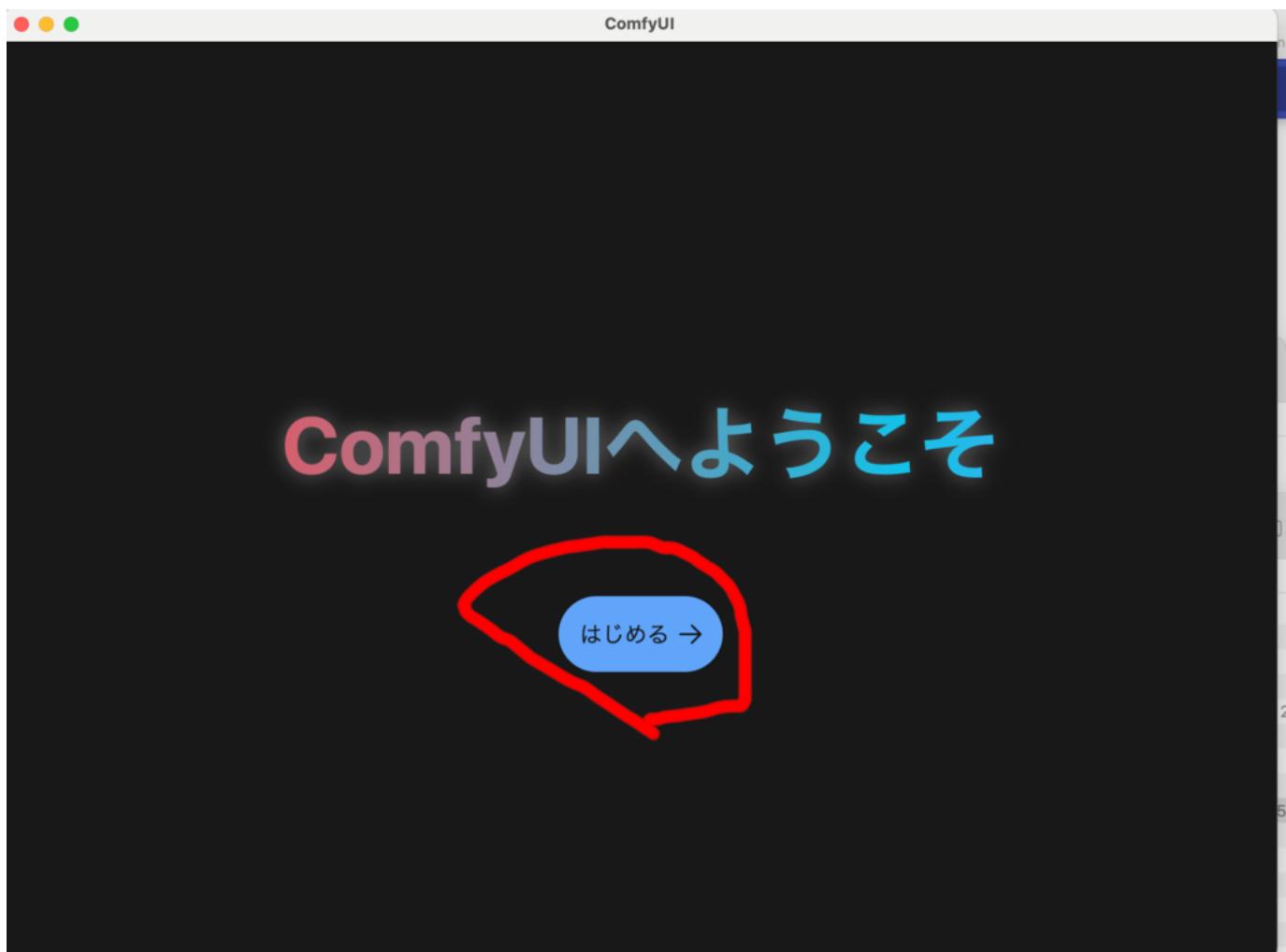
デスクトップの『ComfyUI 0.4.34 - arm64.dmg』を右クリックして "ComfyUI 0.4.34-arm64" を取り出すを押してディスクを取り外し、ダウンロードフォルダの『ComfyUI 0.4.34 - arm64.dmg』はゴミ箱にいれて削除してください。



次に LaunchPad から ComfyUI を起動してください。



初回は警告が出ますが、開くを押してください。

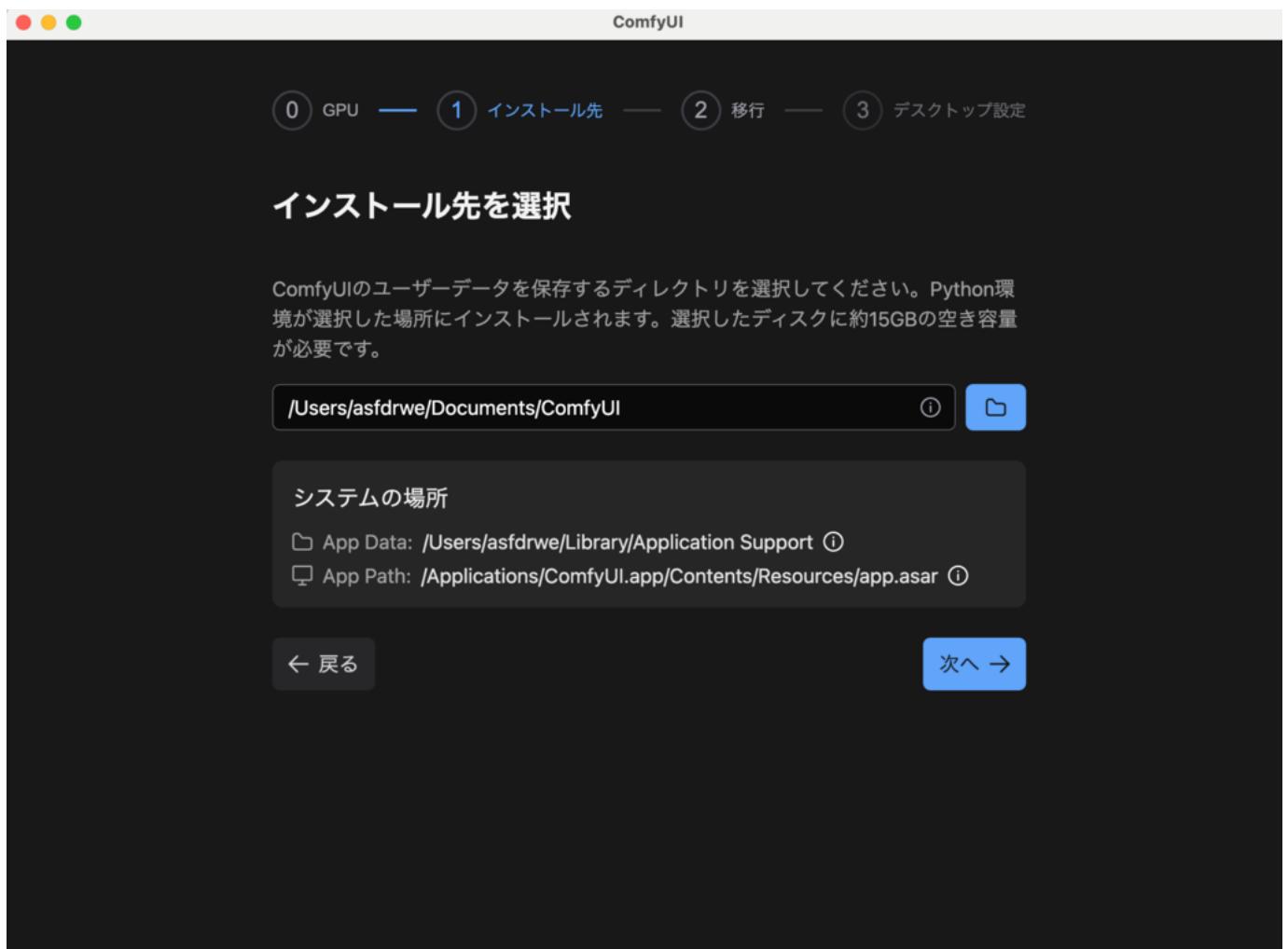


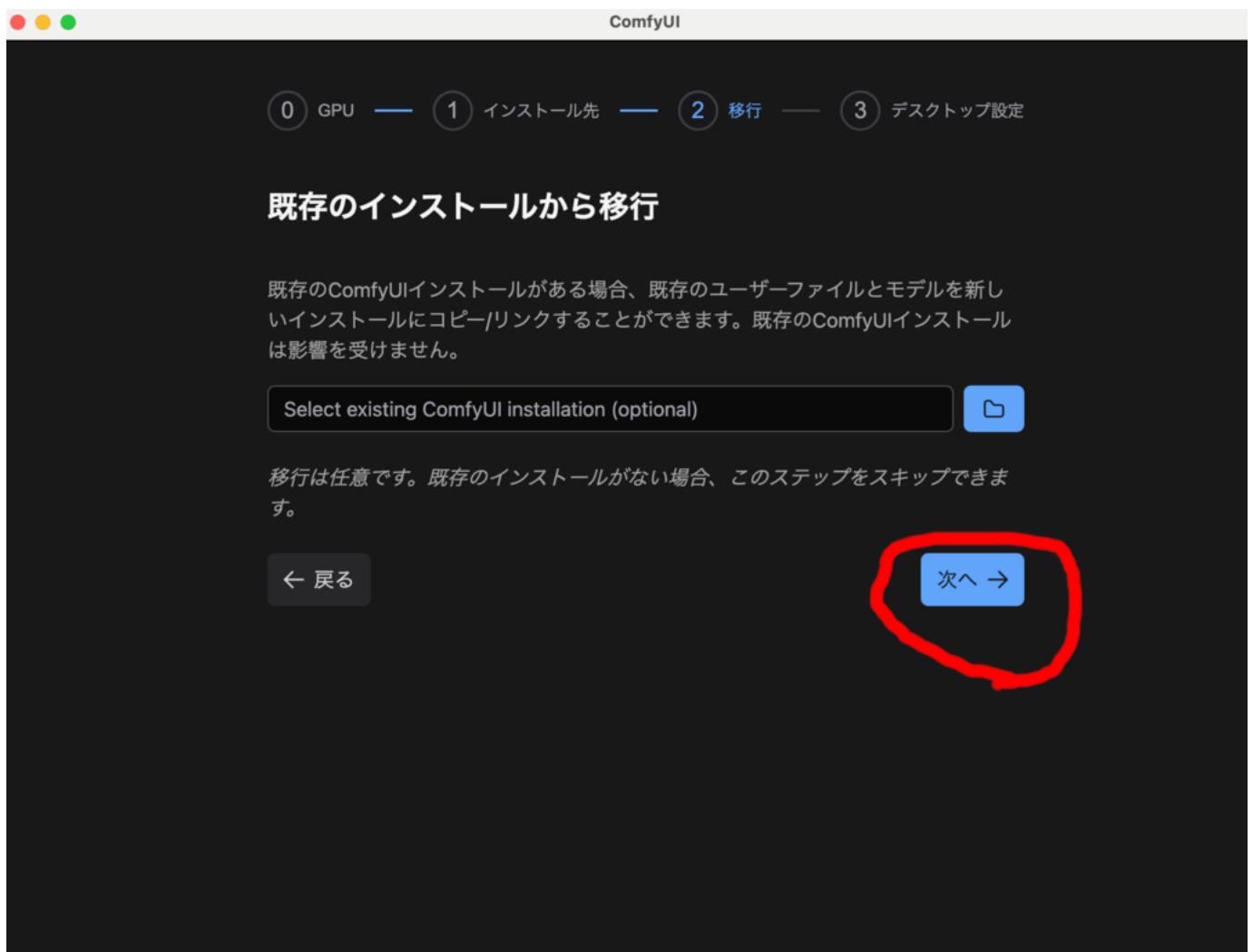
画面の指示に従い ComfyUI のインストールを行います。



標準では書類フォルダ(/User/ユーザ名/Documents)内に作成される ComfyUI フォルダ以下にプログラムや使用するモデルや生成された画像が置かれます。

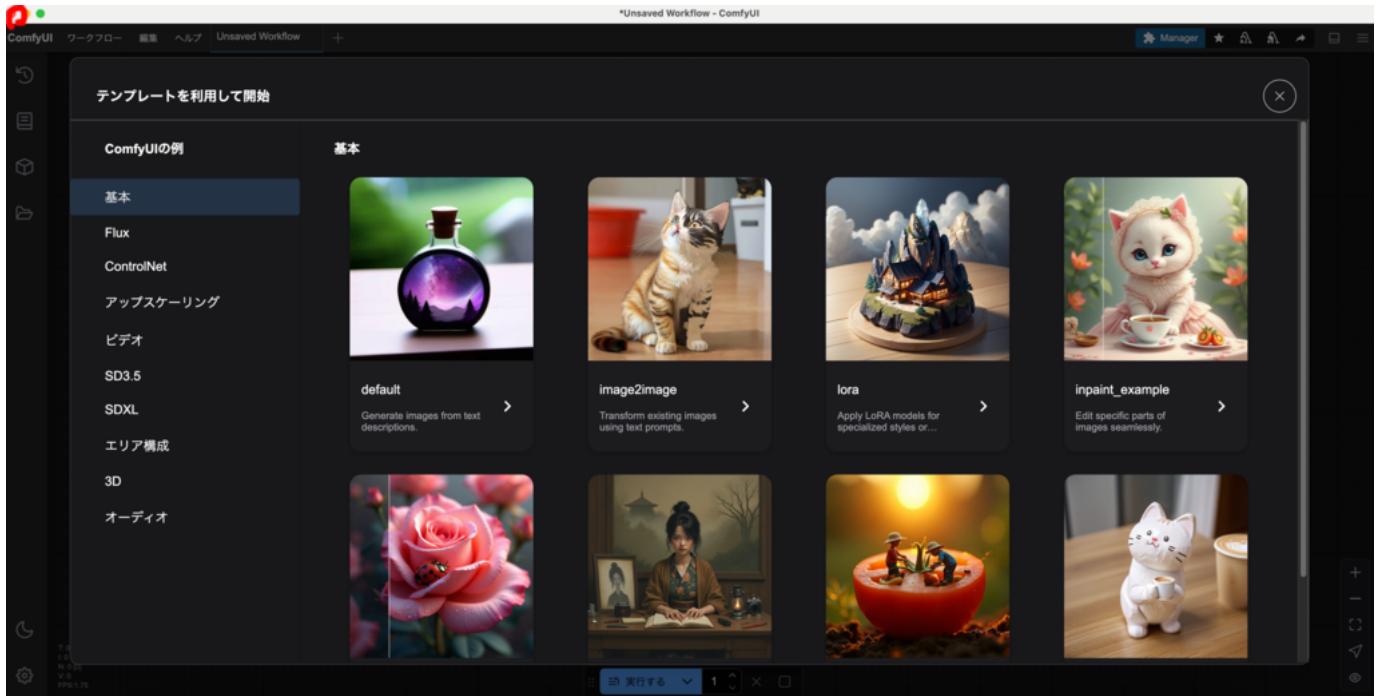
推奨モデルを利用する場合モデルだけで数十 GB 消費するので、ストレージが足らない場合は外付け SSD 等を指定してそちらに置くようにしてください。あとからフォルダを移動させると正常に動作しないので、最初に外付け SSD 等を指定してください。あとから移動させたい場合は、ComfyUI Desktop を再インストールしてもう一度設定してください。







画面の指示に従いインストールを完了させてください。



起動したら一旦なにもせずに閉じてください。

推奨モデルのダウンロードと用語解説

ワークフロー

ComfyUIで実行する処理手順を記載したものがワークフローです。処理単位にあたるノードをつないで作成します。

ワークフローは本来自身で作成して実行するものです。ただ、生成AIの仕組みがわからないと一から自力で作成するのは難しいので、M4 Mac Miniでも実用可能な速度で動くよう調整済みで、さまざまな用途に対応するワークフローを用意しました。

catAIToolsフォルダの workflows フォルダに入れてあります。この文書で使い方を順次解説していきます。

なお、こちらの用意したワークフローでは、どこを変更する必要があるかわかりやすくするために、処理内容に合わせて設定・記述すべきノードを茶色、必要に応じて変更するノードを緑色、通常変更する必要がないノードを灰色、説明用のノードを黄土色にしてあります。無効なノードは紫色にしています。これは一般的なルールではなくこちらのワークフローのみのローカルルールです。

モデル

ComfyUIでは画像や音声、動画などを生成するためにさまざまなモデルを使用します。

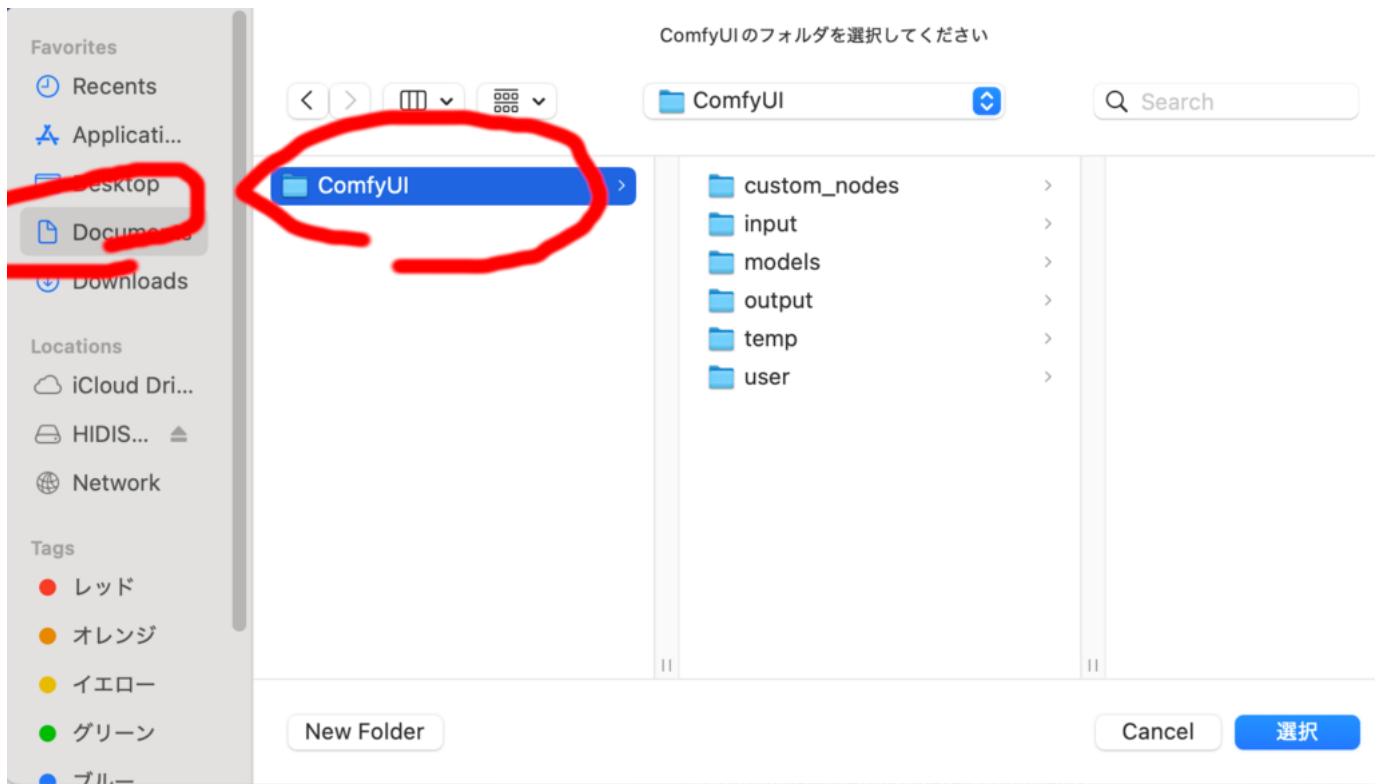
ワークフローのノードで使用するモデルは自動的にダウンロードされるものもありますが、別途ダウンロードしないといけないものが大半です。

推奨モデルをダウンロードするユーティリティを用意しました。

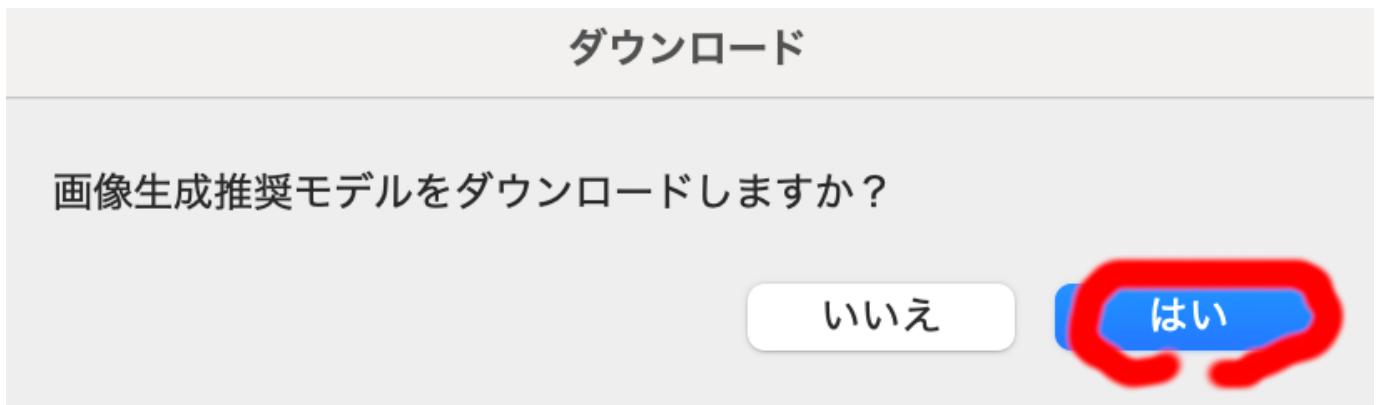
catAIToolsフォルダのComfyUI用ツールフォルダを開いてください。

名前	変更日	サイズ	種類
AudioModel.download.command	今日 13:51	1 KB	ターミ…クリプト
ComfyUI-danbot.install.command	昨日 7:55	727 バイト	ターミ…クリプト
ComfyUI-MMAudio.install.command	昨日 7:54	5 KB	ターミ…クリプト
ComfyUI-RMGB.install.command	昨日 7:55	964 バイト	ターミ…クリプト
ComfyUI-ESPCAN_fix.command	昨日 7:55	1 KB	ターミ…クリプト
ImageModel.download.command	今日 13:46	2 KB	ターミ…クリプト
ViewModel.download.command	今日 14:03	3 KB	ターミ…クリプト

中にある『ImageModel.download.command』をダブルクリックしてください。

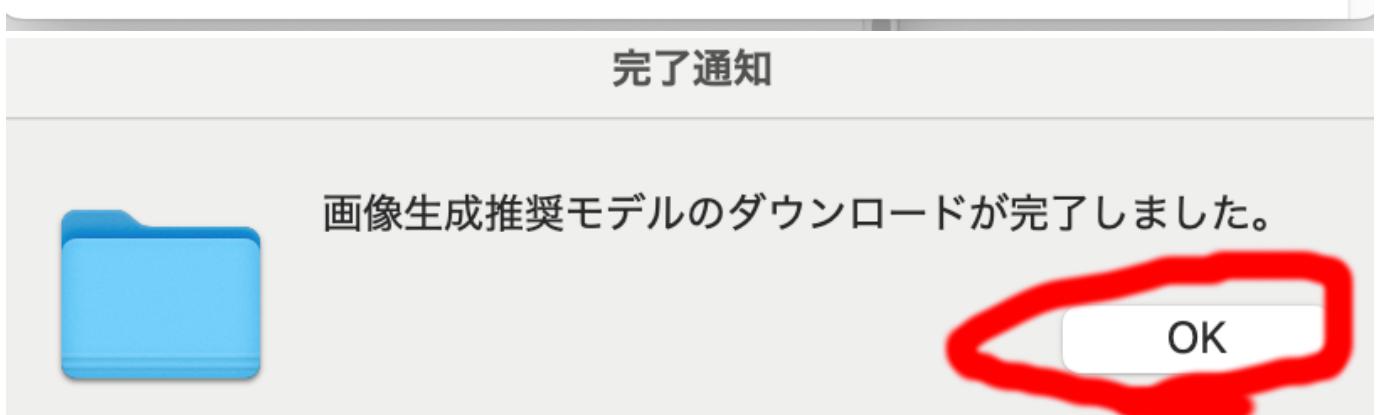


指示に従い、ComfyUI のインストールフォルダを選んでください (標準では書類フォルダの ComfyUI フォルダ)。

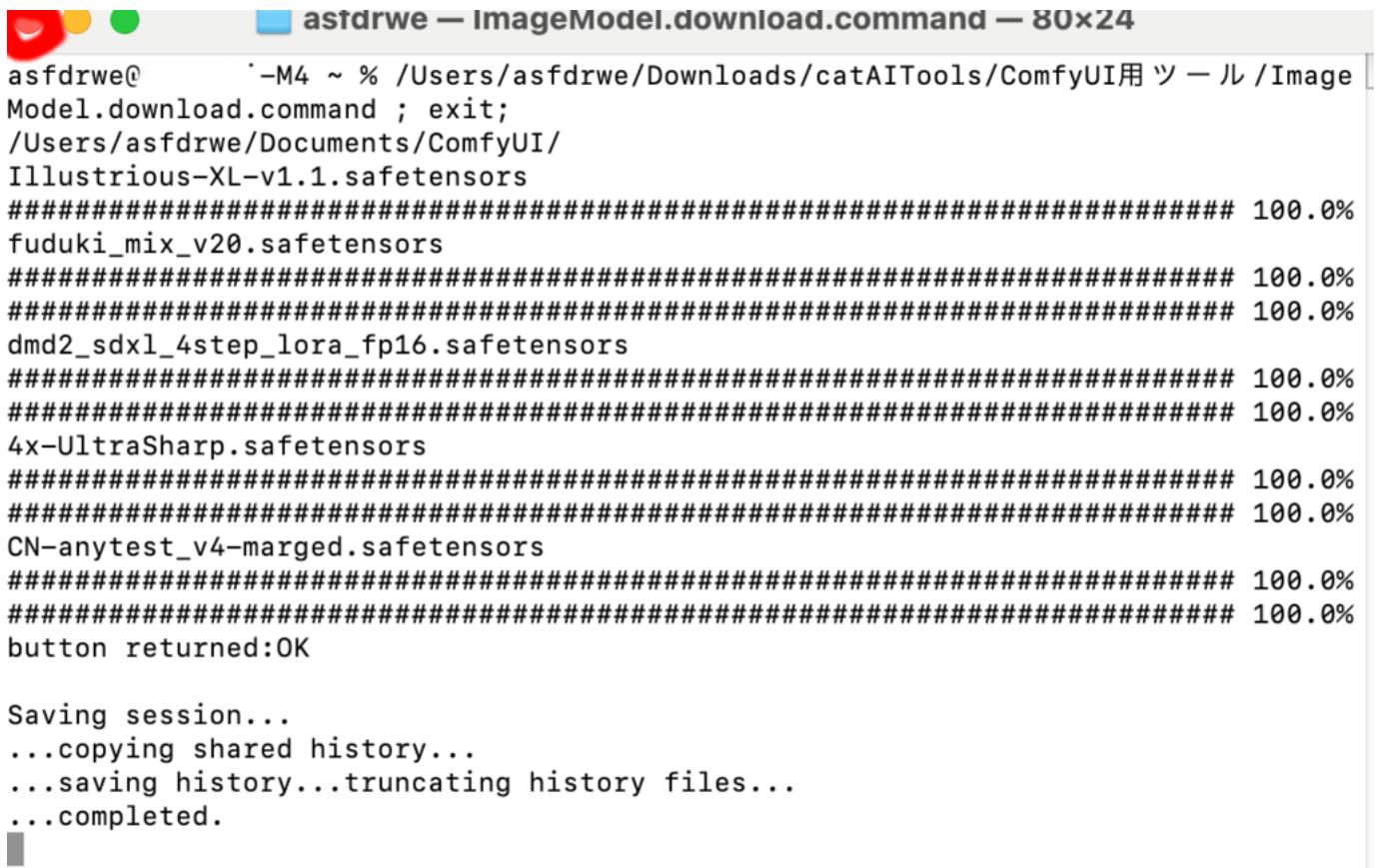


推奨モデルをダウンロードしますか？と聞かれるのではいを押してください。 推奨モデルをダウンロードを開始します。

```
asfdrwe — ImageModel.download.command — curl -I ImageModel.download...  
Last login: Fri Apr 11 17:14:52 on ttys003  
/Users/asfdrwe/Downloads/catAITools/ComfyUI用 ツール /ImageModel.download.command  
; exit;  
asfdrwe@ ~ % /Users/asfdrwe/Downloads/catAITools/ComfyUI用 ツール /Image  
Model.download.command ; exit;  
/Users/asfdrwe/Documents/ComfyUI/  
Illustrious-XL-v1.1.safetensors  
##### 78.0%
```



ダウンロードが終了するとダイアログが出るので OK を押してください。



```

astdrwe@ ~ % /Users/asfdrwe/Downloads/catAITools/ComfyUI用ツール/ImageModel.download.command ; exit;
/Users/asfdrwe/Documents/ComfyUI/
Illustrious-XL-v1.1.safetensors
#####
##### 100.0%
fuduki_mix_v20.safetensors
#####
##### 100.0%
#####
##### 100.0%
dmd2_sdxl_4step_lora_fp16.safetensors
#####
##### 100.0%
#####
##### 100.0%
4x-UltraSharp.safetensors
#####
##### 100.0%
#####
##### 100.0%
CN-anytest_v4-marged.safetensors
#####
##### 100.0%
#####
##### 100.0%
button returned:OK

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

```

さらにスクリプトを実行していたターミナルを閉じてください。

ダウンロードするモデルは全部で5つあり数十GBあります。通信データ量が大きいので時間がかかります。モバイル回線で落とす場合は気をつけてください。

カスタムノード

ComfyUIはノードをつないでワークフローを作ります。標準で提供されているノードの他にさまざまなカスタムノードが存在します。

ComfyUI Desktopでインストール時に自動でインストールされる ComfyUI Manager もカスタムノードの1つで、さまざまなカスタムノードを管理できます。ワークフローに含まれているがインストールされていないカスタムノードを自動的に調べてインストールできる機能があるので、基本的にはワークフローを開いたときの指示に従い ComfyUI Manager で必要なカスタムノードをインストールします。

カスタムノードはさまざまな作者が独立して作成しています。

あるカスタムノードをインストールすると依存するライブラリが競合するため別なカスタムノードが動かなくなったり、同じ機能を持つカスタムノードが何種類もあったり、過去、[情報を盗むなど悪意のあるカスタムノード](#)が発見されたりしています。

むやみやたらとカスタムノードをインストールするのは好ましくないです。

カスタムノード管理する ComfyUI Manager は十分検証されていないカスタムノードを標準ではインストールできないようになっています。

この文書の一部のワークフローに ComfyUI Manager をセキュリティレベルを下げないとインストールできないカスタムノードを含むものがあります。

ComfyUI Managerでインストールする場合はComfyUI フォルダの user フォルダの default フォルダの ComfyUI Manager フォルダの『config.ini』を開き、 security_level を normal から weak に書き換え、セキュリティレベルを下げる必要があります。

また使用するカスタムノードにはそのままでは Apple Silicon Mac で動かないものもあり、コードの修正が必要です。

手動でこれらの作業を行うのは面倒なので、そのようなカスタムノードのためのインストール&修正スクリプトを用意しました。

推奨モデルのダウンロードツールと同様に catAITools の ComfyUI用ツール フォルダに入れてあります。

名前	変更日	サイズ	種類
AudioModel.download.command	一昨日 0:19	1 KB	ターミ…クリプト
ComfyUI-danbot.install.command	2025年4月10日 7:55	727 バイト	ターミ…クリプト
ComfyUI-MMAudio.install.command	2025年4月10日 7:54	5 KB	ターミ…クリプト
ComfyUI-RMGB.install.command	2025年4月11日 23:08	1 KB	ターミ…クリプト
ComfyUI-WanStartEndF...Native.install.command	昨日 22:27	2 KB	ターミ…クリプト
ComfyUI.ESRGAN.fix.command	2025年4月10日 7:55	1 KB	ターミ…クリプト
ImageModel.download.command	今日 13:25	2 KB	ターミ…クリプト
VideoModel.download.command	昨日 12:43	5 KB	ターミ…クリプト

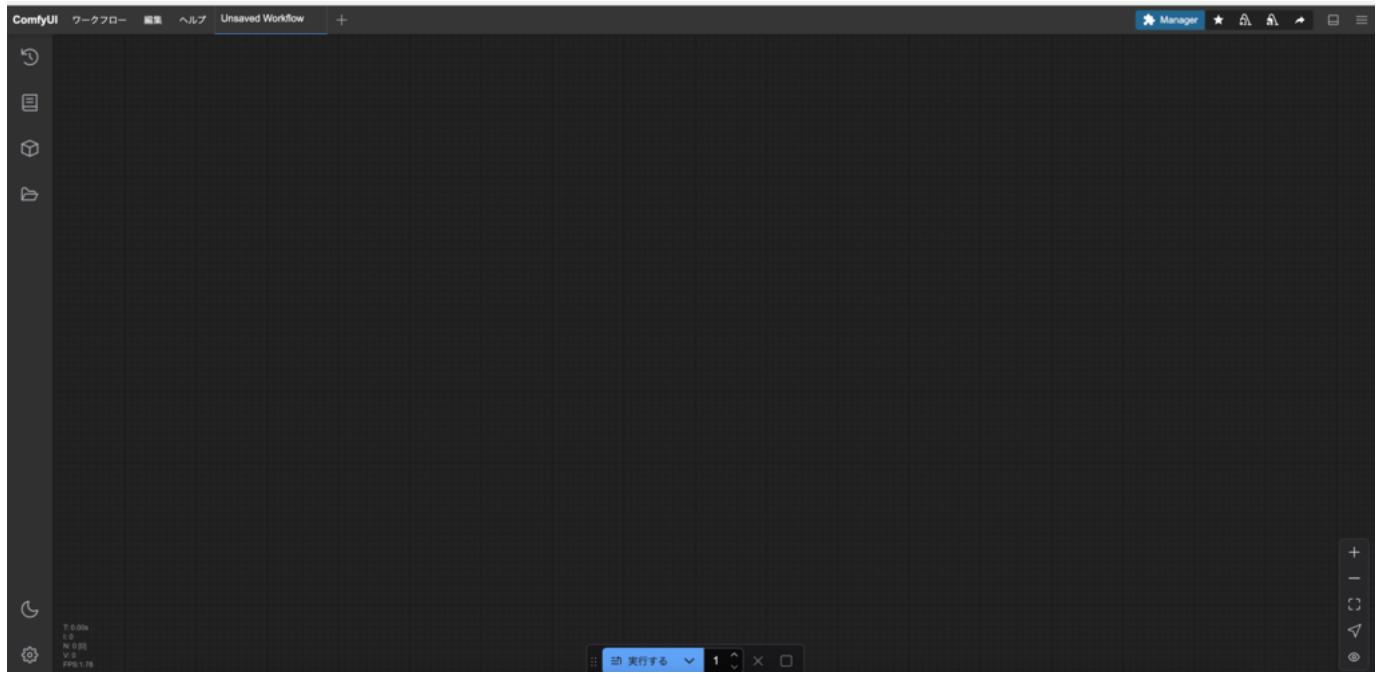
この文書のワークフローでは必要なカスタムノードをこのツールでインストールしてください。

なお、こちらが作成したワークフローはなるべくカスタムノードを使わず標準のノードメインで実行できるよう構成しています。

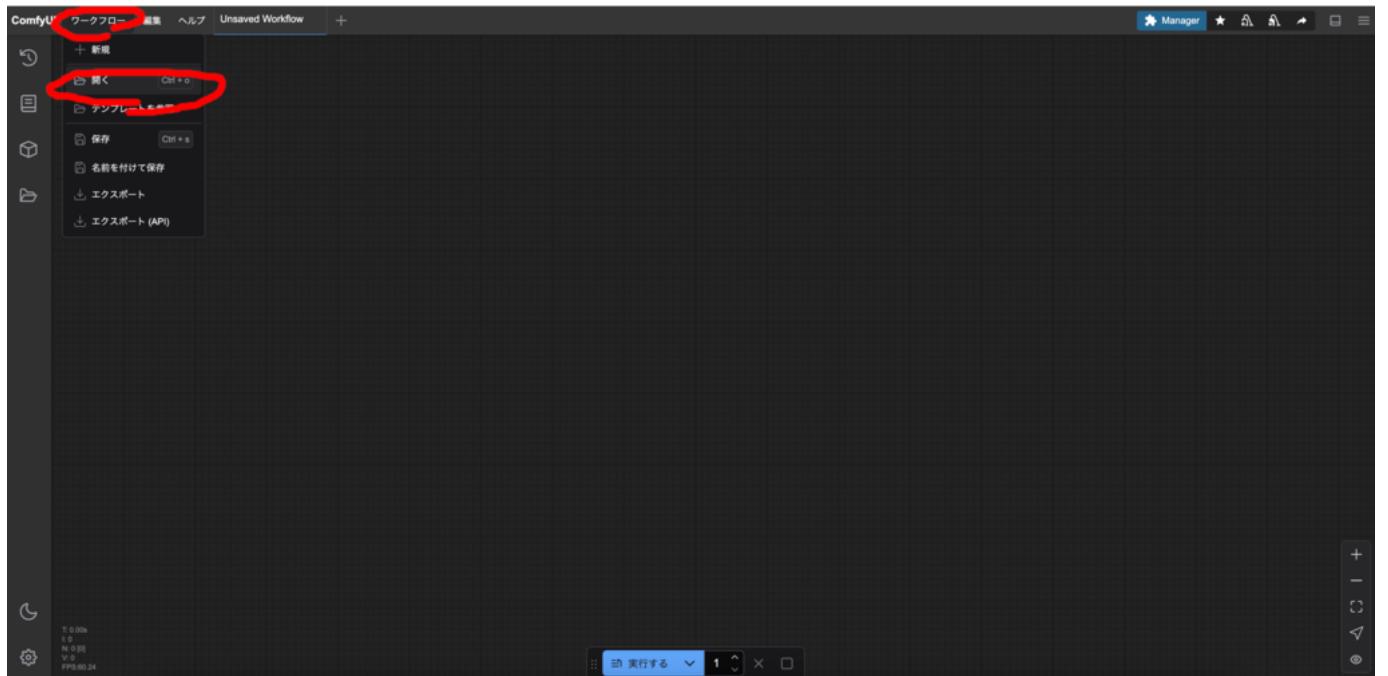
3.2 文章に基づく画像作成

文章に基づく画像生成

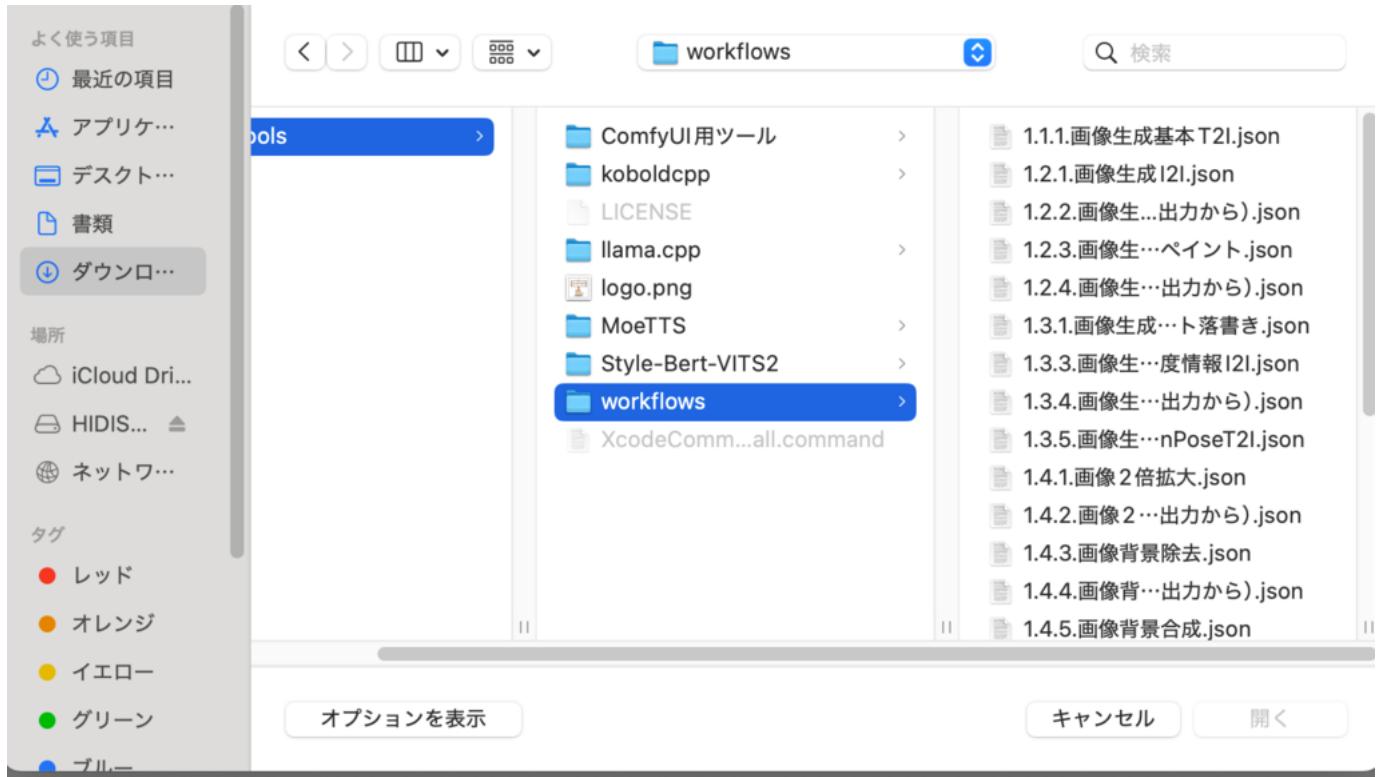
LaunchPad から ComfyUI Desktop を起動してください。



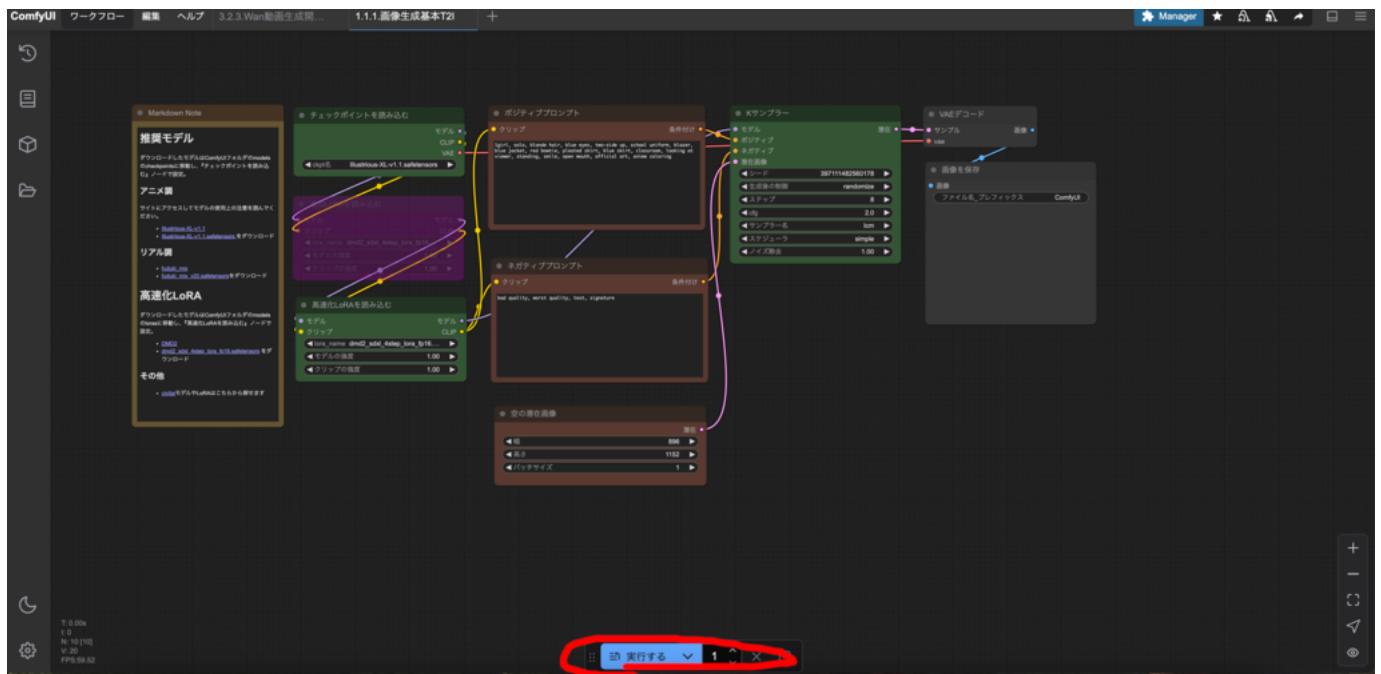
はじめての画像生成



メニュー左上のワークフローから開くを押してください。

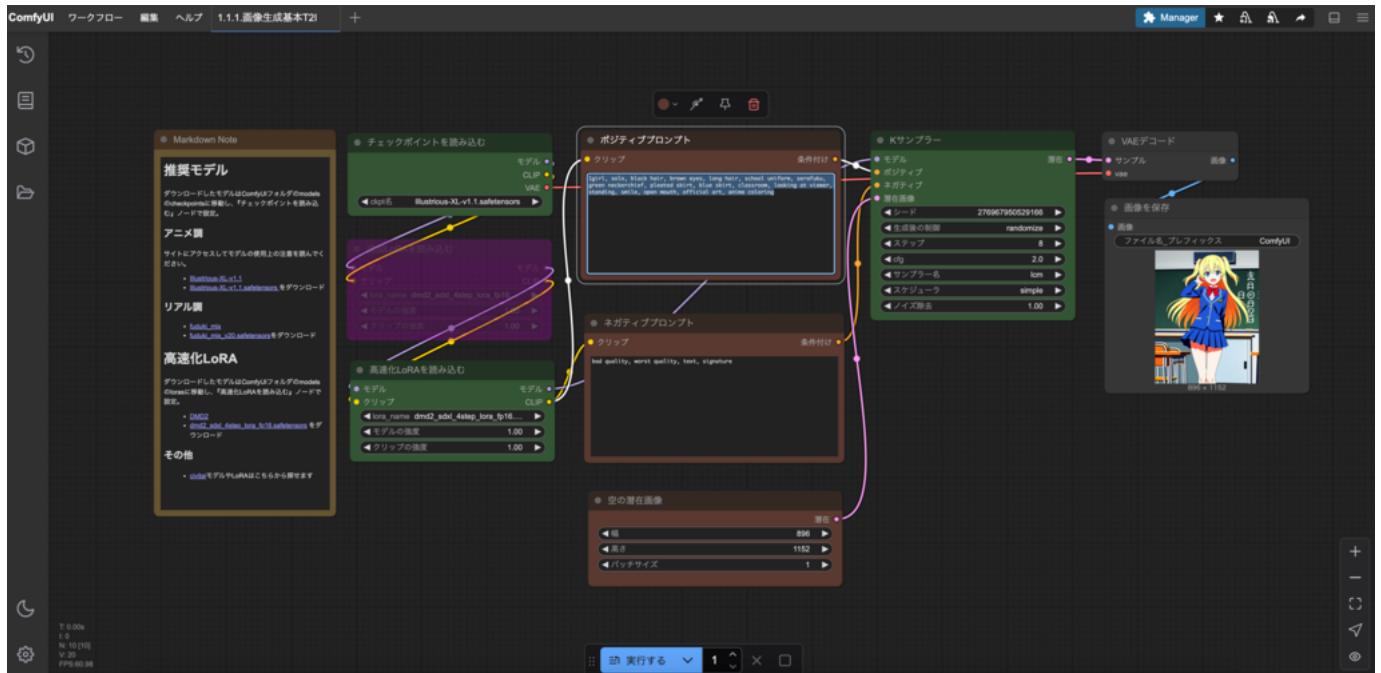


ComfyUI フォルダの workflows フォルダの『1.1.1.画像生成基本T2I.json』を開いてください。



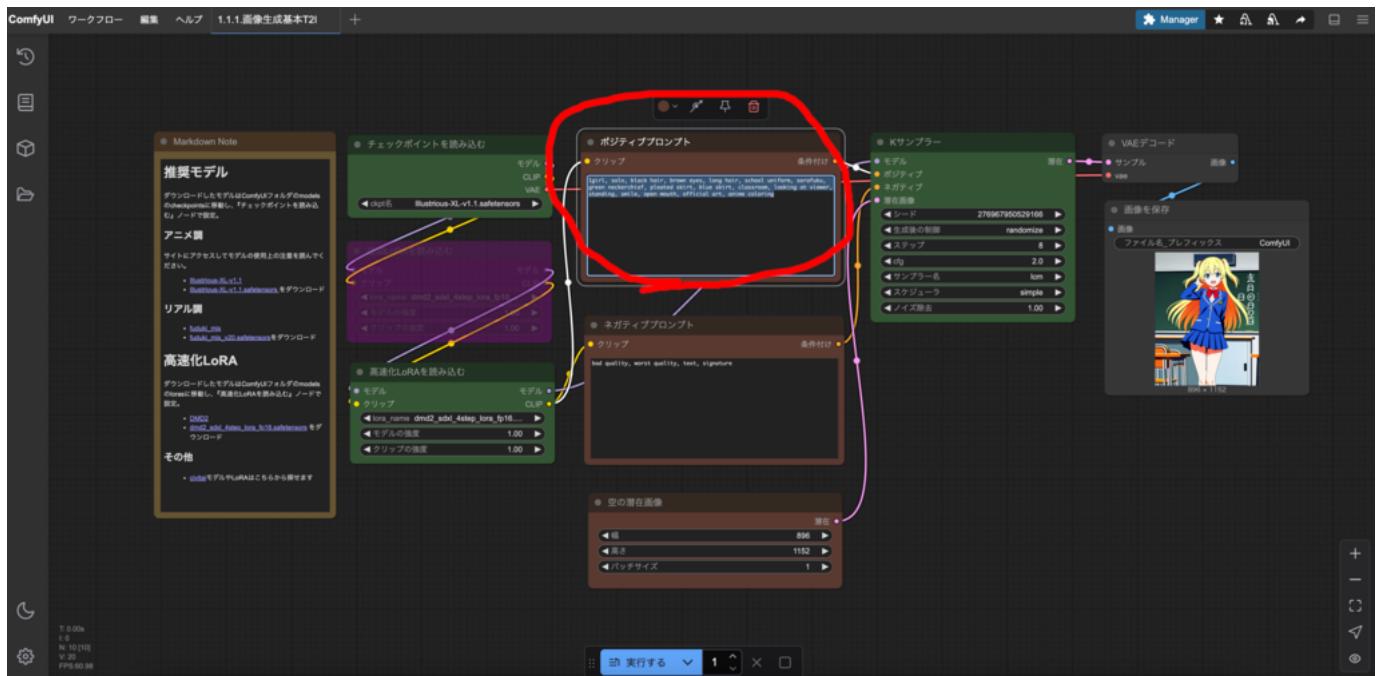
推奨モデルをダウンロードしている場合はそのまま実行できます。下の実行するを押してください。

金髪ツインテール青目で青ブレザー赤タイ青いスカートの制服を着た女の子が教室で口を開けてこちらを向いて微笑んでいるアニメ風の 896x1152 サイズの画像が 1 分程度で生成されるはずです。





生成画像への指示



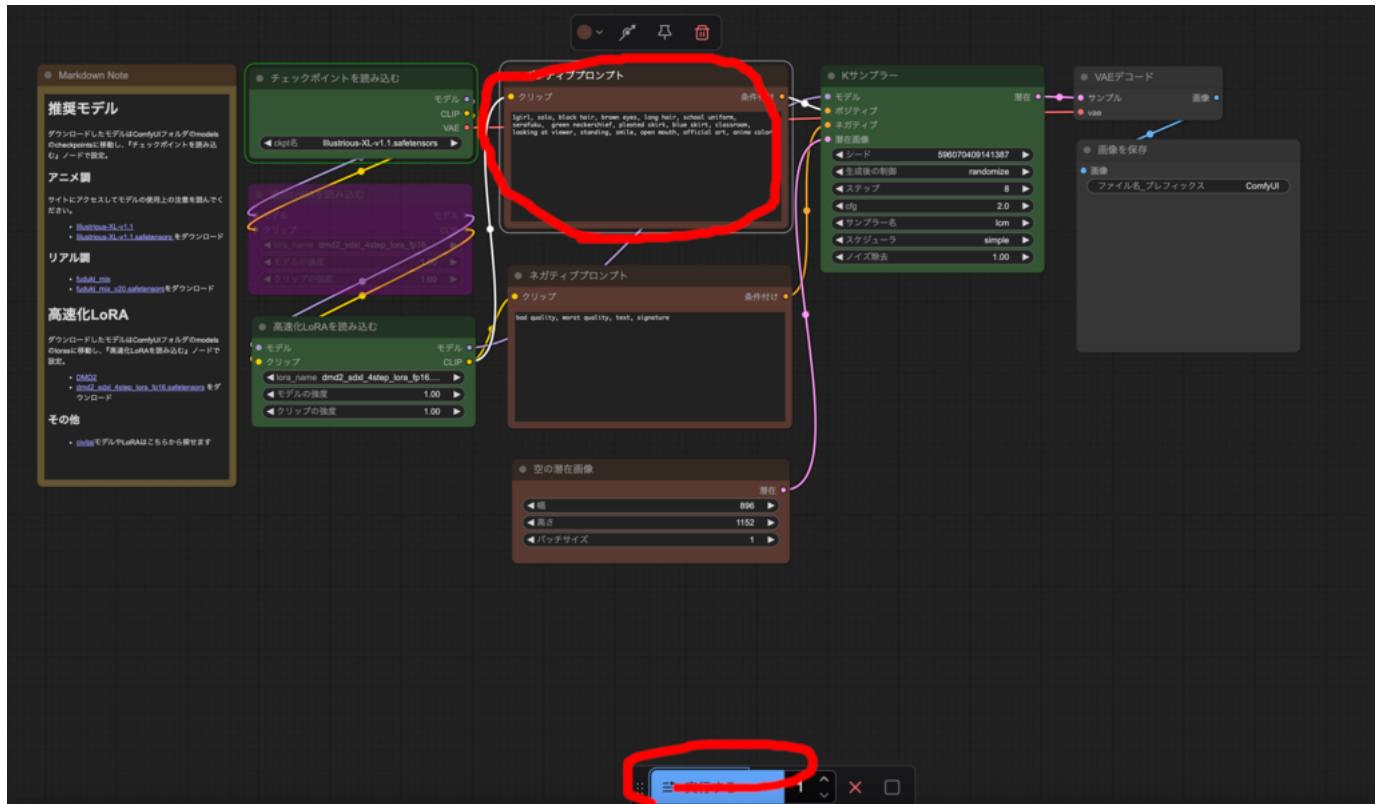
中央のポジティブプロンプトがどんな画像を生成するかの指示になります。文章(Text)から画像(Image)を生成するので、Text to Image(T2I)といいます。

今使用しているモデル(Illustrious-xl-v1.1)では英語で単語に区切って与えるのが通常です。

```
1girl, solo, blonde hair, blue eyes, two-side up, school uniform,
blazer, blue jacket, red bowtie, pleated skirt, blue skirt,
classroom, looking at viewer, standing, smile, open mouth,
official art, anime coloring
```

- 1girl => 少女1人
- solo => 他に人がいない
- blonde hair => 金髪
- blue eyes => 青い目
- ...

