💻 Applied Generative AI

Applied Generative AI

Local Image Generation with ComfyUI

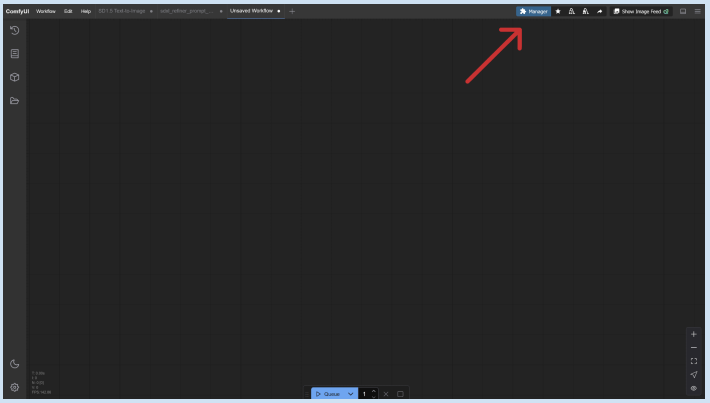
Please bring a laptop (preferably with a modern NVIDIA GPU). If this is unavailable, CPU-only inference is also possible but very slow. (An alternative to running inference locally is GPU rental services that offer ComfyUI deploys.)

Set-up

Introduction to Diffusion Image Generation	• How AI Image Generators Work (Stable Diffusion / Dall-E)

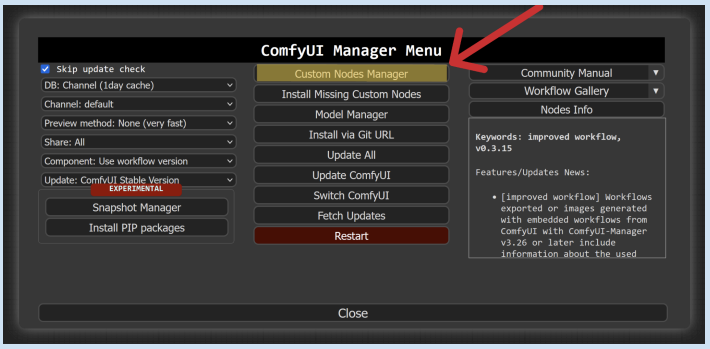
ComfyUI Installation	**Base Installation**
	
	GitHub - ComfyUI
	
	• **Download** and **install ComfyUI** (https://github.com/comfyanonymous/ComfyUI)
	  ◦ Portable (Windows)
	  ◦ Application (Windows/MacOS)
	  ◦ Manual install (Windows/Linux)
	
	Note: Portable installation is recommended as it contains the latest commits. Do not try to install on a network drive, such as OneDrive - it will not work.
	
	ComfyUI Manager Installation
	
	GitHub - ComfyUI-Manager
	
	• **Install** Git: Git - Downloads
	• **Open a Terminal** in the directory **ComfyUI\custom_nodes** (right-click in the directory and select **Terminal**)

	• Run the following command: **git clone https://github.com/ltdrdata/ComfyUI-Manager comfyui-manager** • **Restart** ComfyUI (if running) ***Note:*** *You can find alternative installation procedures here: GitHub - ComfyUI-Manager* **Starting ComfyUI** • ComfyUI Portable can be started with **run_nvidia.bat** (NVIDIA GPU) or **run_cpu.bat** (no NVIDIA GPU) in the base folder where you unpacked the installation. ***Note:*** *When the server starts, it will open in a web browser. The default address is: http://127.0.0.1:8188/*
Model Installation	***Note:*** *If you don't have time or space to download all models below, please prioritize Stable Diffusion 1.5.* **Stable Diffusion 1.5** • **Download** the following files and put them in the following sub-directories in your installation folder. Rename files and create the directory if necessary: • v1-5-pruned-emaonly.fp16.safetensors **(ComfyUI\models\checkpoints)** • sd-v1-5-inpainting.ckpt **(ComfyUI\models\checkpoints)** • vae-ft-mse-840000-ema-pruned.safetensors **(ComfyUI\models\VAE)** **Stable Diffusion XL** • **Download** the following files and put them in the following sub-directories in your installation folder. Rename files and create the directory if necessary: • sd_xl_base_1.0.safetensors **(ComfyUI\models\checkpoints)** • sd_xl_refiner_1.0.safetensors **(ComfyUI\models\checkpoints)** • sdxl_vae.safetensors **(ComfyUI\models\vae)** **Controlnet** • **Download** the following files and put them in the following sub-directories in your installation folder. Rename files and create the directory if necessary:

	• Install **control_v11f1p_sd15_depth_fp16.safetensors** and **control_v11p_sd15_scribble_fp16.safetensors**: • ControlNet-v1-1_fp16_safetensors **(ComfyUI\models\controlnet)** • controlnet-union-sdxl-1.0 **(ComfyUI\models\controlnet)** **LTX Video** • **Download** the following files and put them in the following sub-directories in your installation folder. Rename files and create the directory if necessary: • ltx-video-2b-v0.9.1.safetensors **(ComfyUI\models\checkpoints)** • t5xxl_fp16.safetensors **(ComfyUI\models\text_encoders)**
ComfyUI Manager	**Updating ComfyUI** • Select **Update ComfyUI** in **ComfyUI Manager** • **Restart ComfyUI**  ***Note:*** *Alternatively, you can run batch files (i.e.* ***update_comfyui.bat****) in the* ***update*** *directory.*

Installing custom nodes

Custom nodes can be installed manually (same method as for **ComfyUI Manager** above), or using **ComfyUI Manager:**



Note: *If you import a workflow with missing custom nodes (marked in* ***red****), you can select* ***Install Missing Custom Nodes*** *in* ***ComfyUI Manager*** *to try to install them automatically.*

Image Generation

Building Workflows

Adding nodes

- **Add a node** by **right-clicking** in the workspace and select **Add Node** *or* **double-click** and **Search** for nodes to add.
- You can also add nodes by **dragging a node port from an existing node** into the workspace.
  Note: *Only compatible nodes will show up for selection.*

Connecting nodes

- **Connect nodes** by dragging connections between existing nodes.

Note: *Connection types are color-coded.*

- **Break connections** by clicking the dot in the middle of a connection line and selecting **Delete**.
- **Add Reroute nodes** to tidy up node connections.

Importing workflows

- **Import workflows** by dragging a **JSON** file or **PNG** file containing an embedded workflow into ComfyUI's workspace.

  Note: *Workflows are embedded into generated images by default!*

  Note: *User workflows are saved in* ***ComfyUI\user\default\workflows*** *by default.*

Grouping nodes

- You can group nodes by **right-clicking** and selecting **Add group**. If you change the position of a node touching the group, it will move with the group.

Commenting code

- You can add comments by adding a **Note** node.
- Nodes can be color-coded by right-clicking and selecting **Colors**.

Bypassing and collapsing nodes

- You can bypass a node by right-clicking on it and selecting **Bypass**.

  Note: *If combined with* ***Rerouting*** *nodes, this can be used to alternate between alternative workflow branches.*

- Nodes can be collapsed by right-clicking and selecting **Collapse**.

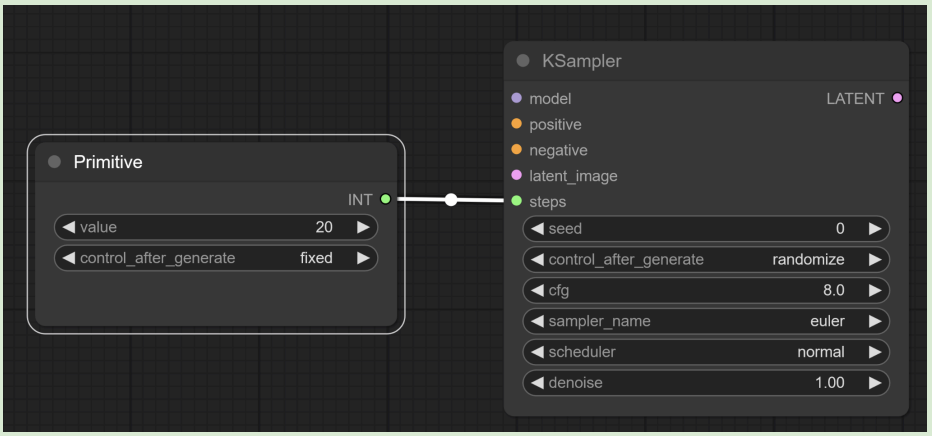
  Note: *Some workflows you download might have collapsed nodes that need to be expanded before you can configure them.*

Converting node widgets to input ports

- Node widgets can be converted to input nodes by **right-clicking** on them and selecting **Convert [widget name] to**

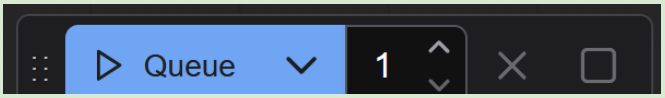
input. The node can then be connected to other nodes.