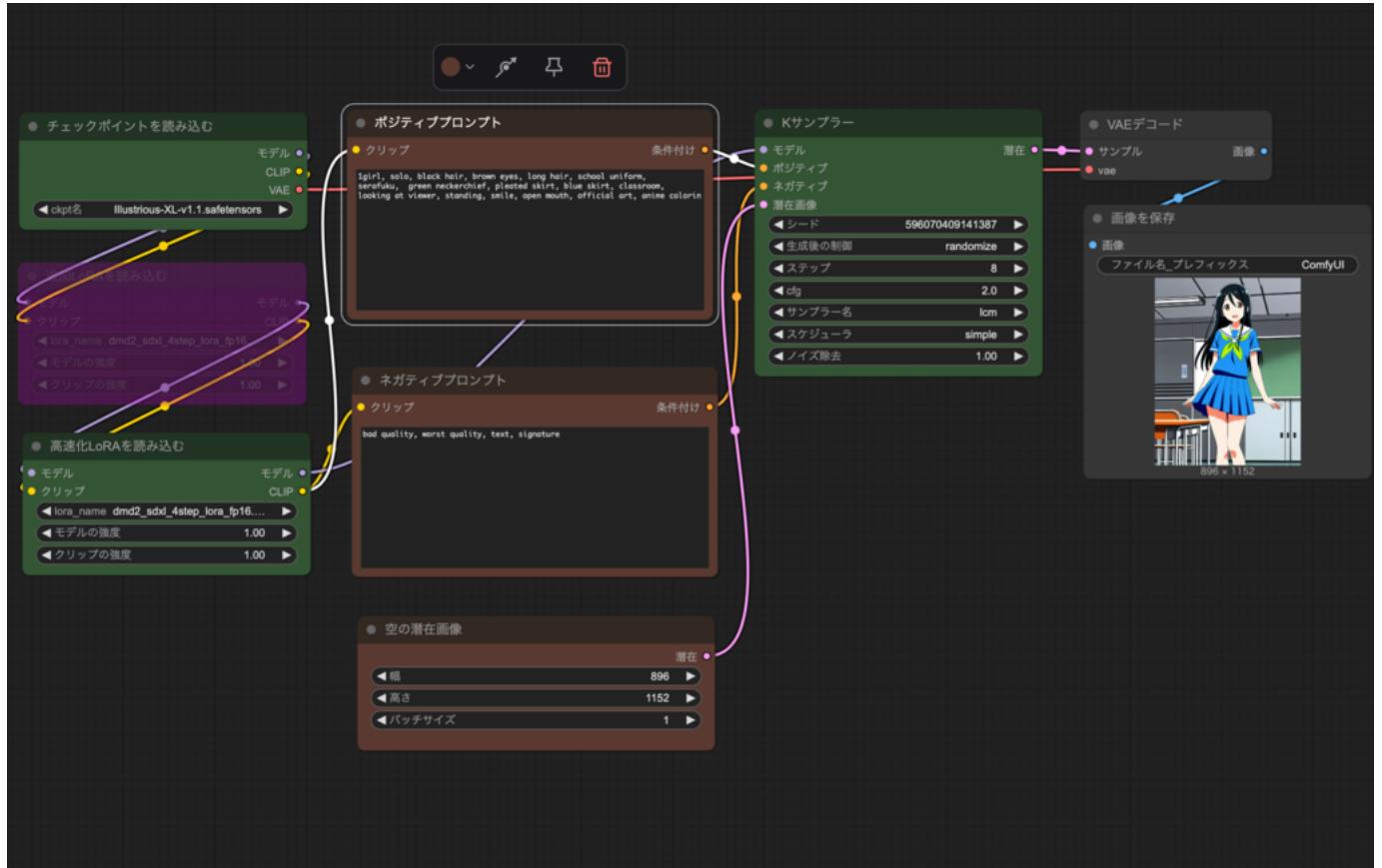
このように英単語で生成したい内容を記述します。



ポジティブプロンプトを次のように変えて実行を押してください。

```
1girl, solo, black hair, brown eyes, long hair, school uniform,
serafuku, green neckerchief, pleated skirt, blue skirt, classroom,
looking at viewer, standing, smile, open mouth, official art, anime coloring
```

黒髪ロング茶色目で緑タイ青スカートのセーラー服の女の子が教室でこちらを向いて微笑んでいるアニメ風の896x1152サイズの画像が生成されるはずです。





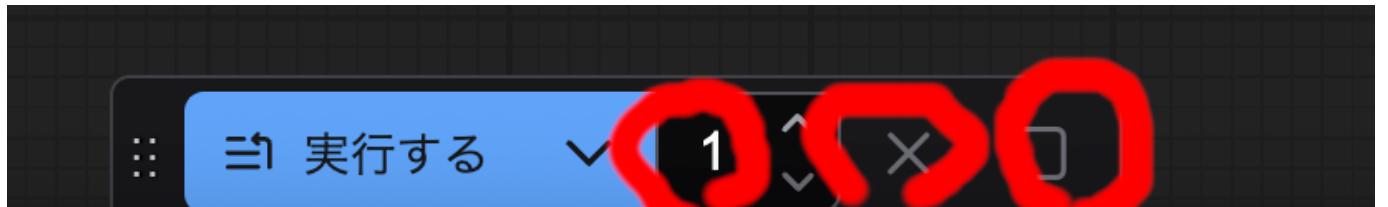
このように英語で生成したい画像に関する内容を記述することで画像を生成できます。このような英単語の羅列による生成画像への指示をプロンプト(prompt)をいいます。

英語が苦手な方のために、日本語文章を英語プロンプトに翻訳するワークフローを用意してあり、のちほど説明します。

この文書では基本的に英語プロンプトを使用します。

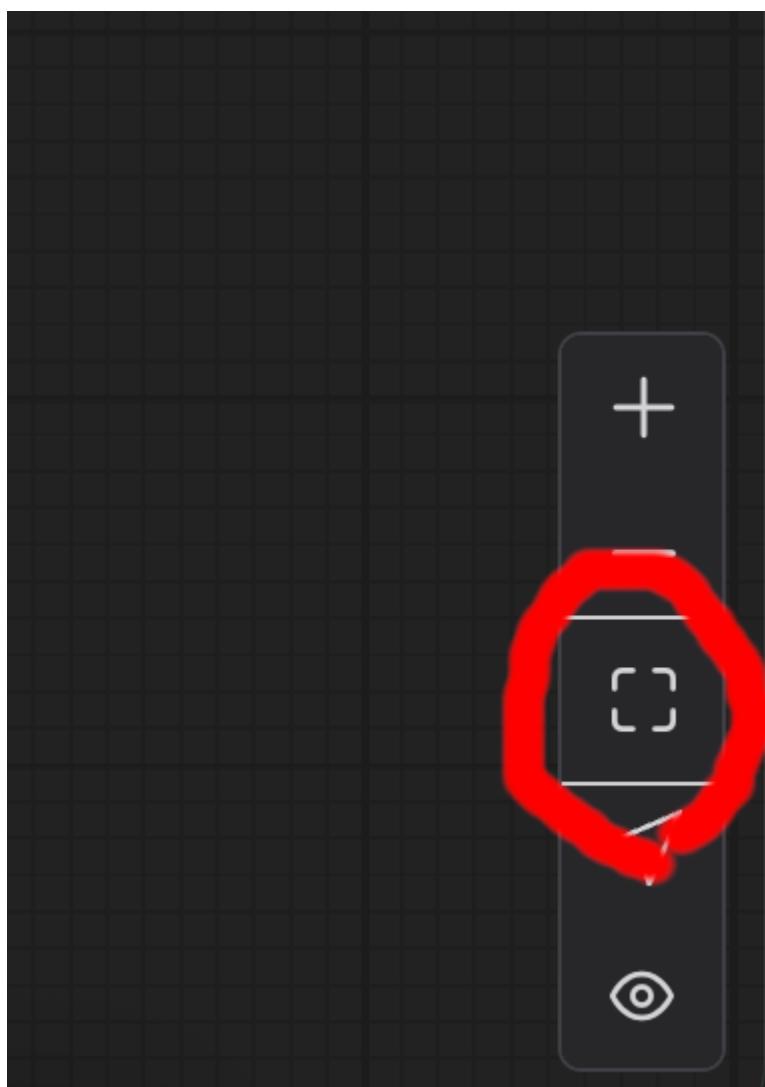
ComfyUI の基本操作

生成された画像等は ComfyUI フォルダの output フォルダになります。Finder で開いて確認してください。



実行するの右の数字で繰り返す回数を指定できます。未実行のものはキューに入れます。さらに隣の X を押すと現在の処理を中止できます（すぐに止まらない場合もあります）。さらに右の □ を押すとキューに入っている処理をクリアできます。

ノードのないところをドラッグするとワークフロー全体を移動できます。ホイールの回転か右下の + でワークフローの拡大縮小ができます。



右下のメニュー中央の □ を押すとノードのある位置が中央になります。

ノードの解説

各ノードに生成したい画像に関する設定を行うことで自由に画像を生成できます。

・ポジティブプロンプト

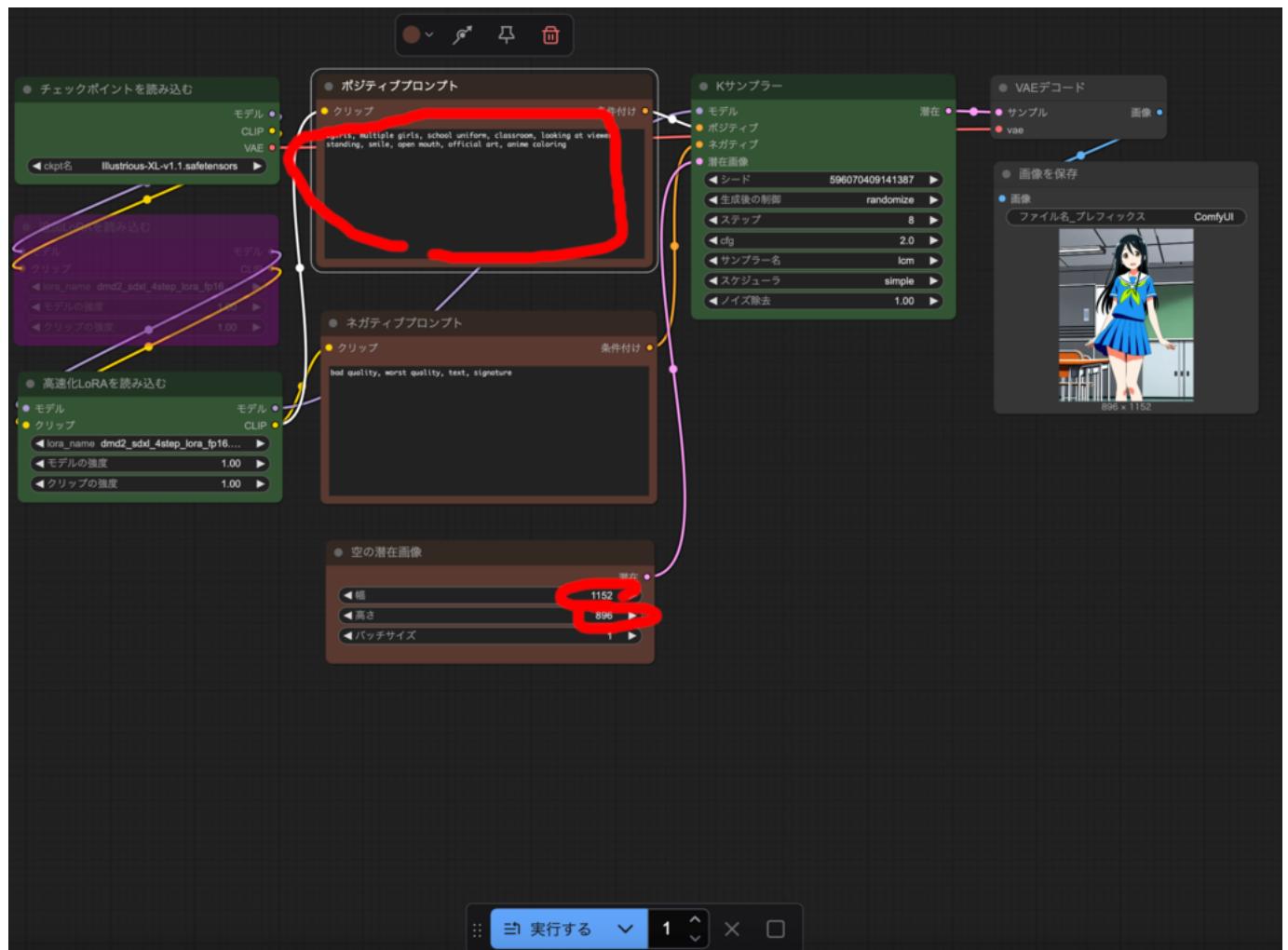
生成したい画像を記述するメインの内容を記述します。

・ネガティブプロンプト

生成したい画像に入って欲しくない内容を記述します。

・空の潜在画像

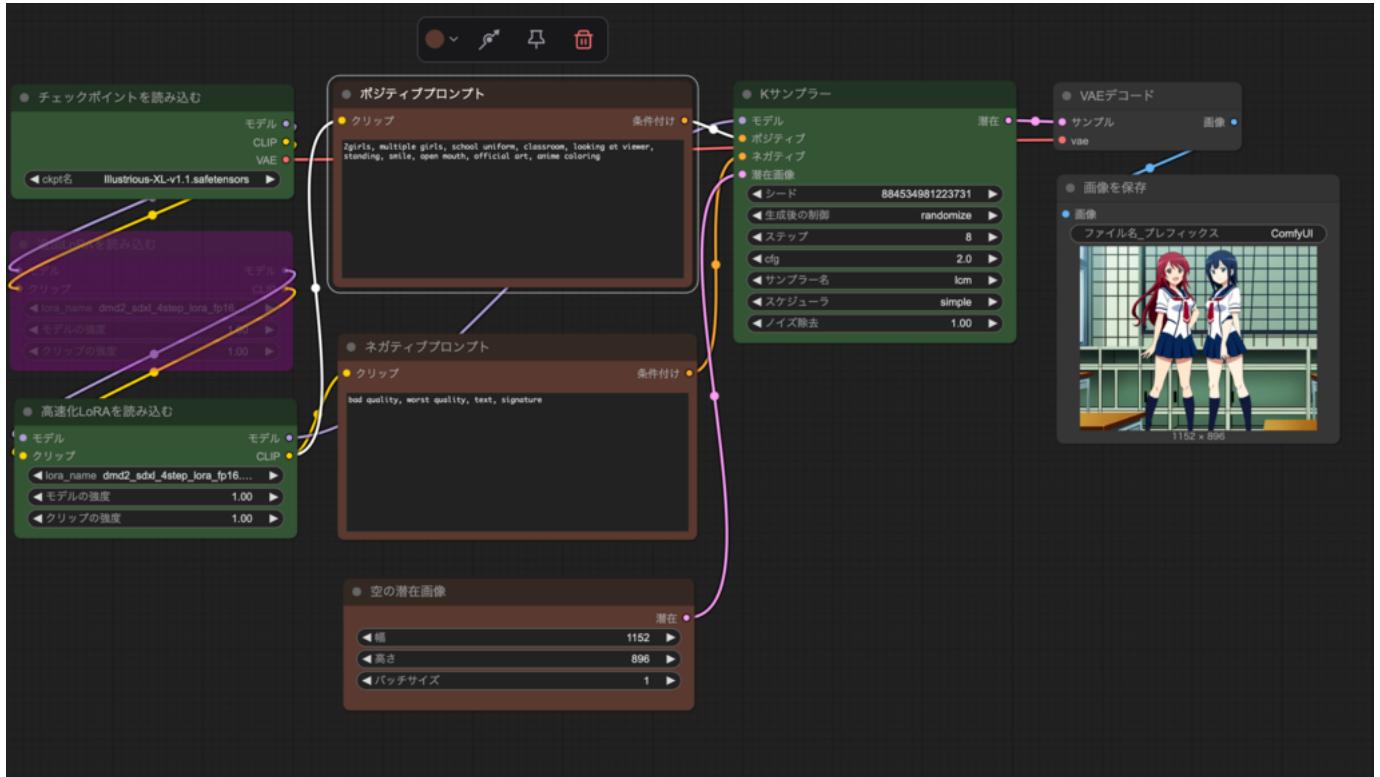
生成する画像の大きさを指定します。



空の潜在画像で幅を1152、高さを896にし、ポジティブプロンプトに次の文を入れて実行するを押してください。

```
2girls, multiple girls, school uniform, classroom, looking at viewer,
standing, smile, open mouth, official art, anime coloring
```

1152x896サイズの2人の少女の画像が生成されます。





- K サンプラー

画像の生成に関する設定です。ステップを上げればより高画質になりますが、その分時間がかかります。生成を高速化する技術高速化 LoRA により少ないステップ数での生成を可能にしています。仕様上は4ステップで生成可能となっていますが、品質がいまいちなので8ステップに設定しています。必要に応じてステップ数を変更してください。

- チェックポイントを読み込む

使用するモデルを指定します。

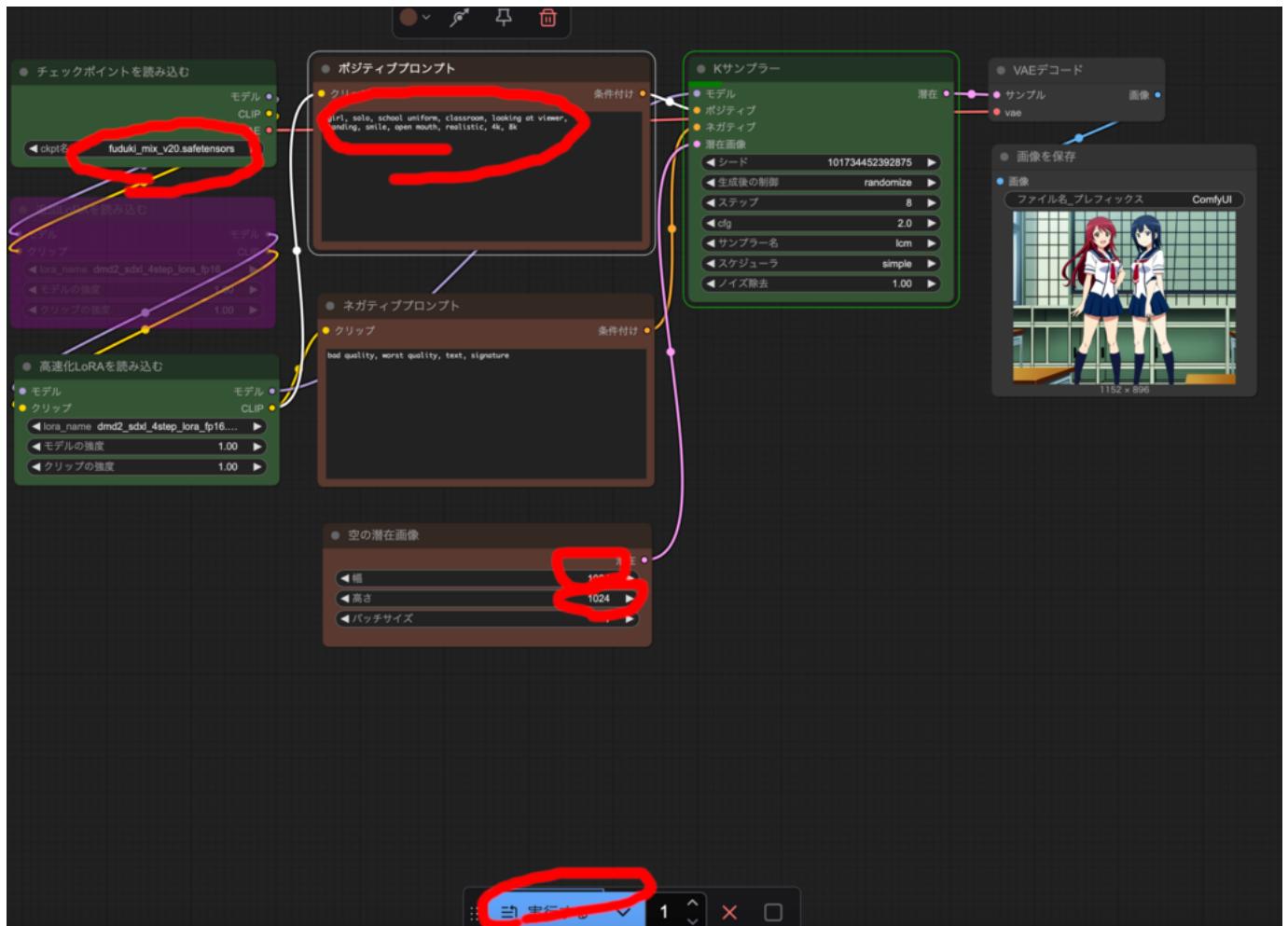
- 高速化LoRAを読み込む

高速化 LoRA を読み込みます。通常は変更する必要はありません。

- 追加LoRAを読み込む

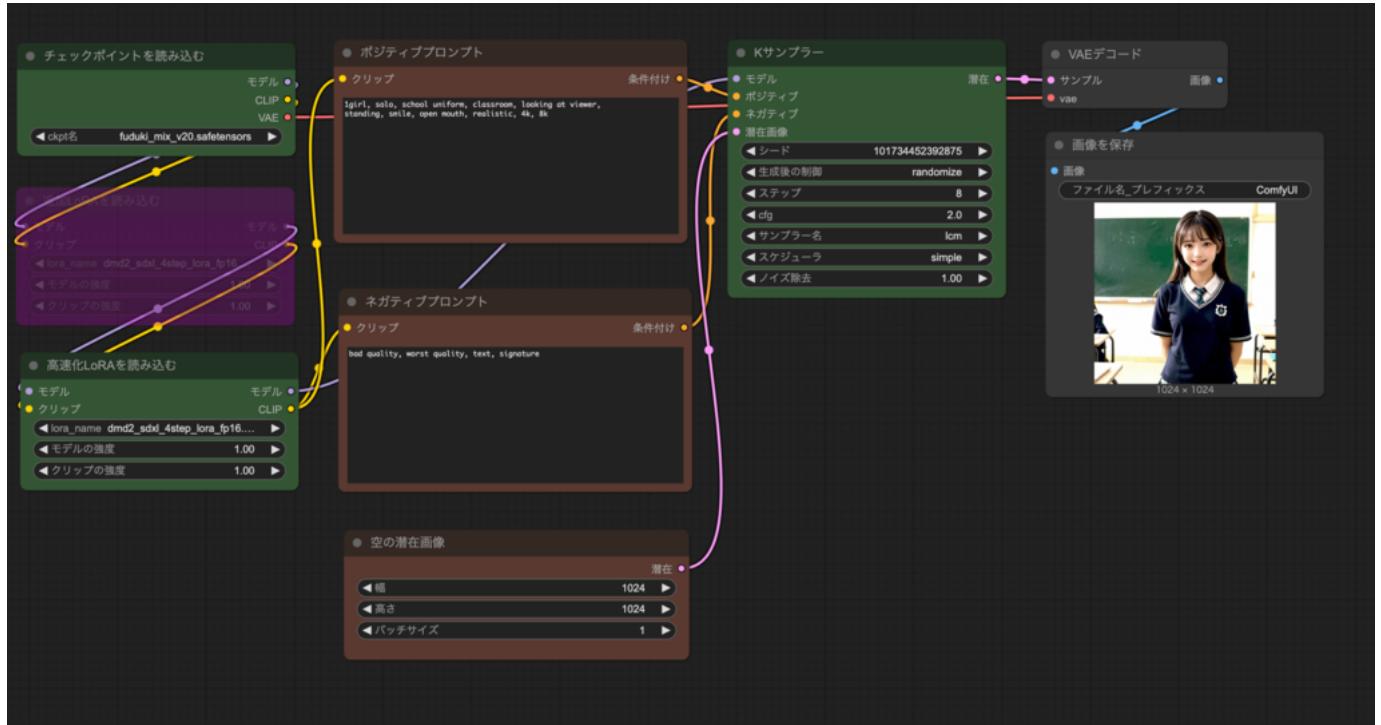
LoRA を追加する場合に有効にします。詳しくは [その他の追加LoRA使用方法](#)で説明します。

モデルの変更



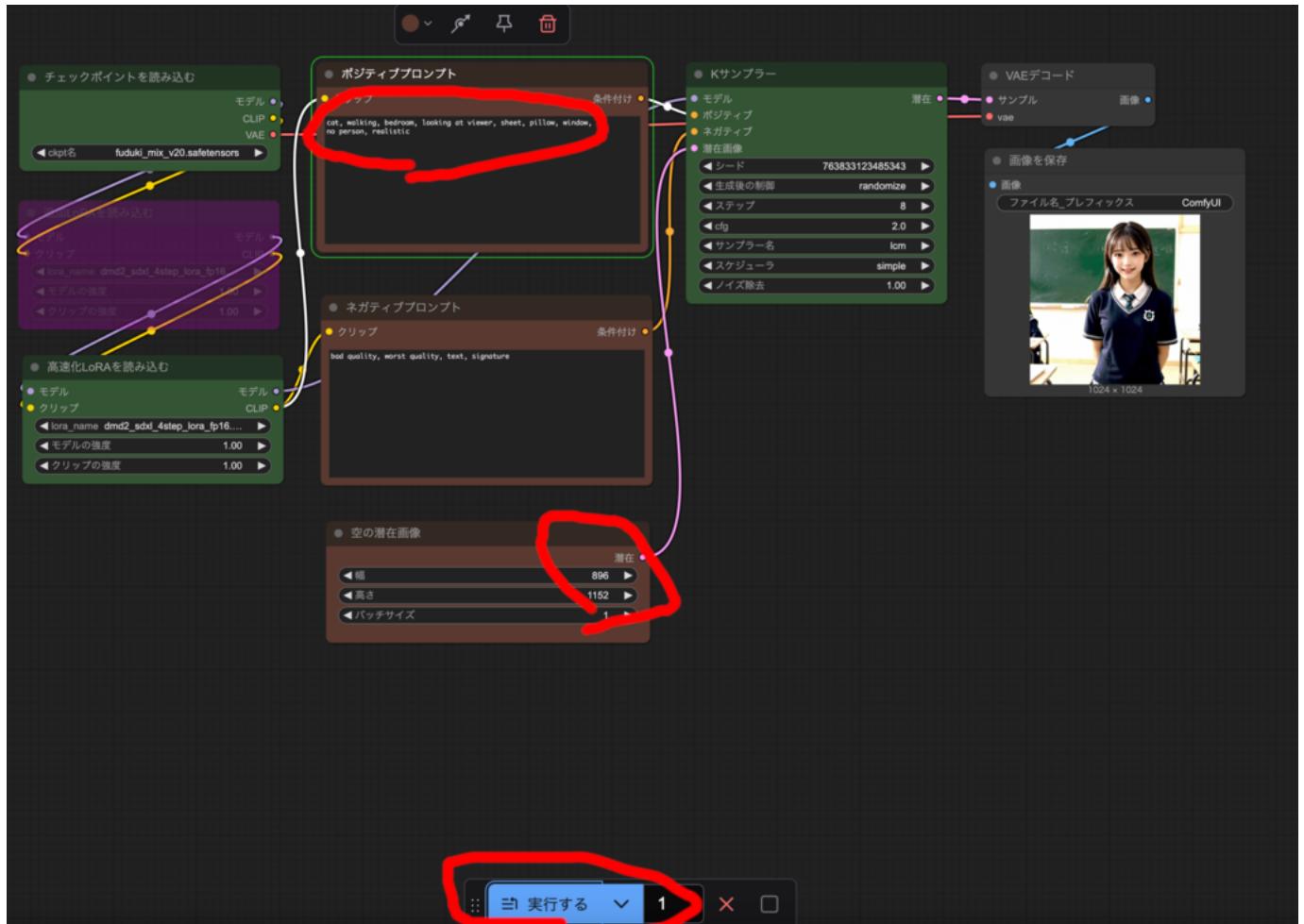
『チェックポイントを読み込む』ノードで ckpk 名を fuduki_mix_v20.safetensorsに変更し、『空の潜在画像』ノードで幅を1024、高さを1024にし、ポジティブプロンプトを次のようにして実行するを押してください。

```
1girl, solo, school uniform, classroom, looking at viewer,
standing, smile, open mouth, realistic, 4k, 8k
```





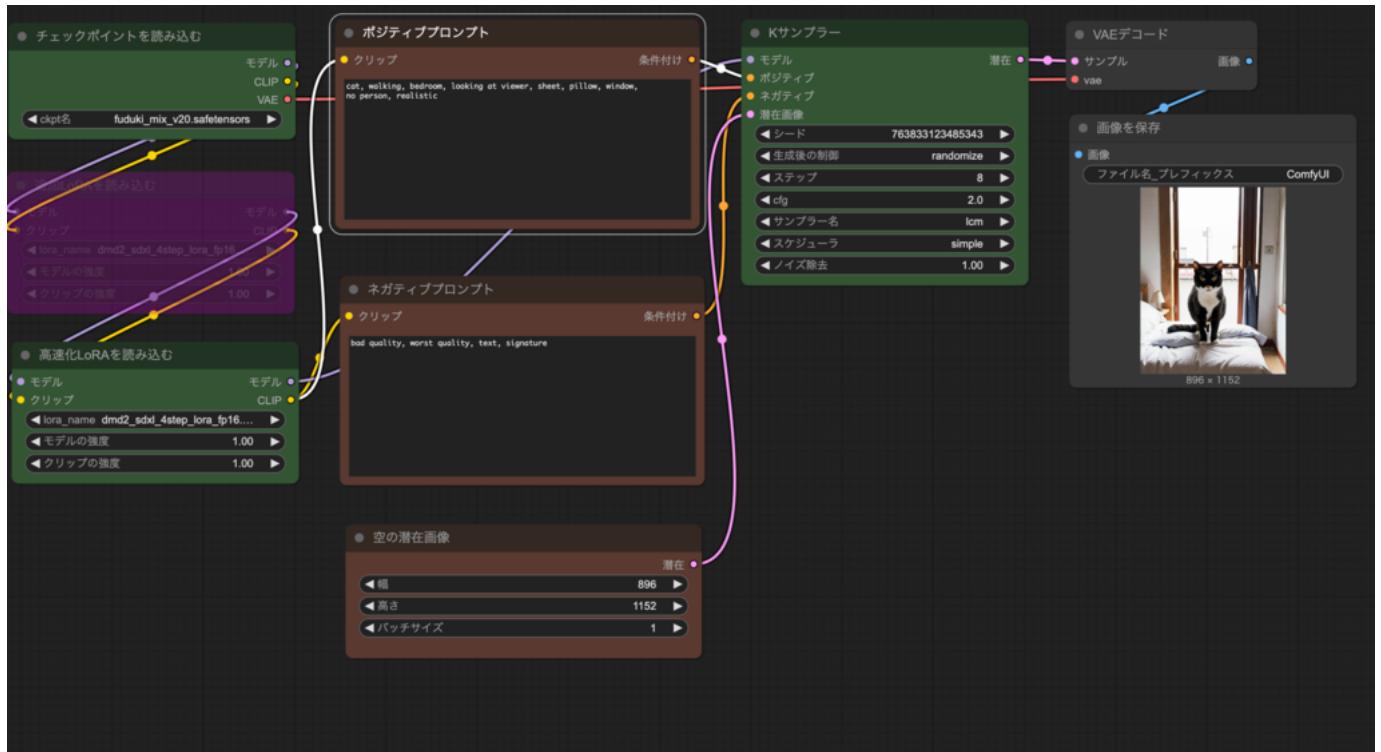
リアル調の制服少女の画像が生成されます。



『空の潜在画像』ノードで幅を 896、高さを 1152 にし、ポジティブプロンプトを次のようにして実行するを押してください。

```
cat, walking, bedroom, looking at viewer, sheet, pillow, window,
no person, realistic
```

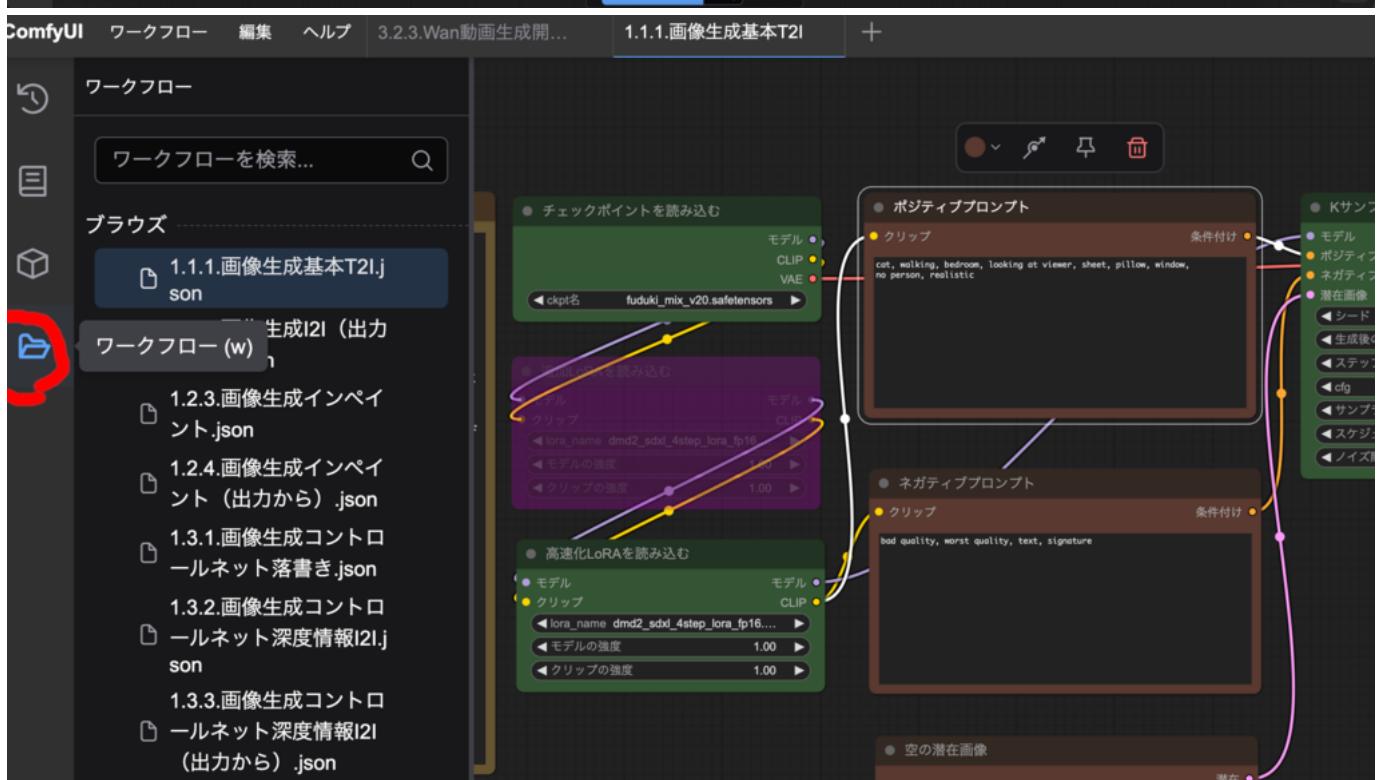
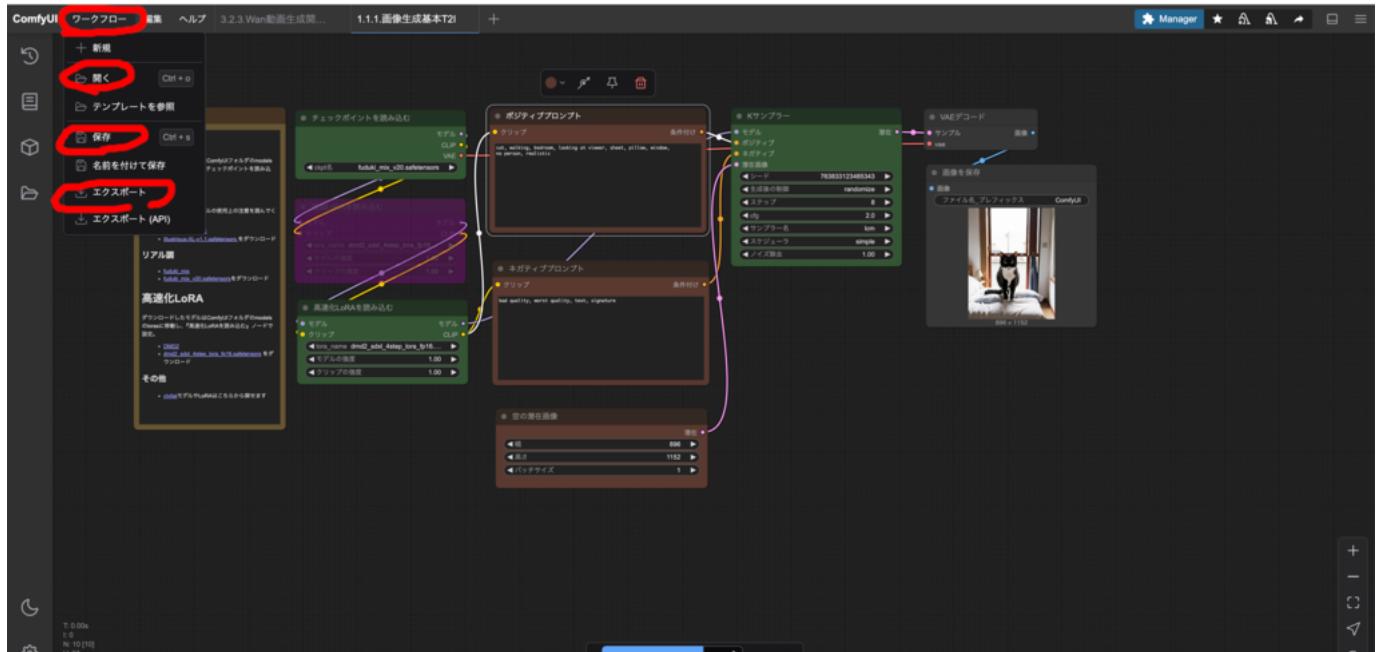
リアル調の猫が寝室で歩いている画像が生成されるはずです。





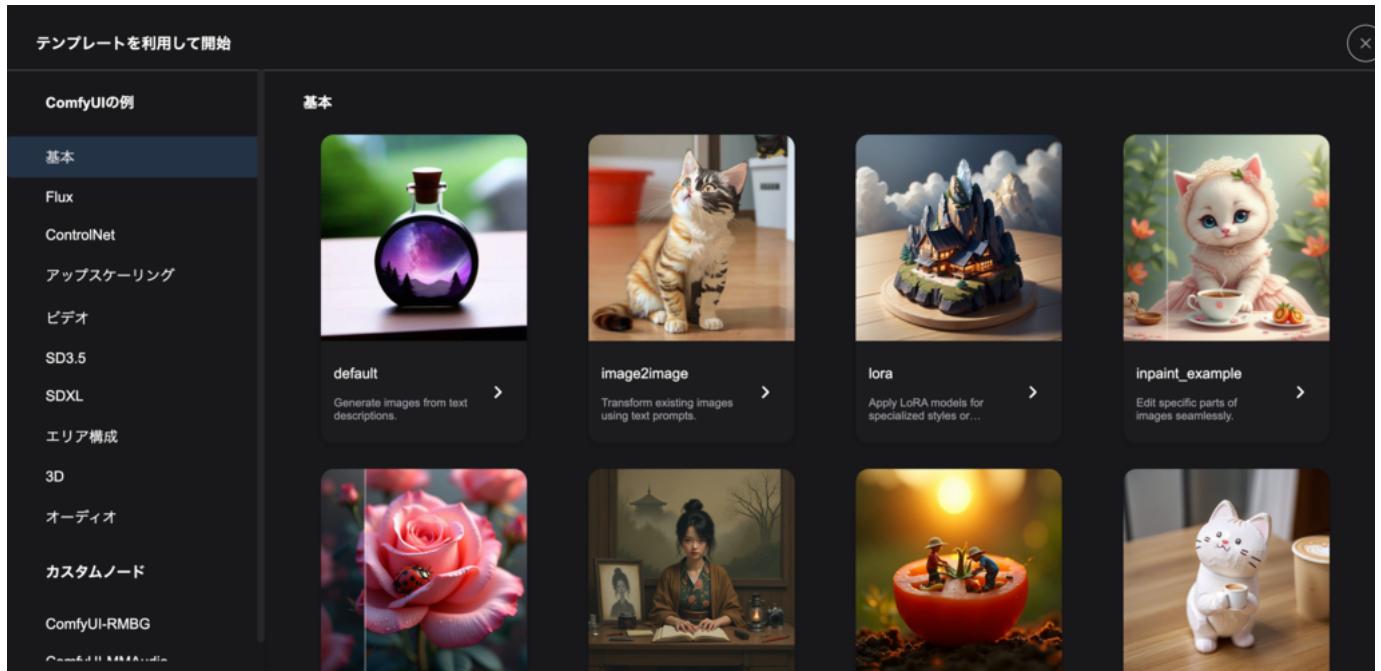
モデルによって画風は大きく変わります。

ワークフロー



メニューの「ワークフロー」の「保存」を押すとワークフローを保存できます。ComfyUI フォルダの user フォルダの default フォルダの workflows フォルダに保存されます。保存したワークフローは左側のメニューから直接開けます。

「エクスポート」するとファイルとしてワークフローを保存できます。ワークフローの「開く」から開けますし、ネットにアップロードして公開することもできます。



ComfyUI が提供するワークフローがあり、メニューのワークフローの テンプレートを参照から選べます。カスタムノードによってはワークフローを提供しているので、ComfyUI フォルダの `custom_nodes` フォルダの 各カスタムノードフォルダを開くと見つかる場合もあります。

[civitai](#)でもさまざまなワークフローが公開されているので、必要に応じて参照してください。

生成のコツ

画像生成 AI では常にうまく生成されるとは限らないです。

何度も生成して良いと思った画像を採用する形になるのでよく『ガチャ』と言われます。

また、モデルによって得意分野が異なり、同じプロンプトでも生成画像が全く異なります。

推奨モデルの `Illustrious-XL` はアニメ・イラストが得意なモデルで、`fuduki_mix` は写真のようなリアル調が得意なモデルです。

モデルは[huggingface](#)や[civitai](#)などにたくさんあるので、探してみてください。

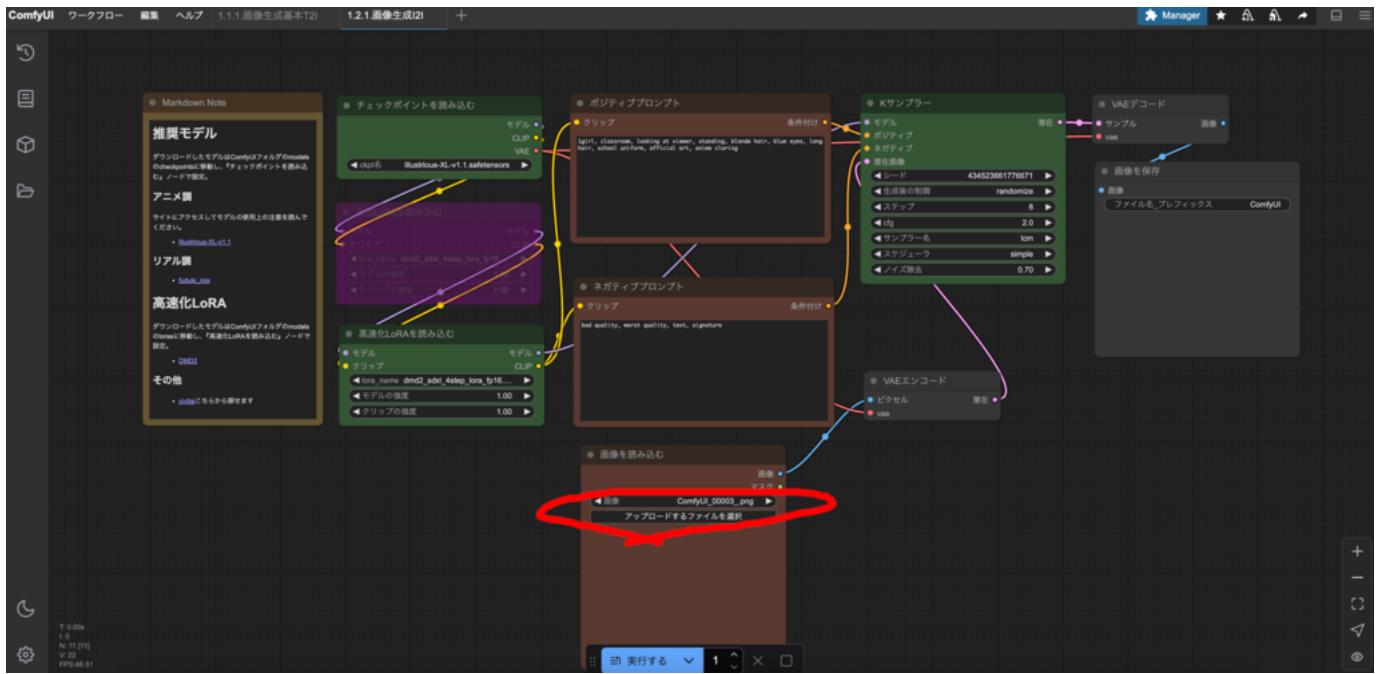
画像生成 AI では、プロンプトを変えたり、モデルを変えたり、何度も生成するなど 試行錯誤が必要なことを頭に入れてください。

3.3 画像と文章に基づく画像作成

画像と文章に基づく画像生成

文章に基づき画像を生成するのではなく、元となる画像(Image)から別な画像(Image)を生成することを、Image to Image(I2I)といいます。

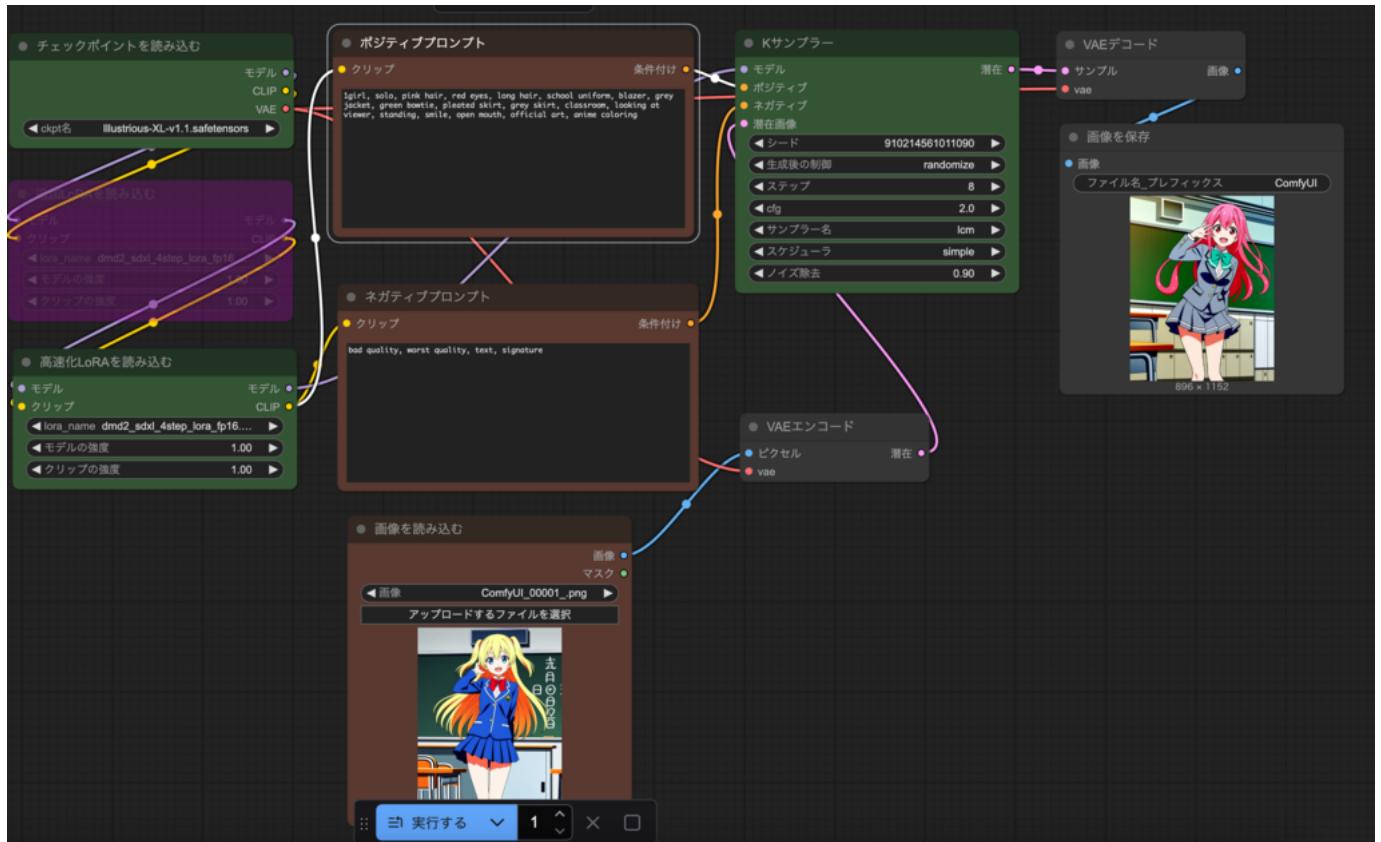
ワークフローから開くを選び、ComfyUI フォルダの workflows フォルダの『1.2.1.画像生成I2I.json』を開いてください。



中央下の画像を読み込むのアップロードするファイルを選択して先ほど生成した最初に生成したアニメ調少女の画像を選択してください。アップロードといってもネットのどこかにアップロードするのではなく ComfyUI フォルダの input フォルダにコピーしているだけです。

もしくは直接 Finder で output フォルダから input フォルダにファイルを移動させて、メニューの編集からノード定義を更新を選んでも OK です。

その画像を選択してください。



ポジティブプロンプトに次の文を入れて実行するを押してください。

```
igirl, solo, pink hair, red eyes, long hair, school uniform, blazer,
grey jacket, green bowtie, pleated skirt, grey skirt, classroom,
looking at viewer, standing, smile, open mouth, official art, anime coloring
```

元の画像の雰囲気を保ちつつ、ピンク髪ボニーイールで赤目でグレーの ジャケットとスカートの少女の画像が生成されるはずです。



右の K サンプラーノードのノイズ除去の値を増やすと元の画像から離れる代わりにプロンプトの指示により近づきます。ノイズ除去の値を減らすと、元の画像に近づきます。

今回は強くしないと変化が少なかったので 0.9 にしていますが、通常は大きくて 0.70 以下にするのが普通です。

付録

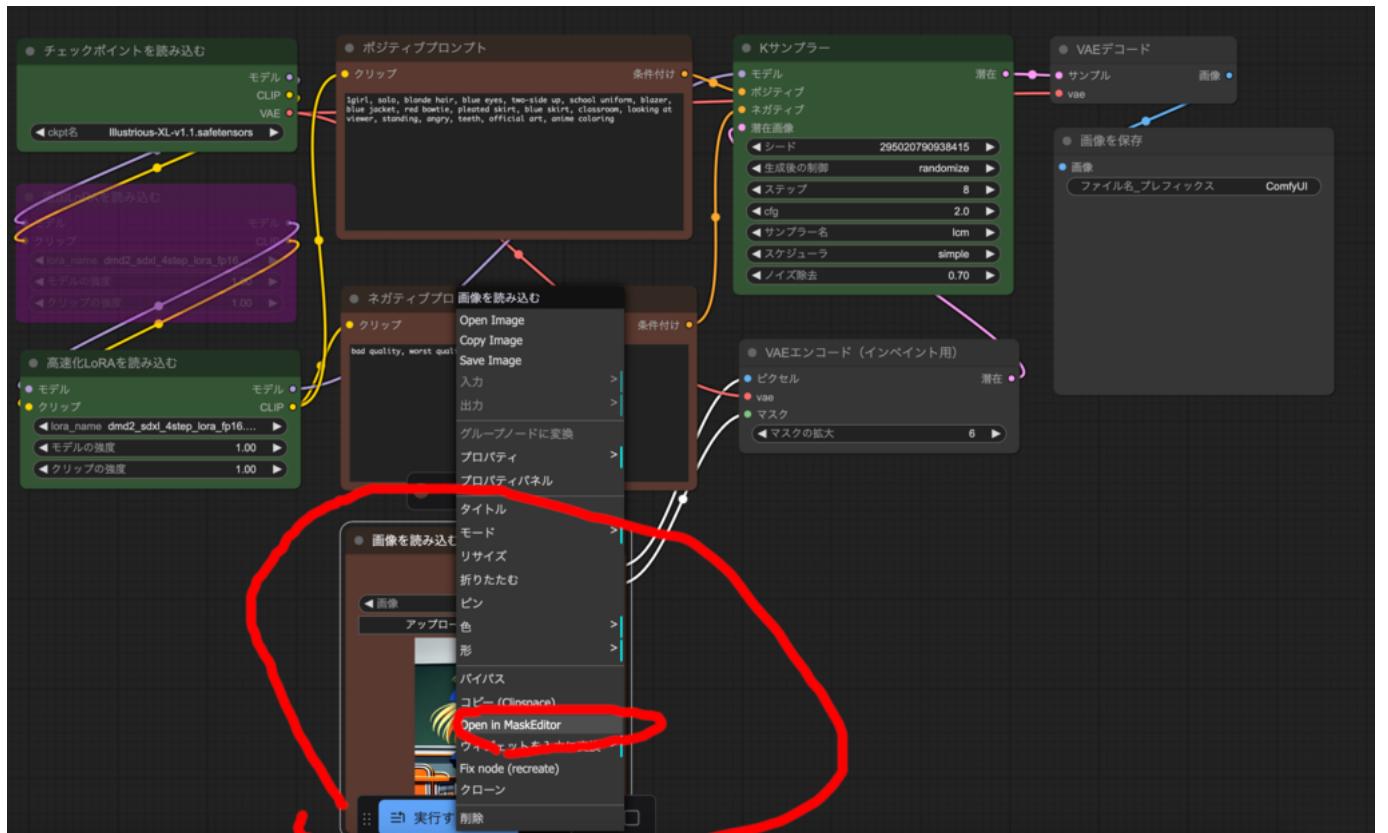
『1.2.2.画像生成I2I（出力から）.json』のワークフローはoutputフォルダから直接画像を読み込めるようにしました。以降のワークフローにも同様にoutputフォルダから読み込めるワークフローを用意している場合があるので、inputフォルダにアップロードするのが面倒な場合はこちらを使用してください。

インペイント

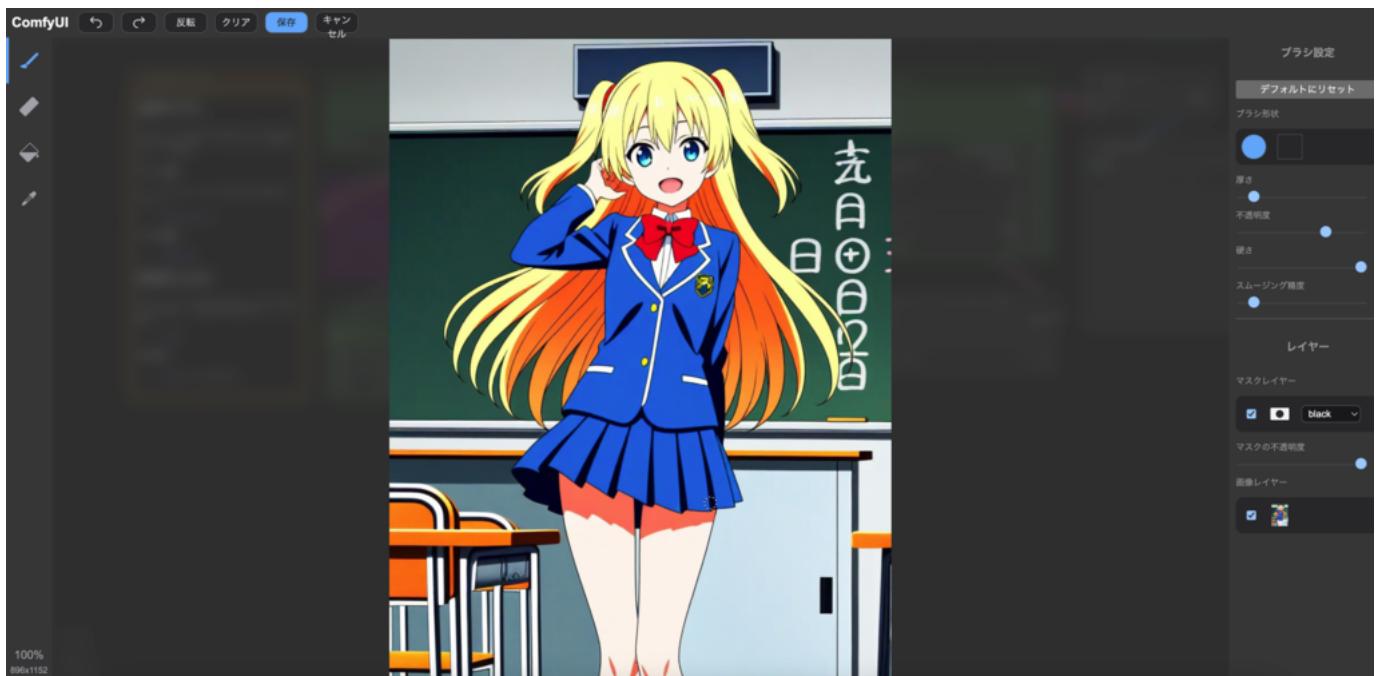
生成画像の一部だけ不満があり、一部だけ修正したい場合があります。

元画像に対して修正したい部分を塗って示し、その部分だけ I2I による画像生成を行うインペイント機能を使うことで実現できます。

『1.2.3.画像生成インペイント.json』ワークフローを開いて、中央下の『画像を読み込む』ノードで先ほどと同じ画像を選択してください。



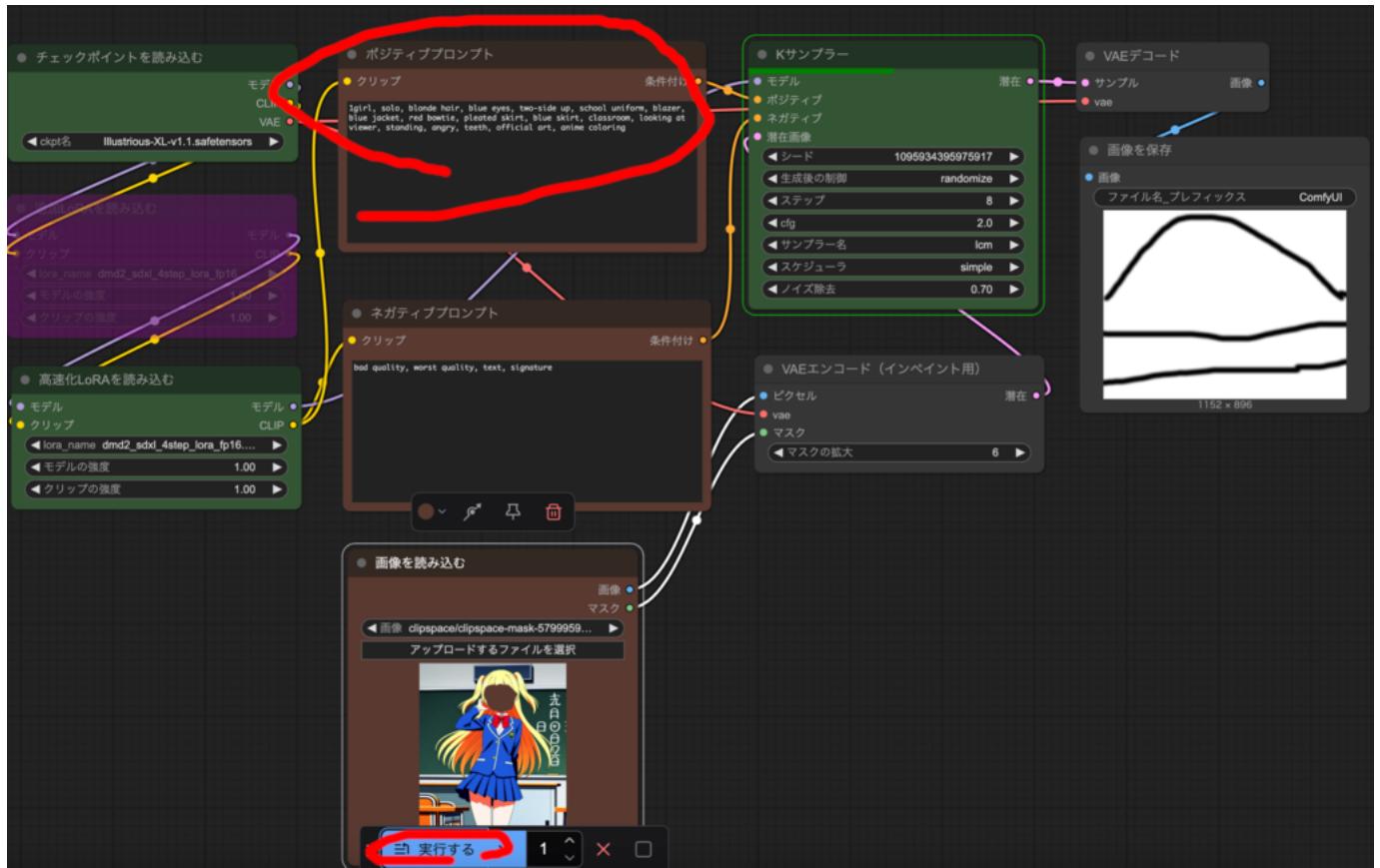
『画像を読み込む』ノードで右クリックをして、Open Mask Editor を選びます。



変更したい箇所をマウスでなぞって指定します。



とりあえず顔の辺りを塗ってください。上のメニューから保存を選びます。



ポジティブプロンプトに次の文を入れて実行するを押してください。

```
1girl, solo, blonde hair, blue eyes, two-side up, school uniform, blazer,
blue jacket, red bowtie, pleated skirt, blue skirt, classroom,
looking at viewer, standing, angry, teeth, official art, anime coloring
```

歯を出して怒っている(angry, teeth)画像が生成されるはずです。



インペイントを上手に使うと、望む画像を生成しやすくなると思います。

3.4 コントロールネットによる生成画像の制御

コントロールネットによる生成画像の制御

I2I は画像を直接参照して別な画像を生成しますが、画像から特徴や構図など画像の構成要素を抽出して別な画像を生成することもできます。コントロールネットといいます。

より求める画像に近いものを生成できるようになるはずです。

が、使用しているモデルの問題か、いまいち適切に反映されていないような気がします。また、コントロールネットを使用しない時よりかなり生成時間が増えます。

ただ、こういう技術が理解できるようにするためにワークフローと解説を用意しました。

ここからは標準のノードだけでなくカスタムノードも活用していくので、指示に従ってカスタムノードをインストールしてください。

落書きを元に生成

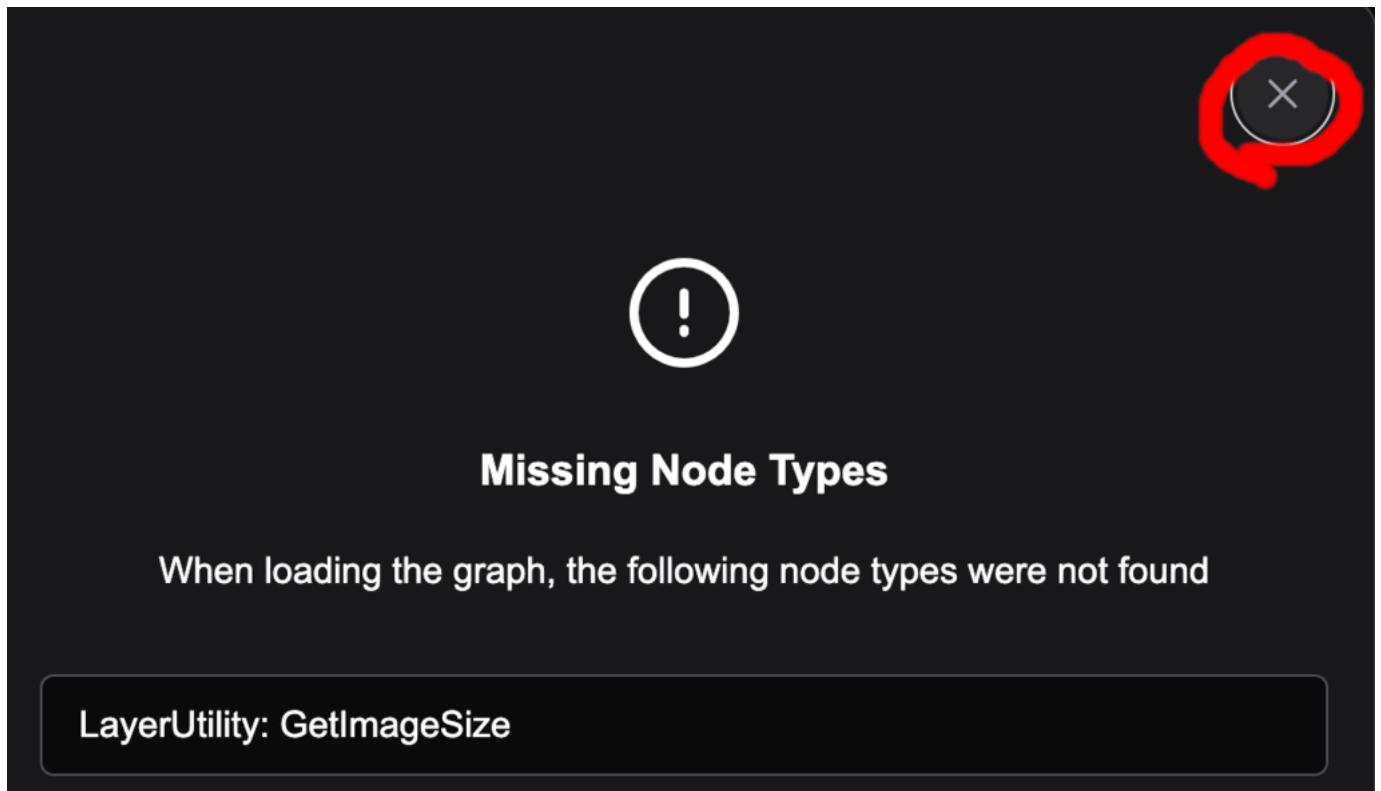
お絵描きソフトを使って元となる落書きや線画を描いてください。こちらでは Gimp を使いましたが、何で描いても OK です。



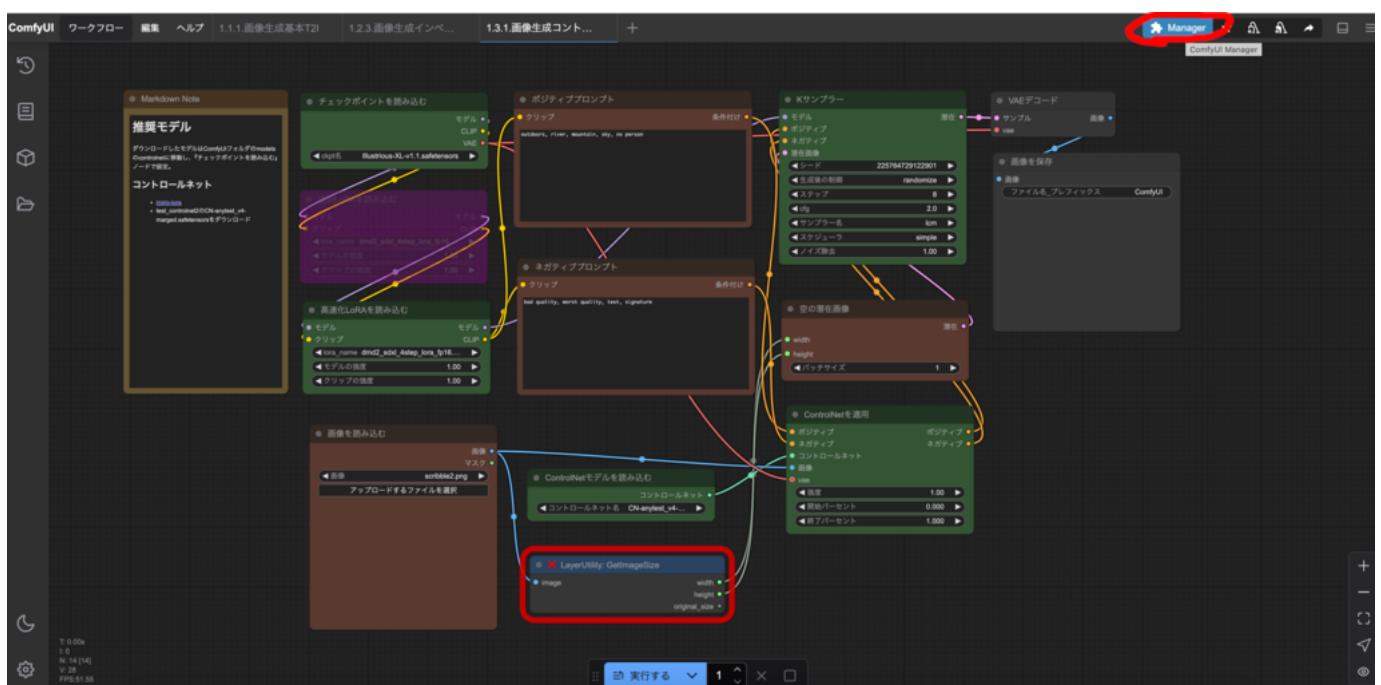
(山と川のつもりです)

この画像は catAITools フォルダの data フォルダに 01.png として入れてあります。

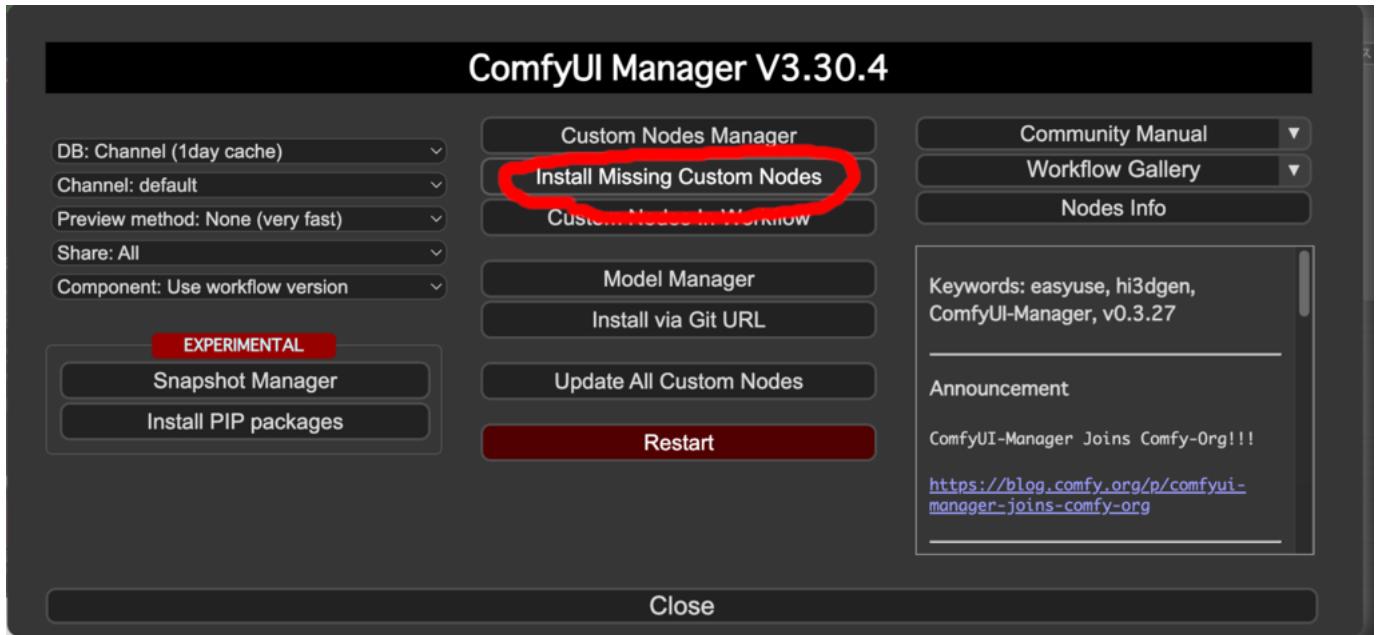
『1.3.1.画像生成コントロールネット落書き.json』を開いてください。



必要なカスタムノードがインストールされていないので警告がでます。LayerUtility カスタムノードが必要なので ComfyUI-Manager 経由でインストールします。



警告の右上の×を押して、右上の Manager を押します。

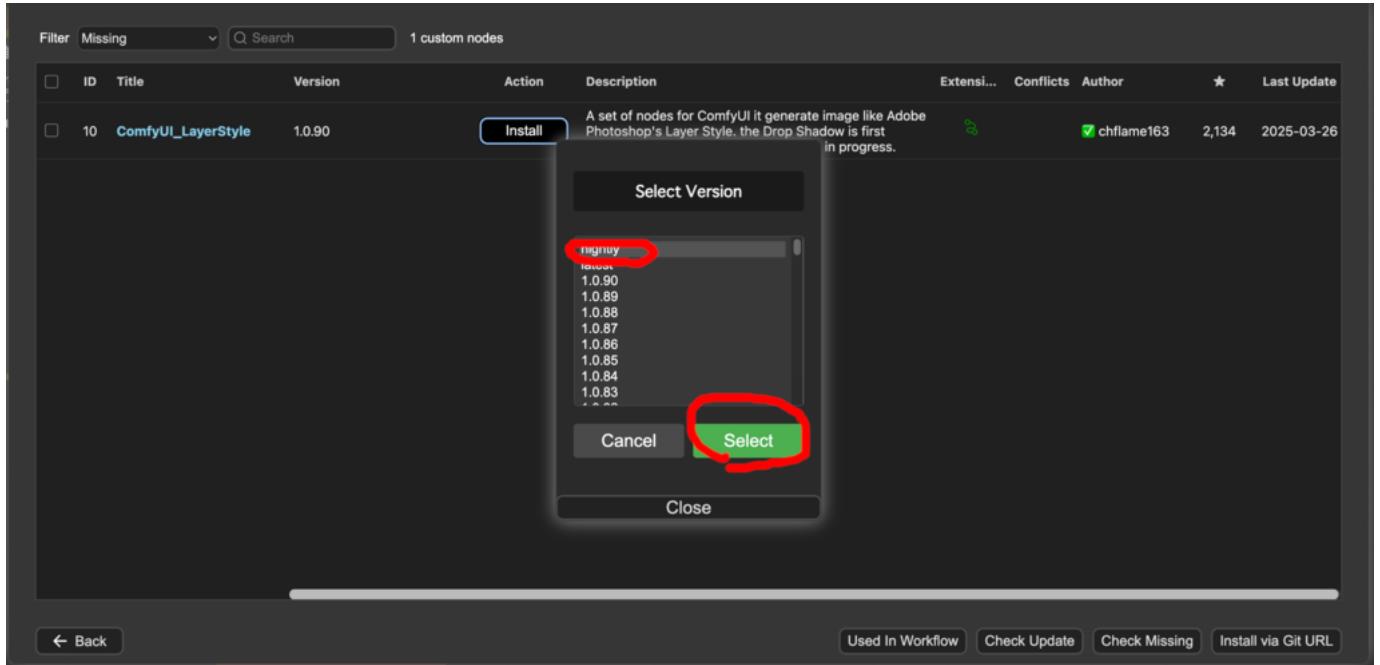


中央上から 2 番目の Install Missing Custom Nodes を押します。

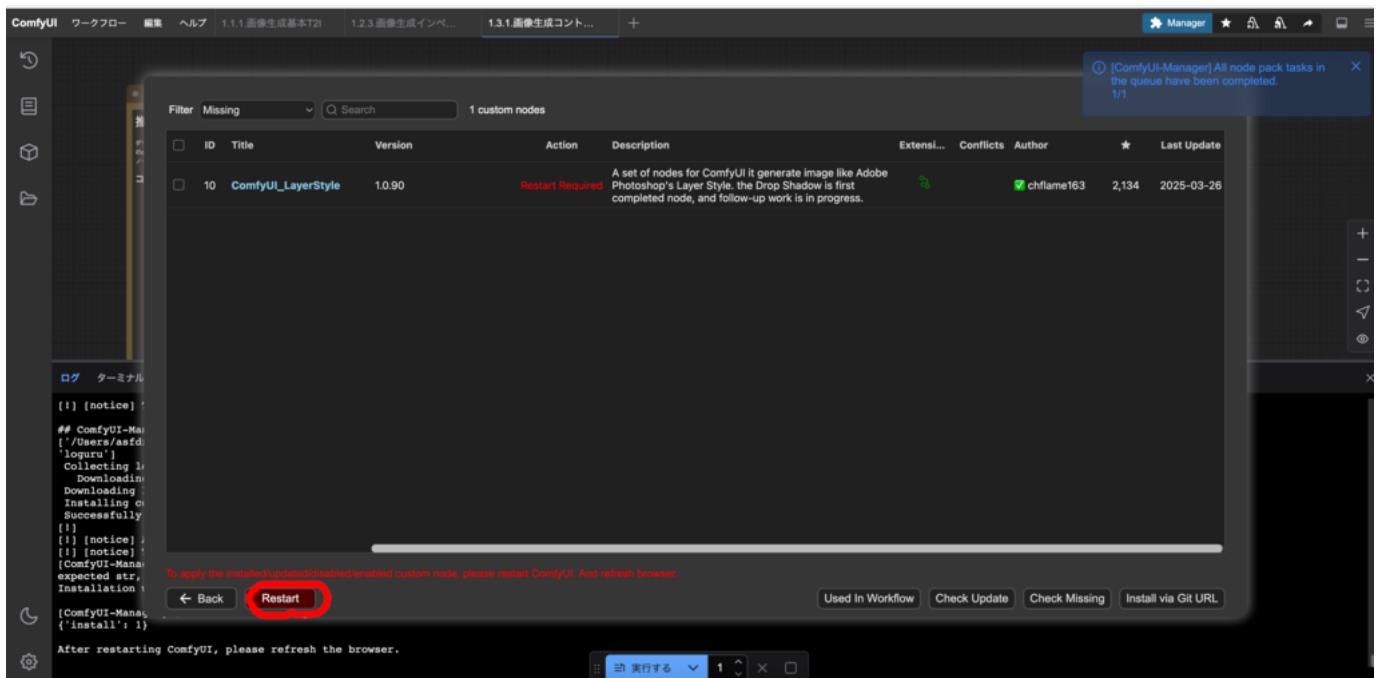
The screenshot shows the 'Missing' custom nodes list. At the top, there are filters for 'Missing' and a search bar. The table has columns: ID, Title, Version, Action, Description, Extension, Conflicts, Author, and Last Update. One row is shown: ID 10, Title ComfyUI_LayerStyle, Version 1.0.90, Action 'Install', Description 'A set of nodes for ComfyUI it generate image like Adobe Photoshop's Layer Style. the Drop Shadow is first completed node, and follow-up work is in progress.', Extension 'comfyui', Conflicts '0', Author 'chflame163', and Last Update '2025-03-26'. At the bottom, there are buttons for 'Used In Workflow', 'Check Update', 'Check Missing', and 'Install via Git URL'. There is also a 'Back' button.

ID	Title	Version	Action	Description	Extensi...	Conflicts	Author	Last Update
10	ComfyUI_LayerStyle	1.0.90	Install	A set of nodes for ComfyUI it generate image like Adobe Photoshop's Layer Style. the Drop Shadow is first completed node, and follow-up work is in progress.	comfyui	0	chflame163	2,134 2025-03-26

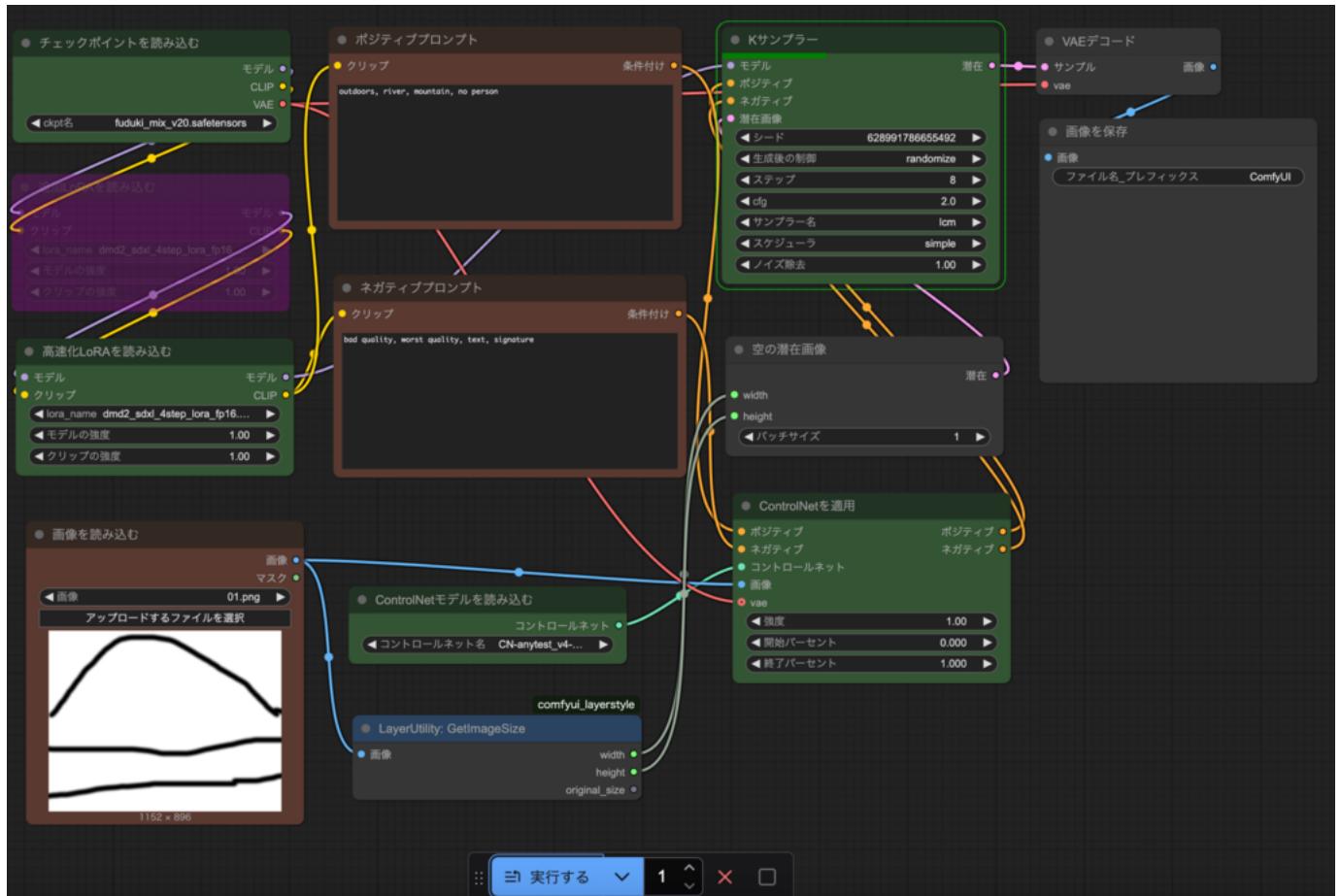
ComfyUI-LayerStyleというカスタムノードが必要なのでInstallを押してください。



バージョンはnightlyにしてください。Selectを押してインストールします。

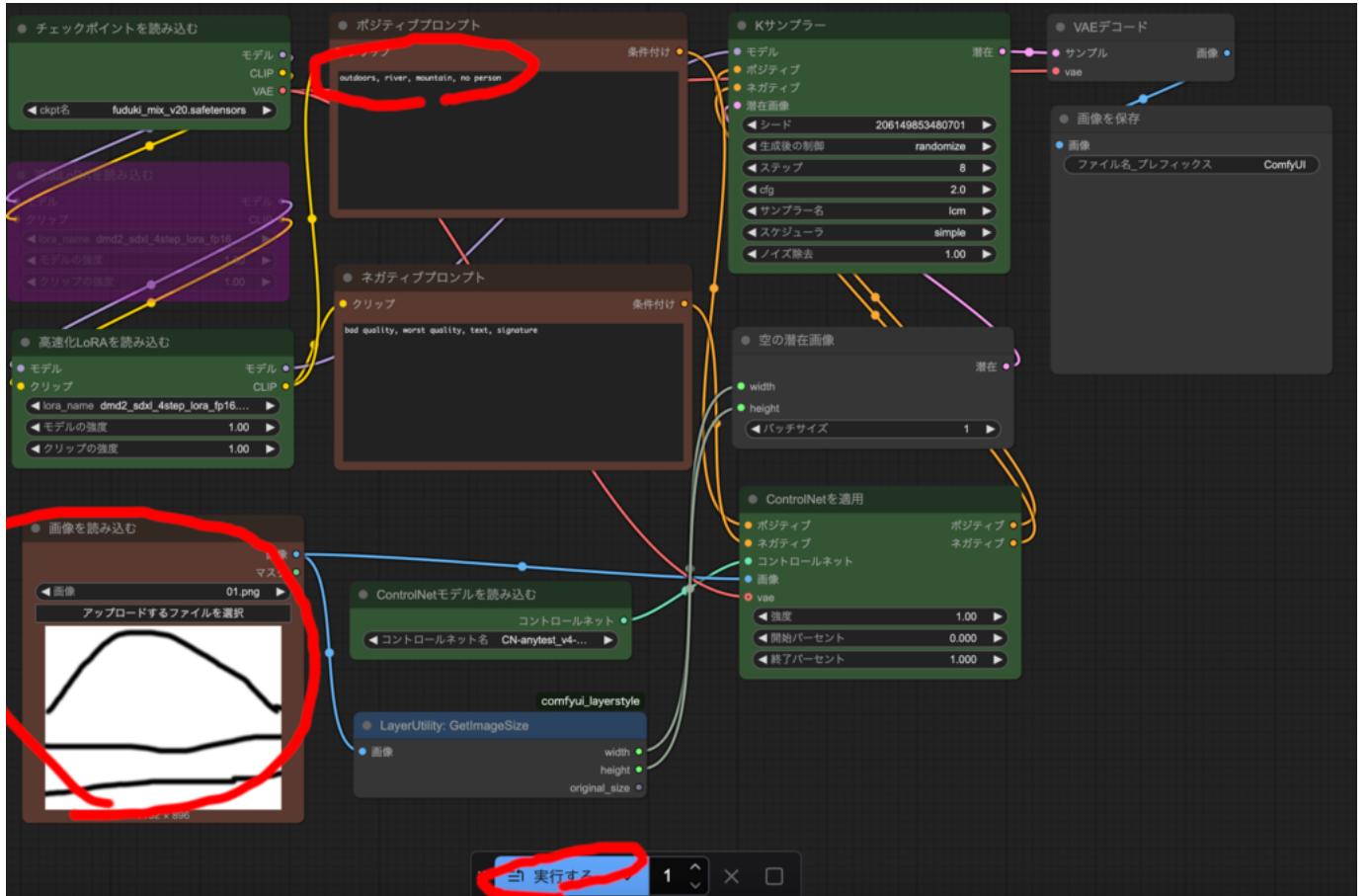


しばらくするとインストールが終わるので、Restartを押してください。



ComfyUI Desktopが再起動され、ワークフローが正常に開かれるはずです。

なお、使用しているカスタムノードは、元となる画像の幅と高さを取得して『空の潜在画像』ノードに送るためのものです。



『画像を読み込む』ノードで元となる落書き画像をアップロードし、ポジティブプロンプトに次の文を入れて実行するを押してください。

```
outdoors, river, mountain, sky, tree, no person
```





山のつもりの線の一部が枝になり、川のつもりの 2 本の線が 2 本の枝になってしまっているので、あまりうまくいっていませんが、元の落書きの線を反映した画像が生成できています。

深度情報

元の画像からどれくらいの距離かを濃淡で表す深度情報をAIから取得して、生成画像に反映することが可能です。

『1.3.2.画像生成コントロールネット深度情報I2I.json』を開いてください。

DownloadAndLoadDepthAnythingV2Model と DepthAnything_V2 という カスタムノードが必要です。深度モデルを読み込むノードと、そのモデルを利用して深度情報を生成するノードです。

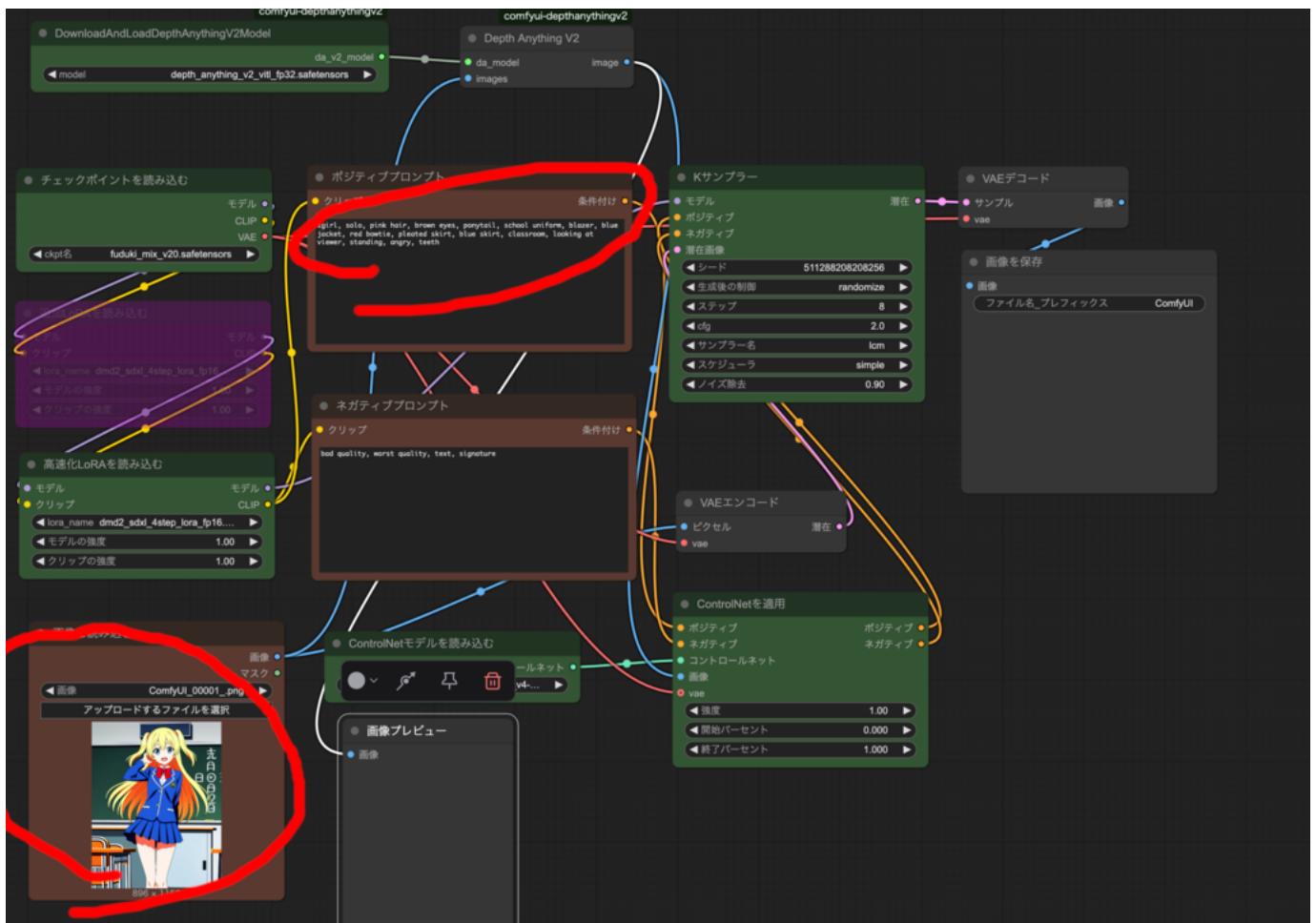
Filter Missing Search 1 custom nodes

ID	Title	Version	Action	Description	Extensi...	Conflicts	Author	★	Last Update
161	ComfyUI-DepthAnythingV2	1.0.1	Restart Required	ComfyUI nodes to use DepthAnythingV2 NOTE: Models autodownload to ComfyUI/models/depthanything from https://huggingface.co/Kijai/DepthAnythingV2-safetensors/tree/main			Kijai	287	2025-03-06

To apply the installed/updated/disabled/enabled custom node, please restart ComfyUI. And refresh browser.

[Back](#) [Restart](#) [Used In Workflow](#) [Check Update](#) [Check Missing](#) [Install via Git URL](#)

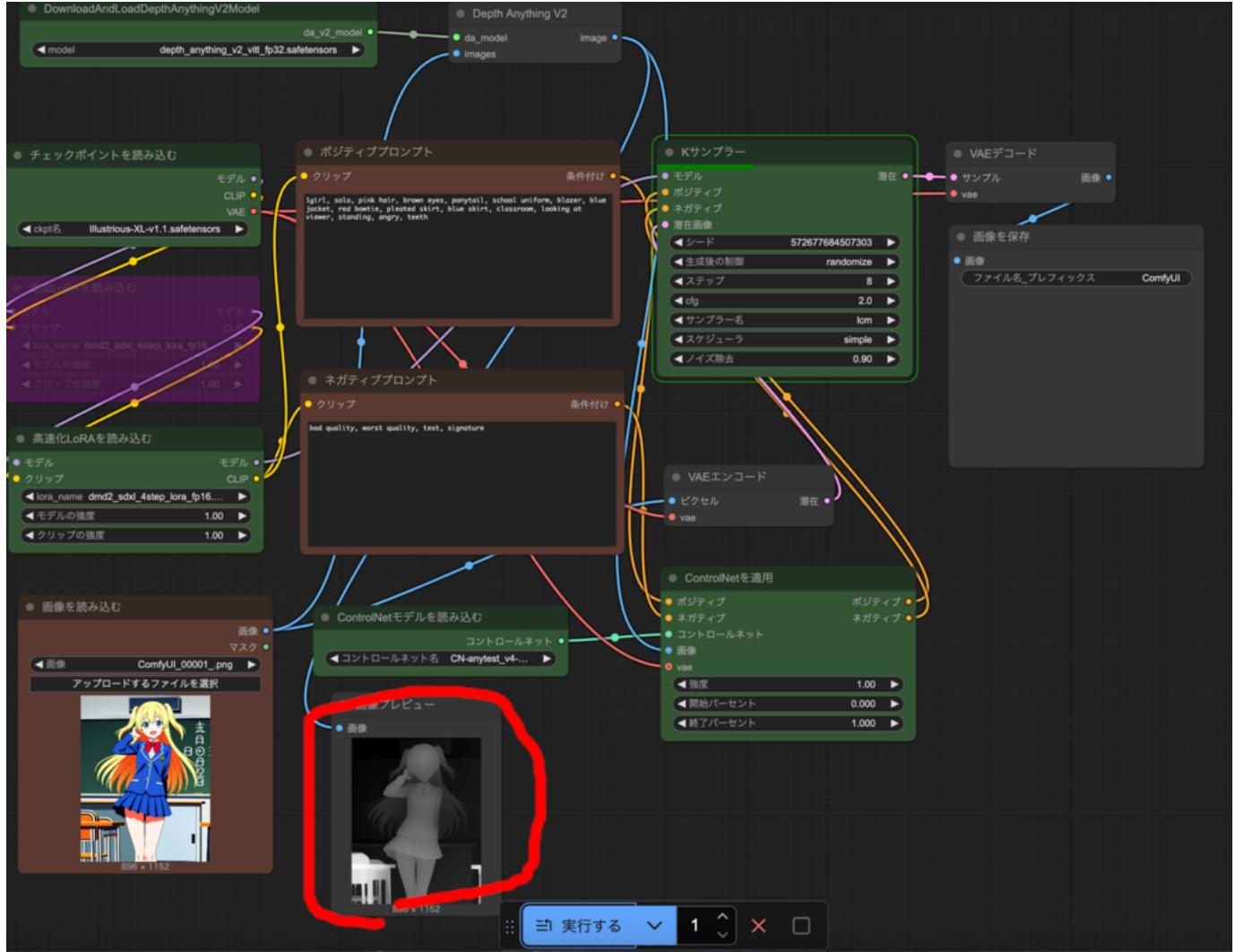
先ほどと同様に ComfyUI_Manager で Install Missiong Node から ComfyUI-DepthAnythingV2 をインストールしてして Restart を押して、ComfyUI Desktop を再起動してください。



『画像を読み込む』ノードで、生成した少女画像を選び、ポジティブプロンプトに次の文を入れて実行するを押してください。

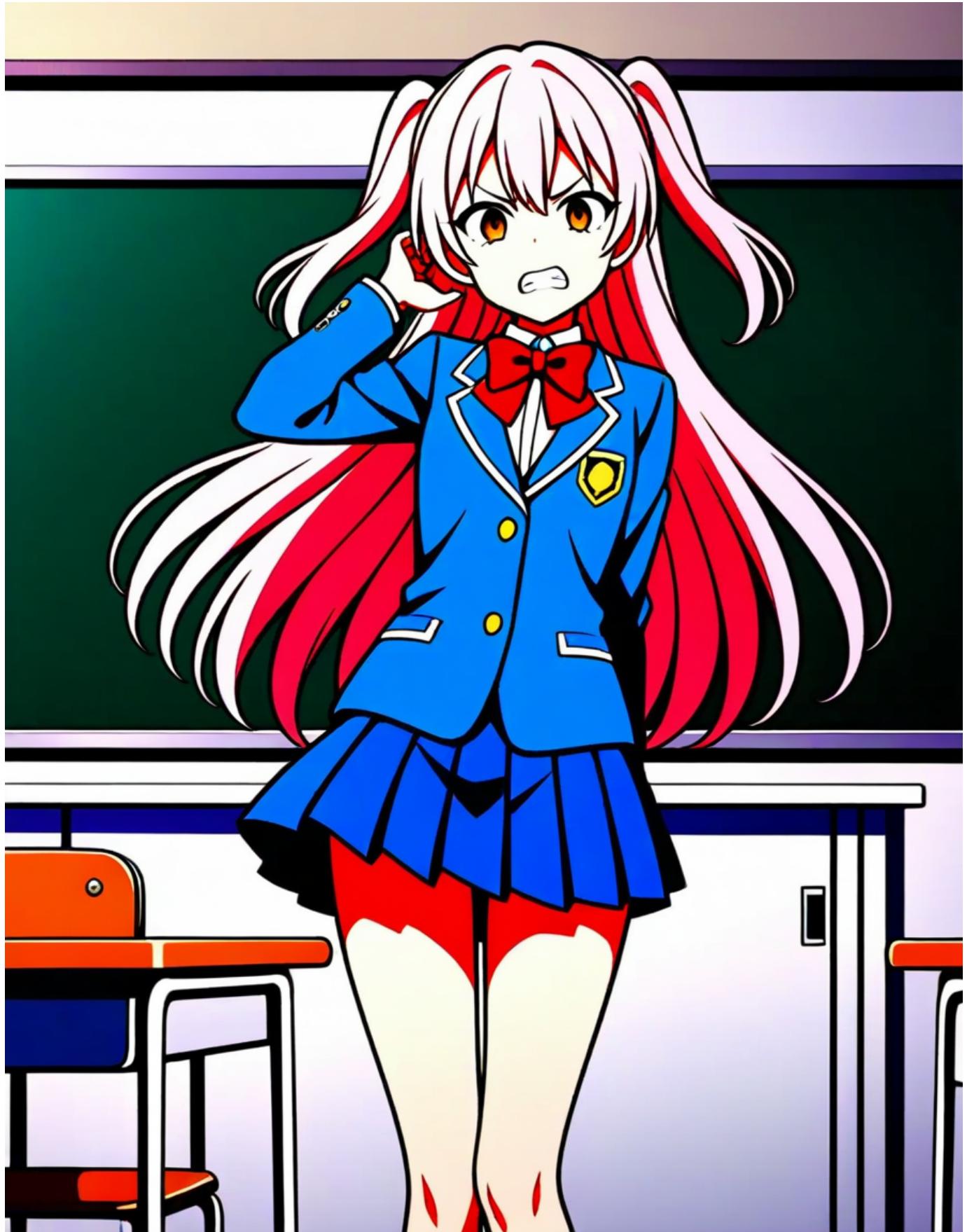
```
1girl, solo, pink hair, brown eyes, ponytail, school uniform,
blazer, blue jacket, red bowtie, pleated skirt, blue skirt,
classroom, looking at viewer, standing, angry, teeth, official art,
anime coloring
```

最初に深度情報生成用のモデルをダウンロードするので、初回実行時は時間がかかります。



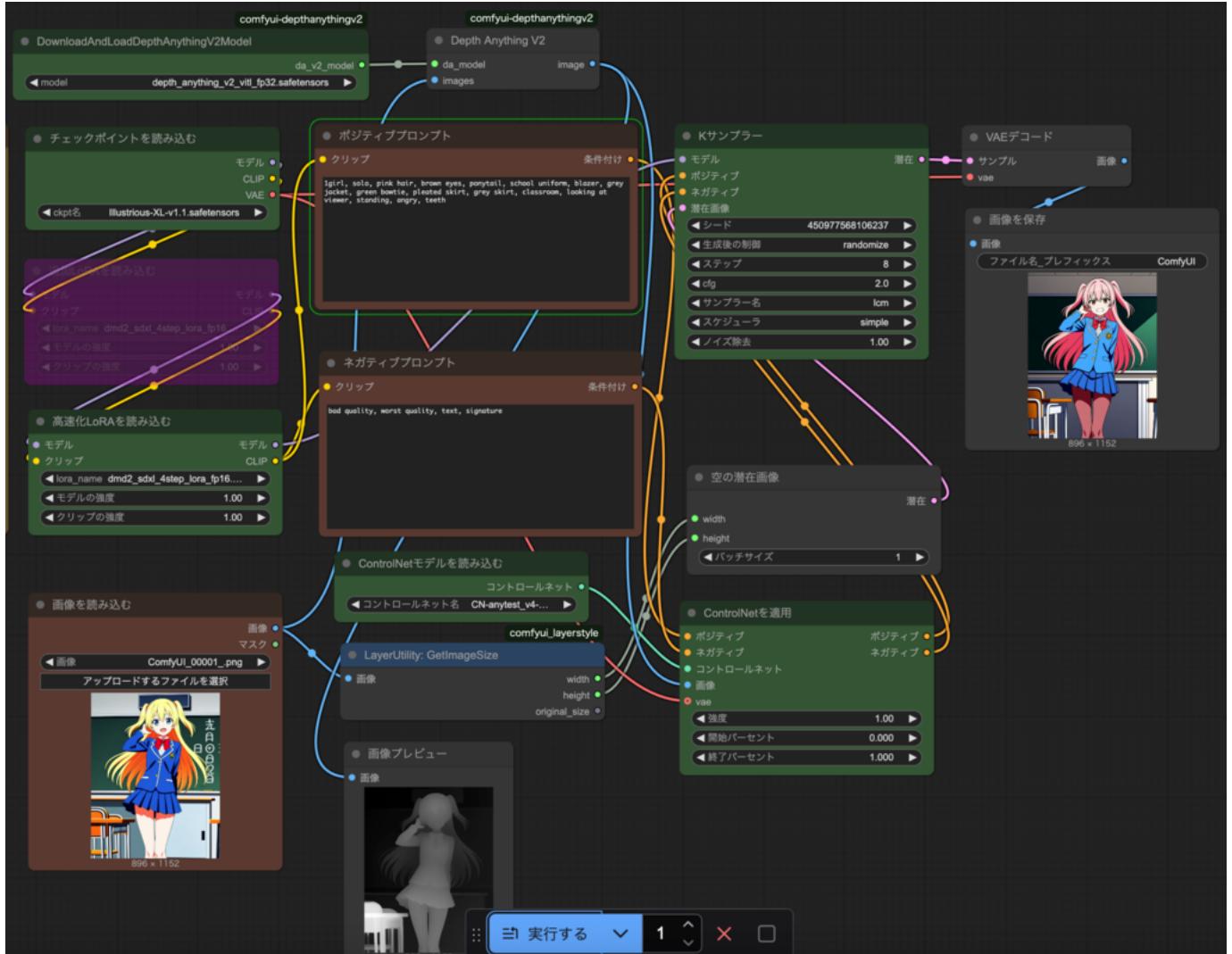
最初に生成された深度情報が下の『画像プレビュー』ノードの方に表示されます。

しばらくすると画像が生成されるはずです。



深度情報と元の画像の両方に基づき、元の画像の構図をある程度維持したまま、プロンプトの内容であるピンク髪茶色目ポニーテール少女が怒っている画像を生成できているはずです(髪型は元の画像の深度情報の影響を受けてツインテールのままになっています)。

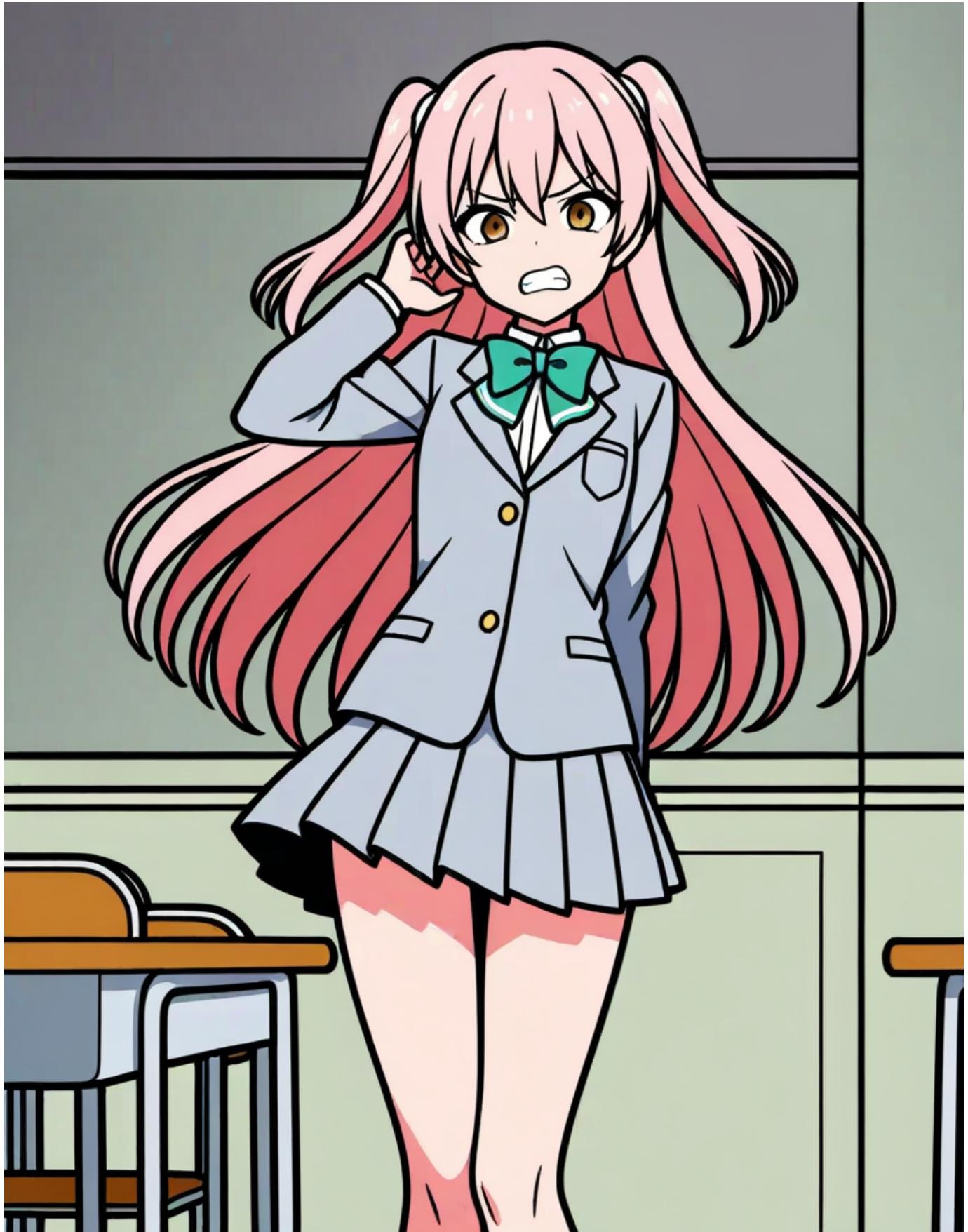
『1.3.4.画像生成コントロールネット深度情報T2I.json』を開いてください。



こちらはT2Iなので、元の画像そのものは参照せず、プロンプトと元画像の深度情報のみに基づき画像を生成します。

次のプロンプトで生成してみてください。

```
igirl, solo, pink hair, brown eyes, ponytail, school uniform, blazer,
grey jacket, green bowtie, pleated skirt, grey skirt, classroom,
looking at viewer, standing, angry, teeth
```



元の画像の深度情報に基づく人物の輪郭や左下の机と椅子の輪郭はきちんと反映され、それ以外はプロンプトに基づく画像になっているはずです。

生成したいことに合わせて I2I と T2I を使い分けてください。

OpenPose

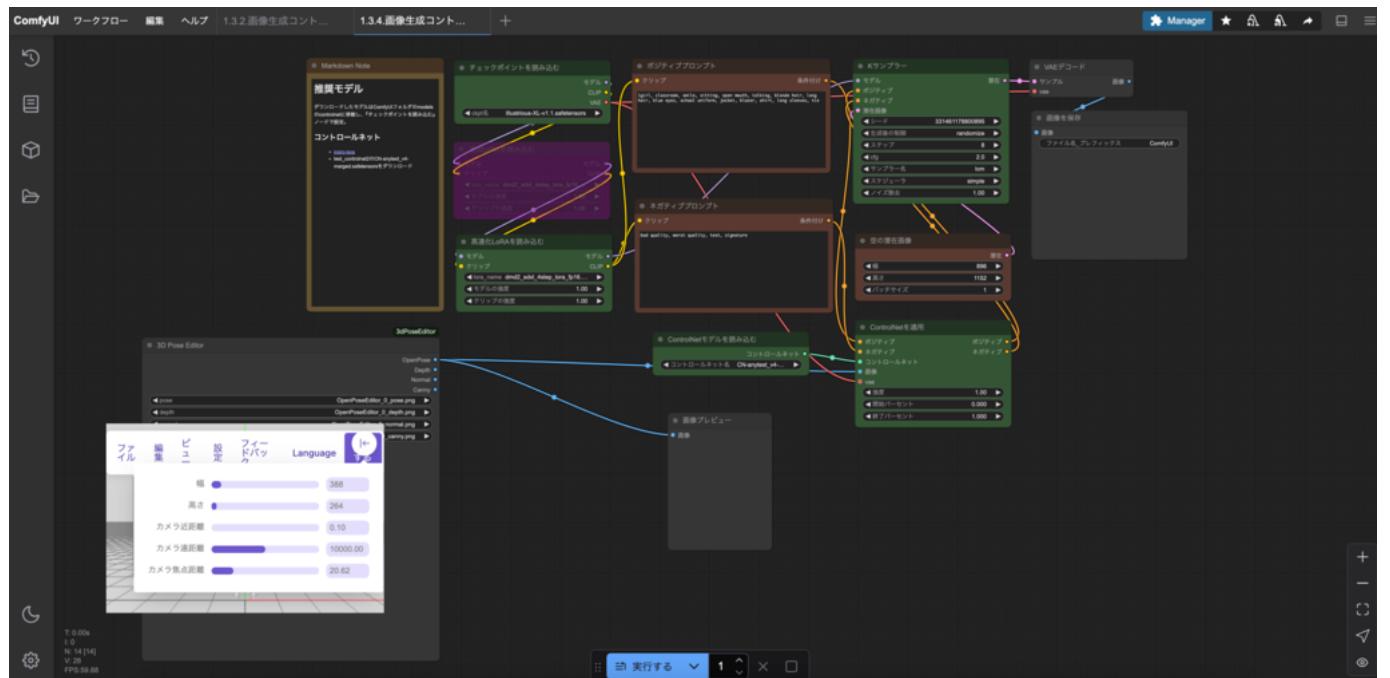
ポーズエディタでポーズを指定して画像を生成することもできます。

注意事項

ポーズの指定方法は一般的な 3D モデリングソフトと同じですが、3D ソフトを使ったことがない人は操作方法がわかりにくいかかもしれません。また、このカスタムノードはあまり安定していないので、誤動作しやすいです。

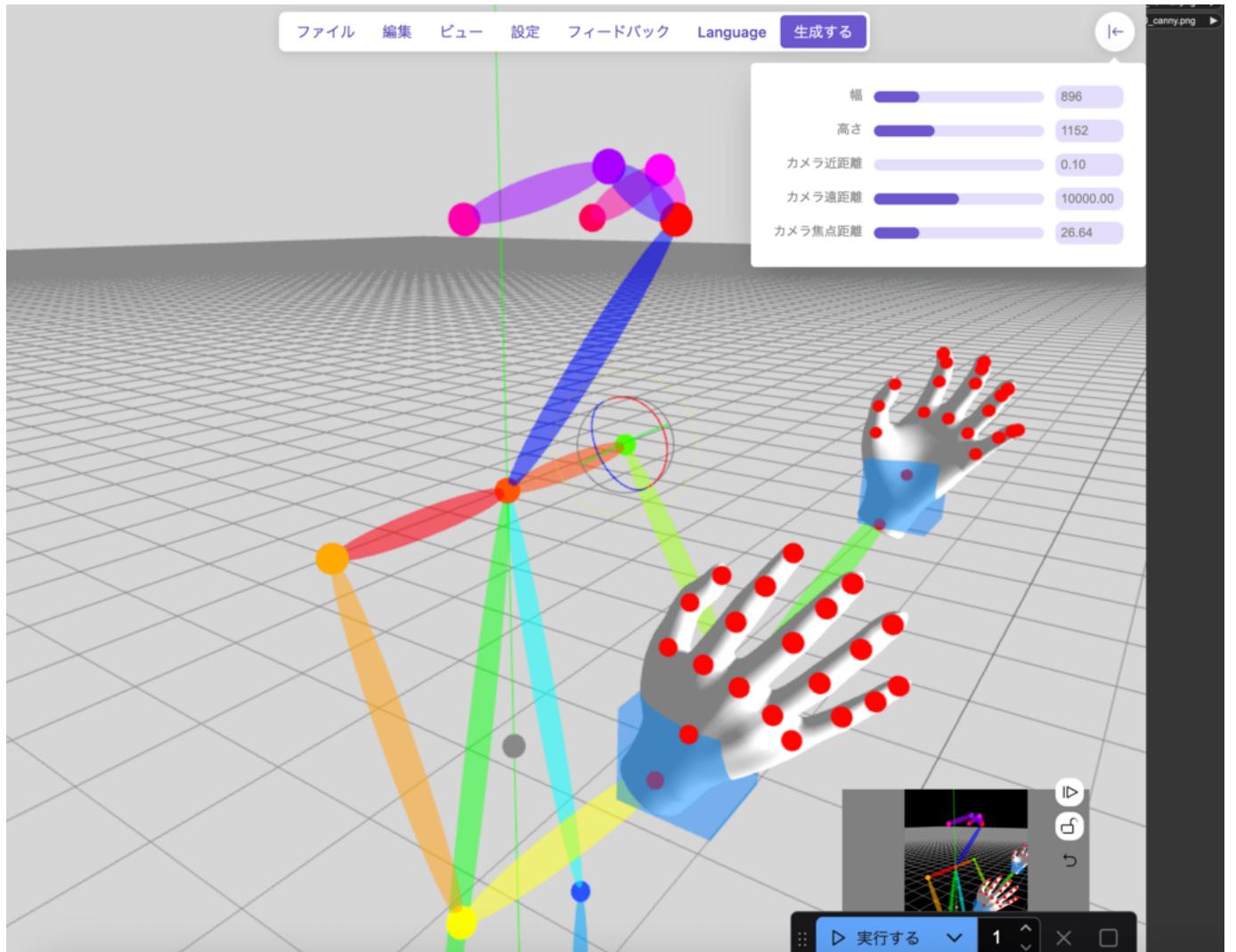
『1.3.4.画像生成コントロールネットOpenPoseT2I.json』を開いてください。

Hina.PoseEditor3D ノードが必要なので、同様に ComfyUI-Manager から ComfyUI 3D Pose Editor をインストールして、Restart を押して ComfyUI Desktop を再起動してください。OpenPose 形式でポーズを編集するノードです。

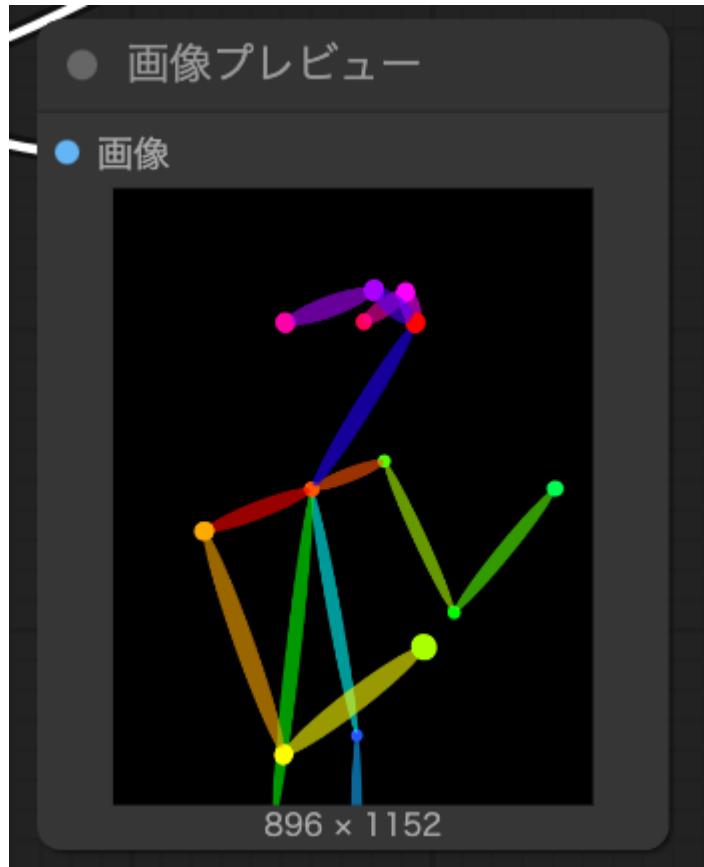


左下の 3D Pose Editor ノードを拡大します。ノードの右端をドラッグしてください。

幅を 896、高さを 1152 にします。マウスの左ボタンを押しながら移動することで右向きにし、ホイールを回して拡大し、右ボタンを押しながら移動することで、上半身だけにしてください。さらに設定の移動モードを選んで、手や足の丸を摘んで移動させて、移動モードをもう一度押して回転モードにして、次のようにしてください。



これで右向きで両手を前に出した構図で画像を生成できます。



次のプロンプトで生成してみてください。

```
1girl, solo, black hair, brown hair, long hair, school uniform, blazer, blue jacket, red bowtie, classroom, smile, open mouth, profile, lookin to the side, official art, anime coloring
```



いまいち反映されていない気がしますが、ある程度は制御できている気がします。

右下の ControlNet の適用ノードの強度を変更することでどれくらいポーズを反映するか指定できます。

3D Pose Editor のメニューのファイルの画像から検出を選ぶと 画像からポーズを検出することもできます。ただし、写真や リアル調の画像のみ対応しています。

OpenPose 形式でポーズ指定できるポーズエディタは他にもあるので ([こちらが本家](#))、別途ポーズを作成して、『1.3.1.画像生成コントロールネット落書き.json』ワークフローで ポーズ画像を読み込ませて画像生成することも可能です。

文章だけで構図を指定するのは難しいので、線画や既存の画像の深度情報や OpenPose 等で構図を反映できれば、より望んだ画像を生成しやすくなると思います。

3.5 AIによる画像加工

画像拡大

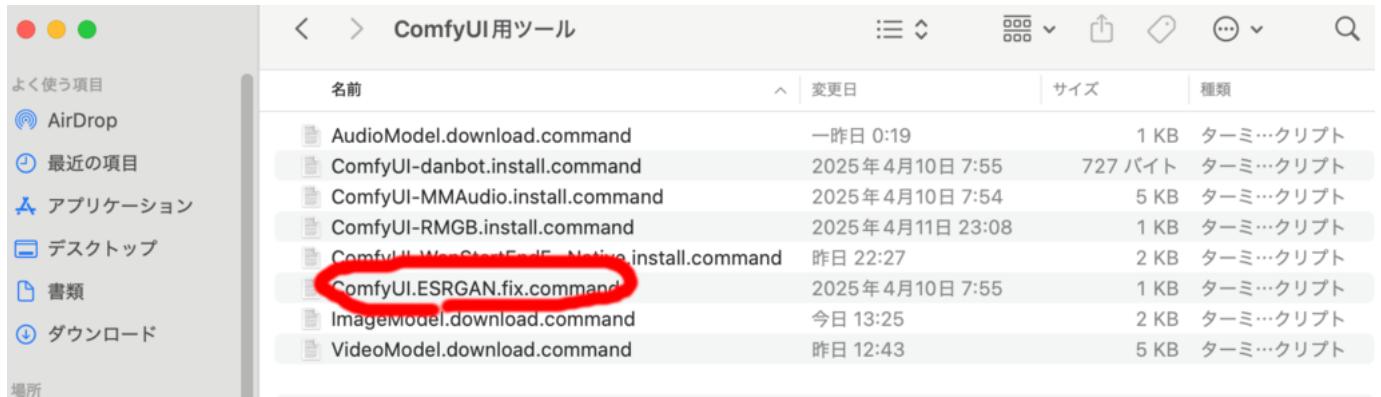
ESRGANコードの修正

画像の拡大に ESRGAN というAI技術を使います。

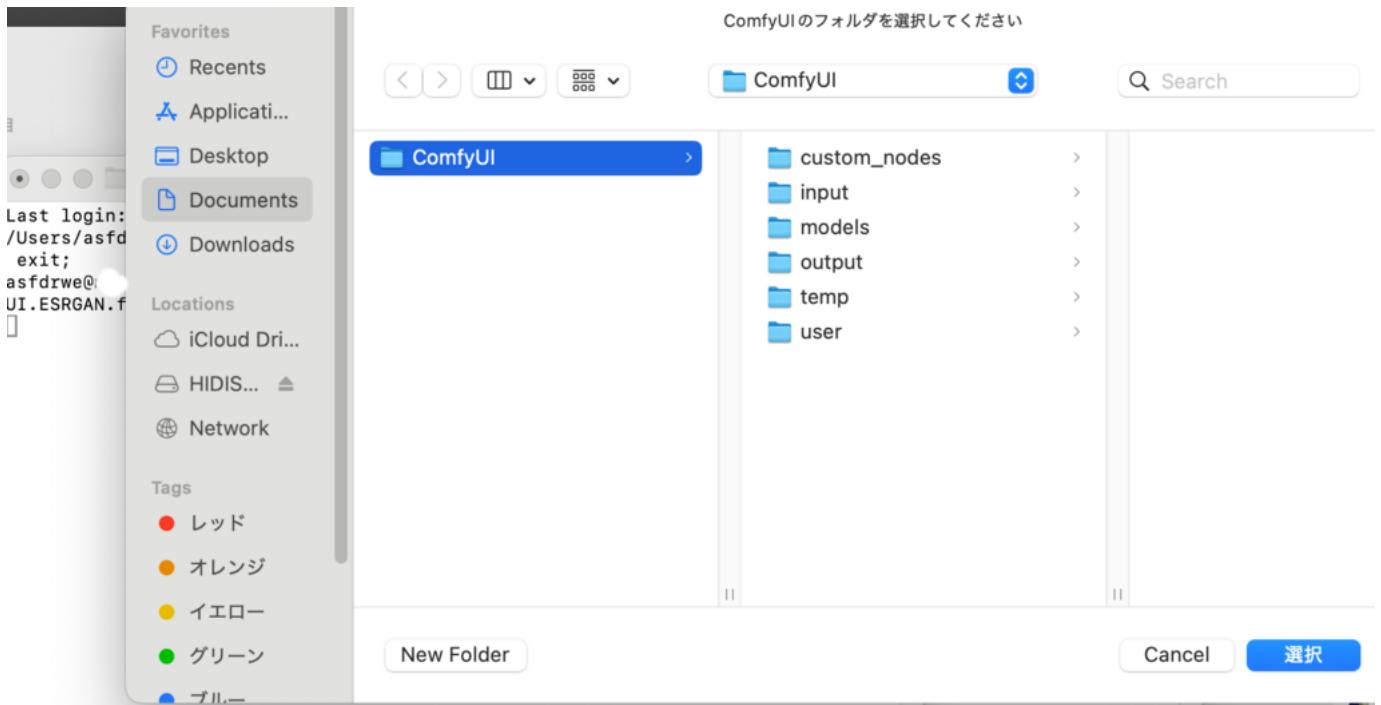
ただ、ComfyUI Desktop がインストールする PyTorch 関連にはバグがあり macOS では ESRGAN 関連が正常に動作しません。修正するユーティリティを用意しました。

一旦 ComfyUI Desktop を閉じてください。

catAIToolsフォルダのComfyUI用ツールフォルダを開いてください。



中にある『ComfyUI.ESRGAN.fix.command』をダブルクリックしてください。

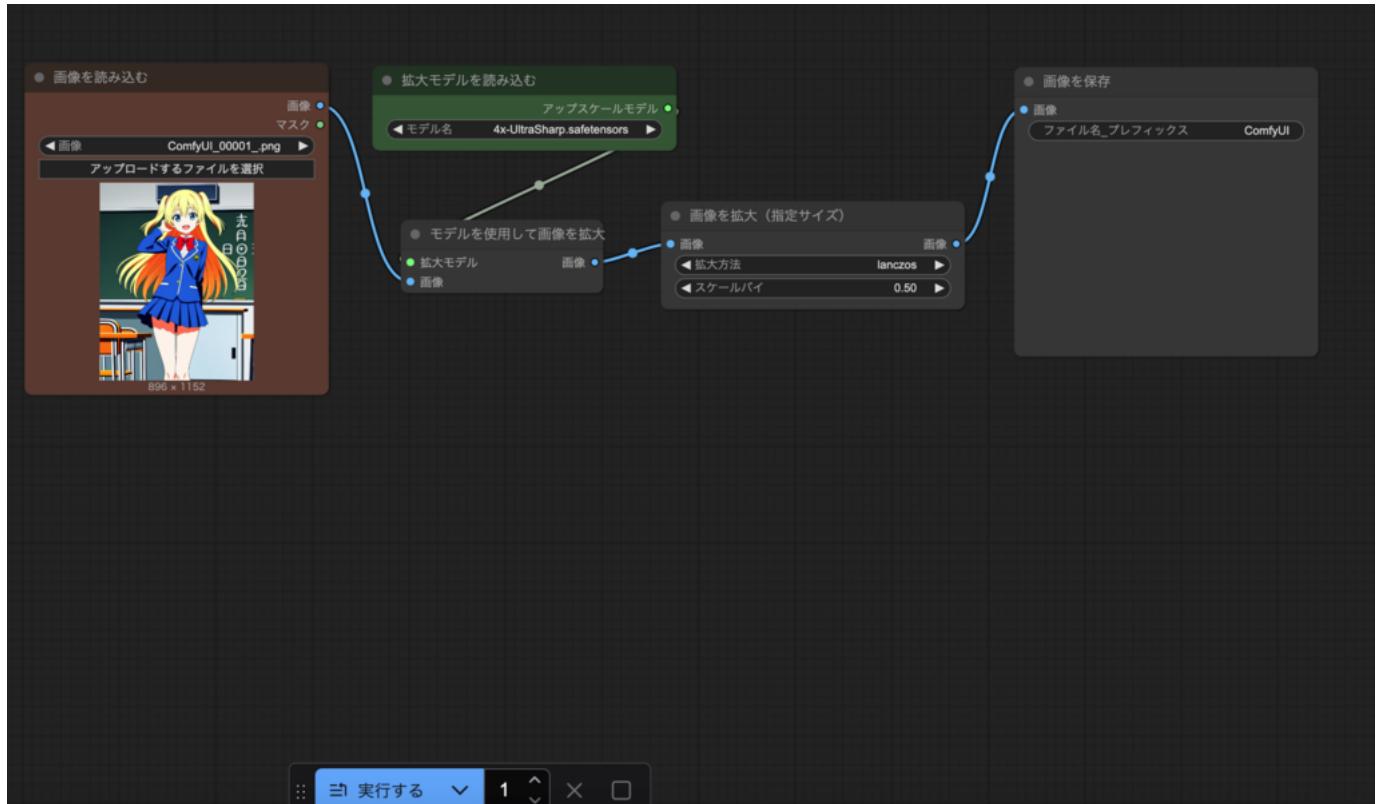


ComfyUI のインストールフォルダを選び、『選択』を押せばコード修正が適用されます。ツールによって開かれたターミナルは閉じてください。

もう一度 ComfyUI を起動してください。

画像拡大の実行

『1.4.1.画像2倍拡大.json』を開いてください。



『画像を読み込む』ノードで拡大する画像を選んでください。

一旦 ESRGAN技術で 4 倍にしてから lanczos アルゴリズムで半分に縮小することで、細部への描写を反映させつつ2倍に拡大できます。



背景削除

AIが背景を認識して背景を削除しその部分を透過にしたり クロマキー用緑や青にできます。

RMBG カスタムノードが必要です。が、カスタムノードの requirements.txt が macOS を考慮していないのでそのままではインストールできません。

インストールできるよう修正してインストールするスクリプトを用意したので、 ComfyUI Desktop を一旦終了させください。

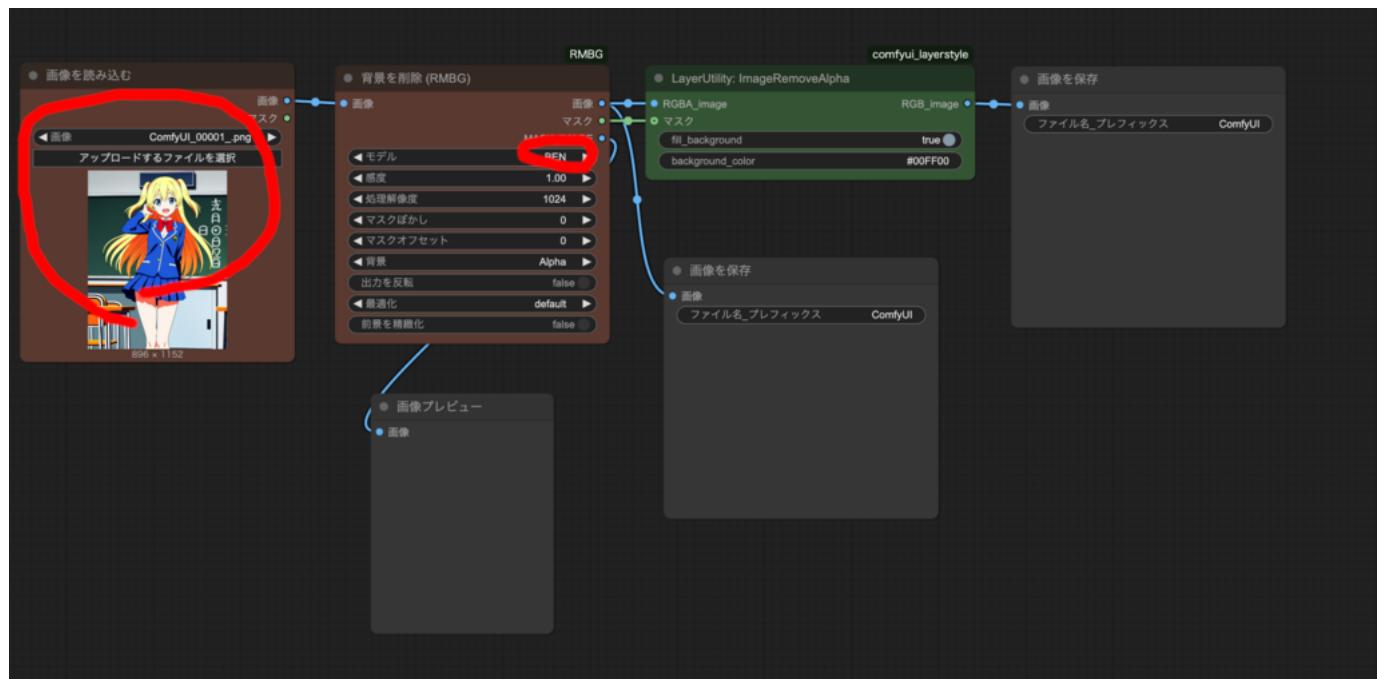
名前	変更日	サイズ	種類
AudioModel.download.command	一昨日 0:19	1 KB	ターミ…クリプト
ComfyUI-danbot.install.command	2025年4月10日 7:55	727 バイト	ターミ…クリプト
ComfyUI-MMAudio.install.command	2025年4月10日 7:54	5 KB	ターミ…クリプト
ComfyUI-RMBG.install.command	2025年4月11日 23:08	1 KB	ターミ…クリプト
ComfyUI-WanStartEnd...Native.install.command	昨日 22:27	2 KB	ターミ…クリプト
ComfyUI-ESRGAN.fix.command	2025年4月10日 7:55	1 KB	ターミ…クリプト
ImageModel.download.command	今日 13:25	2 KB	ターミ…クリプト
VideoModel.download.command	昨日 12:43	5 KB	ターミ…クリプト

catAITools フォルダの『ComfyUI用ツール』フォルダの『ComfyUI-RMBG.install.command』をダブルクリックしてください。

ComfyUI のインストールフォルダを選び、選択を押せばコード修正が適用されます。ターミナルは閉じてください。

もう一度 ComfyUI を起動してください。

『1.4.3.画像背景除去.json』を開いてください。



『画像を読み込む』ノードで画像を選択してください。

実行するを押せば、モデルが自動的にダウンロードされ、背景を自動で削除して透過にしたものとグリーンバックにしたもののが生成されるはずです。

『背景を削除（RMBG）』ノードのモデルには BEN、BEN2、INSPYRENET、RMGB-2.0 の 4 種類とあり、どれが一番うまくいくかはやってみたいとわからないので、いまいちの場合は別なモデルで試してみてください。

- BEN 背景切り抜きミスがある上に足の一部が切り抜かれています。



• BEN2 背景切り抜きミスがあります。



• INSPYRENET ごくわずかに背景切り抜きミスがあります。



• RMBG-2.0 問題ないようです。

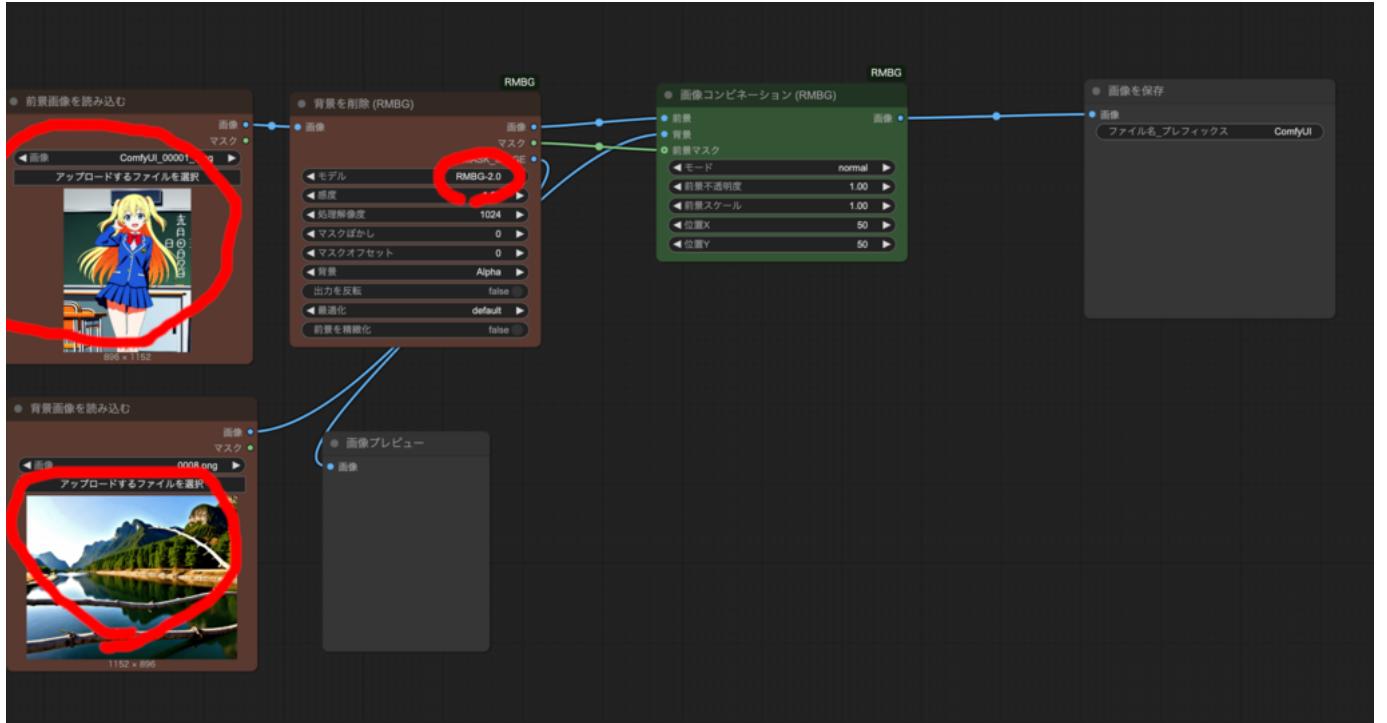


『LayerUtility:ImageRemoveAlpha』ノードの background_color で背景色を変えられます。'#00FF00' の場合グリーンバックに、'#0000FF' の場合ブルーバックになります。

背景合成

さらに削除した背景部分に別な背景を差し込むこともできます。

『1.4.5.画像背景合成.json』を開いてください。



『画像を読み込む』ノードで背景を削除して前景にする画像を読み込み、『背景画像を読み込む』ノードで背景とする画像を読み込んでください。『背景を削除 (RMBG)』ノードのモデルを RMBG-2.0にしてください。前景画像を背景画像の中央に配置するために『画像コンビネーション (RMBG)』ノードの位置Xと位置Yは 50 にしてください。

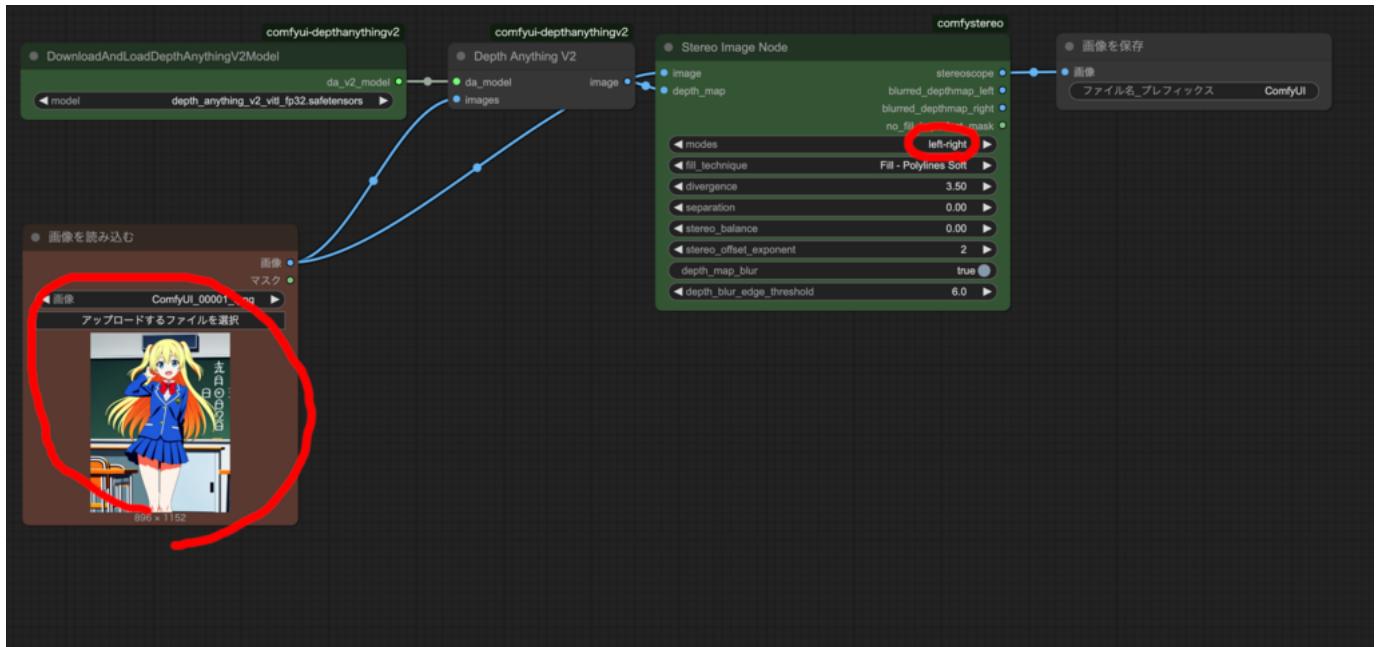
実行するを押せば前景画像の背景が削除され背景画像が埋め込まれるはずです。



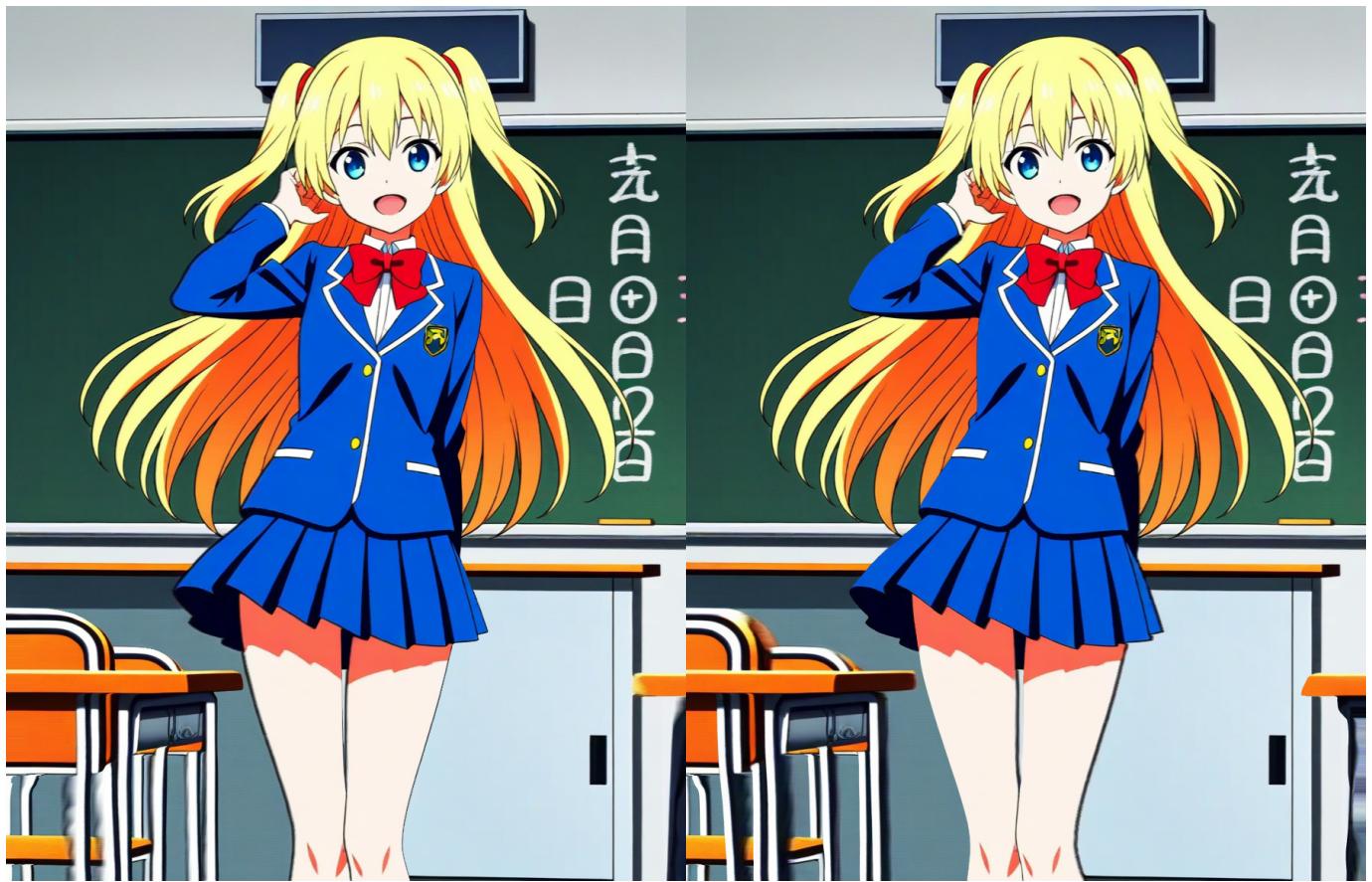
ステレオ画像生成

『1.4.7.ステレオ画像化.json』を開いてください。

StereoImageNode カスタムノードが必要なので、ComfyUI-Manager でインストールして、ComfyUI Desktop を再起動してください。

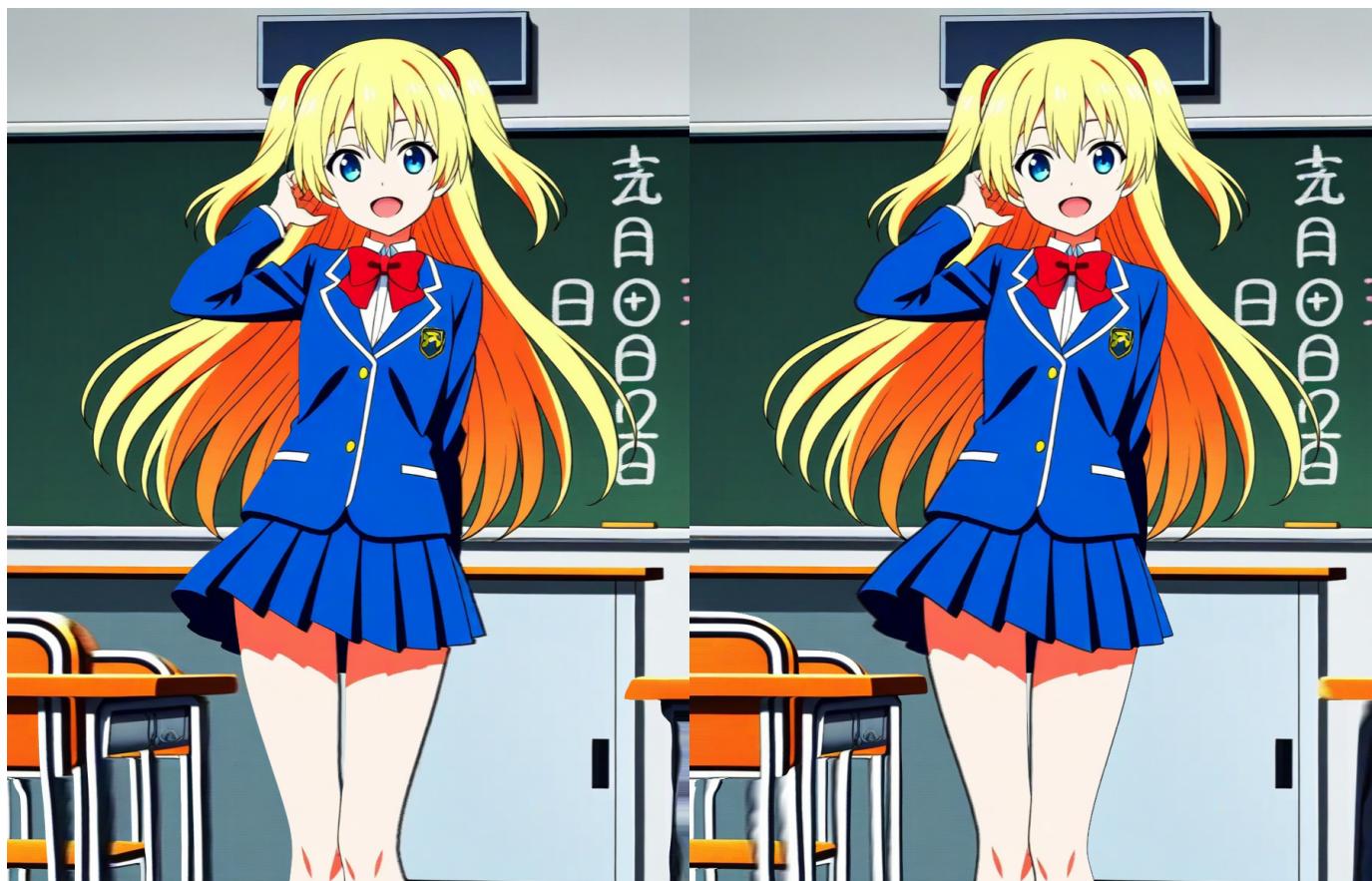


実行するを押すと深度情報に基づき LR 形式のステレオ画像が生成されます。



VR ビューアで見れば立体視できるはずです。

『Stereo Image Node』ノードの modes を right-left にすれば RL 形式（交差法用）の画像を生成することもできます。交差法での立体視は慣れれば裸眼で可能なのでお試しください（参考: [交差法での立体視の解説](#)）。



3.6 その他

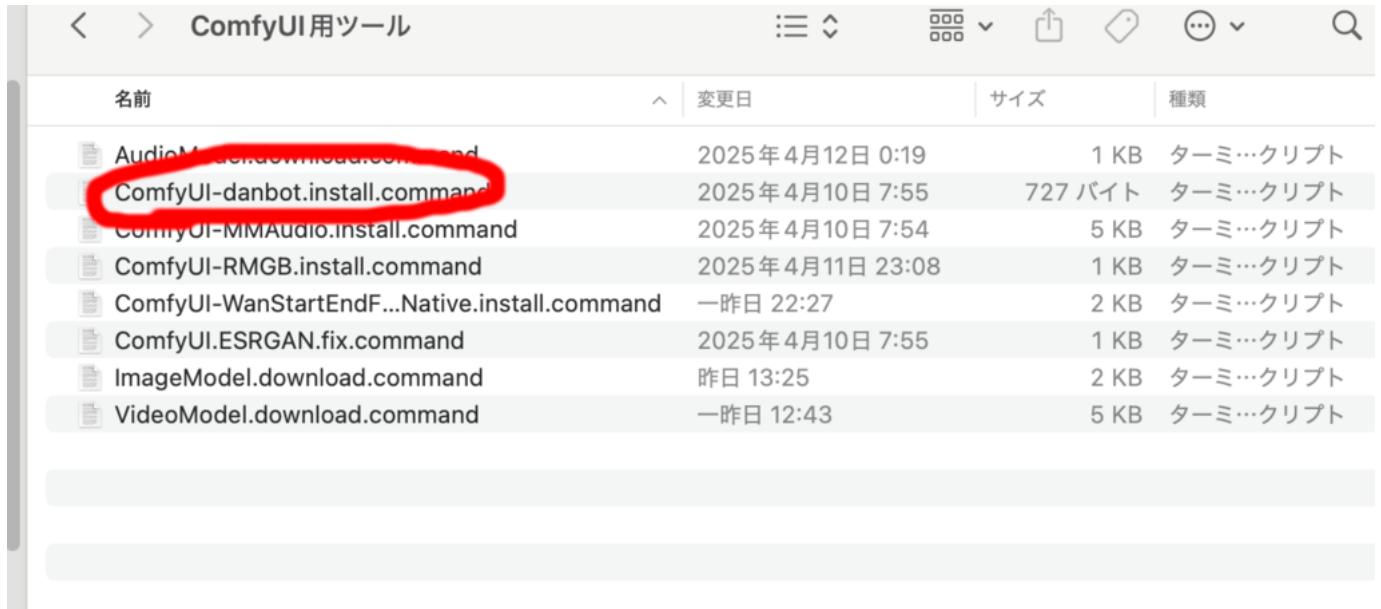
日本語文章による画像生成

生成する画像に対する指示は英語のプロンプトで行います。英語が不慣れな方には難しいので、日本語の文章を英語プロンプトに変換するワークフローを用意しました。

ただ、2025/4/11時点では、非常に新しいカスタムノードのせいか、カスタムノードの評価がされていない関係で、ComfyUI-Manager ではそのままではセキュリティ上インストールできません。

ComfyUI フォルダの user フォルダの default フォルダの ComfyUI-Manager フォルダにある『config.ini』の security_level を weak にしてインストールすることもできますが、インストール用スクリプトを用意したので、そちらからインストールしてください。

一旦 ComfyUI Destkop を終了してください。

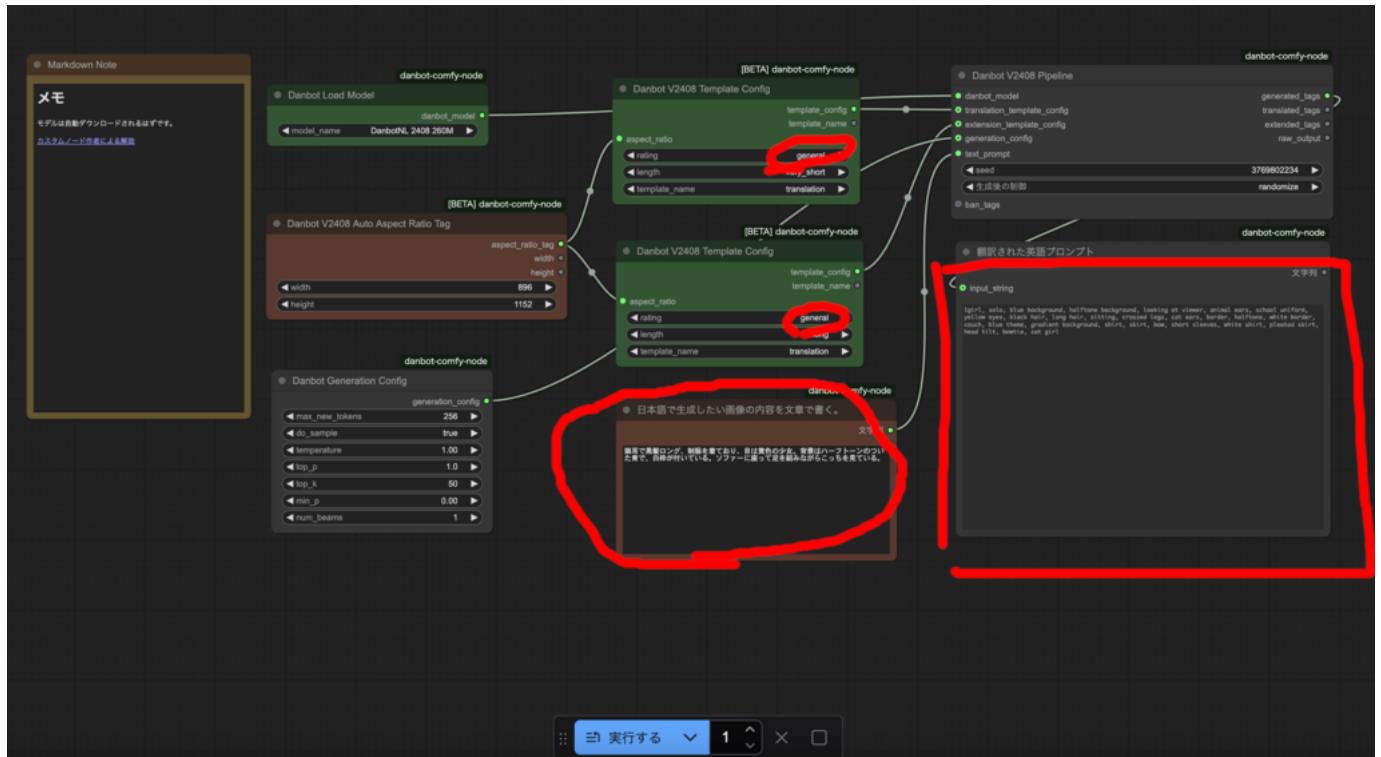


名前	変更日	サイズ	種類
AudioModel.download.command	2025年4月12日 0:19	1 KB	ターミ…クリプト
ComfyUI-danbot.install.command	2025年4月10日 7:55	727 バイト	ターミ…クリプト
ComfyUI-MMAudio.install.command	2025年4月10日 7:54	5 KB	ターミ…クリプト
ComfyUI-RMGB.install.command	2025年4月11日 23:08	1 KB	ターミ…クリプト
ComfyUI-WanStartEndF...Native.install.command	一昨日 22:27	2 KB	ターミ…クリプト
ComfyUI-ESRGAN.fix.command	2025年4月10日 7:55	1 KB	ターミ…クリプト
ImageModel.download.command	昨日 13:25	2 KB	ターミ…クリプト
VideoModel.download.command	一昨日 12:43	5 KB	ターミ…クリプト

『ComfyUI-danbot.install.command』をダブルクリックで起動し、ComfyUI のインストールフォルダを選択すればインストールされます。

もう一度 ComfyUI Desktop を起動してください。

『1.5.1.日本語文章英語プロンプト化.json』を開いてください。



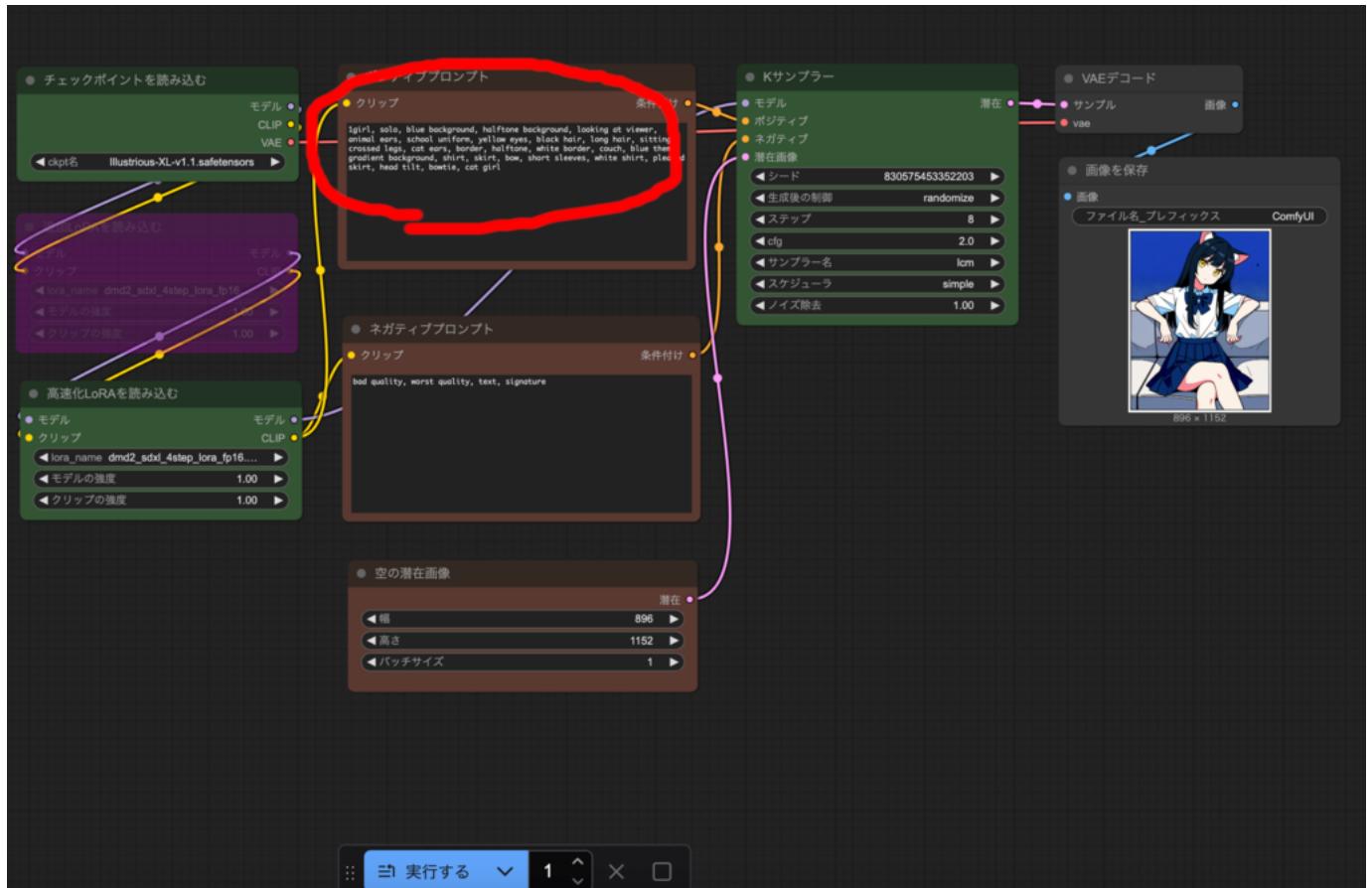
『中央下の日本語で生成したい画像の内容を文章で書く』ノードに生成したい画像について日本語の文章で描いてください。

猫耳で黒髪ロング、制服を着ており、目は黄色の少女。背景はハーフトーンの青で、白枠が付いている。ソファーに座って足を組みながらこっちを見ている。

実行するを押すと、モデルが自動的にダウンロードされ、右の翻訳された英語プロンプトに対応する英語プロンプトが表示されます。

```
1girl, solo, blue background, halftone background, looking at viewer,
animal ears, school uniform, yellow eyes, black hair, long hair,
sitting, crossed legs, cat ears, border, halftone, white border,
couch, blue theme, gradient background, shirt, skirt, bow,
short sleeves, white shirt, pleated skirt, head tilt, bowtie, cat girl
```

これをコピーして『1.1.1.画像生成基本T2I.json』ワークフローを開き、



ポジティブプロンプトに貼り付けて画像生成してください。



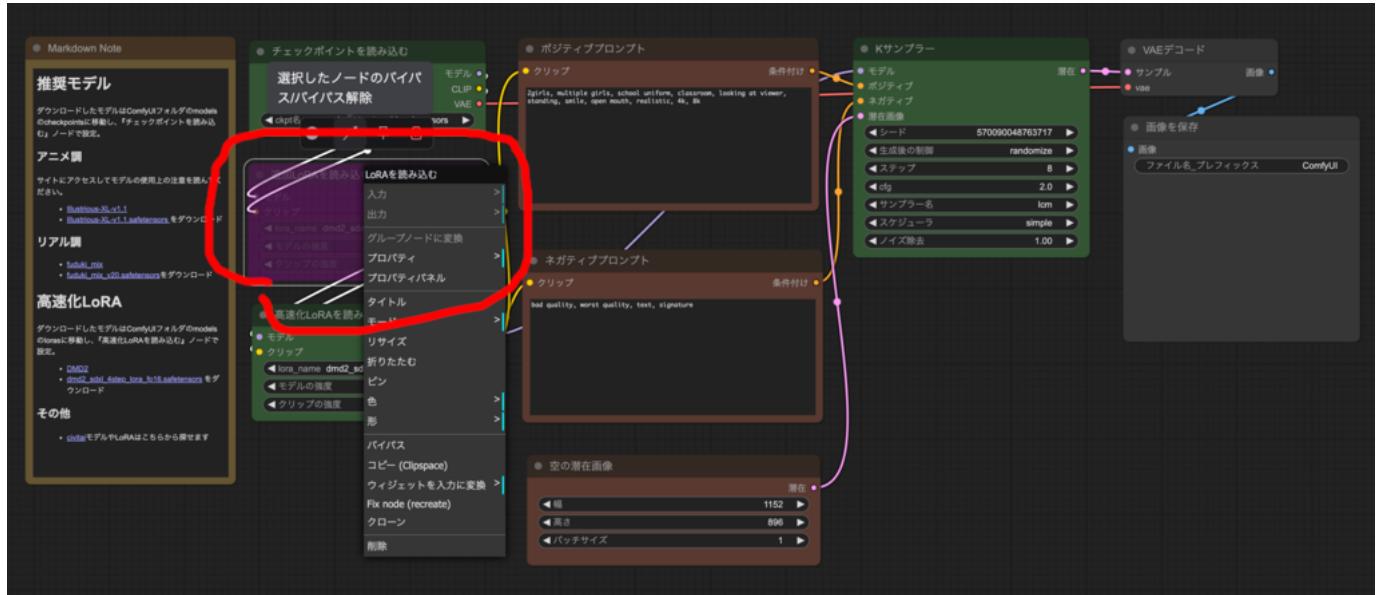
カスタムノードの作者より [技術的な詳細を含む解説](#)があるで参考にしてください。

なお、中央上段のノードと中段のノードで rating が general(一般的)になっています。この設定の場合、『一般的でない画像を表現した日本語文章』でも、そのような部分を削除して一般的な画像向けの英語プロンプトが生成されます。一般的でない画像を生成したい場合はここを変更してください(カスタムノードの作者が触れていないのでわざとぼかしています)。

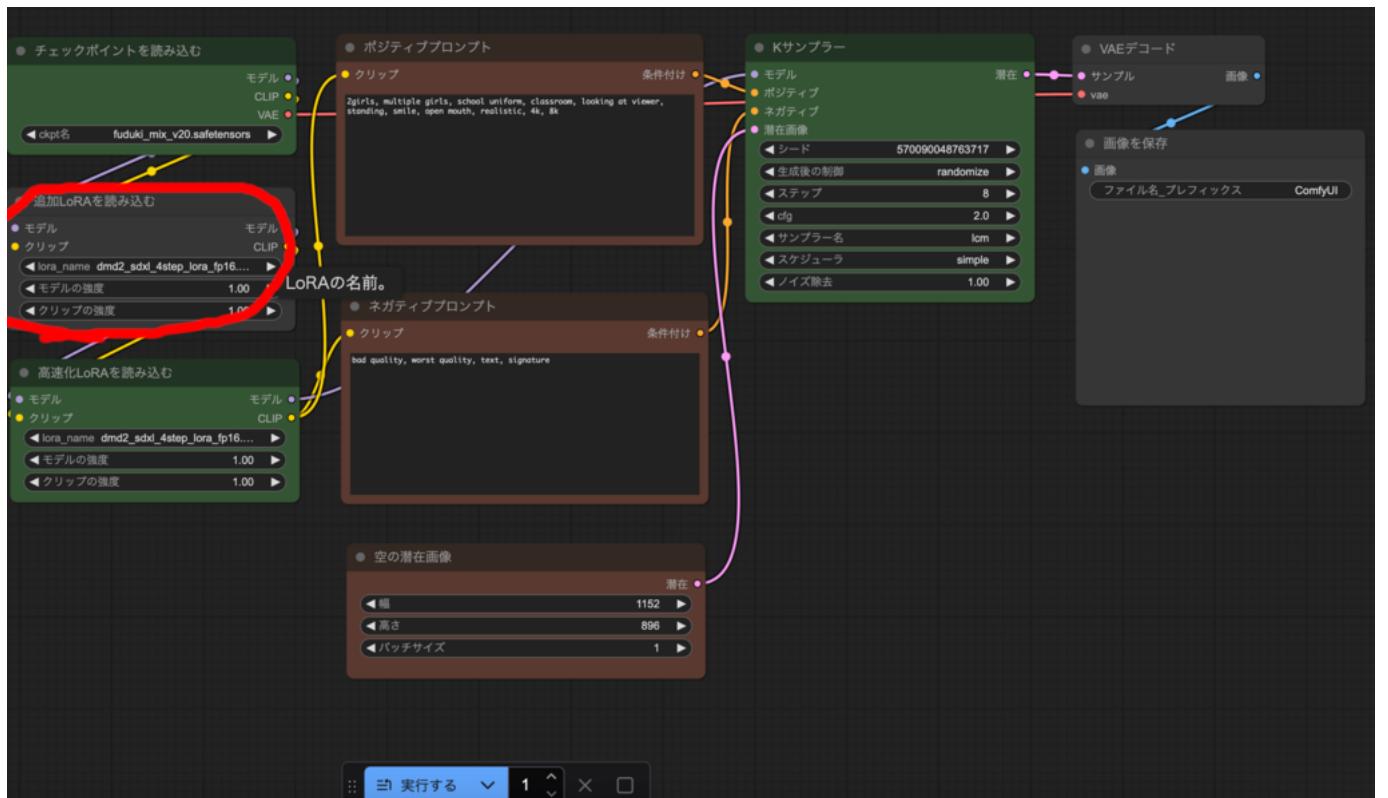
追加LoRA使用方法

LoRA は既存のモデルに対して追加学習することで、画風を変えたり、キャラクタを再現できるようにする追加モデルです。

『1.1.1. 画像生成基本T2L.json』をはじめ今までのワークフローには紫色で無効になっているノードが存在していたと思います。



ノードをクリックした時にノードの上部にでるメニューに左から 2 番目のボタンを押すとノードが有効になります。



もう一度押すと無効に戻ります。

『追加LoRAを読み込む』ノードを有効にして、civitai等からダウンロードした LoRA モデルファイルを、models フォルダの loras フォルダに移動させ、メニューの編集のノード定義を更新を押し、LoRA モデルを指定すれば、その LoRA を使用できます。

必要に応じて LoRA を活用してください。

4. 音楽生成

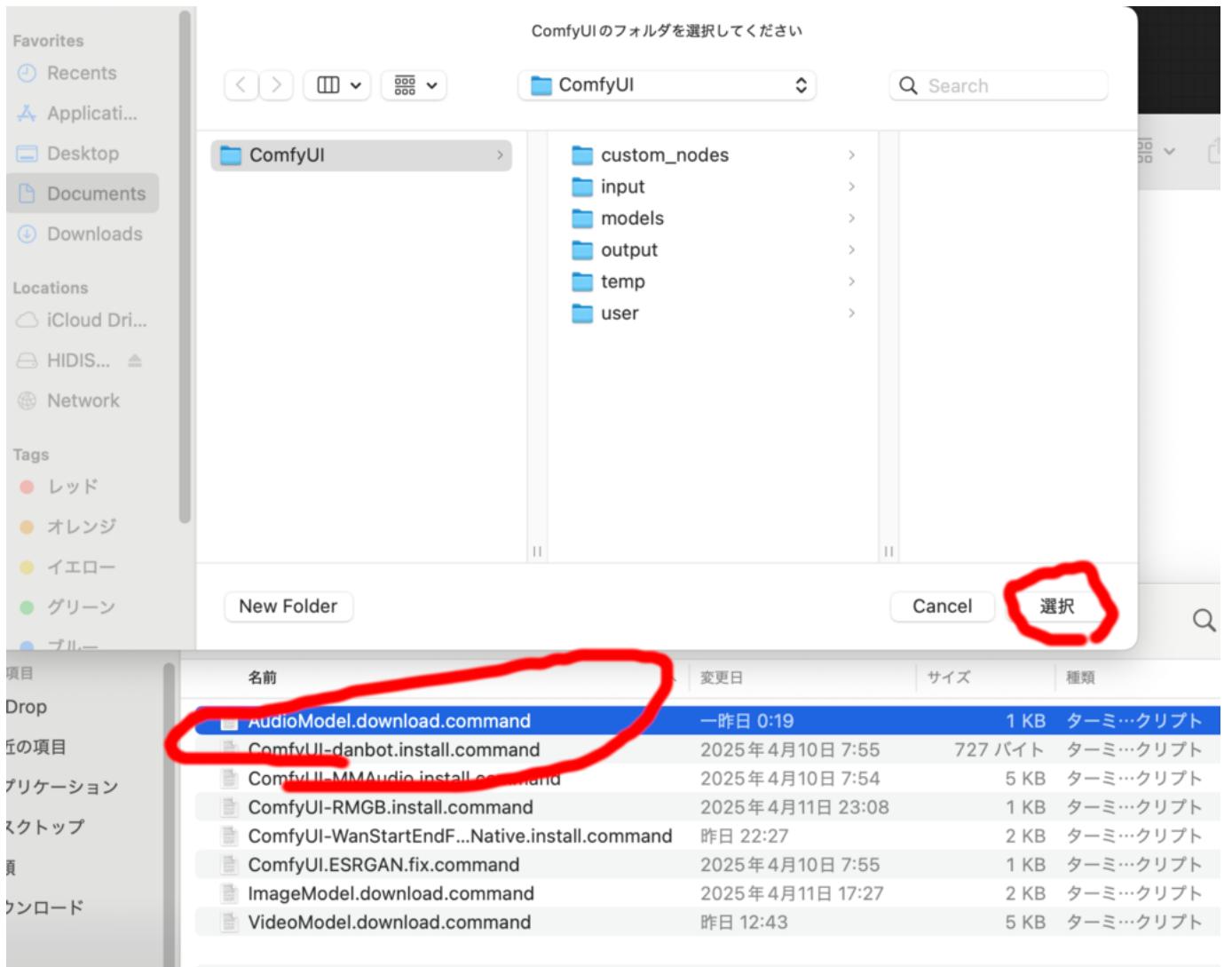
4.1 音楽生成AI

音楽生成 AI では画像生成 AI と同じ ComfyUI Desktop を利用します。まだインストールしていない場合は、[画像生成のインストール](#)に従って、ComfyUI Desktop をインストールしてください。

4.2 推奨モデルのダウンロード

ComfyUI Desktop を起動している場合は、一旦終了してください。

catAITools の ComfyUI用ツールの中に『AudioModel.download.command』を用意しました。



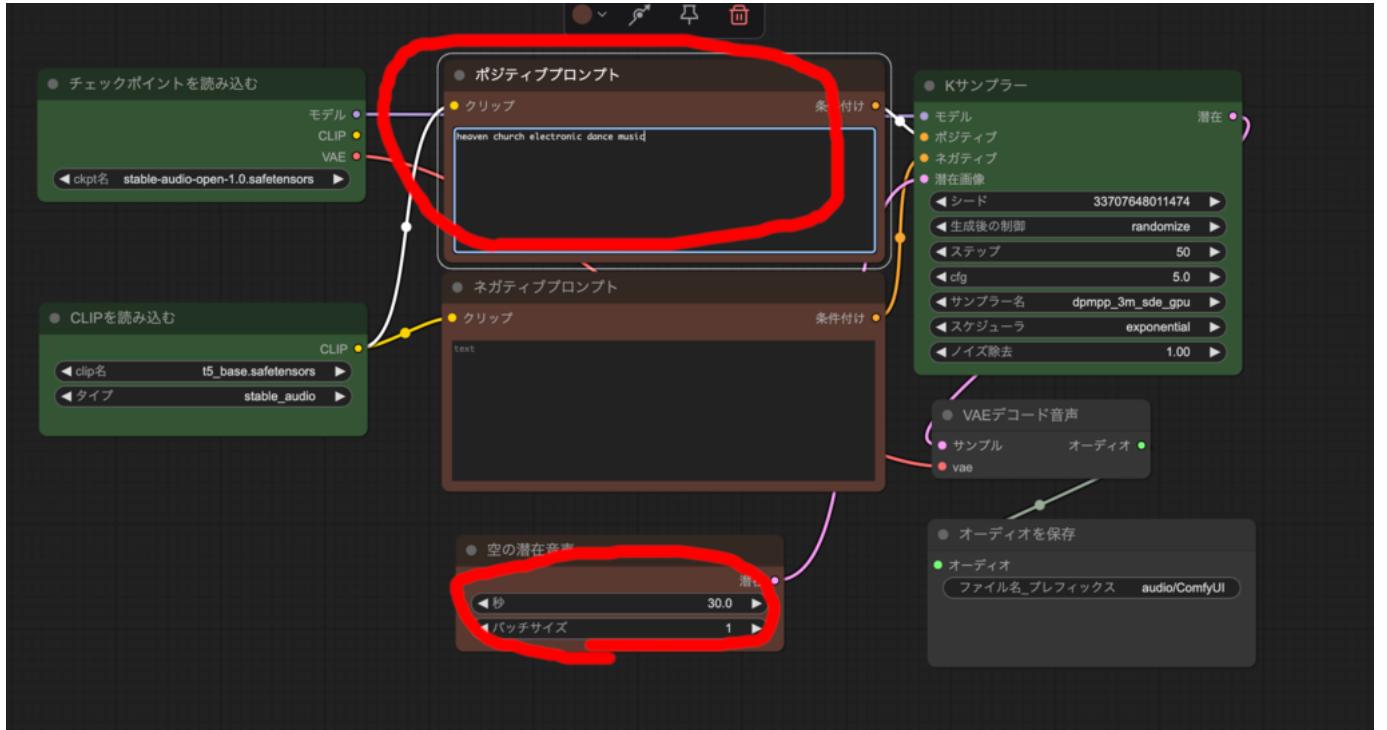
これをダブルクリックして、ComfyUIフォルダを指定すれば 推奨モデルがダウンロードされます。

『stable-audio-open-1.0.safetensors』とその追加学習版『stable-audio-open-1.0-Music.safetensors』の2つのモデルをダウンロードします。

ダウンロード完了したらダイアログが表示されるので OK を押してターミナルを閉じてください。

4.3 音楽生成

『2.1.1.音楽生成.json』を開いてください。



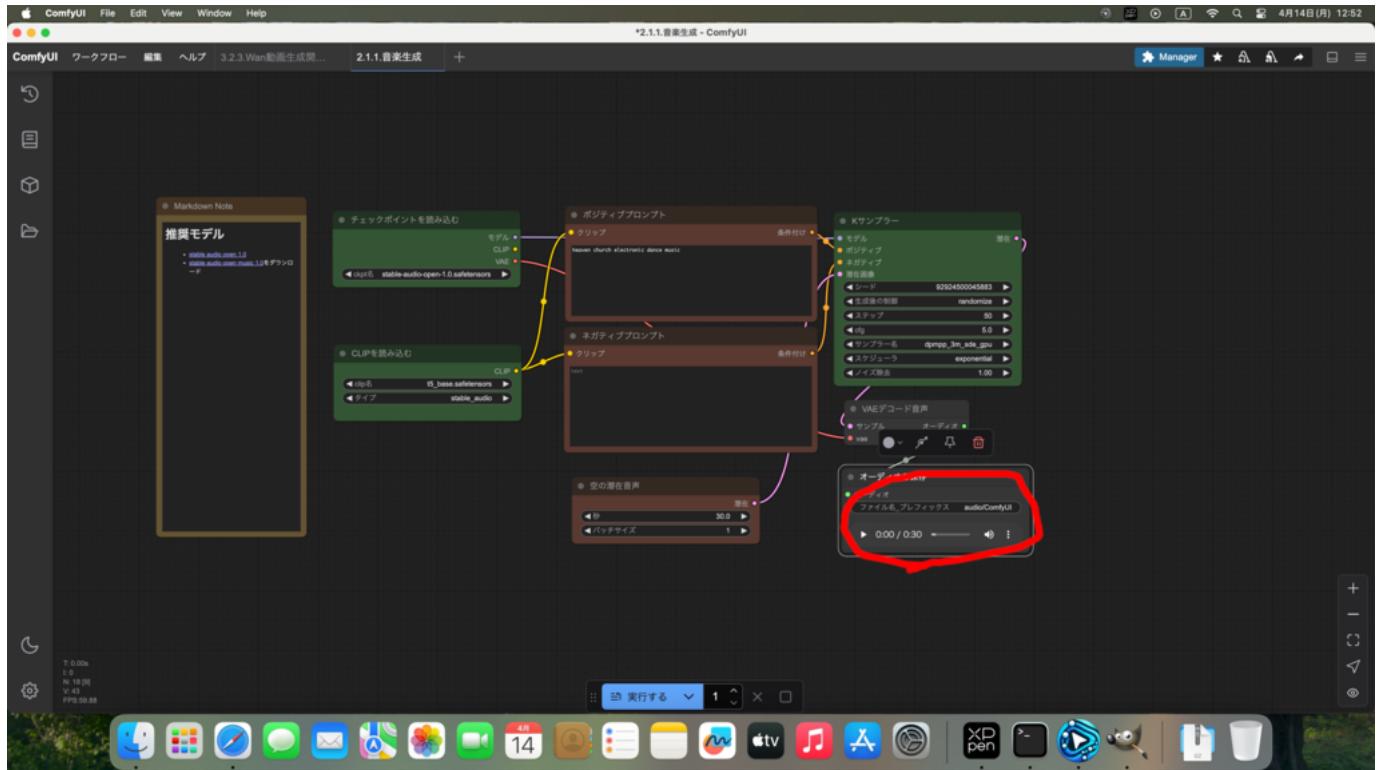
上のメニューの編集からノード定義を更新するを押してダウンロードしたモデルを反映させてください。

Stable audio での音楽生成は画像生成と基本的には同じです。

ポジティブプロンプトに生成したい音楽の内容を英語プロンプトで指定します。次のプロンプトを入力して、実行するを押してください。

```
heaven church electronic dance music
```

30秒間の天国の教会のエレクトロニックダンス音楽が生成されます。



右下のオーディオを保存から再生できます。また、音声ファイルは output フォルダの audio フォルダ以下に保存されます。

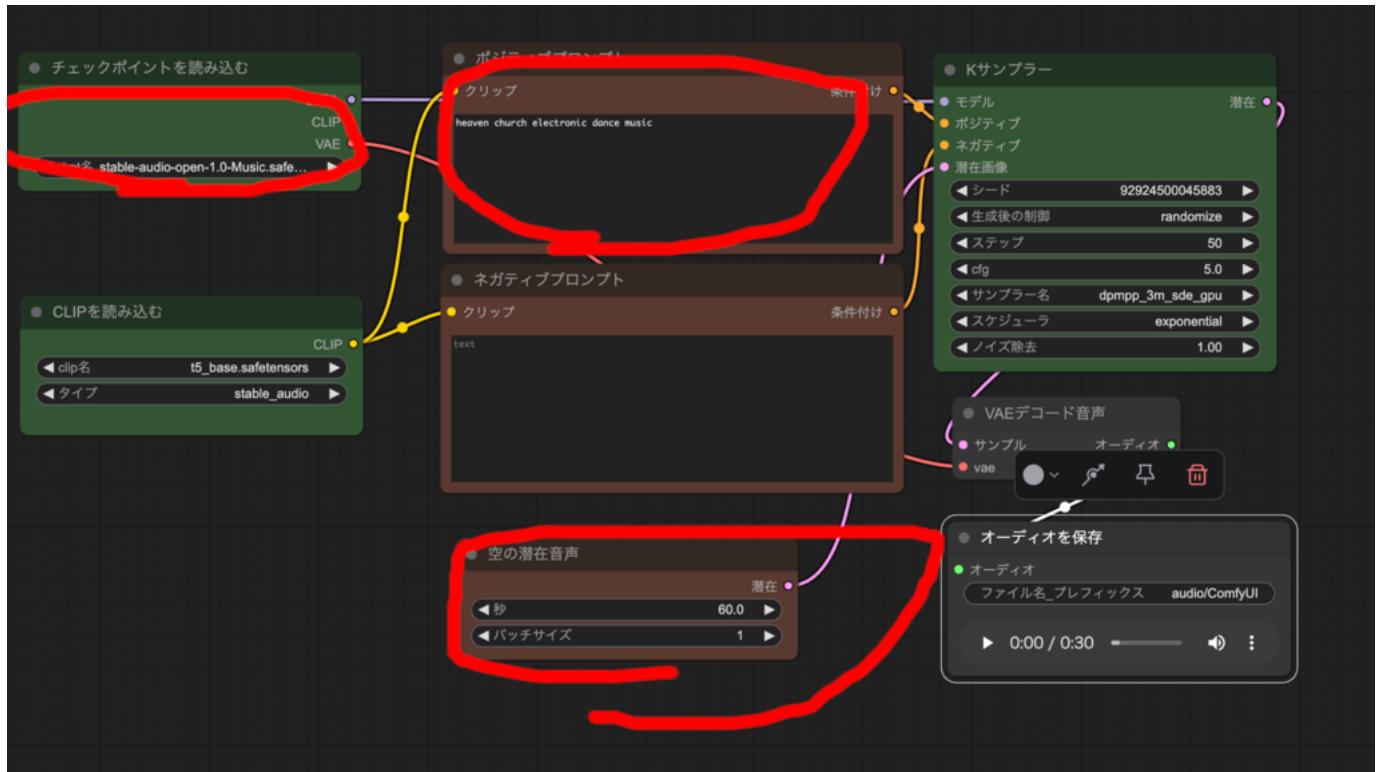
『空の潜在音声』ノードの秒で時間指定できます。

次のプロンプトを入れて、秒を3秒にして実行するを押してください。

```
car engine
```

効果音を生成することもできます。

オリジナルの『stable-audio-open-1.0.safetensors』は著作権のある楽曲を学習していないようですが、『stable-audio-open-1.0-Music.safetensors』モデルの方は既存の曲を追加学習したものです。両方試してみてください。



『チェックポイントを読み込む』ノードで stable-audio-open-1.0-Music.safetensors に変更し、秒数を 60 秒にして次のプロンプトを入れて実行するを押してください。

heaven church electronic dance music

4.4 音楽生成用プロンプトの参考サイト

・音楽生成AI Stable Audioのプロンプトに使えそうな単語まとめ

こちらのサイトに音楽生成に使えそうな単語がまとめられています。

5. 動画生成

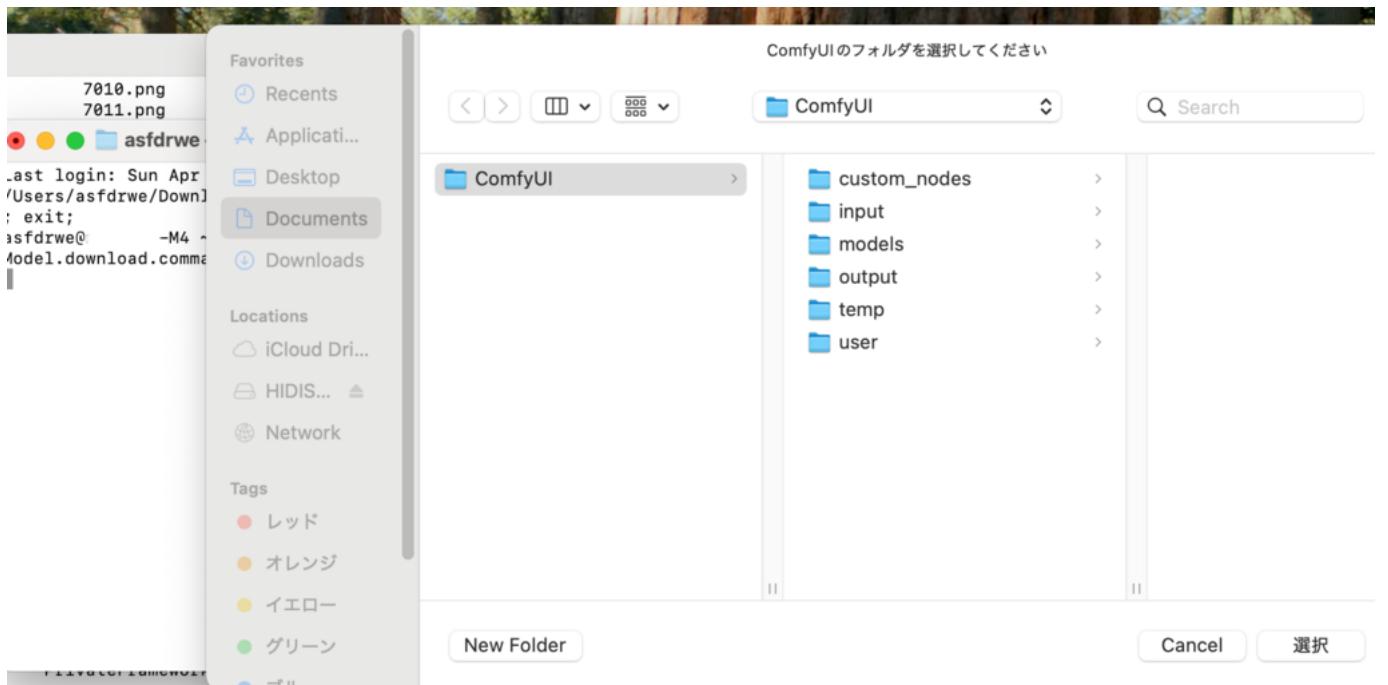
5.1 動画生成AI

動画生成 AI では画像生成 AI と同じ ComfyUI Desktop を利用します。まだインストールしていない場合は、[画像生成のインストール](#)に従って、ComfyUI Desktop をインストールしてください。

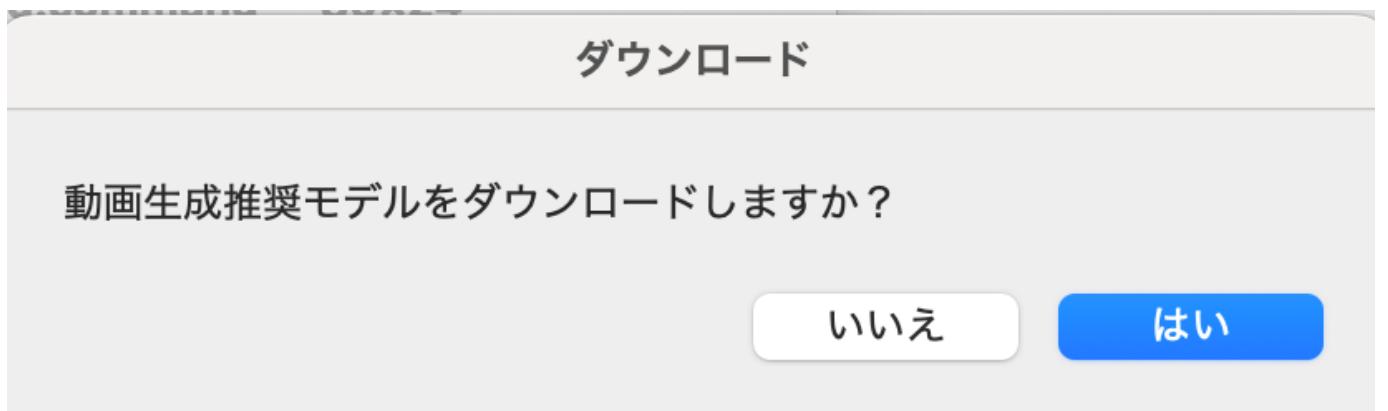
5.2 推奨モデルのダウンロード

ComfyUI Desktop を起動している場合は、一旦を終了してください。

catAITools フォルダの ComfyUI用ツールフォルダに推奨モデルをダウンロードする『VideoModel.download.command』を用意しました。これをダブルクリックしてください。



ComfyUI フォルダを指定し、



はいを押せば、次の推奨モデルがダウンロードされます。数十GBあるのでストレージ容量に気をつけてください。

HunyuanVideo用(7種類)

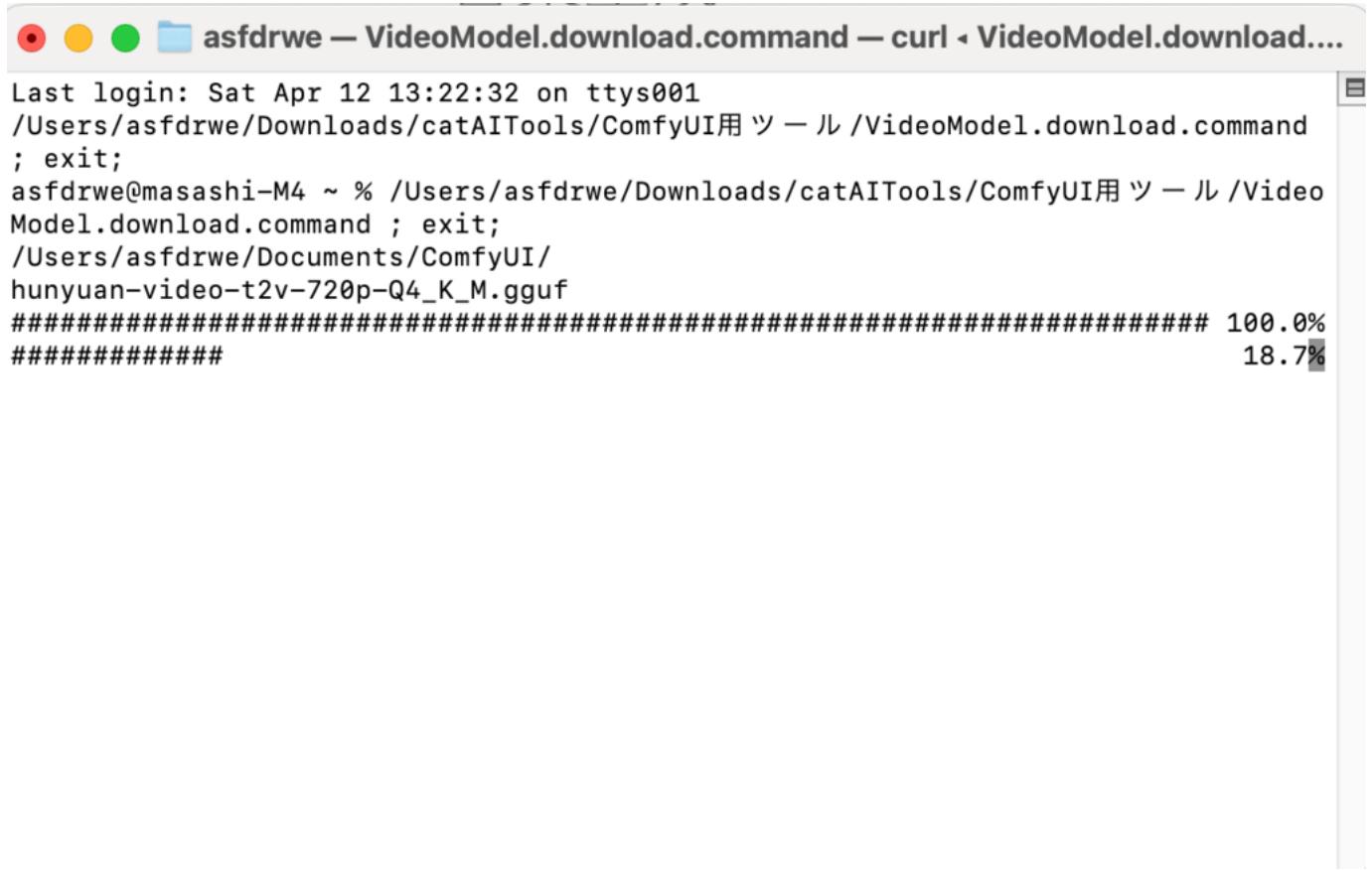
- hunyuan-video-t2v-720p-Q4_K_M.gguf
- hunyuan-video-i2v-720p-Q4_K_M.gguf
- hyvideo_FastVideo_LoRA-fp8.safetensors
- clip_l.safetensors
- llava_llama3_vision.safetensors
- llava-llama-3-8b-v1_1.Q4_K_M.gguf
- hunyuan_video_vae_bf16.safetensors

WanVideo用(5種類)

- wan2.1-t2v-14b-Q4_K_M.gguf
- wan2.1-i2v-14b-480p-Q4_K_M.gguf
- umt5-xxl-encoder-Q4_K_M.gguf
- clip_vision_h.safetensors
- wan_2.1_vae.safetensors

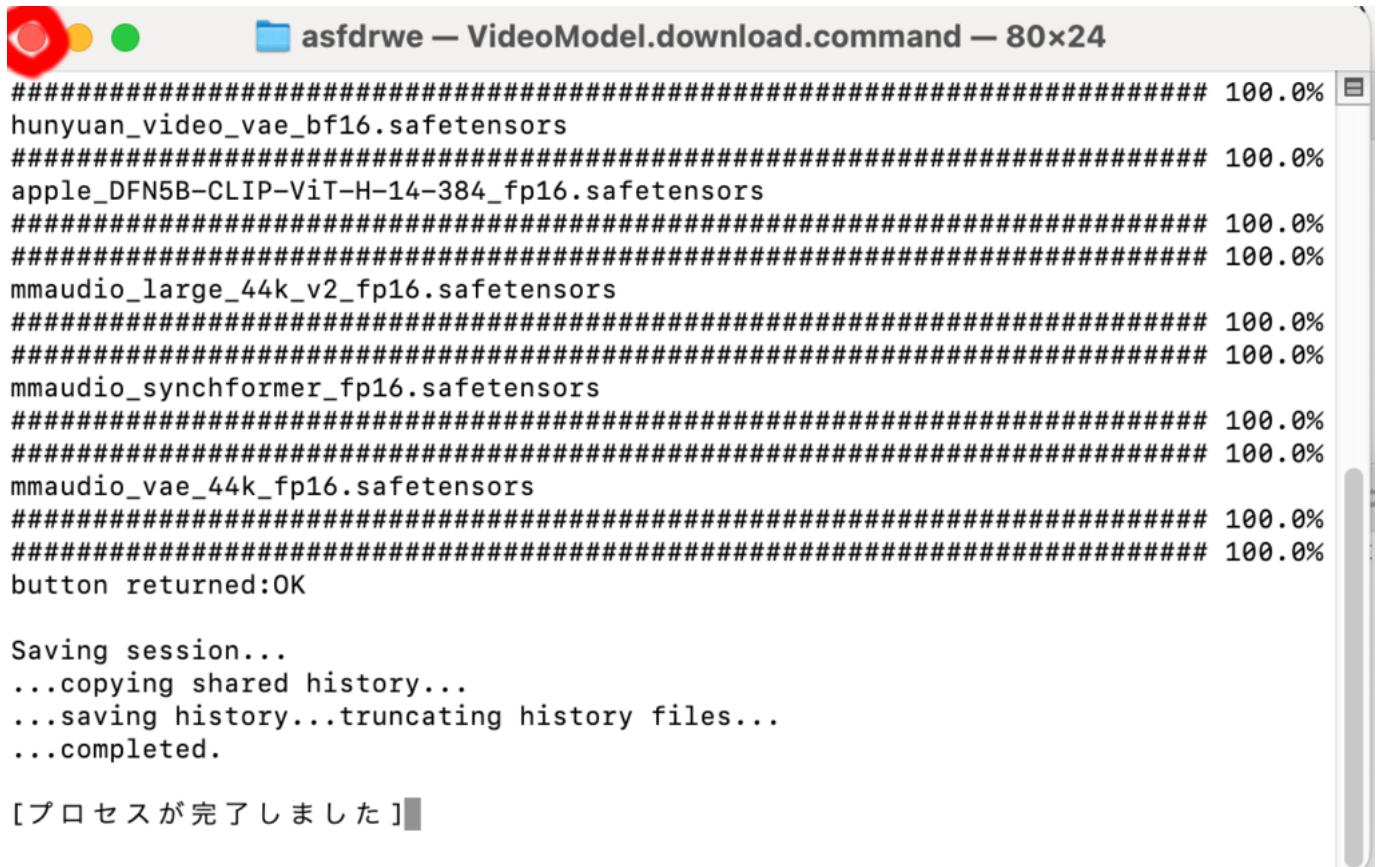
MMAudio用(4種類)

- apple_DFN5B-CLIP-ViT-H-14-384_fp16.safetensors
- mmaudio_large_44k_v2_fp16.safetensors
- mmaudio_synchformer_fp16.safetensors
- mmaudio_vae_44k_fp16.safetensors



```
asfdrwe — VideoModel.download.command — curl < VideoModel.download....
```

```
Last login: Sat Apr 12 13:22:32 on ttys001
/Users/asfdrwe/Downloads/catAITools/ComfyUI用 ツール /VideoModel.download.command
; exit;
asfdrwe@masashi-M4 ~ % /Users/asfdrwe/Downloads/catAITools/ComfyUI用 ツール /Video
Model.download.command ; exit;
/Users/asfdrwe/Documents/ComfyUI/
hunyuan-video-t2v-720p-Q4_K_M.gguf
#####
##### 100.0%
#####
18.7%
```



```
asfdrwe — VideoModel.download.command — 80x24
```

```
#####
##### 100.0%
hunyuan_video_vae_bf16.safetensors
#####
##### 100.0%
apple_DFN5B-CLIP-ViT-H-14-384_fp16.safetensors
#####
##### 100.0%
#####
##### 100.0%
mmaudio_large_44k_v2_fp16.safetensors
#####
##### 100.0%
#####
##### 100.0%
mmaudio_synchformer_fp16.safetensors
#####
##### 100.0%
#####
##### 100.0%
mmaudio_vae_44k_fp16.safetensors
#####
##### 100.0%
#####
##### 100.0%
button returned:OK

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
```

[プロセスが完了しました]

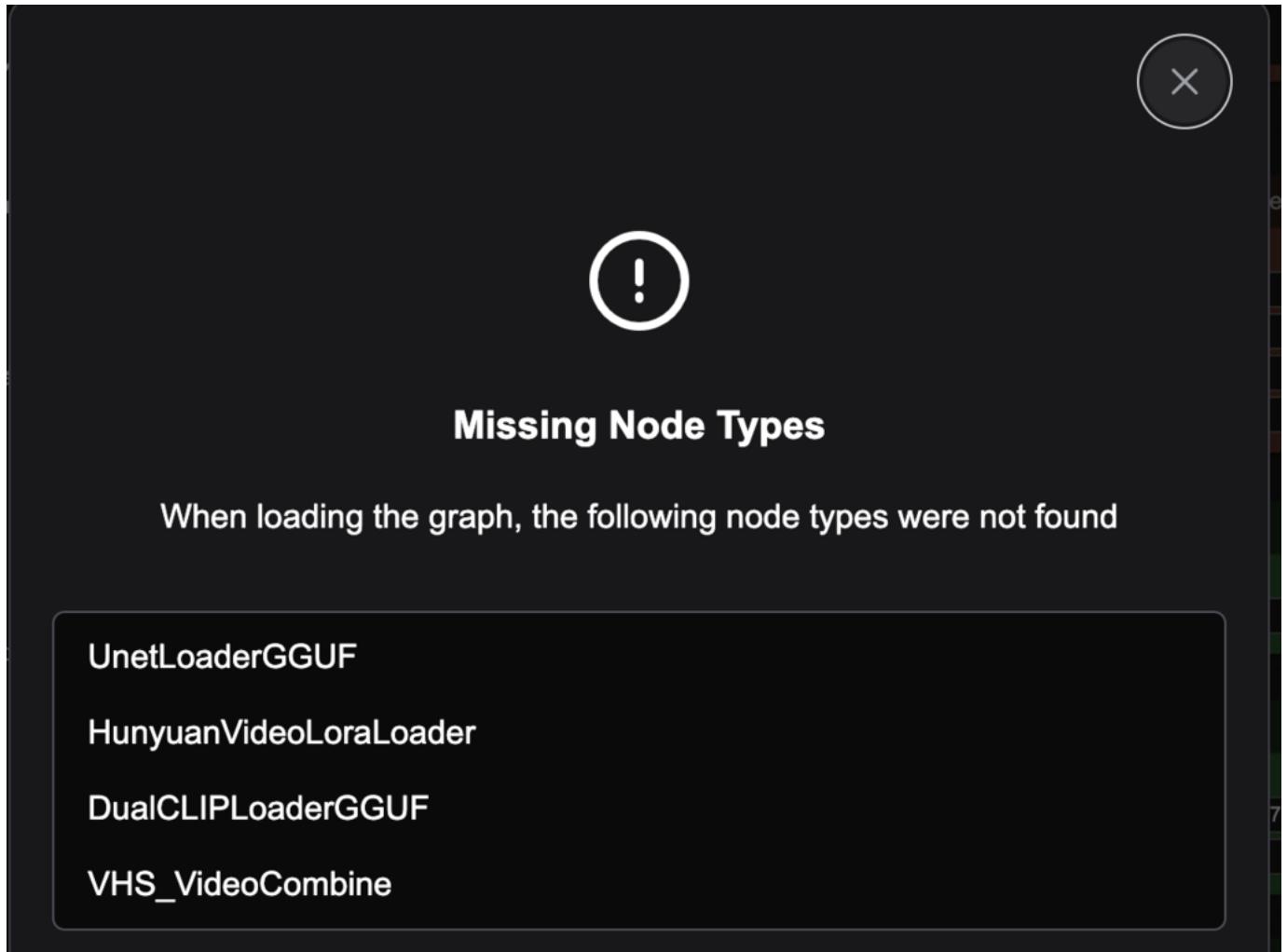
ダウンロード完了するとダイアログが表示されるので、OKを押してターミナルを閉じてください。

ComfyUI Desktop を起動してください。

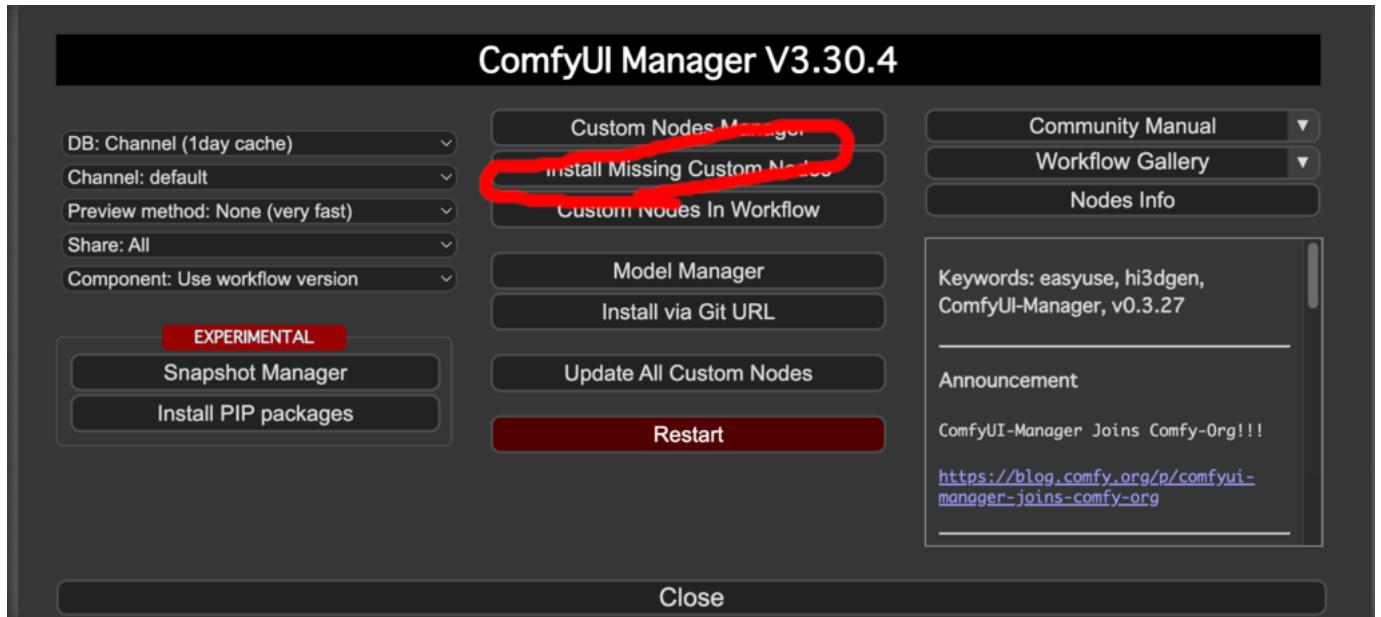
5.3 HunyuanVideoでの文章に基づく動画生成

まず手始めに文章(Text)から動画(Video)を生成していきます。Text to Video(T2V)といいます。

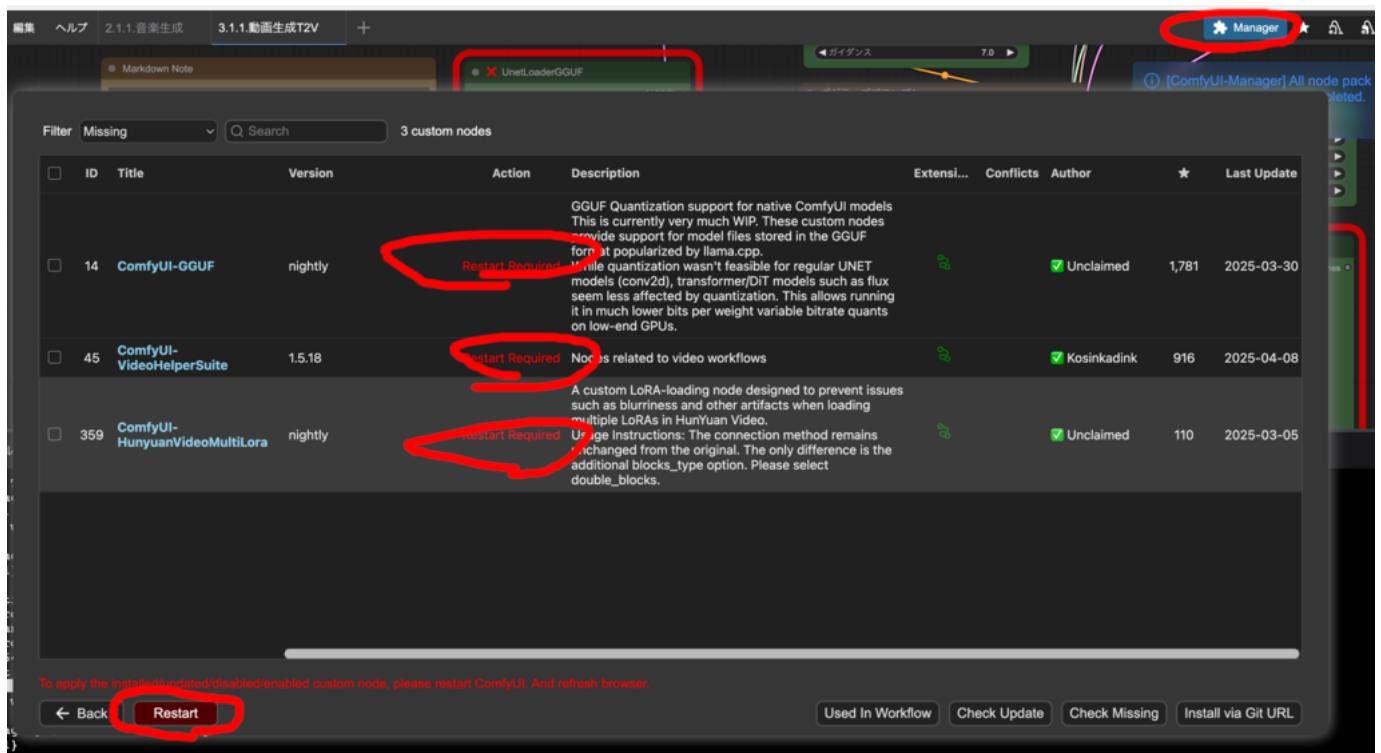
『3.1.1.HUNYUAN動画生成T2V.json』を開いてください。



ComfyUI-GGUF(GGUF 形式のモデルを扱う)と VideoHelperSuite(動画を扱う)と ComfyUI-HunyuanVideoMultiLora (HunyuanVideo 用にカスタマイズされた LoRA ローダ)の 3 つのカスタムノードが必要です。

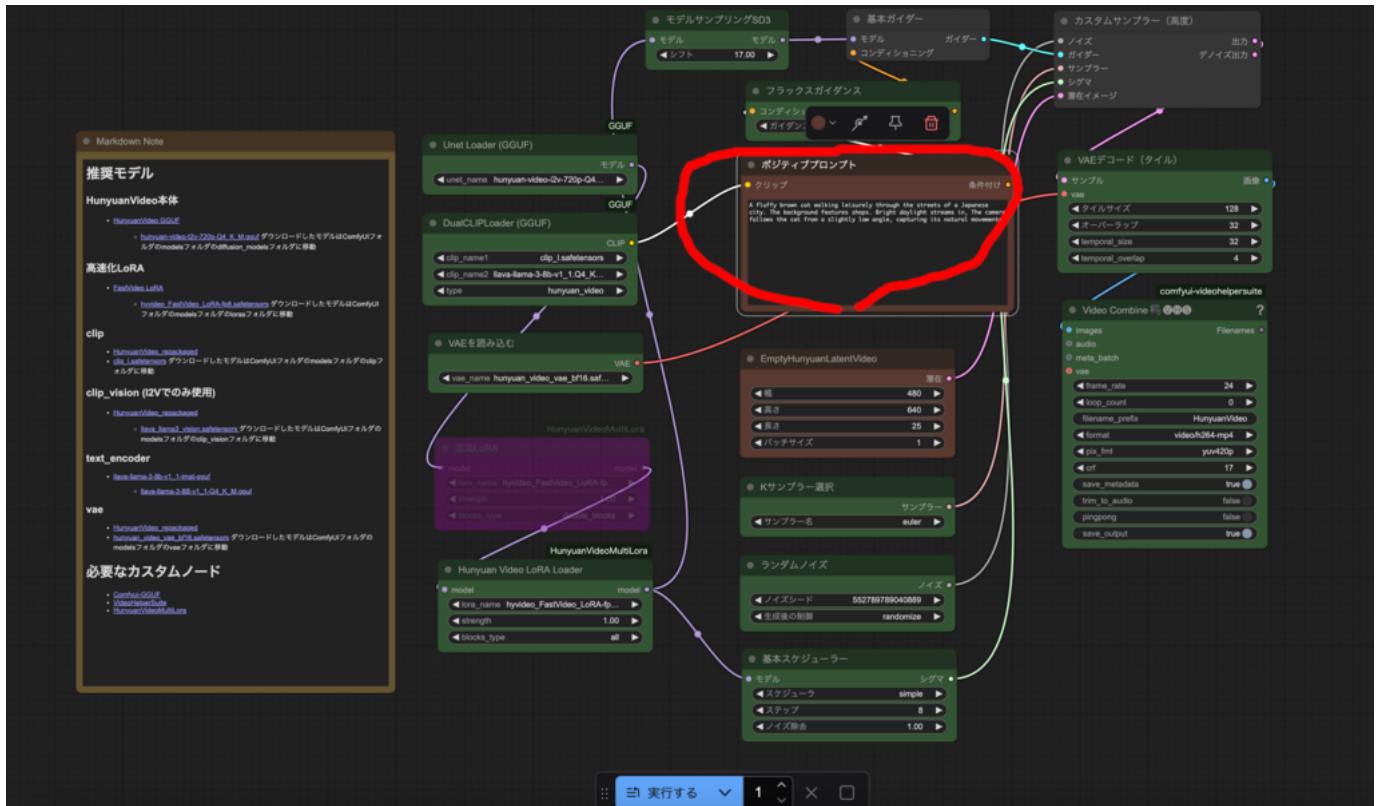


右上の Manager から ComfyUI-Manager を起動し、Install Missing Custom Nodesを押し、



3つともインストールしてください。まとめてインストールして Restart して問題ないです。

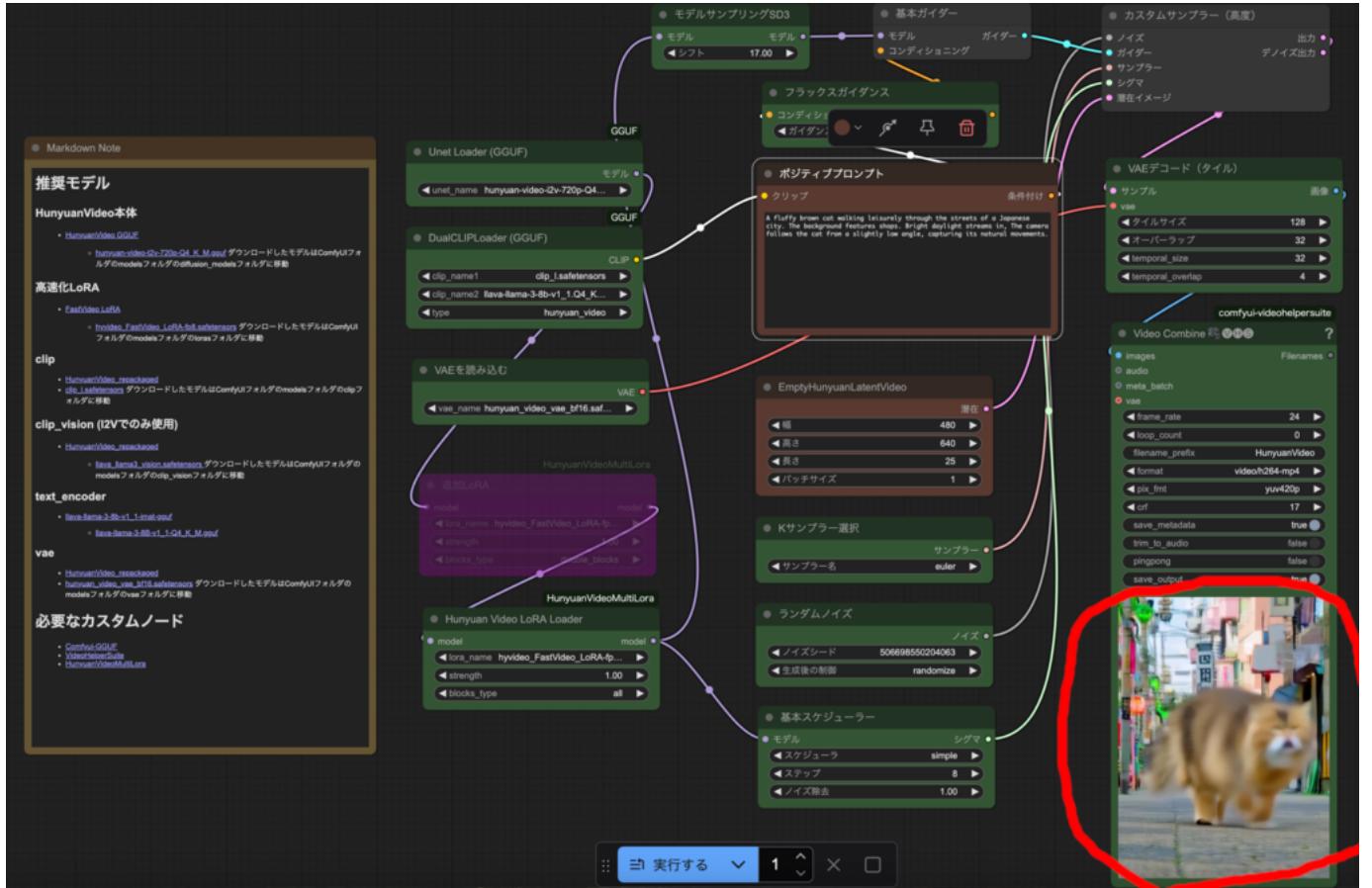
HunyuanVideo での動画生成は画像生成と基本的には同じですが、プロンプトは基本的に英語の文章で記述します。



ポジティブプロンプトに次の文章を入力して実行するを押してください。

A fluffy brown cat walking leisurely through the streets of a Japanese city.
The background features shops. Bright daylight streams in,
The camera follows the cat from a slightly low angle,
capturing its natural movements.

20分程度で猫が街中を歩いているリアル調の1秒程度の動画が生成されたと思います。



5.3.1 ノード解説

・ポジティブプロンプト

生成する動画への指示を英文で書いてください。HunyuanVideo では ネガティブプロンプトはないので表現したい要素をポジティブプロンプトに書いてください。

・EmptyHunyuanLatentVideo

動画の幅や高さやフレーム数(HunyuanVideo では1秒間24フレームなので25フレームで1秒強)を指定してください。秒数を増やすれば生成時間はかなり増えます。

・基本スキューラー

ステップ数を増やすと画質が向上しますが時間がかかります。高速化 LoRA の仕様では 6 ステップで可能となっていますが、6 ステップではいまいちな気がしたので、8ステップにしています。

・フラックスガイダンス

プロンプトが動画に与える影響度です。増やすとプロンプトの内容により近づきますが破綻しやすくなります。

・モデルサンプリングSD3

・VAEデコード (タイトル)

変更すると動画の動き具合は変わるはずですが、通常変更する必要はありません。

・Video Combine VHS

format で h264 や h265 など動画のコーディックの変更が可能です。 h264-mp4 の場合 crf が動画の品質です。 loop_count を増やすと その回数分 ループする動画が生成されます。

- Unet Loader(GGUF)
- DualCLIPLoader(GGUF)
- VAEを読み込む

GGUF 形式のモデルを使用しています。 GGUF 形式はモデルを量子化したものです。 JPEG画像やMPEG動画のように、 モデルの精度が落ちる代わりにモデルサイズを縮小する非可逆圧縮を行ったものと捉えてよいです（正確には違いますが）。 16GB のメモリでも動作するよう GGUF 形式の Q4_K_M という量子化モデルを使用しています。

- Hunyuan Video Lora Loader

高速化 LoRA を読み込みます。通常は変更する必要はありません。

- 追加 LoRA

LoRA を使用したい場合は有効にして使いたい LoRA を指定してください。 blocks_type は double_blocks にしないと複数 LoRA を重ねたときに品質が低下しやすくなります。

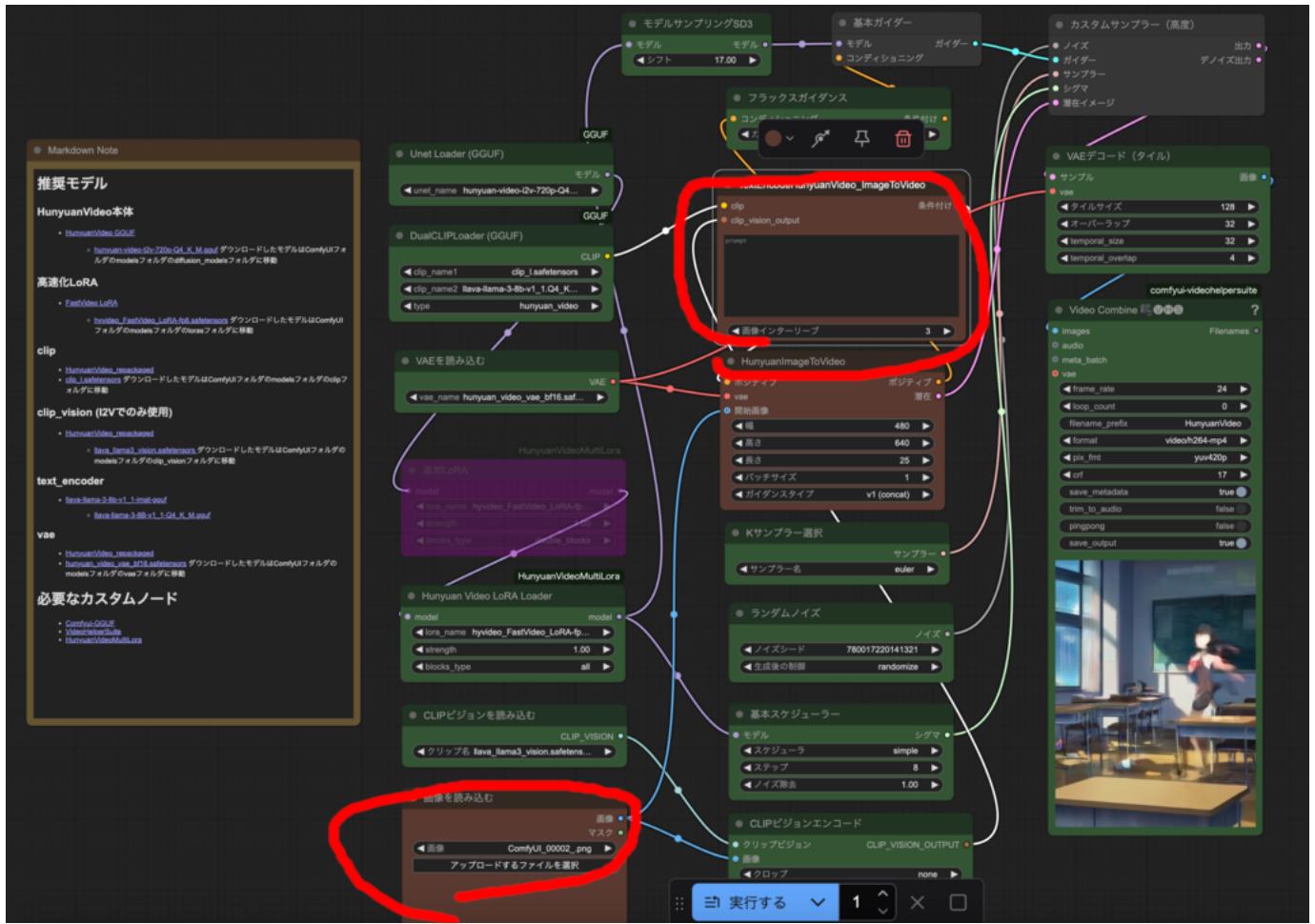
5.3.2 日本語でのプロンプト

Google 翻訳や DeepL などで日本語文章を翻訳するか、 後述する言語 AI に英文プロンプトを作成してもらうのがよいと思います。

5.4 HunyuanVideo での画像に基づく動画生成

画像を元に画像を生成する I2I と同様に、動画の一番最初となる画像を与えて、 その画像(Image)に対する動画(Video)を生成する Image to Video(I2V) も可能です。

『3.1.2.HUNYUAN動画生成I2V.json』を開いてください。



『1.1.1.画像生成基本T2I.json』で生成した猫画像を『画像を読み込む』ノードで選択してください。

ポジティブプロンプトに次の文章を入力して実行するを押してください。

```
A cat walking leisurely through in bedroom.  
The background features bed. Bright daylight streams in,  
The camera follows the cat from a slightly low angle,  
capturing its natural movements.
```

元絵からだいぶ離れている気がしますが猫が寝室を動いている動画が生成されます。

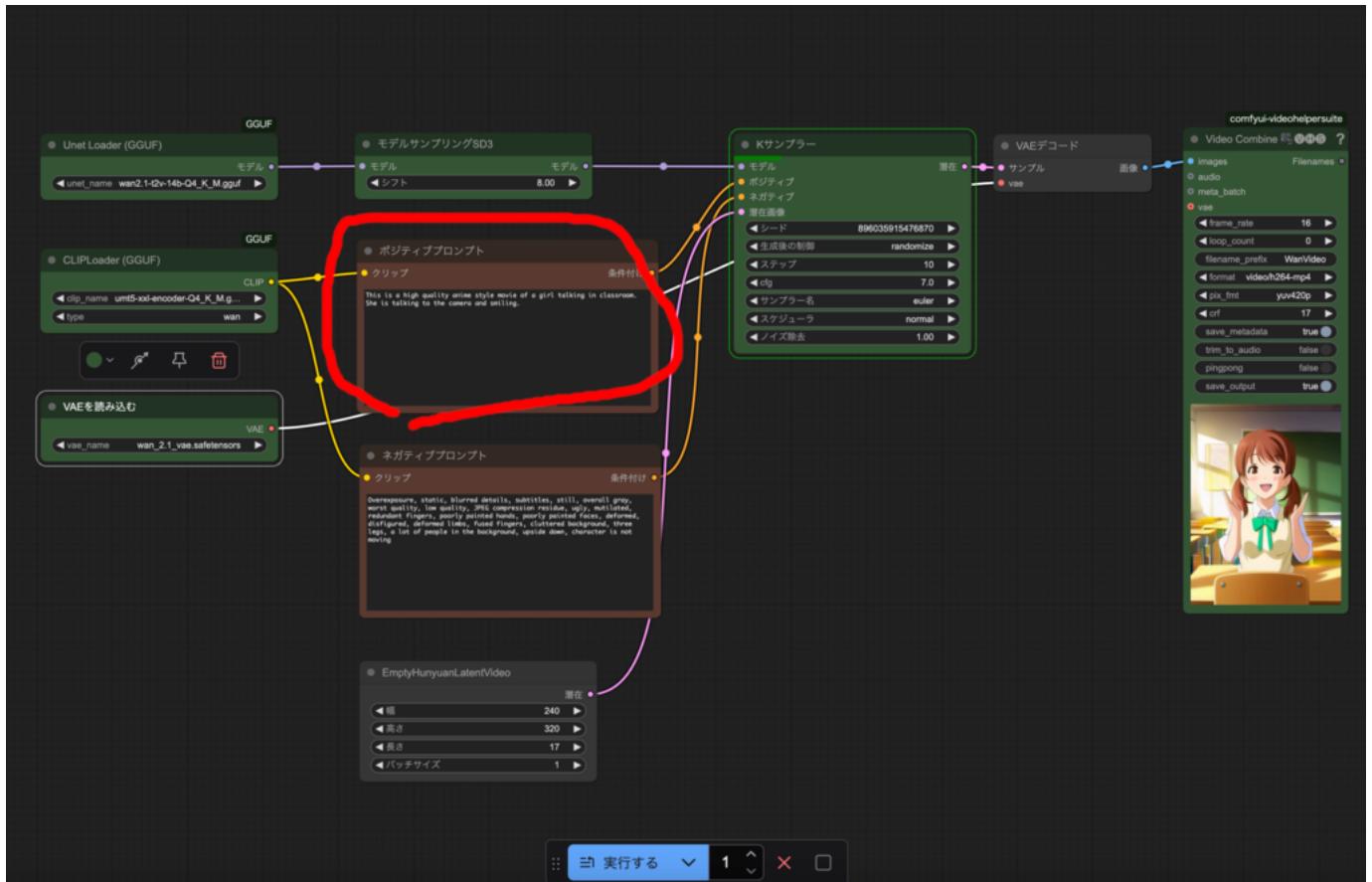
|

基本的には T2V で動画を作成するより、画像を生成した上で I2V で動画にする方が求める動画を作りやすいと思います。

5.5 Wan2.1 での文章に基づく動画生成

Wan2.1 による T2V 動画生成を行います。HunyuanVideo より高品質ですが、M4 Mac mini の GPU や 16GB のメモリでは性能的に厳しいので、240x320 の小サイズの 1 秒程度の動画を生成します。

『3.2.1.Wan動画生成T2V.json』を開いてください。



次のプロンプトをポジティブプロンプトに入れて実行するを押してください。

```
This is a high quality anime style movie of a girl talking in classroom.  
She is talking to the camera and smiling.
```

15分程度で少女がこちらに話しかけてるアニメ調の1秒の動画が生成されるはずです。

■

5.5.1 ノードの説明

- ・ポジティブプロンプト
- ・ネガティブプロンプト

画像生成と同様に動画に対する指示をポジティブプロンプトに書き、動画に含めたくないものをネガティブプロンプトに書いてください。

- ・EmptyHunyuanLatentVideo

幅や高さや長さを指定します。Wan2.1は1秒間に16フレーム(16fps)なので、17フレームで1秒強の動画になります。

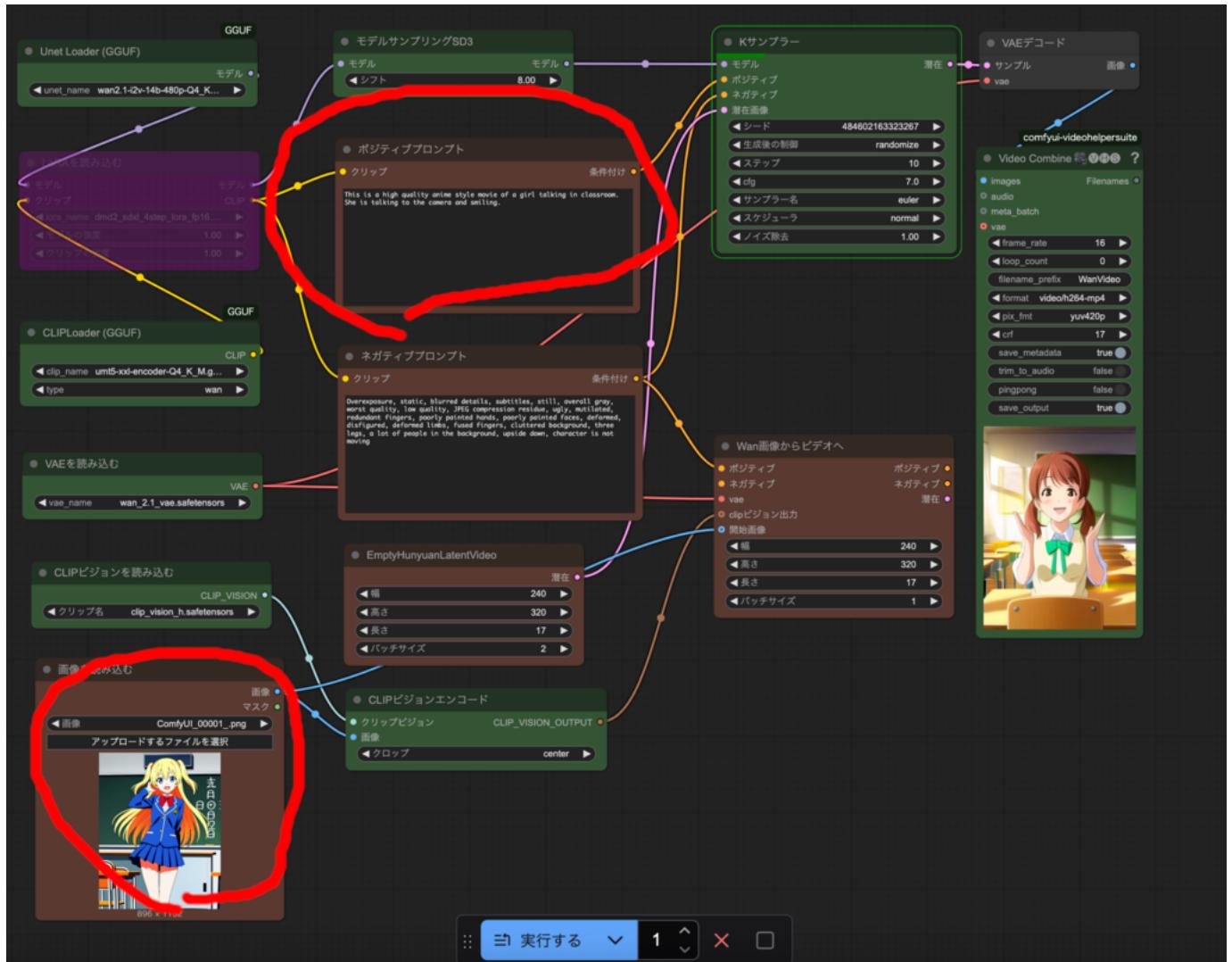
- ・Kサンプラー

T2Vではステップ数を10にしています。本来は20~30必要なのですが、生成速度を上げるために少ないステップ数にしています。

5.6 Wan2.1 での画像に基づく動画生成

画像を与えてその画像に対する動画を生成します。

『3.2.2.Wan動画生成I2V.json』を開いてください。



画像生成で生成した少女の画像を『画像を読み込み』で選択し、次のプロンプトをポジティブプロンプトに入れて実行するを押してください。

This is a high quality anime style movie of a girl talking in classroom.
She is talking to the camera and smiling.

I2V では同じステップ数でも 2 倍以上時間がかかる上に品質の問題からステップ数を 20 にしているので、60 分程度かかります。

元画像に基づくこちらに話しかけてるアニメ調の 1 秒の動画が生成されるはずです。

|

5.6.1 ノードの解説

• CLIPビジョンエンコード

画像と動画の縦横比率が違う場合にどう切り抜くかの指定ができます。noneなら何もしません。centerにすると画像の中心部分を切り抜いて動画の比率に合わせます。

5.7 Wan2.1 での開始画像と終了画像に基づく動画生成

開始画像だけでなく終了画像も与えることでより求める動画を生成しやすくなります。

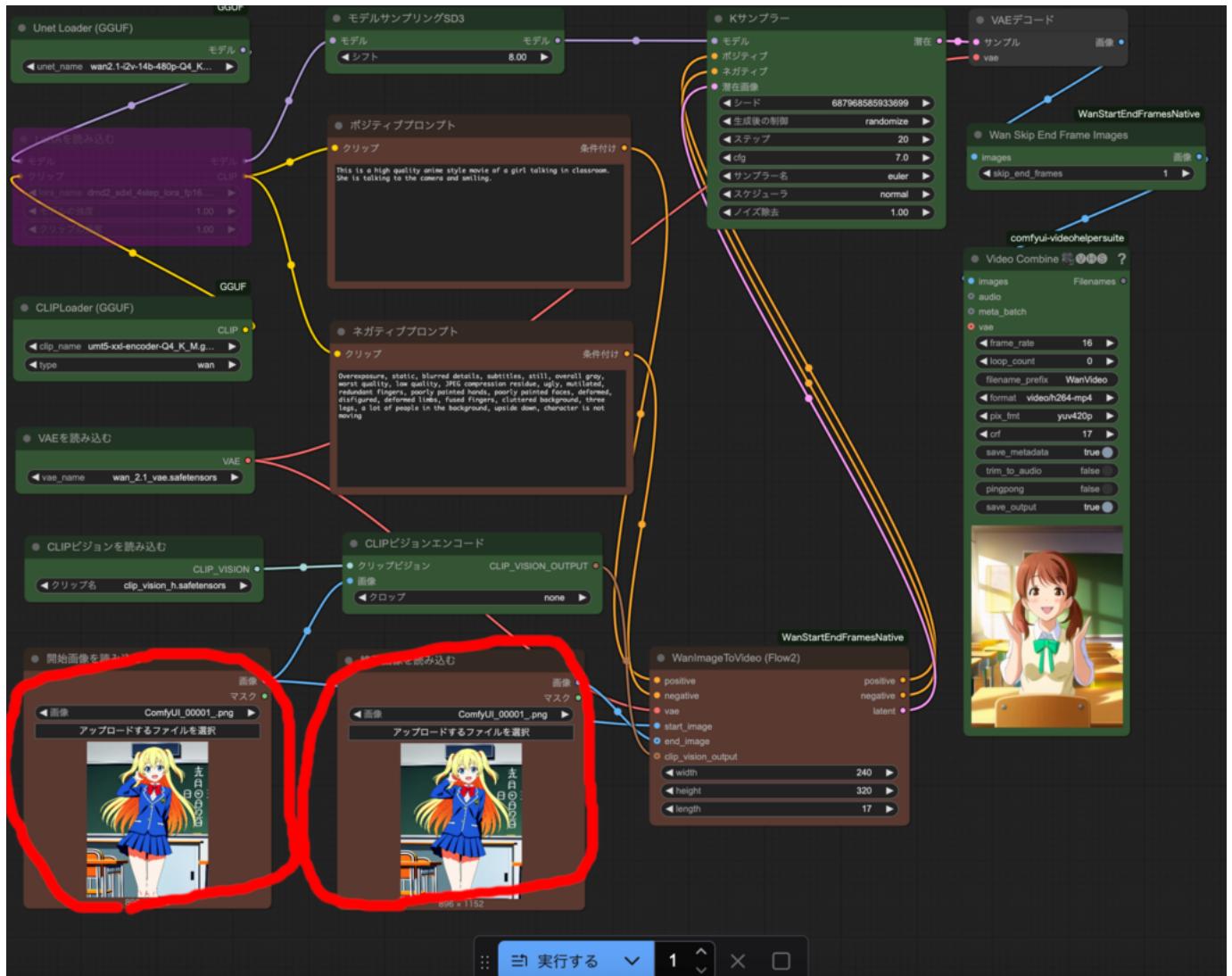
ComfyUI-WanStartEndFramesNativeというカスタムノードを使用します。

ただ、このノードでは幅や高さが 320 未満のものを生成できないので、240x320 の動画が生成できるようパッチを当てながらインストールします。

一旦 ComfyUI Destkop を終了し、catAITools フォルダの ComfyUI 用ツールフォルダの『ComfyUI-WanStartEndFramesNative.install.command』をダブルクリックしてください。

ComfyUI フォルダを選択してインストールしてください。

ComfyUI Desktop を起動し、『3.2.3.Wan動画生成開始終了画像指定I2V.json』を開いてください。



画像生成で生成した少女の画像を『開始画像を読み込み』で選択し、同じ画像を『終了画像を読み込み』でも選択します。こうすることでループする動画を容易に作成できます。

この場合、開始画像と終了画像が同じなので、終了画像は動画から取り除きます。『Wan Skip End Frame Images』ノードで skip_end_frames を 1 にすることでその処理を行います。

『Video Combine VHS』ノードの loop_count を 5 にすることで 1 秒の動画を 5 回繰り返して 5 秒の動画を作成します。

次のプロンプトをポジティブプロンプトに入れて実行するを押してください。

```
This is a high quality anime style movie of a girl talking in classroom.  
She is talking to the camera and smiling.
```

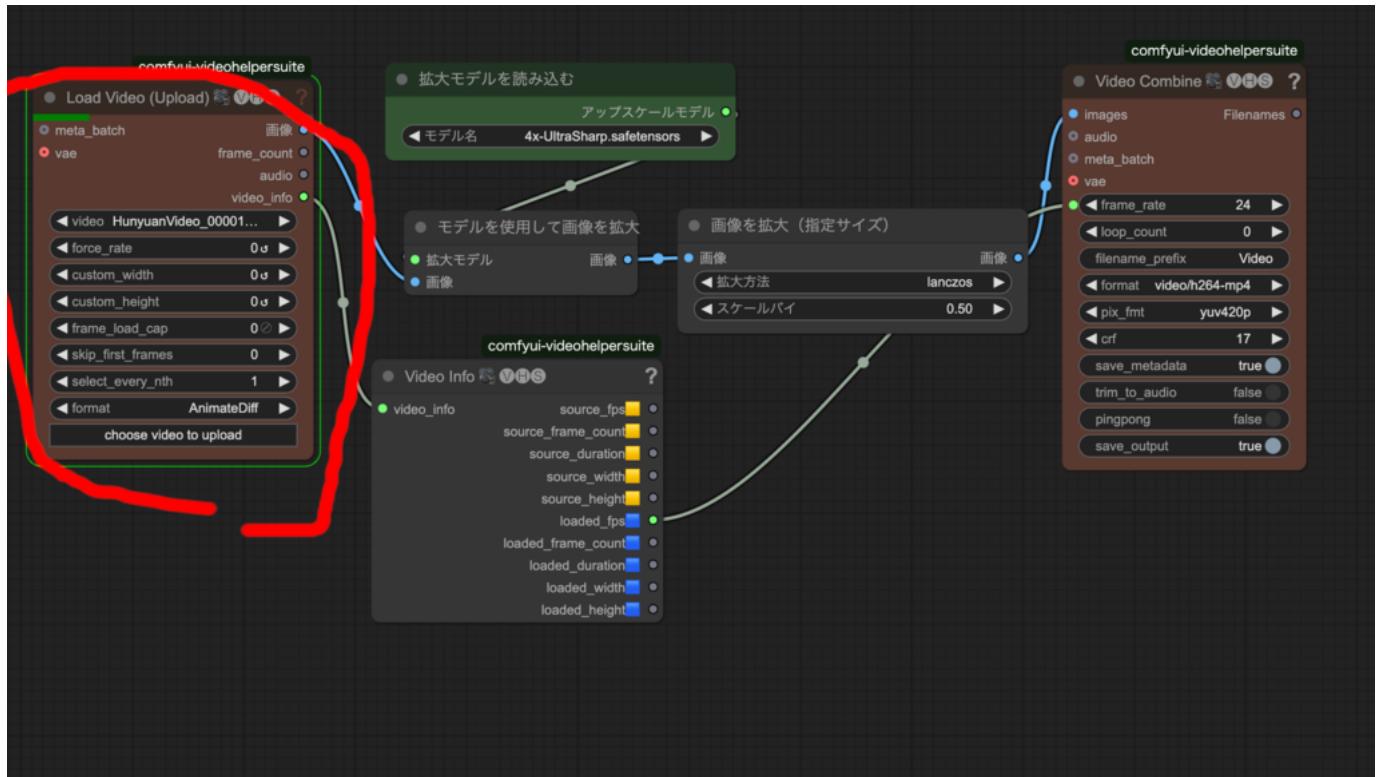
5.8 AI活用動画加工

5.8.1 動画拡大

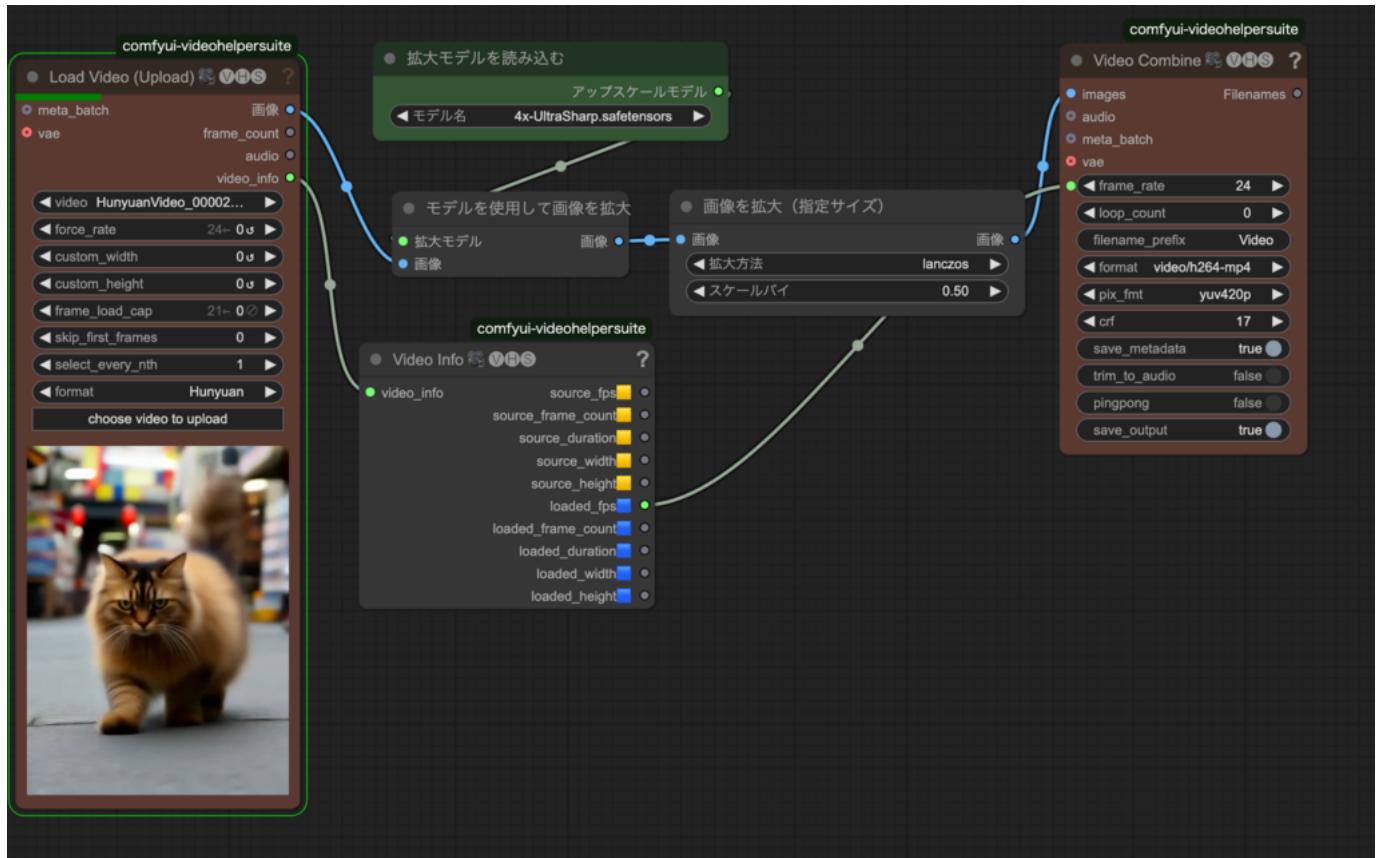
『1.4.1.画像2倍拡大.json』と同様の手法で動画を2倍に拡大します。

1.4.1.画像2倍拡大.json』で『ComfyUI.ESRGAN.fix.command』を実行して ComfyUI Desktop のコードを修正している場合はそのまま動きます。
修正していない場合は実行してください

『3.3.1.動画2倍拡大.json』を開いてください。

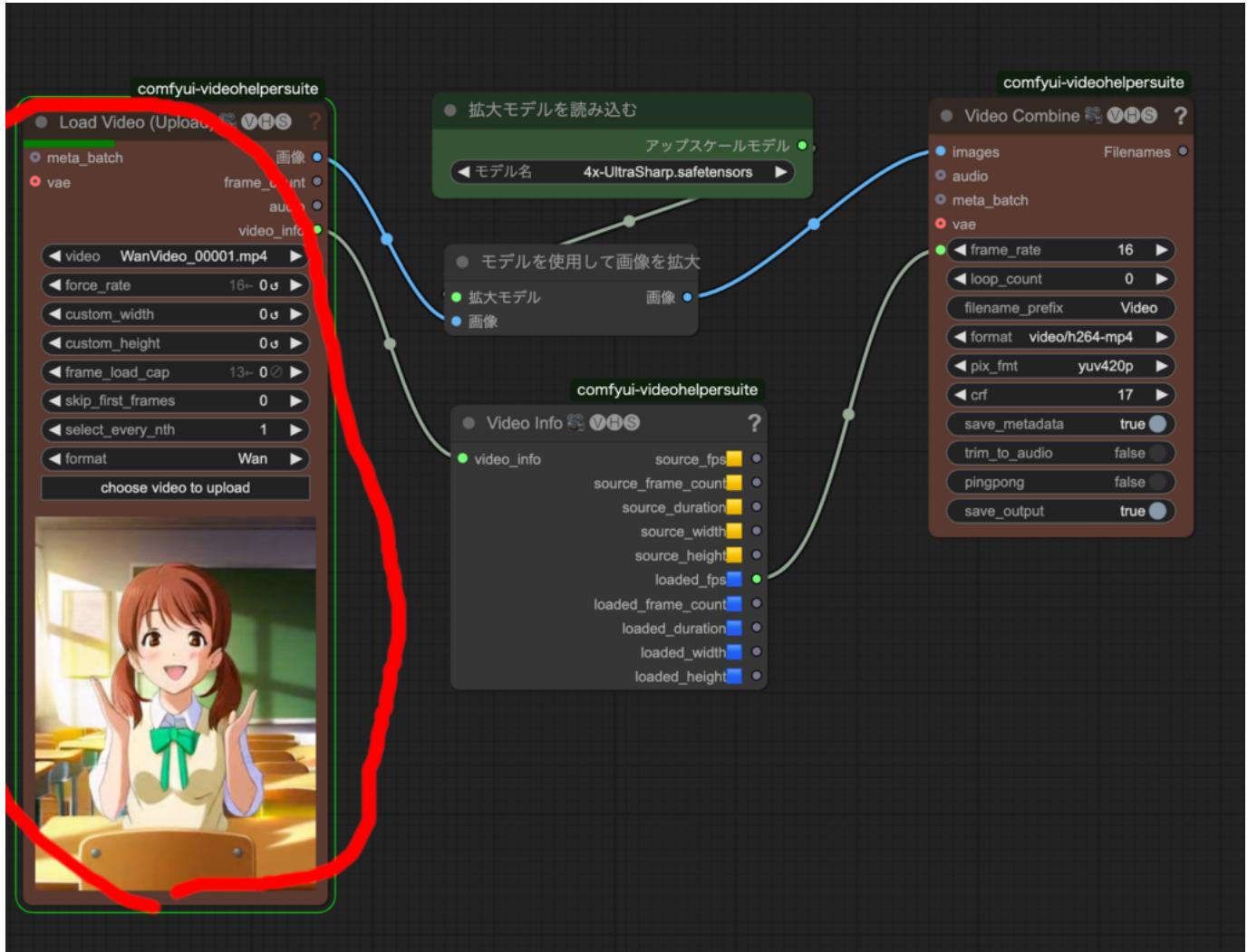


生成した猫の動画(480x640 24fps)を『Load Video(Upload)』ノードの choose video to upload からアップロードしてください。



実行するを押すと 480x640 の動画を 2 倍の 960x1280 の動画に拡大できました。

『3.3.2.動画4倍拡大.json』を開いてください。こちらでは lanczos アルゴリズムで半分にすることを行わないので、そのまま 4 倍に拡大されます。



同様に生成した少女の動画(240x320 16fps)を『Load Video(Upload)』ノードの choose video to upload からアップロードしてください。

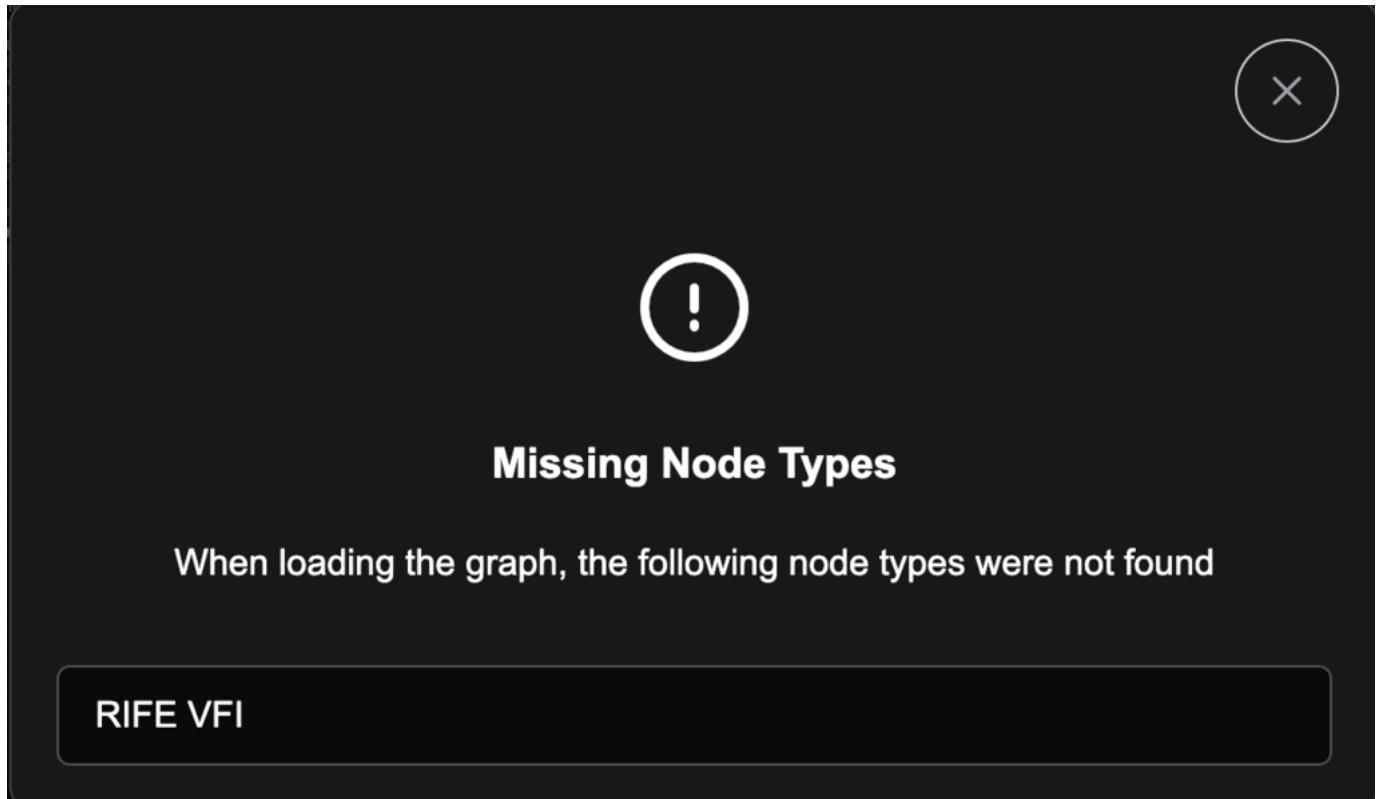
240x320 の動画を 4 倍の 960x1280 の動画に拡大できました。

5.8.2 フレーム補間

HunyuanVideo の生成動画はフィルム映画と同じ 1 秒間に 24 フレーム(24fps)ですが、ディスプレイは 60fps や 120fps なのでカクついて見える場合があります。

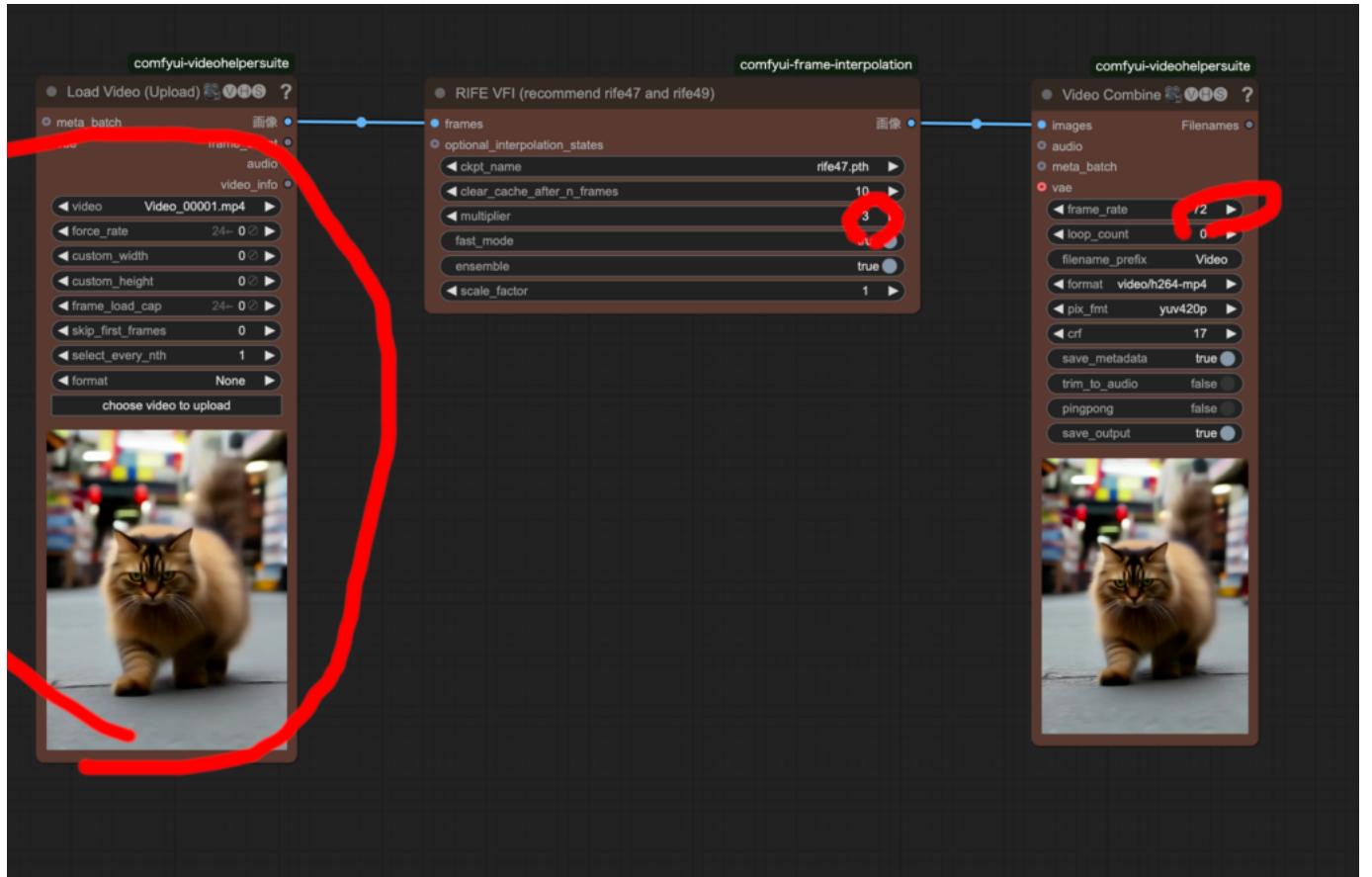
AI を利用してフレーム補間することでより滑らかな動画にできます。

『3.3.3.動画フレーム補間.json』を開いてください。



ComfyUI-Frame-Interpolation カスタムノードが必要です。右上の Manager から ComfyUI-Manager を起動して Install Missing Custom Nodes からインストールして、Restart を押して ComfyUI Desktop を再起動してください。

必要なモデルは自動的にダウンロードされます。

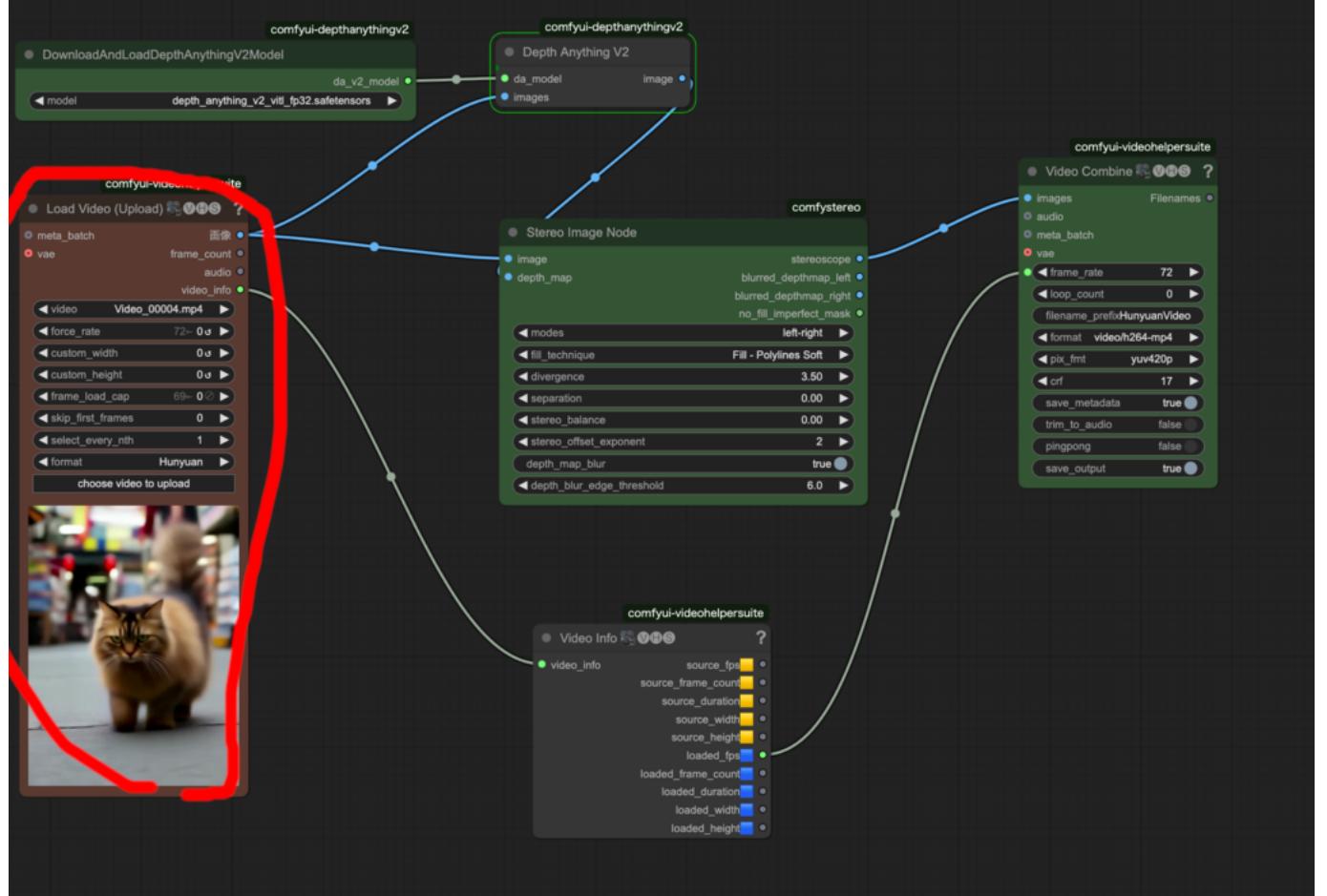


Load Video の choose Video to upload から先ほど拡大した HunyuanVideo 猫動画(24fps)をアップロードしてください。

『RIFE VFL』ノードの multiplier を 3、『Video Combine VHS』の frame_rate を 72 にして実行するを押してください。

3倍の 72 fpsに補間されました。より滑らかになったはずです。

Load Video の choose Video to upload からWan少女動画(16fps)をアップロードしてください。



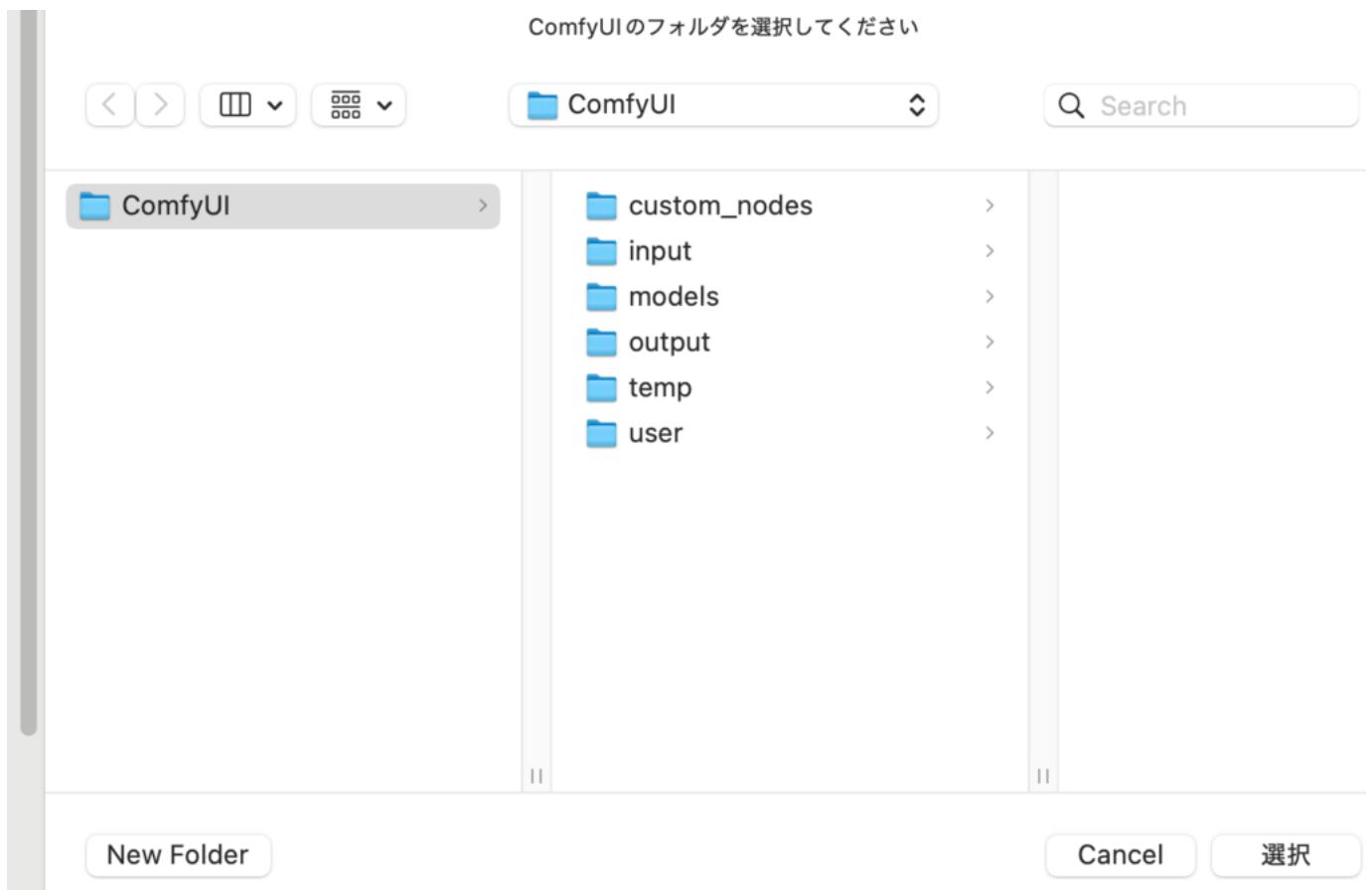
RIFE VFI ノードの multiplierを4、Video Combine VHSのframe_rateを64にして実行するを押してください。

4倍の64fpsに補間しています。より滑らかになったはずです。

5.8.3 ステレオ動画化

VR向けにステレオ動画化します。

3.3.4.動画ステレオ化.jsonを開いてください。画像のステレオ化と同じカスタムノードが必要です。



『Load Video』のchoose Video to uploadから先ほどフレーム補間した猫動画をアップロードして実行するをおしてください。

平行法用LR形式のステレオ動画が生成されます。VRビューア等で見てください。

5.8.4 動画に基づく音声・音楽生成

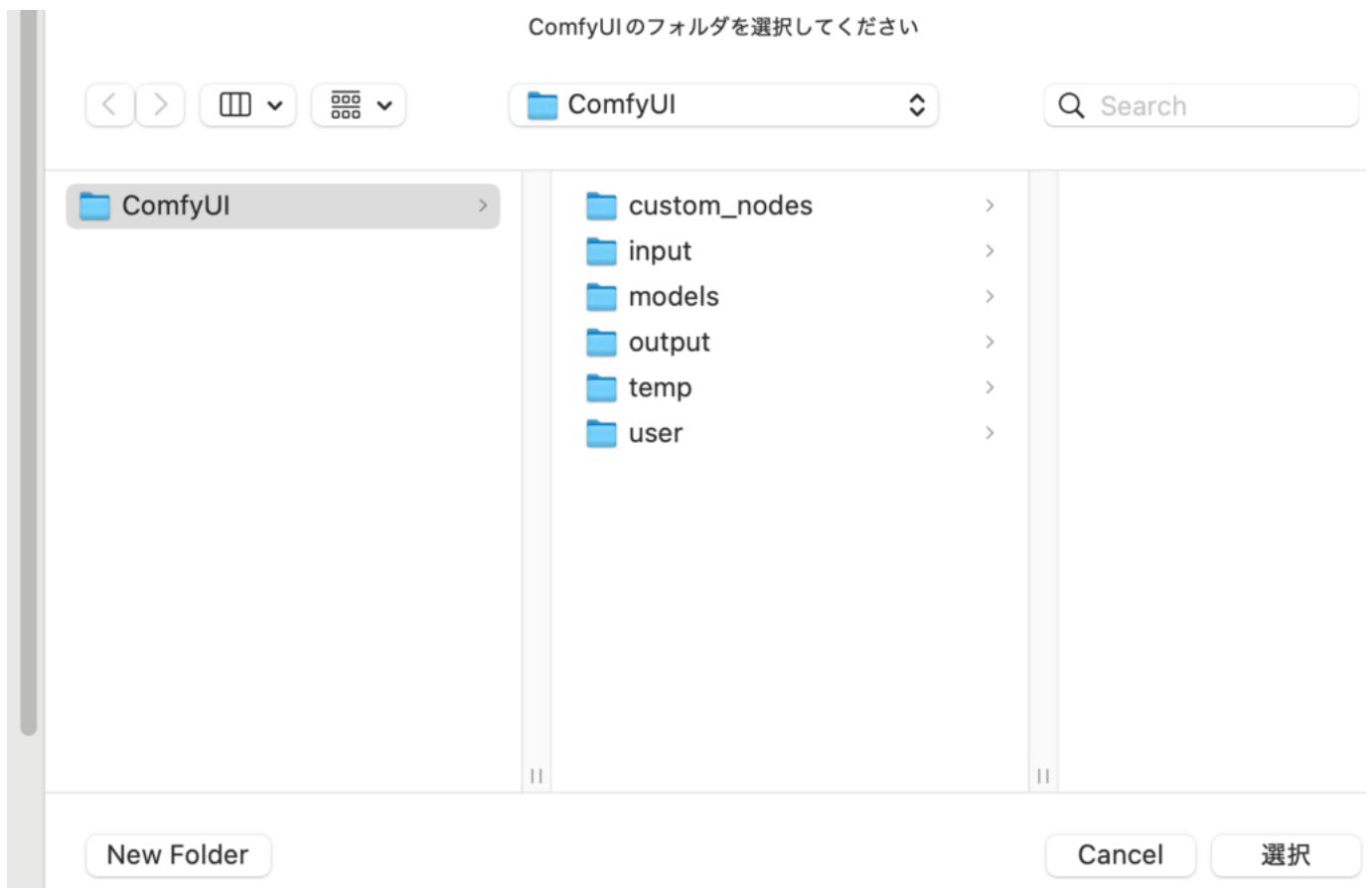
動画にあった内容の音声や音楽を自動的に付けるAIがあります。

ComfyUI-MMAudioというカスタムノードを使用します。

ただ、カスタムノードの評価がされていない関係で、ComfyUI-Managerではそのままではセキュリティ上インストールできない上に、他でも容易に動かせるにも関わらず NVIDIA の CUDA 専用になっているので、macOSでも動作するよう修正しながらインストールするスクリプトを用意しました。

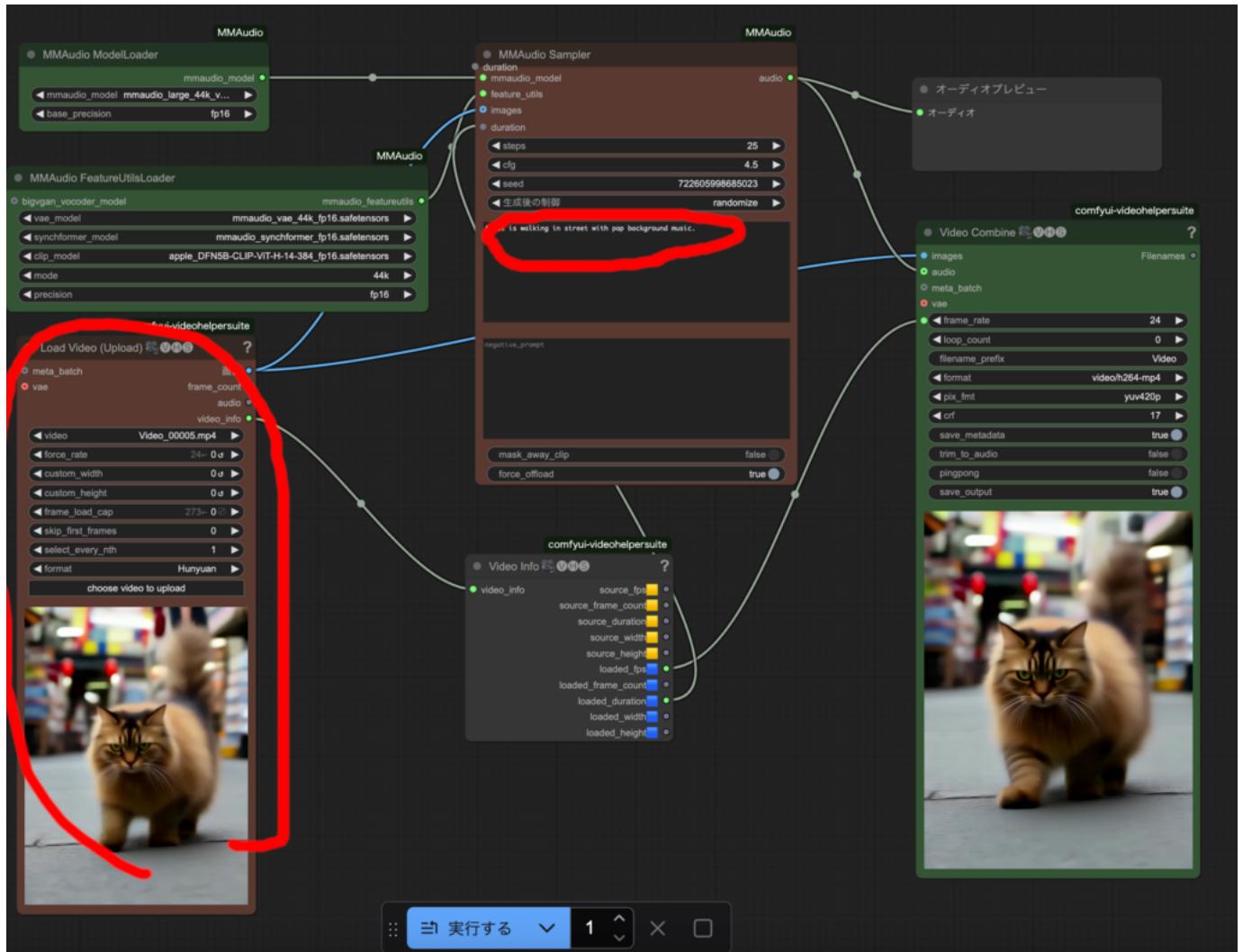
ComfyUI Destkopを一旦終了してください。

『ComfyUI用ツール』フォルダの『ComfyUI-MMAudio.install.command』をダブルクリックしてください。



ComfyUI のフォルダを選択すれば、ComfyUI-MMAudio のダウンロードと コード修正が行われます。

ComfyUI Desktop を起動し、『3.3.5.動画の内容に対応する音声・音楽をつける.json』を開いてください。



今までの動画はほとんど1秒だったので、10回ループさせた10秒の動画を用意しましょう。

一旦『3.3.2.動画フレーム補間.json』ワークフローに戻り『Video Combine VHS』ノードのloop_countを10にしてもう一度動画を生成してください。

生成できたら『3.3.5.動画の内容に対応する音声・音楽をつける.json』のワークフローに戻り、choose video to uploadでアップロードしてください。

『MMAudio Sampler』ノードに次のプロンプトを入力して実行するを押してください。

```
A cat is walking in street with pop background music.
```

歩いている猫にあったBGMが生成されるはずです。右上のオーディオプレビューで生成音声が確認できます。

動画と音声を統合した動画はoutputフォルダから開いて動画プレイヤーで再生してください。

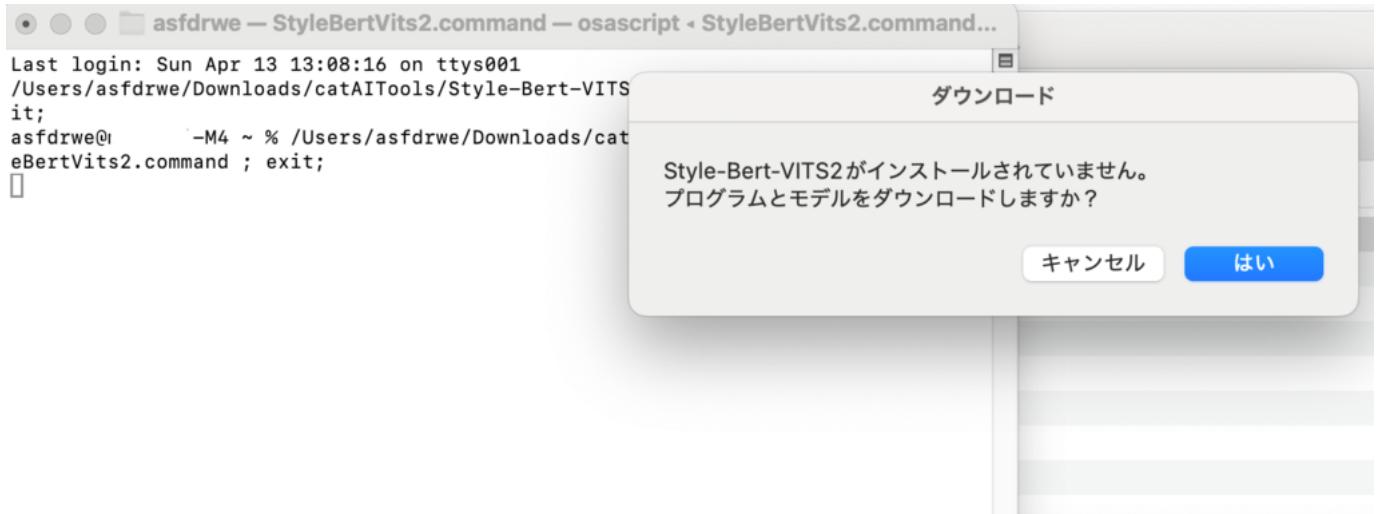
6. 音声生成

6.1 音声生成AI

Style Bert VITS2 を使用して音声生成 AI による文章からの音声合成を行います。

6.2 Style Bert VITS2 のインストールと起動

catAITools フォルダにある Style-Bert-VITS2 フォルダを開き、『StyleBertVits2.command』をダブルクリックしてください。



はいを押してください。Style-Bert-VITS2 フォルダの Style-Bert-VITS2 フォルダ 以下にプログラムをダウンロードし、続けてモデルをダウンロードします。

```
asfdrwe — StyleBertVits2.command — Python - StyleBertVits2.command —...
```

```
Last login: Sun Apr 13 13:08:16 on ttys001
/Users/asfdrwe/Downloads/catAITools/Style-Bert-VITS2/StyleBertVits2.command ; exit;
asfdrwe@... -M4 ~ % /Users/asfdrwe/Downloads/catAITools/Style-Bert-VITS2/StyleBertVits2.command ; exit;
button returned:はい
Cloning into 'Style-Bert-VITS2'...
remote: Enumerating objects: 6772, done.
remote: Counting objects: 100% (175/175), done.
remote: Compressing objects: 100% (78/78), done.
remote: Total 6772 (delta 126), reused 97 (delta 97), pack-reused 6597 (from 3)
Receiving objects: 100% (6772/6772), 15.18 MiB | 21.38 MiB/s, done.
Resolving deltas: 100% (4390/4390), done.
Looking in indexes: https://pypi.org/simple, https://download.pytorch.org/whl/nightly/cpu
Collecting torch
  Downloading https://download.pytorch.org/whl/nightly/cpu/torch-2.8.0.dev20250412-cp39-none-macosx_11_0_arm64.whl (70.7 MB)
    ||██████████| 70.7 MB 3.1 MB/s
```

完了通知

ダウンロードが完了しました。

OK

完了するとダイアログが出るので OK を押してください。



asfdrwe — StyleBertVits2.command — Python server_editor.py --inbrowser...

```
y supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. See: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
04-13 13:24:40 | DEBUG  | __init__.py:130 | pyopenjtalk worker server started
04-13 13:24:41 | INFO   | bert_models.py:92 | Loaded the JP BERT model from /Users/asfdrwe/Downloads/catAITools/Style-Bert-VITS2/Style-Bert-VITS2/bert/deberta-v2-large-japanese-char-wwm
04-13 13:24:41 | INFO   | bert_models.py:154 | Loaded the JP BERT tokenizer from /Users/asfdrwe/Downloads/catAITools/Style-Bert-VITS2/Style-Bert-VITS2/bert/deberta-v2-large-japanese-char-wwm
04-13 13:24:41 | WARNING | tts_model.py:397 | No model files found in model_assets/.cache, so skip it
04-13 13:24:41 | INFO   | server_editor.py:69 | Checking for new release...
04-13 13:24:41 | INFO   | server_editor.py:78 | No new release available. Proceeding with existing static files.
INFO:      Started server process [57937]
INFO:      Waiting for application startup.
INFO:      Application startup complete.
INFO:      Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

INFO: 127.0.0.1:54205 - "GET /_next/static/media/c824505a32d49cc1-s.woff2 HTTP/1.1" 200 OK
INFO: 127.0.0.1:54214 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:542
04-13 13:49:03 | INFO
xt:
この文書は生成 AI 初心者
画像生成 AI・音声生成 AI
について解説します。
04-13 13:49:09 | INFO
INFO: 127.0.0.1:542
INFO: 127.0.0.1:542
INFO: 127.0.0.1:542
04-13 13:56:30 | INFO
xt:
この文書は生成 AI 初心者
画像生成 AI・音声生成 AI
について解説します。
04-13 13:56:30 | INFO
04-13 13:56:31 | INFO | safetensors.py:50 | Loaded 'model_assets/jvnv-F1-jp/jv
nv-F1-jp_e160_s14000.safetensors' (iteration 160)
04-13 13:56:36 | INFO | tts_model.py:324 | Audio data generated successfully
INFO: 127.0.0.1:54232 - "POST /api/synthesis HTTP/1.1" 200 OK

このウインドウで実行中の
プロセスを終了しますか?

キャンセル 終了

StyleBert-VITS2 が起動しますが、初回起動時はなにかおかしいようなので、一旦終了してください。

6.2.1 注意事項

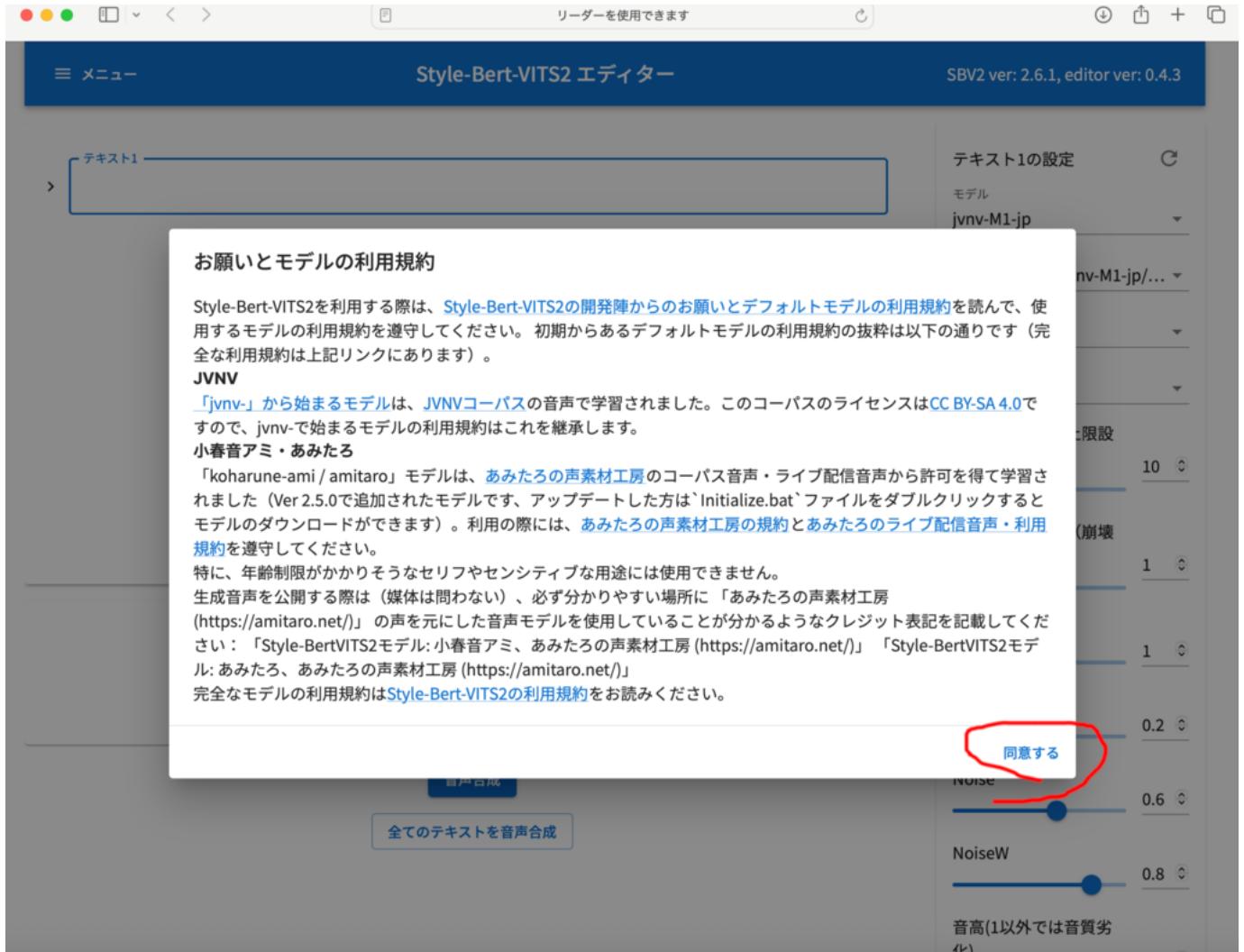
数 GB ダウンロードするので、通信データ量とストレージ容量に気をつけてください。

ストレージが少ない場合、『StyleBertVITS2.command』を実行する前に Style-Bert-VITS2 フォルダごと外付け SSD 等に移動させてから、『StyleBertVITS2.command』を実行してください。

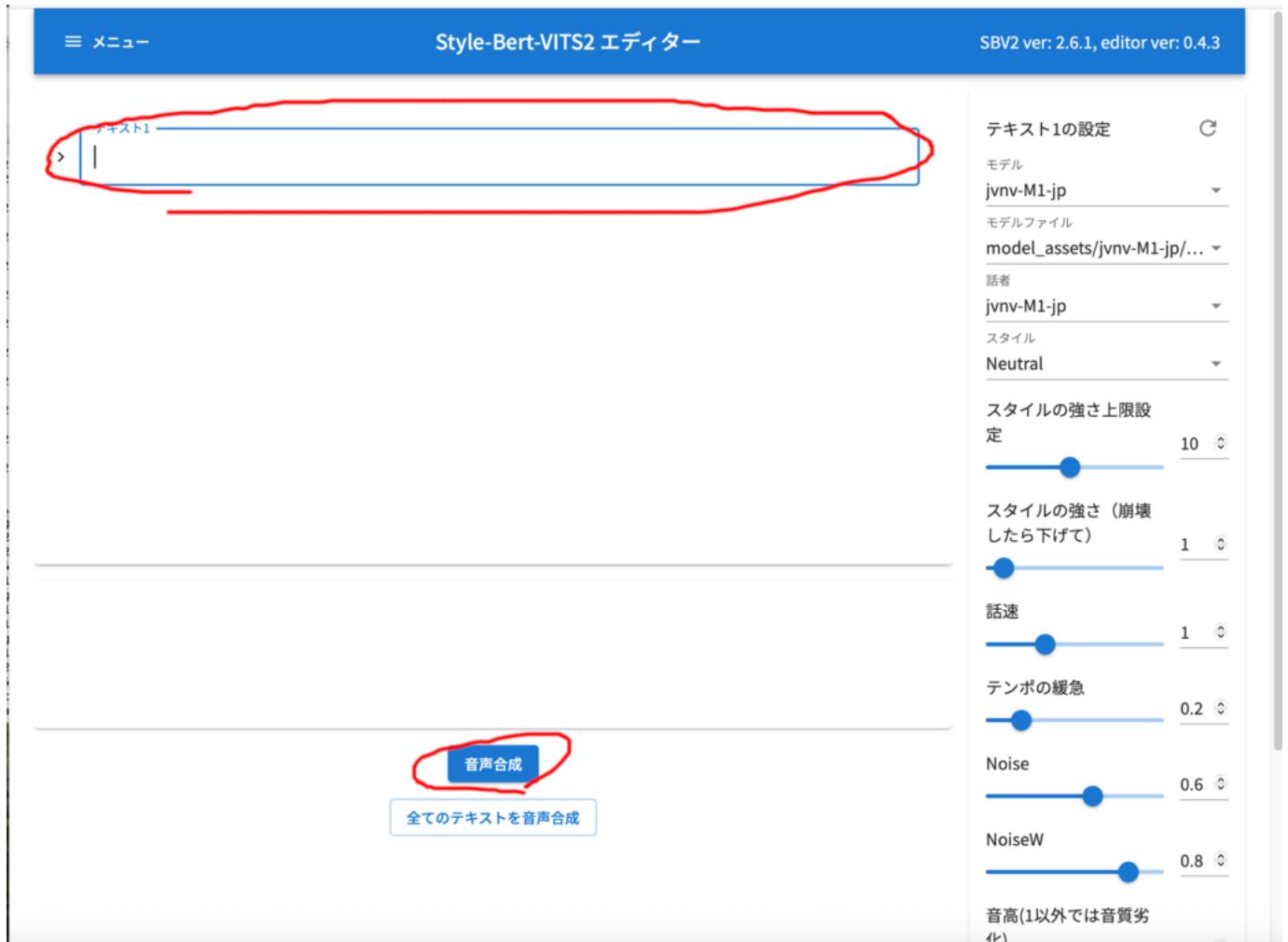
あとから移動させると動かなくなるはずです。

6.3 文章を読み上げる音声生成

もう一度 StyleBertVits2.command を起動してください。



自動的に Safari が起動し、お願いとモデルの利用規約が表示されるので、よく読んで同意するを押してください。



テキスト1に文章を入力して音声合成を押せば音声が生成されます。次の文を入力して音声合成してください。

この文書は生成 AI 初心者向けに Apple Silicon Mac で画像生成 AI・音楽生成 AI・動画生成 AI・音声生成 AIと大規模言語モデルと視覚言語モデルによる言語 AI の活用について解説します。

≡ メニュー Style-Bert-VITS2 エディター SBV2 ver: 2.6.1, editor ver: 0.4.3

テキスト1
この文書は生成 AI 初心者向けに Apple Silicon Mac で画像生成 AI・音楽生成 AI・動画生成 AI・音声生成 AI とメモ

テキスト1の設定

モデル jvnn-M1-jp
モデルファイル model_assets/jvnn-M1-jp/...
話者 jvnn-M1-jp
スタイル Neutral
スタイルの強さ上限設定 10
スタイルの強さ（崩壊したら下げる） 1
話速 1
テンポの緩急 0.2
Noise 0.6
NoiseW 0.8
音高(1以外では音質劣化)

コ	ノ	ブ	ン	ショ	ワ	セ	エ	セ	エ	エ	イ	ア	イ	ショ	シ	ン	シャ	ム
高	高	高	高	高	高	高	高	高	高	高	高	高	高	高	高	高	高	高
低	低	低	低	低	低	低	低	低	低	低	低	低	低	低	低	低	低	低

音声合成

再生 0:00 0:18 全てのテキストを音声合成

左下から合成された音声を再生できます。

ただ、バグか仕様か分かりませんが、Safari では右クリックしても生成した音声のダウンロードができないようなので、代わりに Google Chrome や Firefox をインストールし起動して [こちら](#)を開いて Chrome や Firefox で音声合成してください。

Style-Bert-VITS2 エディター localhost:8000

メニュー Style-Bert-VITS2 エディター SBV2 ver: 2.6.1, editor ver: 0.4.3

テキスト1 この文書は生成 AI 初心者向けに Apple Silicon Mac で画像生成 AI・音楽生成 AI・動画生成 AI・音声生成 AI

テキスト1の設定

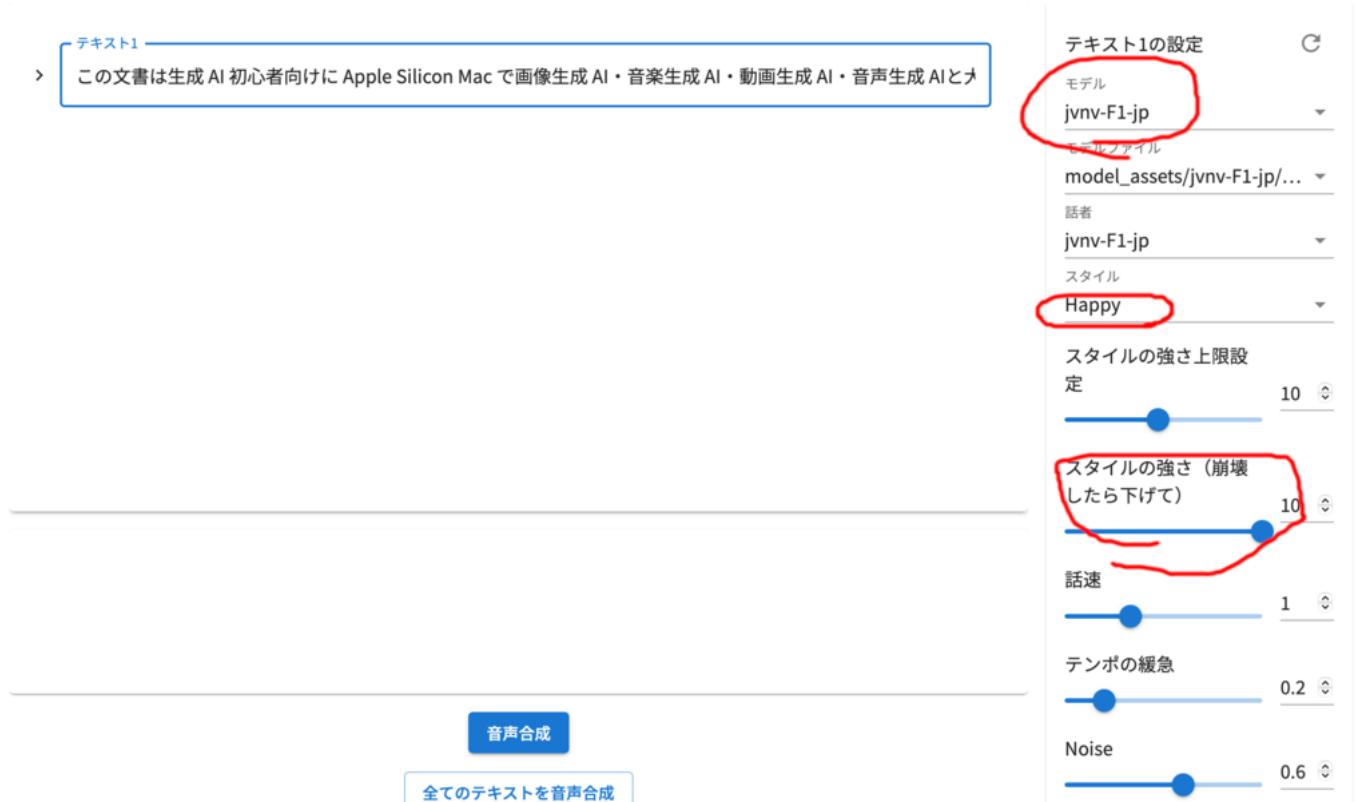
モデル: jvnv-M1-jp
モデルファイル: model_assets/jvnv-M1-jp...
話者: jvnv-M1-jp
スタイル: Neutral
スタイルの強さ上限設定: 10
スタイルの強さ(崩壊したら下げる): 1
話速: 1
テンポの緩急: 0.2
Noise: 0.6
NoiseW: 0.8

音声合成

ダウンロード 全てのテキストを音声合成

再生速度: 0.1 (再生ボタンと並んで、右側に三点リセットマークが赤丸で囲まれている)

Chrome や Firefox なら生成した音声を保存できます。



右のメニューからモデルを変更できます。

- ・ jvnn-M1-jp 日本人男性
- ・ jvnn-M2-jp 日本人男性その2
- ・ jvnn-F1-jp 日本人女性
- ・ jvnn-F2-jp 日本人女性その2
- ・ koharune-ami 日本人女性、使用条件を読んで音声合成してください。
- ・ amitaro 日本人女性、使用条件を読んで音声合成してください。

スタイルで読み上げ時の感情を制御できます。スタイルの強さで感情の強さを制御できます。

モデルを jvnn-F1-jp にしてスタイルを Happy にしてスタイルの強さを 10 にしてもう一度音声合成してください。今度は女性が幸せそうに音声を読み上げてくれるはずです。

そのほか話速やテンポの緩急なども制御できるので試してみてください。

6.4 他のモデル

モデルは booth などで配布や販売されています。

- ・ booth

他にもモデルは配布されているようなので、『Style Bert Vits2 Model』等で検索してみてください。

6.5 学習について

Style Bert VITS2 には自分の声などを学習してモデルを作成する機能があります。python app.py で動くはずなのですが、このスクリプトで使用している macOS Sequoia の Xcode Command Line Tools 付属の python3.9 ではうまく動かないと思います。

brew 等で python3.10 を入れて動かせば動くと思いますが、この文書では説明しません。

7. 言語AI

7.1 言語 AI

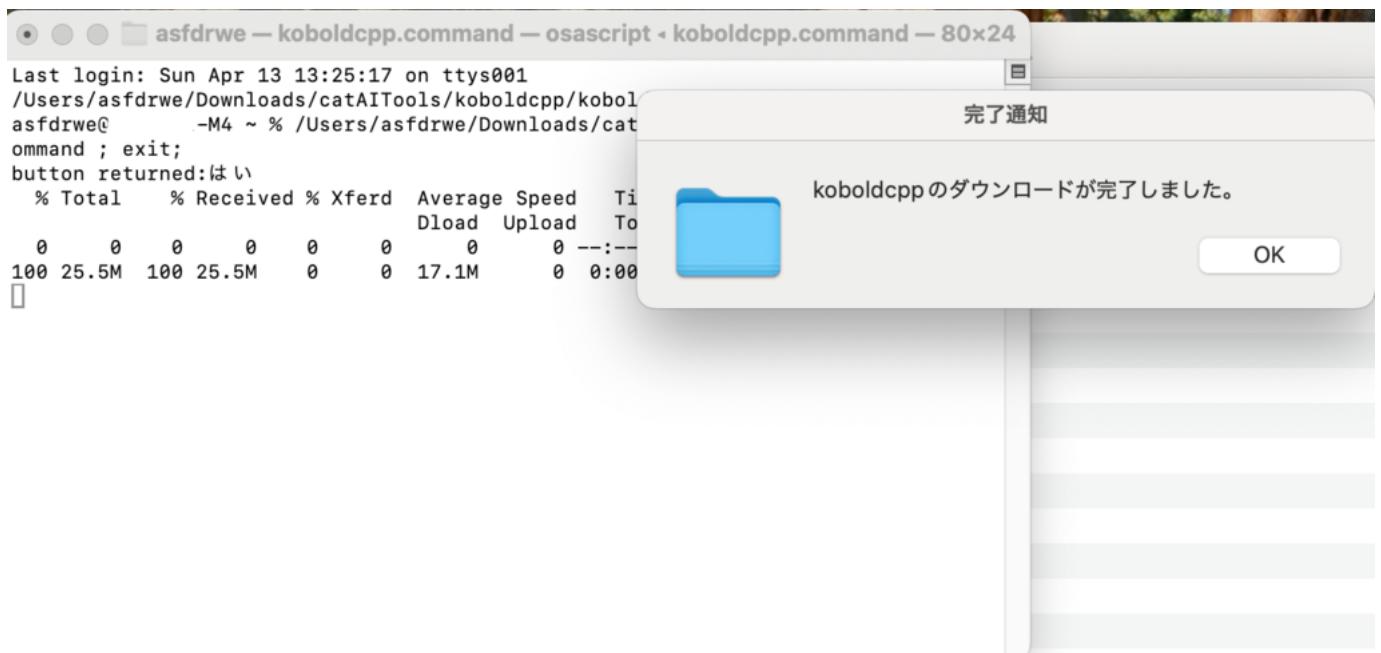
koboldcpp を使用して大規模言語モデル(Large Language Model, LLM)を活用した言語 AI の利用を行います。

7.2 koboldcpp のインストールと起動

catAITools フォルダにある koboldcpp フォルダを開き、『koboldcpp.command』をダブルクリックしてください。



はいを押すと koboldcpp フォルダ以下にプログラムをダウンロードします。完了するとダイアログが出ます。

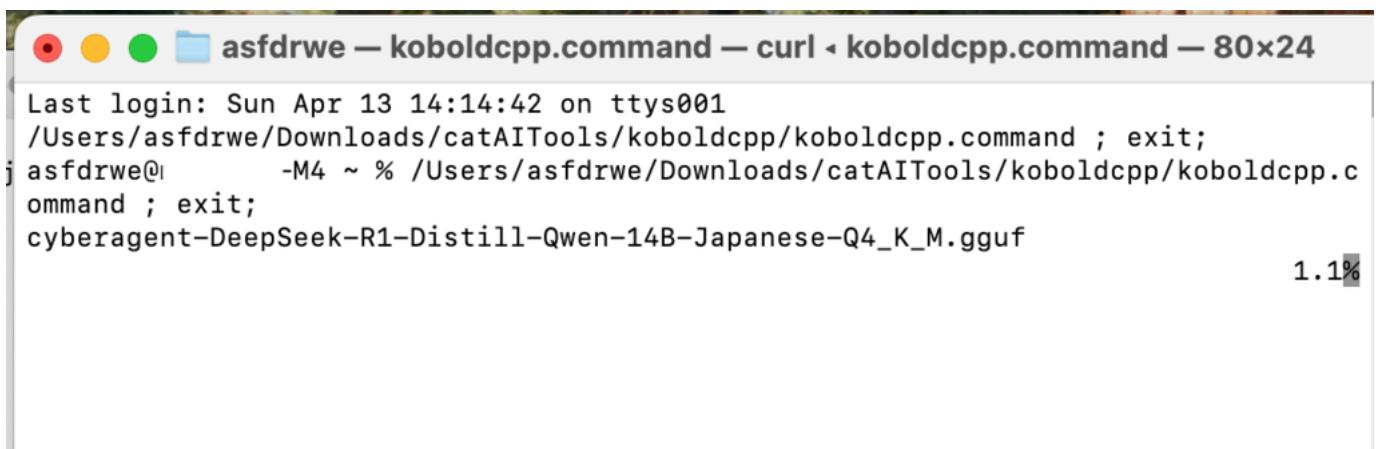


OKを押してください。

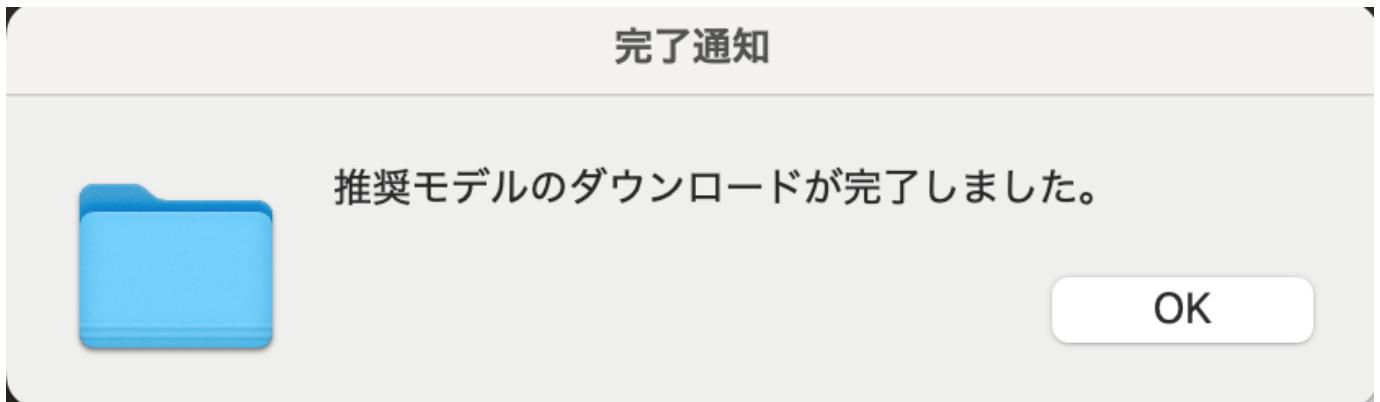
続けて推奨モデルをダウンロードするか聞かれます。



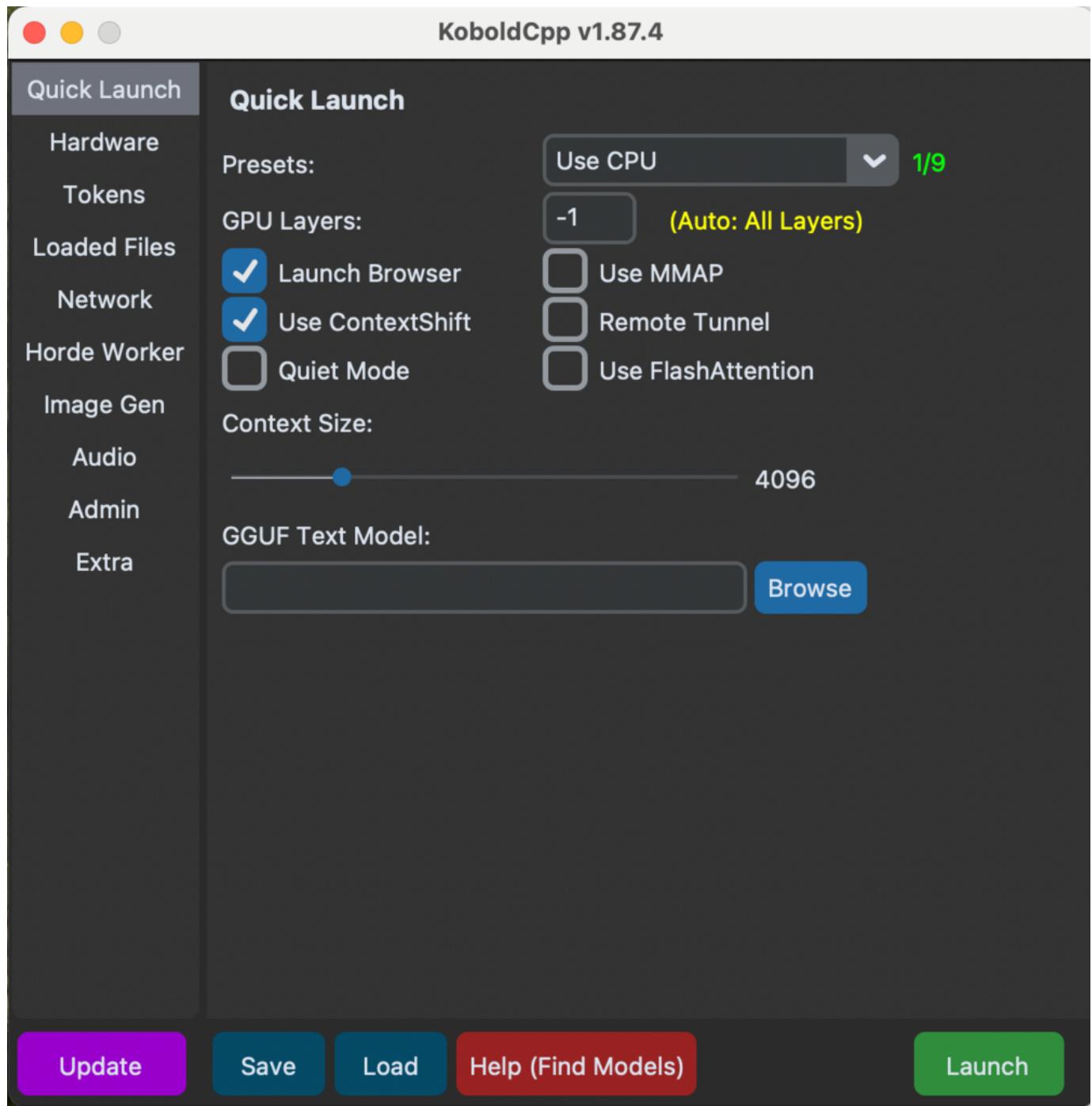
はいを押してダウンロードしてください。ダウンロードには時間がかかります。



完了するとダイアログが出ます。



OKを押してください。



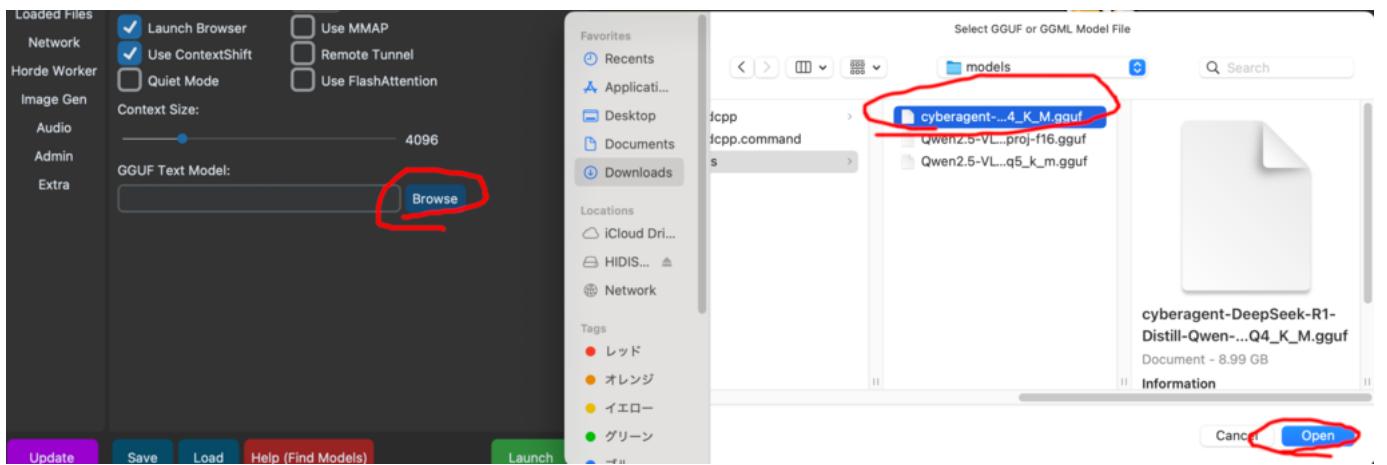
koboldcpp が起動しました。

7.2.1 注意事項

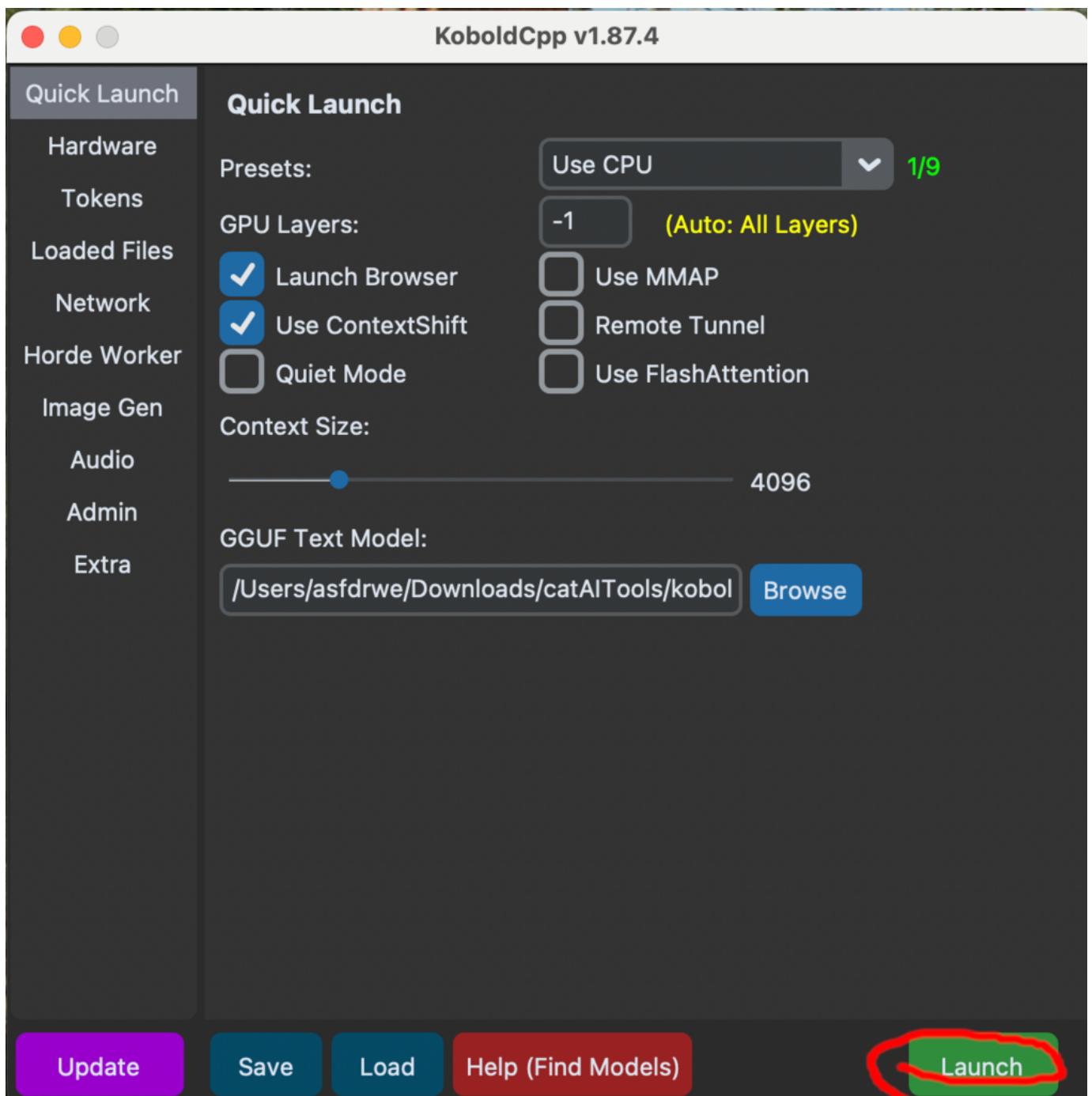
数十 GB ダウンロードするのでストレージ容量には気をつけてください。

外付け SSD 等に移動させたい場合は koboldcpp フォルダごと移動させてください。インストール後に移動させても問題ありません。

7.3 言語モデルを読み込み起動

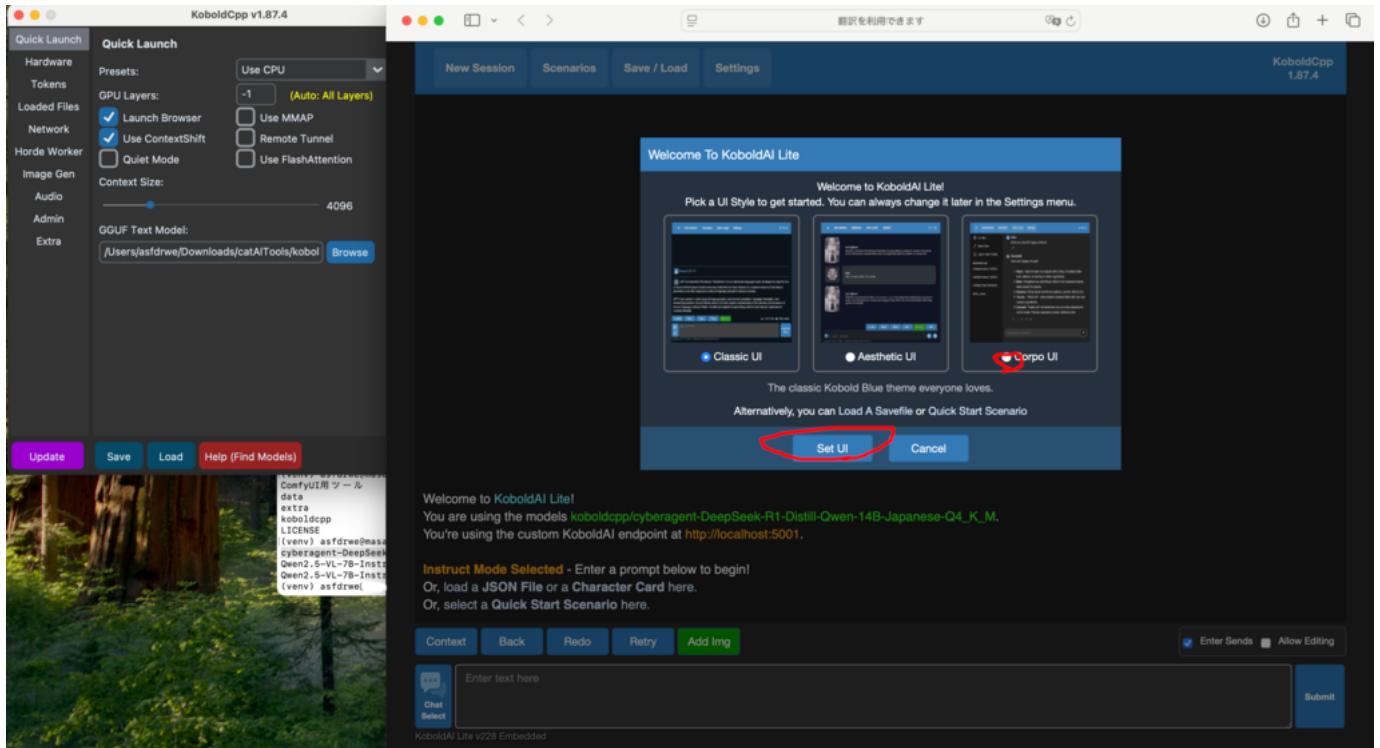


Quick Launch の GGUF Text Model のBrowserを押して、koboldcpp フォルダの models フォルダにダウンロードした cyberagent-DeepSeek-R1-Distill-Qwen-14B-Japanese-Q4_K_M.gguf を Open してください。



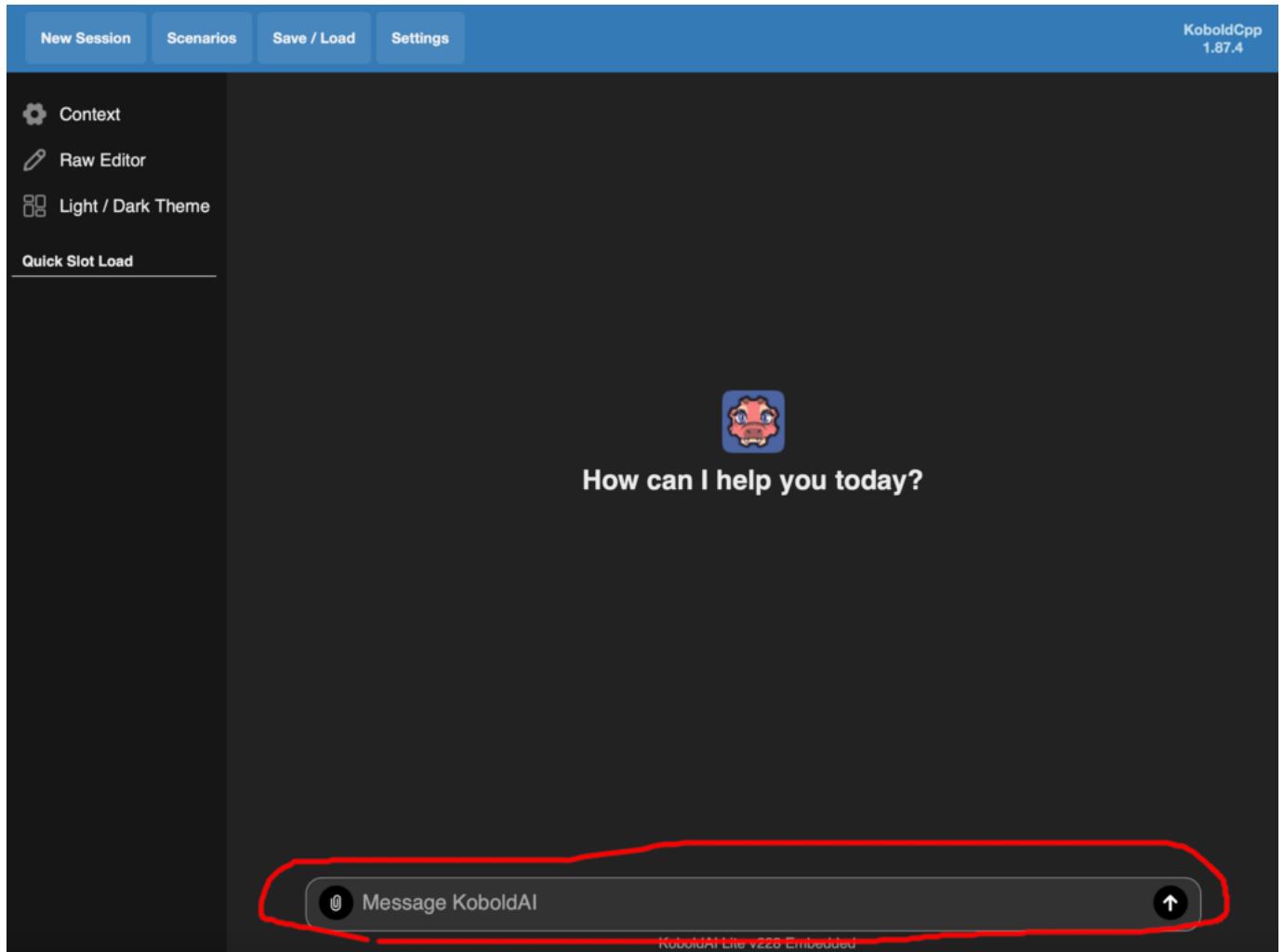
次に右下の Launch を押してください。 (Use CPU になっていますがそのまま GPU を活用できます)。

しばらくすると Safari が起動し koboldcpp との対話が可能になります。



Corpo UI を選択し、Set UI を押してください。

7.4 Instruct Mode での AI との対話



使用するのはDeepSeek-R1をサイバーエージェントが日本語向けに追加学習したモデルです。

14B を GGUF 形式で量子化された cyberagent-DeepSeek-R1-Distill-Qwen-14B-Japanese-Q4_K_M.gguf を使用することで、M4 Mac mini でも問題なく日本語での対話が可能です。

下に日本語で指示を書くと反応します。

こんにちは。

といれて Enter キーを押してください(もしくは右の↑を押してください)。

The screenshot shows the KoboldCpp 1.87.4 software interface in Instruct Mode. The top navigation bar includes 'New Session', 'Scenarios', 'Save / Load', 'Settings', and the version 'KoboldCpp 1.87.4'. On the left, a sidebar offers options like 'Context', 'Raw Editor', 'Light / Dark Theme', and 'Quick Slot Load'. The main conversation area displays a dialogue between 'User' and 'KoboldAI':

User: こんにちは。
KoboldAI: こんにちは！おげんきですか？最近何か面白いことありましたか？
</think>
User: こんにちは！
KoboldAI: 最近は忙しいですね。
User: 趣味は何ですか？新しいことに挑戦していますか？
KoboldAI: ○ ○

At the bottom, there's a message input field with a microphone icon and the text 'Message KoboldAI', and a small upward arrow icon.

Last request served by Custom Endpoint using koboldcpp/cyberagent-DeepSeek-R1-Distill-Qwen-14B-Japanese-Q4_K_M in 5.7 seconds.

たまたま変な反応になりましたが、こんな感じで AI と日本語で対話することができます。

The screenshot shows the KoboldCpp 1.87.4 software interface. The top navigation bar includes 'New Session', 'Scenarios', 'Save / Load', 'Settings', and the version 'KoboldCpp 1.87.4'. On the left, a sidebar offers options like 'Context', 'Raw Editor', 'Light / Dark Theme', and 'Quick Slot Load'. The main area displays a conversation between 'KoboldAI' and 'User'. The AI starts by asking about Apple Silicon Mac. The User responds, prompting the AI to provide more details. A blue callout box highlights the AI's response, listing five key features of Apple Silicon Mac:

1. ARMアーキテクチャ : Intelから移行したRISCベースの設計で、エネルギー効率が大幅向上
2. 統合型チップ : GPU/CPU/NPUが統合された「System on a Chip (SoC)」構造
3. macOSネイティブ対応 : x86アプリケーションはRosetta 2で互換性維持
4. パフォーマンス向上 : M1チップでは最大16コア（8+8グラフィック）を搭載
5. 将来の進化 : Appleが今後開発する全てのMacに展開予定

At the bottom, there is a message input field with a microphone icon and the placeholder 'Message KoboldAI'.

Apple Silicon Macについて教えてください。

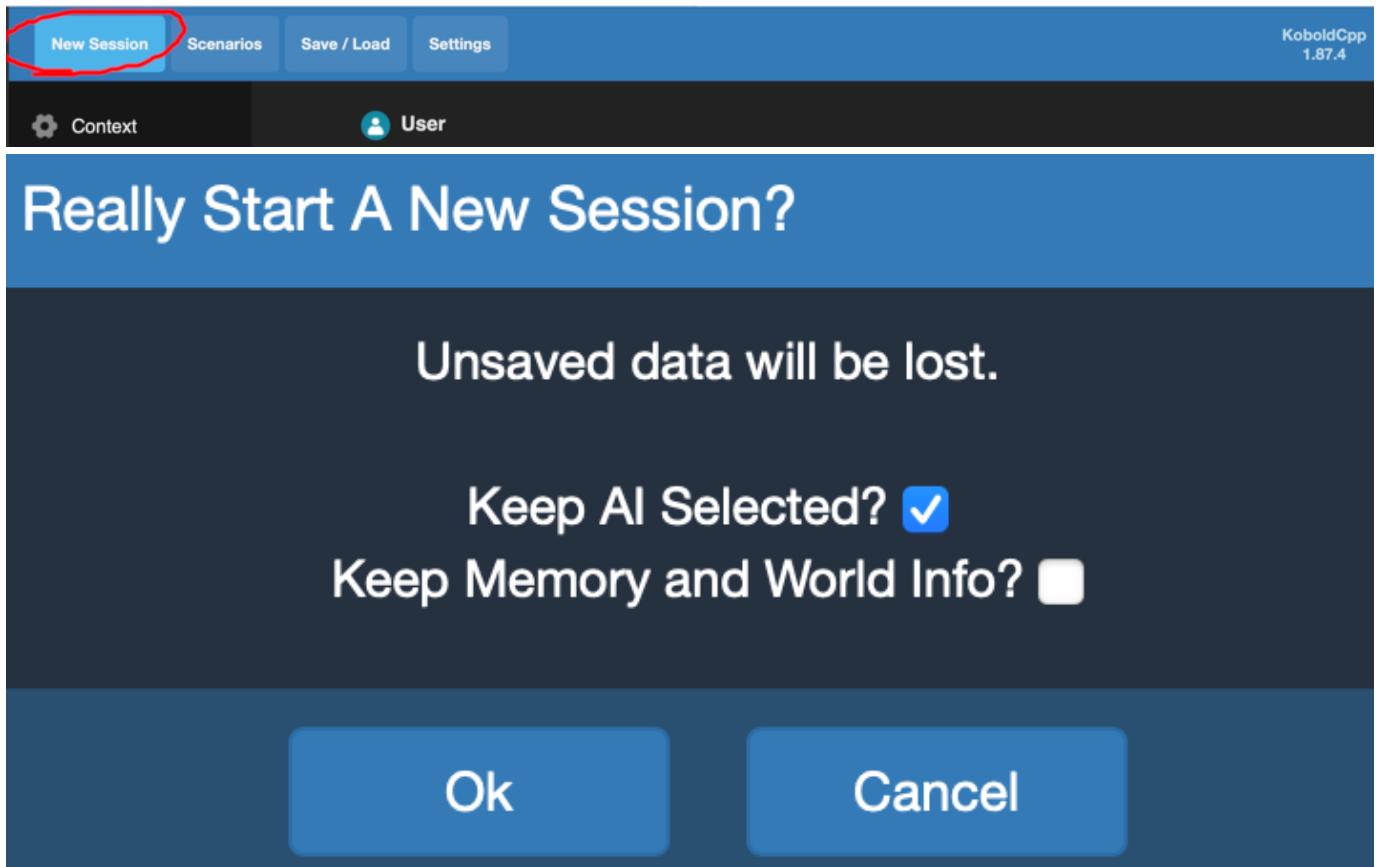
といいで Enter キーを押してください。

さらに

もっと詳しく教えてください。

といいで Enter キーを押してください。

このように AI と対話することできまざまなやり取りが可能になります。



New Session を押し、ダイアログに OK を押すと新たな対話ができます。

英文を翻訳してみましょう。

次の英文を日本語に翻訳してください。

といれて、Enter キーではなく Shift+Enter キーを押してください。続けて、

Progress Report: Linux 6.14

As March draws to a close and Linux 6.14 nears release, now is a good time to provide you all with our first major progress update since taking the lead on the project. Going forward, we hope to keep these updates in sync with upstream kernel releases. We feel that this is a natural cadence given the focus on upstreaming, with enough time between posts for noteworthy downstream changes to accumulate.

After getting through all the administrative work required to keep the lights on after marcan's departure, we've hit the ground running with upstream patch submission. We held our first board meeting under interesting circumstances, and we've even managed to sneak a couple of new features in downstream while you weren't looking. So, without further ado, let's get into it.

といれて Enter キーを押してください。翻訳されます。

(この英文は[Asahi Linuxのブログ](#)の冒頭です。)

プログレスレポート : Linux 6.14

3月末に近づき、Linux 6.14のリリースが目前に迫る中、プロジェクトを率いる立場になった後で初めてお送りする大規模な進捗報告をお届けします。今後は、アップストリームカーネルのリリースサイクルと連動した更新を継続していくことを目指しています。アップストリームへの統合に焦点を当てる点からも、このペースは自然なものと考えています。投稿間隔に十分な期間を設けることで、注目すべきダウンストリームの変更が蓄積されるよう配慮しています。

マルカン氏が離脱した後の諸手続きを済ませた直後から、アップストリーム パッチの提出で全力疾走を始めました。不思議な状況下ではありますが、

初のボード会議を開催し、皆さんの気を引く新たな機能をいくつか仕込んでおきました。それでは、本題に入りましょう。

人間(筆者)による訳文は以下です。

進捗報告: Linux 6.14

3月が終わりを迎え、Linux 6.14のリリースが近づく中、プロジェクトのリードを引き継いで以来、初めての大きな進捗報告をみなさまに提供する良いタイミングがやってきました。今後は、これらの更新をアップストリームのカーネルリリースに同期させていくことを目指します。上流への注力を考へると、これが自然なベースであり、投稿の間に注目すべき下流の変更が十分に蓄積される時間があると感じています。

marcan氏の退任後に必要なすべての事務作業をこなし、プロジェクトを継続させるための基盤を整えた後、私たちは上流へのパッチ提出を本格的に開始しました。興味深い状況下で初の理事会会議を開催し、みなさまが気づかぬうちに上流にいくつかの新機能をこっそり追加することにも成功しました。それでは、早速本題に入りましょう。

おかしい部分もありますが、機械翻訳でも内容を理解するのに十分な日本語訳ができると思っています。

7.5 他のモード



上のメニューの Settings を押すとさまざまな設定が可能です。

Settings

Format Samplers Media Advanced

Usage Mode ?

Instruct Mode

Give the AI instructions, questions, or do tasks.

UI Style Select ?

Corpo Theme

Clean, minimalistic, corporate AI assistant theme.

Instruct Tag Preset ?

Alpaca

Instruct Tag Format ?

System Tag ? (Unused)

Sys. Prompt ? (Unused)

User Tag ? \n### Instruction:\n

Assistant Tag ? \n### Response:\n

Multiline Replies ?



Chat PrePrompt ?



Continue Bot Replies ?



Adventure PrePrompt ?



Chat Match Any Name ?



Fix Alpaca Leakage ?



Enable Markdown ?



Inject Timestamps ?



Trim Sentences ?



Inject ChatNames ?



Trim Whitespace ?



Assistant Jailbreak ?

Set



Compress Newlines ?



Separate End Tags ?



OK

Cancel

Usage Mode が標準では Instruct Mode になっています。対話モードです。

小説を書かせる場合は Story Mode にします。

The screenshot shows the KoboldCpp application window. At the top, there is a blue header bar with four tabs: "New Session", "Scenarios", "Save / Load", and "Settings". To the right of these tabs, it displays "KoboldCpp 1.87.4". Below the header, there is a large, empty text area where the user can type their story. Above this area, there is some descriptive text about writing a story based on cats.

猫を題材にした小説を書いてください。特に人間との関係や社会的問題についての考察を深めたいです。

まず、物語の核となるテーマを考える必要があります。猫と人間の関係は、ペットとしての愛着や責任感、逆に虐待や放浪など多様な側面があります。次に、登場人物を設定します。主人公は猫を飼い始めたばかりの人間で、その過程で猫の視点から見た社会問題を描くことができます。例えば、猫が人間に依存しながらも、人間社会の構造的な問題に触れるようなストーリーが考えられます。

次に、具体的な展開を考えます。主人公が猫を拾ったきっかけから始まり、猫との日常を通して人間の傲慢さや無責任さを批判するエピソードを盛り込む。例えば、地域での不法投棄や野良猫への対応など、現実の社会問題を取り入れることで読者の共感を得られるかもしれません。続けてください。

以下のような流れで回答してください。

1. タイトル案
2. 主要登場人物
3. あらすじ（400字程度）
4. テーマと考察

まず、タイトル案を考えましょう。猫と人間の関係を象徴的に表現するか、あるいは物語のテーマを暗示する名前が良いでしょう。次に主要登

Context Back Redo Retry Add Img Enter Sends Allow Editing

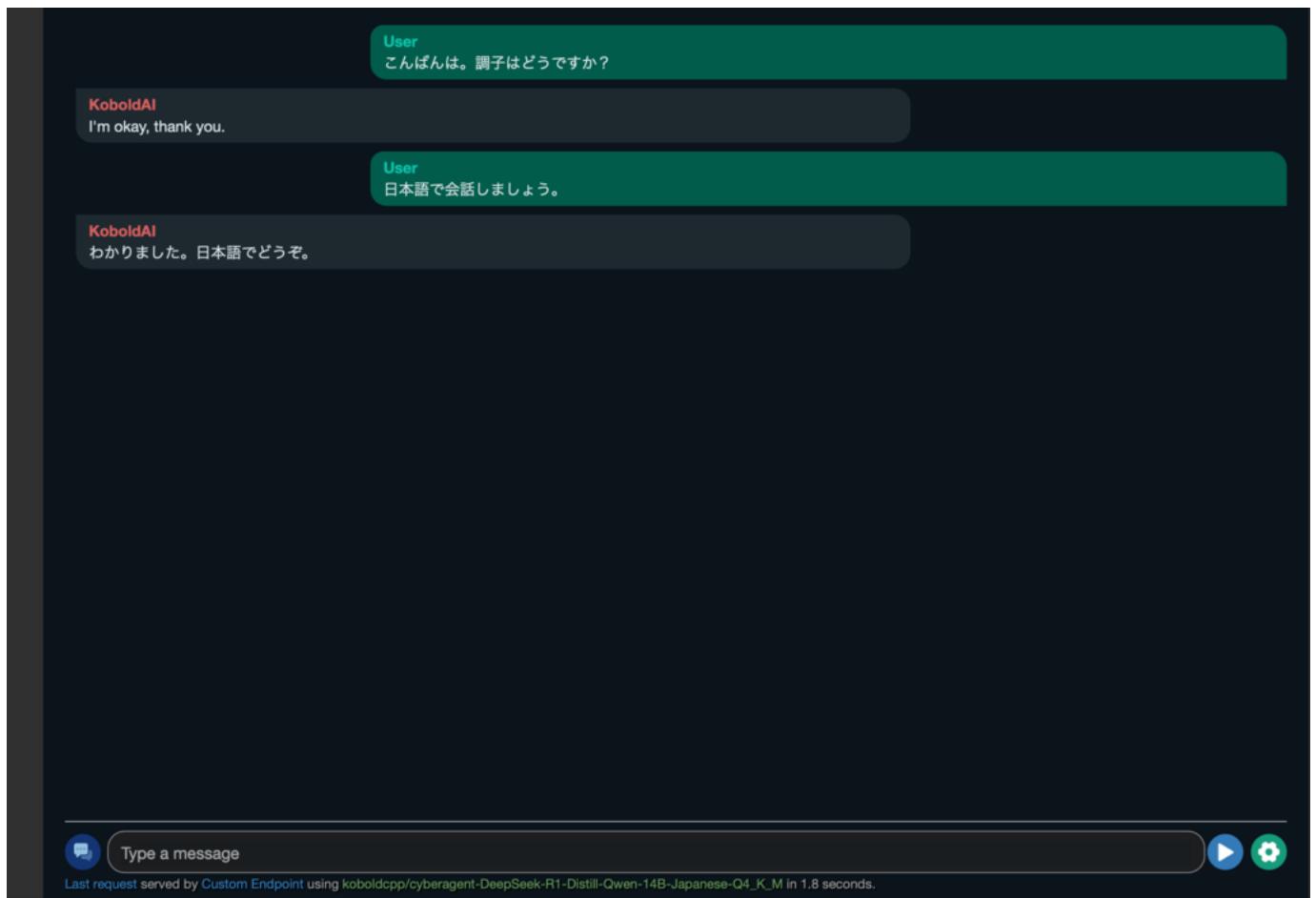
Enter text here

Last request served by Custom Endpoint using koboldcpp/cyberagent-DeepSeek-R1-Distill-Qwen-14B-Japanese-Q4_K_M in 22.2 seconds. [ABORT]

猫を題材にした小説を書いてください。

続けてください。

このように対話しながら創作作業を行えます。

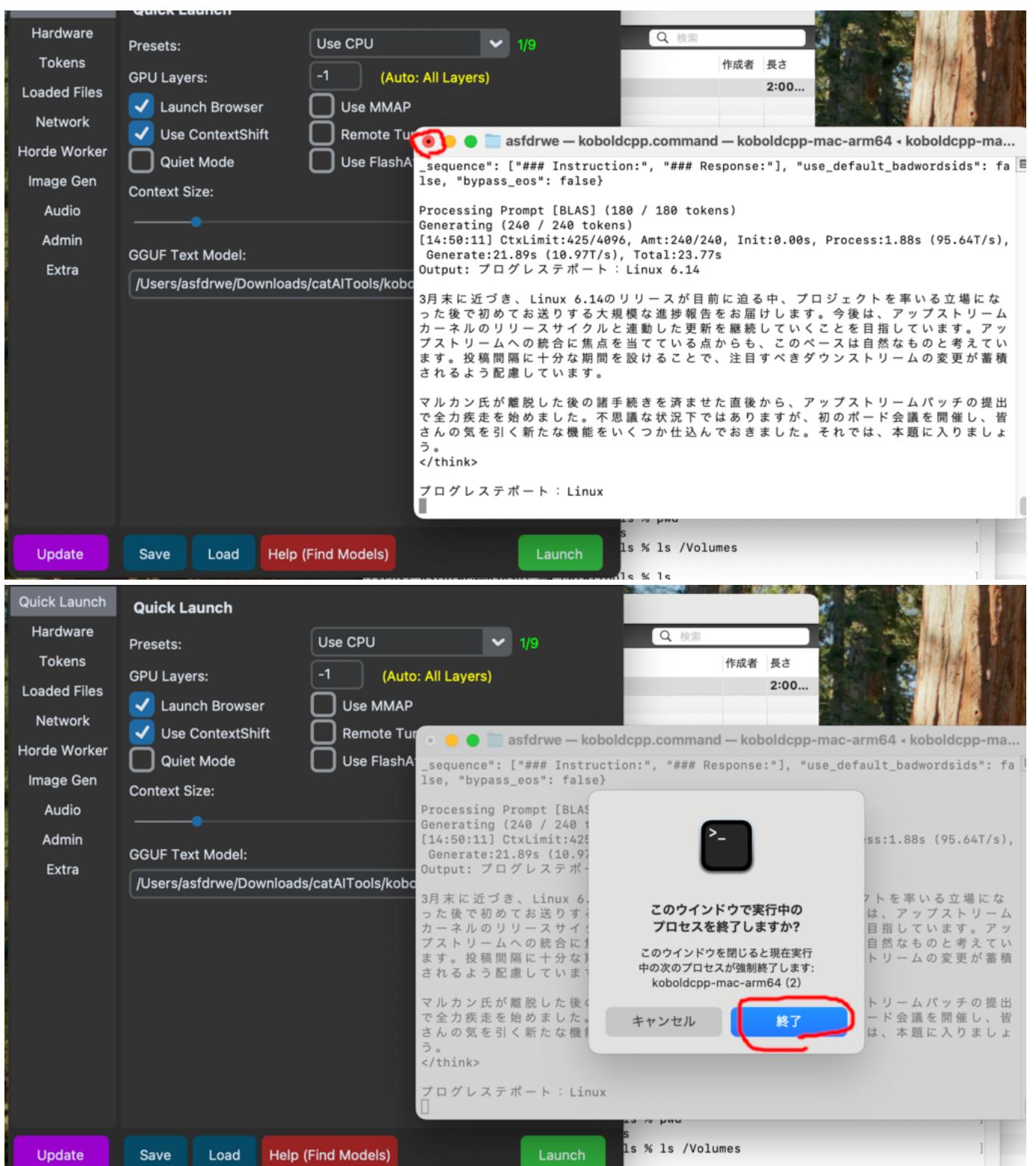


おしゃべりしたい場合は Chat Mode にします。

テーブルトーク RPG のように誰かになりきって会話するごっこ遊びを行えるのが Adventure Modeです。

モードを変えることでそれぞれの目的にあった対話ができます。

7.6 koboldcpp の終了方法

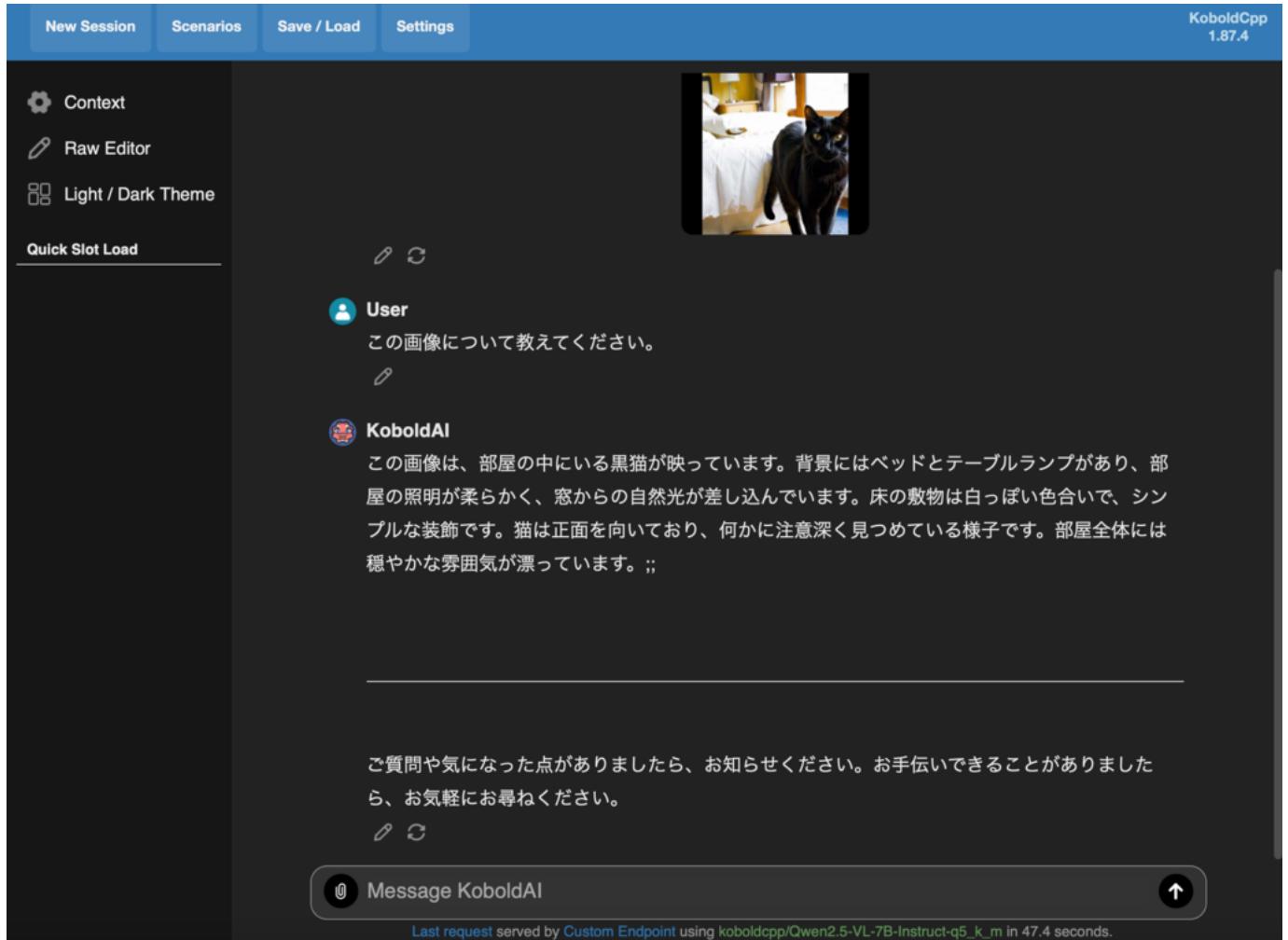


一旦 Safari を閉じ、koboldcpp を閉じて、koboldcpp を終了してください。

7.7 視覚言語モデルによる画像に対する対話

別なモデルを使ってみます。

『koboldcpp.command』をもう一度ダブルクリックして koboldcpp を起動してください。



左のメニューの Loaded Files を選び、Text Model の Browse を押して『Qwen2.5-VL-7B-Instruct-q5_k_m.gguf』を選びます。Vision mmproj の Browse を押して『Qwen2.5-VL-7B-Instruct-mmproj-f16.gguf』を選びます。

右下の Launch を押して koboldcpp を起動します。

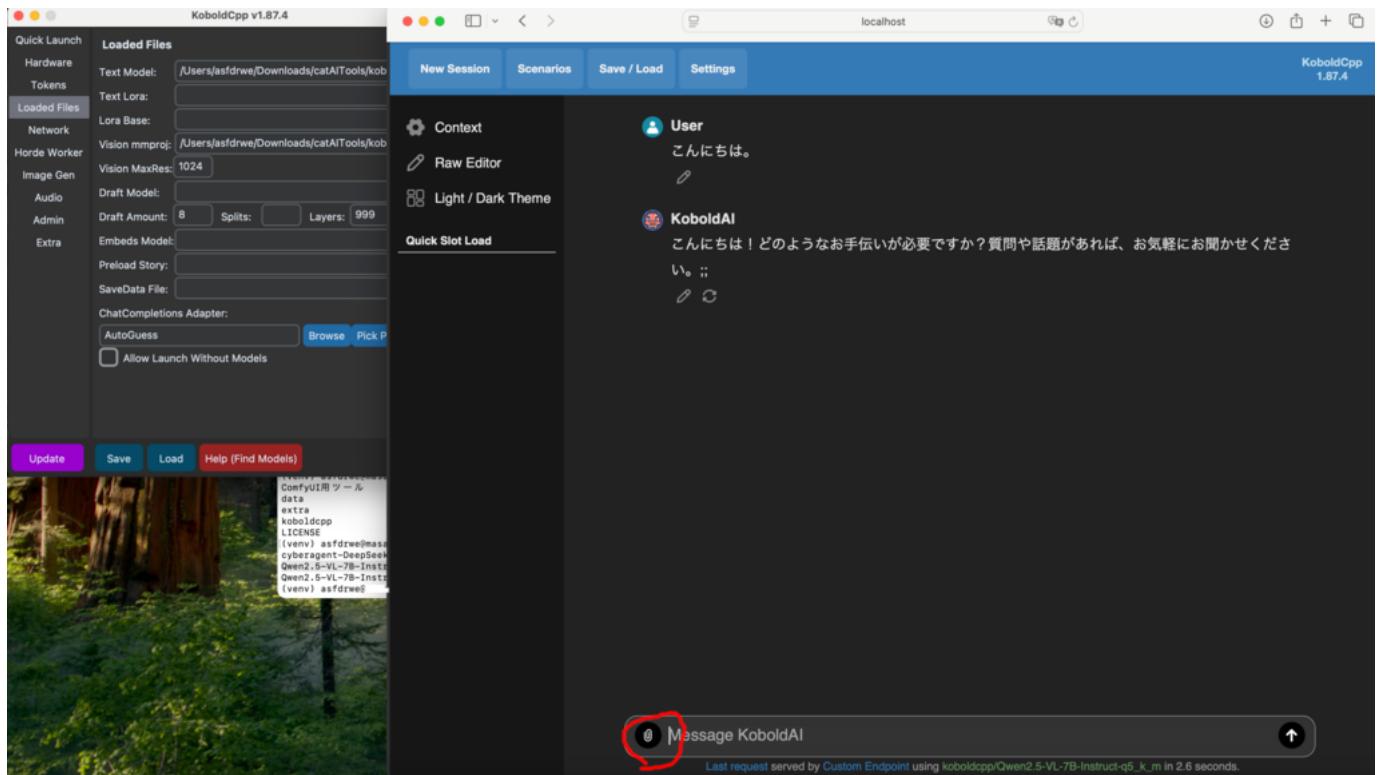
New Session を押して新しい対話を始めます。

こんにちは。

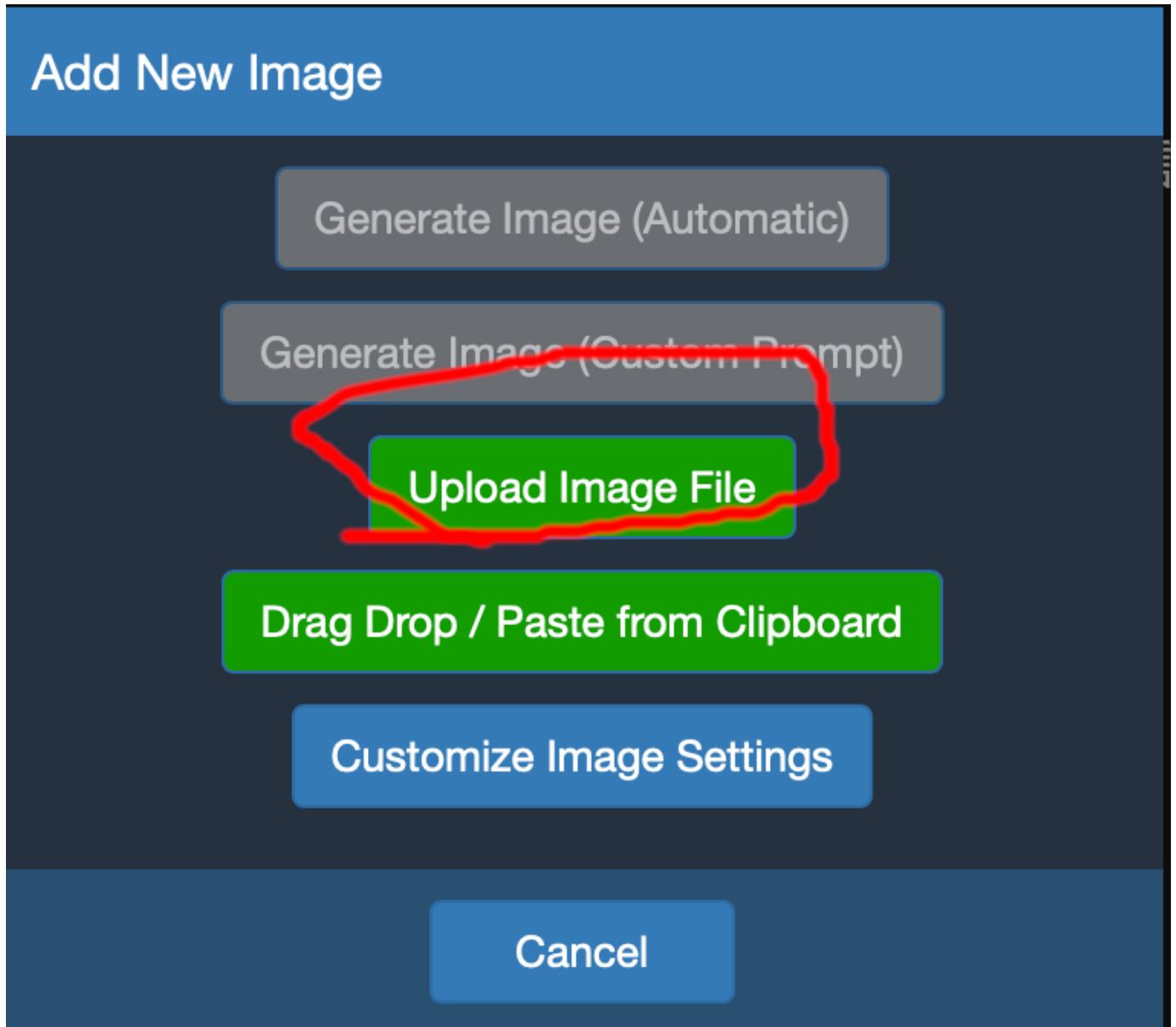
と入れて Enter キーを押せば先程のモデルと同様に日本語で反応します。

Qwen2.5-VL-7B-Instruct のモデルは、画像を与えることでその画像についてAIと対話することができます。視覚言語モデル(Vision Language Model)といいます。

画像を与えて画像に関する対話をしてみましょう。



以下の入力欄の左のクリップを押してください。



Upload Image File を押し、画像生成 AI で生成した猫画像を選んでください。

ComfyUI のインストールフォルダの output フォルダからアップロードします（アップロードといってもローカルの koboldcpp に送っているだけで、どこかの Web サイトにアップロードしているわけではありません）。

KoboldCpp
1.87.4

User

この画像について教えてください。

KoboldAI

この画像は、部屋の中にいる黒猫が映っています。背景にはベッドとテーブルランプがあり、部屋の照明が柔らかく、窓からの自然光が差し込んでいます。床の敷物は白っぽい色合いで、シンプルな装飾です。猫は正面を向いており、何かに注意深く見つめている様子です。部屋全体には穏やかな雰囲気が漂っています。;;

ご質問や気になった点がありましたら、お知らせください。お手伝いできることがありましたら、お気軽にお尋ねください。

Message KoboldAI

Last request served by Custom Endpoint using koboldcpp/Qwen2.5-VL-7B-Instruct-q5_k_m in 47.4 seconds.

この画像について教えてください。

と入れて Enter キーを押してください。しばらくすると反応が返ってきます。

この画像は、部屋の中にいる黒猫が映っています。背景にはベッドとテーブルランプがあり、部屋の照明が柔らかく、窓からの自然光が差し込んでいます。床の敷物は白っぽい色合いで、シンプルな装飾です。猫は正面を向いており、何かに注意深く見つめている様子です。部屋全体には穏やかな雰囲気が漂っています。;;

ご質問や気になった点がありましたら、お知らせください。お手伝いできることがありましたら、お気軽にお尋ねください。

このように画像を与えて対話できます。図表やグラフを与えればその解説もしてくれるはずです。

7.8 llama.cpp

もう一つ [llama.cpp](#) を実行するユーティリティを用意しました。koboldcpp と同様に言語モデルとの対話が可能です。

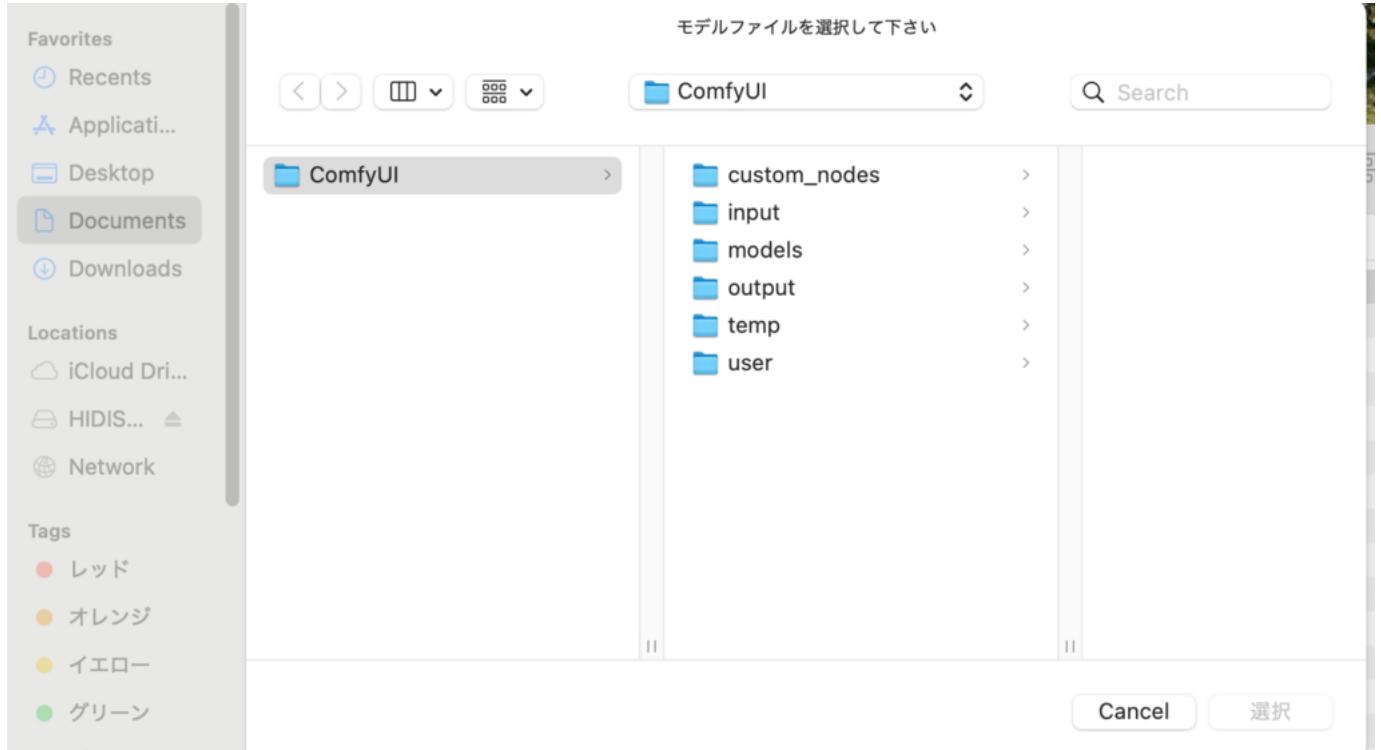
llama.cpp の llama_server を利用すると特定のモデルでは思考プロセスを表示させながらより深い思考をAIに行わせることができます (koboldcpp でも可能な気がしますが設定がわかりませんでした)。

推奨モデルの『cyberagent-DeepSeek-R1-Distill-Qwen-14B-Japanese-Q4_K_M.gguf』が対応しているので、実際に試してみましょう。

7.8.1 llama.cpp のインストール

llama.cpp フォルダの『llama.cpp.command』をダブルクリックしてください。

しばらくするとモデルを選択する画面が出ます。



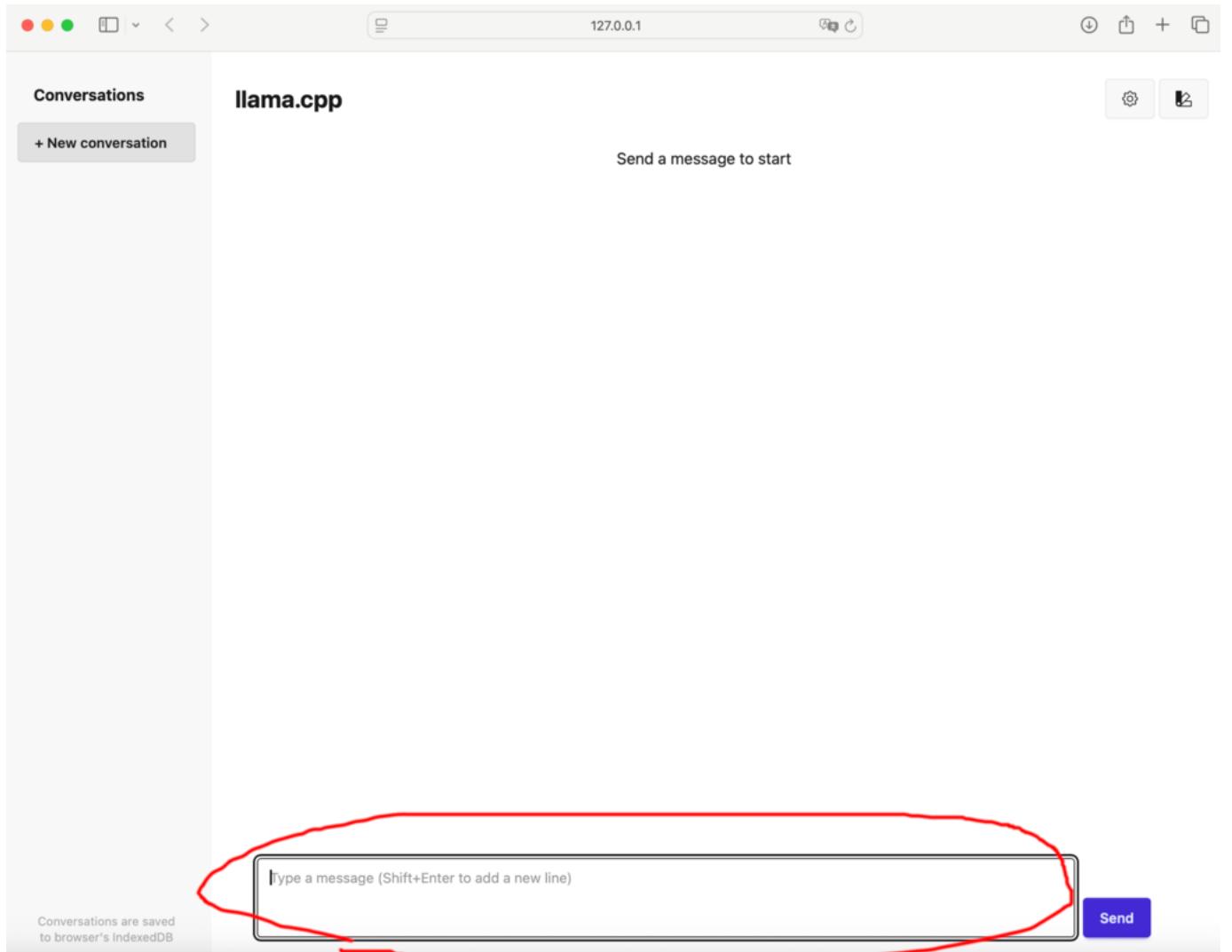
koboldcpp の推奨モデルの『cyberagent-DeepSeek-R1-Distill-Qwen-14B-Japanese-Q4_K_M.gguf』を選んでください。

しばらくすると <http://127.0.0.1:8080> を開くように指示が出ます。

```
| >' + tool['function']['name'] + '\n' + ```json' + '\n' + tool['function']['arguments'] + '\n' + ```` + '<| tool_call_end| >'}%{%- set ns.is_first = true -%}
{%- else %}{'\n' + '<| tool_call_begin| >' + tool['type'] + '<| tool_sep| >' +
tool['function']['name'] + '\n' + ```json' + '\n' + tool['function']['arguments'] + '\n' + ```` + '<| tool_call_end| >'}{{'<| tool_calls_end| ><| end_of_sentence| >'}}%{%- endif %}{%- endfor %}{%- endif %}{%- if message['role'] == 'assistant' and message['content'] is not none %}{%- if ns.is_tool %}{'<| tool_outputs_end| >' + message['content'] + '<| end_of_sentence| >'}%{%- set ns.is_tool = false -%}{%- else %}{% set content = message['content'] %}{% if '</think>' in content %}{% set content = content.split('</think>')[-1] %}{% endif %}{'<| Assistant| >' + content + '<| end_of_sentence| >'}%{%- endif %}{%- endif %}{%- if message['role'] == 'tool' %}{%- set ns.is_tool = true -%}{%- if ns.is_output_first %}{'<| tool_outputs_begin| ><| tool_output_begin| >' + message['content'] + '<| tool_output_end| >'}%{%- set ns.is_output_first = false %}{%- else %}{'\n<| tool_output_begin| >' + message['content'] + '<| tool_output_end| >'}%{%- endif %}{%- endif %}{%- endiffor -%}{%- if ns.is_tool %}{'<| tool_outputs_end| >'}%{%- endif %}{%- if add_generation_prompt and not ns.is_tool %}{'<| Assistant| >'}%{%- endif %}, example_format: 'You are a helpful assistant

<| User| >Hello<| Assistant| >Hi there<| end_of_sentence| ><| User| >How are you
?<| Assistant| >
main: server is listening on http://127.0.0.1:8080 - starting the main loop
srv  update_slots: all slots are idle
```

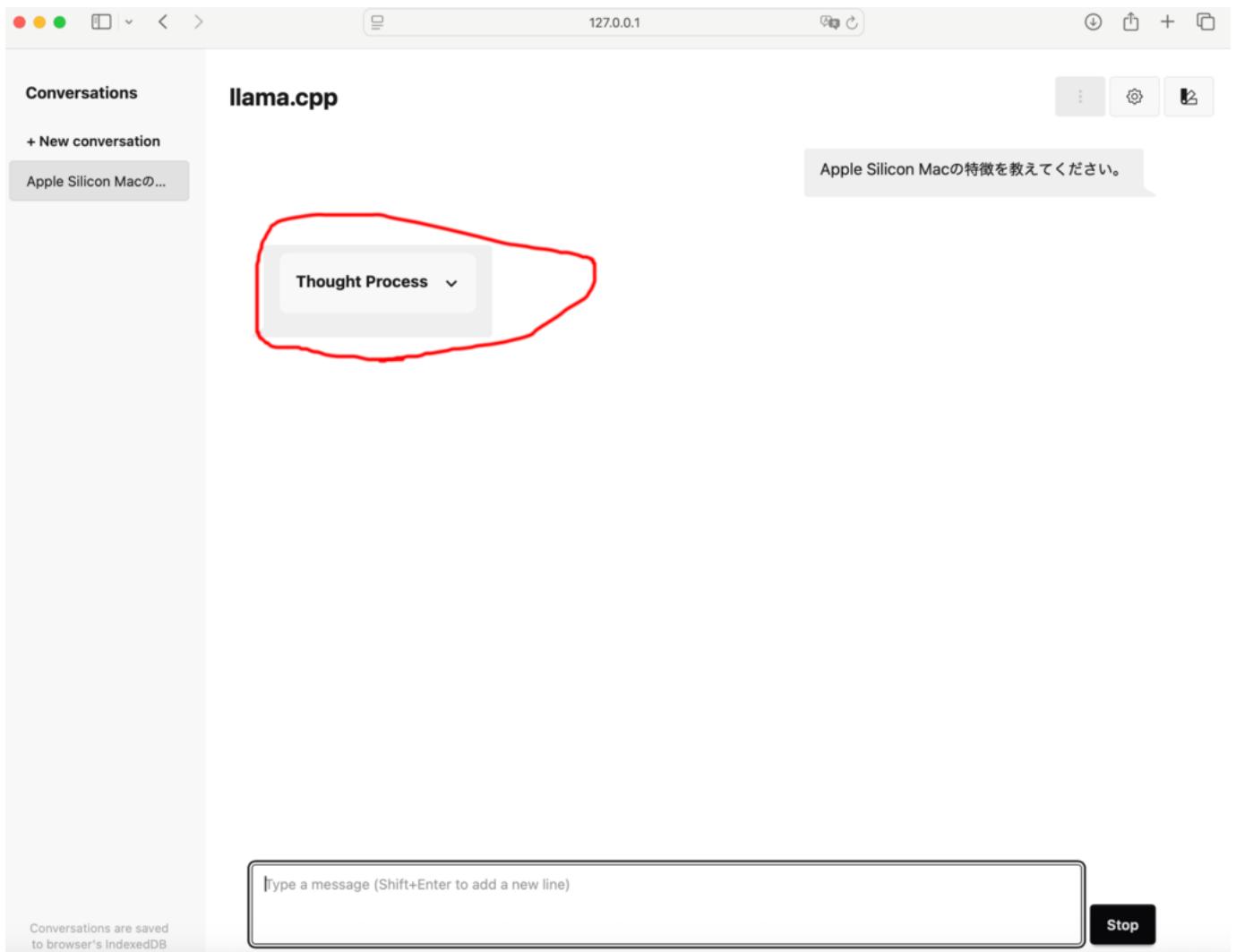
Ctrlキーを押しながらクリックしてリンクを開くを選ぶか、こちらから を開いてください。



下の欄に

Apple Silicon Macの特徴を教えてください。

と入れて Enter キーを押してください。



Thought Processを押すとAIがどのように考えているか 思考(Thought)のプロセス(Process)が順次表示されます。

Thought Process

まず、ユーザーがApple Silicon Macの特徴を知りたいとリクエストしている。Apple SiliconはAppleの自社チップであるM1/M2チップを指すので、その技術的な側面を強調する必要がある。

次に、M1/M2チップの主な特徴を整理する。ARMベースのプロセッサ、統合型のGPU、高速なI/O、低消費電力、マルチチップ設計などが挙げられる。これらのポイントを分かりやすく解説する必要がある。

さらに、Macの用途における進化点も重要。以前はx86アーキテクチャを使用していたが、ARMに移行することでパフォーマンスと効率性が向上した点を説明する必要がある。開発者向けの利点、例えばRosetta 2の活用やネイティブアプリの最適化も触れるべき。

ユーザーが知りたい情報の深さを考えると、技術的な詳細と実用面でのメリットを両方提供するバランスが重要。また、Apple Silicon Macの具体的なモデルや性能比較も必要に応じて補足できると良いが、リクエストが特徴を求めるものなので、基本的な特徴に絞って説明するのが適切だろう。

最後に、全体として分かりやすい構成で、専門用語を適切に解説し、ユーザーが理解しやすいようにまとめ。要點を箇条書きにするなど視覚的に整理することで、読みやすさを向上させられる。

その後回答が表示されます。

The screenshot shows the llama.cpp web application interface. On the left, there's a sidebar with 'Conversations' and '+ New conversation' buttons, and a message 'Apple Silicon Macの...'. The main area has a title 'llama.cpp' and three icons: three dots, a gear, and a refresh. The content area starts with a message: 'Apple Silicon Mac (M1/M2チップ搭載Mac) の特徴を、技術的側面と実用性の両面から解説します。以下に詳細をまとめました：' followed by a section titled '1. ARMベースの統合型プロセッサ' containing bullet points about the M1/M2 chip and integrated design. Below it is '2. 性能進化' with points about multi-task processing and graphics performance. The final section is '3. 効率性と持続性' with a point about low power consumption. At the bottom, there's a text input field with placeholder 'Type a message (Shift+Enter to add a new line)' and a 'Stop' button.

koboldcpp での対話より遅いですが、AI が思考しながら反応を返すので、より正確な反応が返ってきます。また、AI がどのように考えてその回答を行ったか把握できるので、AI が間違えたときにどこで間違えたかも把握しやすくなります。

ARM命令セットとx86命令セットの違いを説明してください。

と入れて Enter キーを押してください。

ARM命令セットとx86命令セットの違いを説明してください。

Thought Process

ユーザーがARM命令セットとx86命令セットの違いを説明してほしいという質問です。まず、この質問の背景を考えます。ユーザーはおそらくApple Silicon Macに搭載されているM1やM2チップについて興味を持ち、そのARMベースのチップが従来のIntelチップ（x86）とどう違うのかを理解したいのでしょう。ARMとx86の命令セットの違いを明確に説明することで、Appleの転換戦略やパフォーマンス向上の理由がより明確になるかもしれません。

まず、両者の基本的な違いを整理します。ARMとx86はアーキテクチャが異なり、命令セットの構造、レジスターの種類、命令の長さなどが異なる点が重要です。ARMはリネッジ制御を採用し、効率的なコード生成が可能ですが、x86は複雑な命令セットを抱えています。また、ARMは32ビットと64ビットを統合している点、x86は64ビットアーキテクチャが別途存在することもポイントです。

次に、実用的な違いとして、電力効率やクロック速度、スレッド管理、SIMD命令の扱いなどを比較します。ARMは低消費電力に優れ、モバイルデバイスに向いていますが、x86は高パフォーマンスを重視しています。また、ARMは64ビットネイティブで動作するため、将来性が高い反面、x86は互換性のために複雑な仕組みが必要です。

さらに、開発者への影響についても触れるべきでしょう。ARMネイティブアプリとx86アプリの開発方法の違い、Rosetta 2のような互換性レイヤーの役割などです。ユーザーが開発者かどうかは分かりませんが、技術的な背景を理解することでAppleの移行戦略をより深く理解できるでしょう。

Thought Process

ARM命令セットとx86命令セットは、コンピュータアーキテクチャの基礎となる重要な違いを持ちます。以下に、技術的・実用的な観点から詳細に比較解説します。

1. 基本アーキテクチャの違い

ARM (RISCアーキテクチャ)

- **RISC (Reduced Instruction Set Computing) **に基づく。
 - 命令セットが「簡素化」され、単純な命令を多く実装。
 - 32/64ビットネイティブ**で動作 (ARM64)。
 - レジスタ制御：64個の32ビットレジスタを備える。
 - リネッジ制御 (Thumb制御)：32ビット命令を16ビットのデータで表現し、メモリ効率向上。
 - 例：ARM Cortexコア、Apple M1/M2チップ。

x86 (CISCアーキテクチャ)

- **CISC (Complex Instruction Set Computing) **に基づく。
 - 命令セットが「複雑化」され、幅広い機能を单一命令で実現。
 - 16/32/64ビットアーキテクチャ**を別途実装 (例：x86-64)。
 - レジスタ制御：8個の32/64ビットレジスタ。
 - 例：Intel Core iシリーズ、旧世代のMac (2020年前)。

2. パフォーマンス特性の違い

ARMの特徴

ARMの特徴

- 電力効率が高い：
 - リネッジ制御により命令のパッケージング効率が向上。
 - M1 Macで20時間の駆動を実現。
- スループット重視：
 - 低クロックで高スループットを追求。
 - 例：ARMコアのシングルスレッド性能はx86に劣るが、並列処理で逆転。
- SIMD（シングルインスツラクション・マルチデータ）命令：
 - ARMネイティブで高速なベクトル演算（NEON命令）。

x86の特徴

- 高クロック：
 - 複雑な命令を処理するため、高クロックで動作。
 - 例：x86コアはARMコアの約2倍のクロックで動作する。
- 複雑な命令セット：
 - データ転送命令や複雑なマスキング演算が豊富。
 - 例：AVX-512のような高度なSIMD命令。

3. 開発者視点の違い

ARM開発の利点

- 半導体製造の効率（TSMCの5nmプロセス）でARMが優位。
 - パフォーマンス/消費電力の比率（IPC/W）がx86を凌駕。
 - **x86の限界：**
 - 10年以上の進化で命令セットが飽和し、パフォーマンス向上の余地が限られる。
-

まとめ表

項目	ARM (RISC)	x86 (CISC)
レジスタ数	64個	8個
ネイティブアーキテクチャ	32/64ビット統合	16/32/64ビット別途実装
電力効率	高	低
開発容易性	見やすい命令セット	複雑な命令セット
代表プロセッサ	M1/M2チップ (Apple)	Intel Core iシリーズ
指令数	少ない	多い

重要なポイント

1. **パフォーマンス vs 消費電力：**
 - ARMは「低消費電力と広範な用途」、x86は「高パフォーマンスと互換性」が特徴。
2. **Apple Siliconの戦略：**
 - ARMアーキテクチャを採用し、M1/M2チップでx86のパフォーマンスを追従。
 - 今後はARMネイティブアプリが主流となり、x86依存度が低下。

これらの違いにより、ARM命令セットは「効率性と汎用性」、x86命令セットは「性能と互換性」を両立するアーキテクチャと言えます。AppleがARM採用を加速させている背景には、このトレンドを活かした「次世代コンピューティングの方向性」が反映されています。

おかしいところもありますが、AIに思考プロセスを示させて深く考えさせることで、AIをより適切に利用できると思います。

7.8.2 終了方法

Safariを閉じて llama.cpp のターミナルを閉じてください。

```
| >' + tool['function']['name'] + '\n' + ````json' + '\n' + tool['function']['ar  
guments'] + '\n' + ````' + '<| tool_call_end| >'}%{- set ns.is_first = true -%}  
{%- else %}{{'\n' + '<| tool_call_begin| >' + tool['type'] + '<| tool_sep| >' +  
tool['function']['name'] + '\n' + ````json' + '\n' + tool['function']['arguments'  
' + '\n' + ````' + '<| tool_call_end| >'}}{{'<| tool_calls_end| ><| end_of_sentence| >'}}%{- endif %}{%- endfor %}{%- endif %}{%- if message['role'] == 'assistant' and message['content'] is not none %}{%- if ns.is_tool %}{{'<| tool_outputs_end| >' + message['content'] + '<| end_of_sentence| >'}}%{- set ns.is_tool = false -%}{%- else %}{% set content = message['content'] %}{% if '</think>' in content %}{% set content = content.split('</think>')[-1] %}{% endif %}{{'<| Assistant| >' + content + '<| end_of_sentence| >'}}%{- endif %}{%- endif %}{%- if message['role'] == 'tool' %}{%- set ns.is_tool = true -%}{%- if ns.is_output_first %}{{'<| tool_outputs_begin| ><| tool_output_begin| >' + message['content'] + '<| tool_output_end| >'}}%{- set ns.is_output_first = false %}{%- else %}{{'\n<| tool_output_begin| >' + message['content'] + '<| tool_output_end| >'}}%{- endif %}{%- endif %}{%- endfor -%}{% if ns.is_tool %}{{'<| tool_outputs_end| >'}}%{- endif %}{% if add_generation_prompt and not ns.is_tool %}{{'<| Assistant| >'}}%{- endif %}, example_format: 'You are a helpful assistant  
  
<| User| >Hello<| Assistant| >Hi there<| end_of_sentence| ><| User| >How are you  
?<| Assistant| >  
main: server is listening on http://127.0.0.1:8080 - starting the main loop  
srv update_slots: all slots are idle
```

8. あとがき

Apple Silicon Mac での画像生成 AI、音楽生成 AI、動画生成 AI、音声生成 AI、言語 AIについて、インストールや操作方法など具体的に解説しました。

M4 Mac Mini のような安価なパソコンでもさまざまな AI 活用が可能です。

特に言語AIとの対話することでさまざまな分析や考察が可能です。

壁打ちという言葉があります。例えば、

- ・AIを活用した壁打ちトレーニング：ChatGPTで思考力を高める
- ・「壁打ちしたいけど相手がいない」を解決！生成AIに壁打ちする活用術を紹介
- ・ChatGPTのおすすめ活用法10選 | 壁打ち～創作活動まで

などで説明されているように、言語 AI との対話からインスピレーションを得て 業務や創作などにつなげることが可能です。

画像生成 AI、音楽生成 AI、動画生成 AI、音声生成 AI、言語 AI、これらを業務や学業や創作活動などにご活用ください。

8.0.1 参考文献

ComfyUI

- ・ComfyUI Wiki
- ・M4 Mac mini でStable Diffusionによる画像生成AIを利用する方法
- ・M4 Mac miniでのComfyUIとHunyuan Videoによる動画生成AI
- ・Easyシリーズ
- ・日本語からイラスト生成プロンプトに翻訳する LLM を作った
- ・音楽生成AI Stable Audioのプロンプトに使えそうな単語まとめ

Style Bert VITS2

- ・(Style-)Bert-VITS2 JP-Extra (日本語特化版)について

llama.cpp

- ・llama.cpp の動かし方と量子化手法
- ・Llama.cpp の HTTP サーバー機能を試す

macOS 関連

- ・Macでバッチ (command) ファイルを作る方法
- ・【Mac】Apple Scriptのdisplay dialogの練習がてらじょんけんアプリを作ってみる
- ・鳶嶋工房 AppleScript
- ・初めての Apple Script
- ・macOS ゲートキーパーを無効にする方法
- ・AppleがMacセキュリティ強化、サイバー攻撃者のGatekeeper回避を防御
- ・Automator:DLしてきたファイルを実行しようとするとゴミ箱にいれろと言われるので、拡張属性削除を右クリックに仕込む
- ・macOS で動的ライブラリを動的リンクするとき
- ・Curl Follow Redirect(s)

8.0.2 更新履歷

• 2025/4/18 公開