Note: *By dragging from a node, you can create a Primitive node to define a value to be reused in several nodes. This is useful if you want to control a value in multiple nodes from a single node. Here, the "steps" widget has been converted to an input node and connected to an integer primitive (which could control several nodes):*



Queuing and controlling inference

- **Start** and **control** inference with this panel:

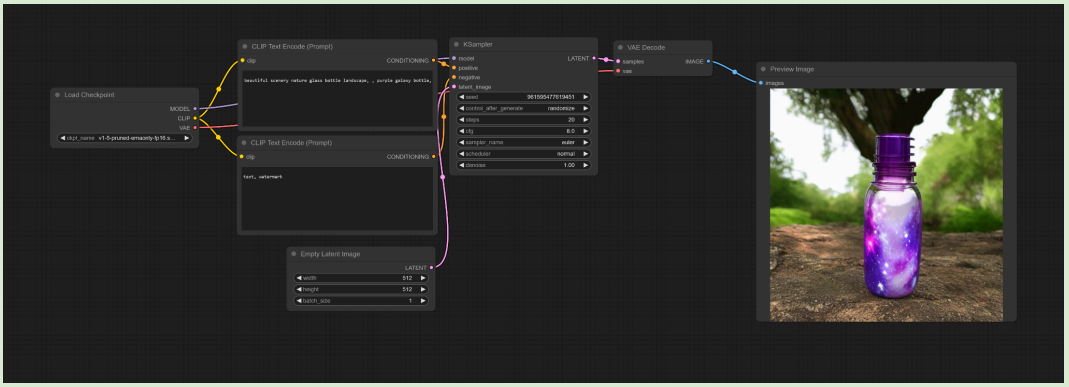


Note: *During interference, output can be seen in the log and terminal window.*

Shortcuts

- Keyboard and Mouse Shortcuts - ComfyUI

Text-to-Image Stable Diffusion 1.5



Simple text-to-image generation with Stable Diffusion 1.5. The model is loaded, CLIP is used to encode text prompts (positive and negative), an empty (random) latent image is defined, a sampler denoises the image, and the denoised latent image is decoded and previewed (or saved).

Nodes

- **Load Checkpoint**
- **CLIP Text Encode**
- **Empty Latent Image**
- **KSampler**
- **VAE Decode**
- **Preview Image / Save Image**

Node Settings Guide

CLIP Text Encode

A CLIP model encodes text into an embedding that guides image diffusion.

- **Positive** and **negative** conditioning.

Empty Latent Image

Creates an empty latent image (random noise) to denoise.

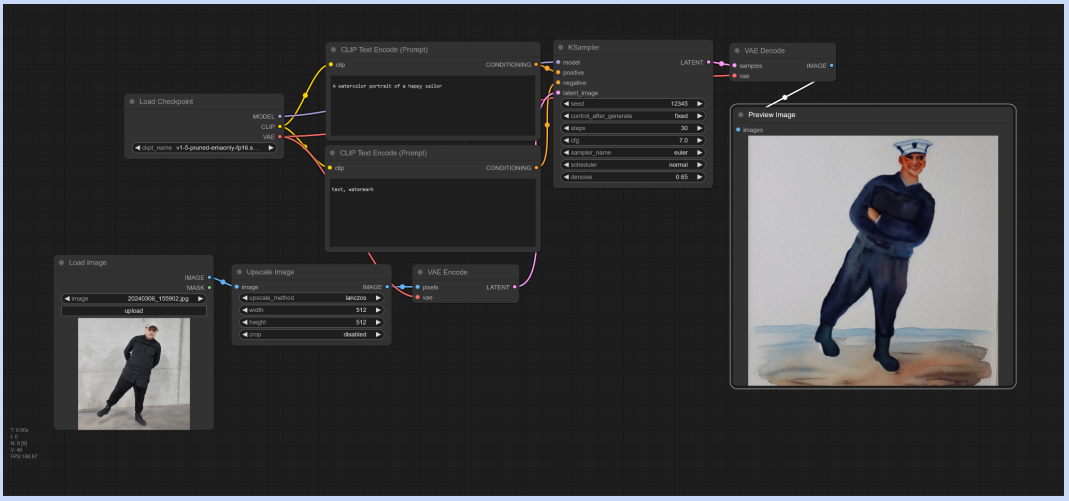
width, height	Pixel width and height of an empty latent image to denoise. **Typical settings:** **SD1.5: 512x512** **SDXL: 1024x1024** **FLUX.dev: 1024x1024, 1920x1080**
batch_size	Size of batch.

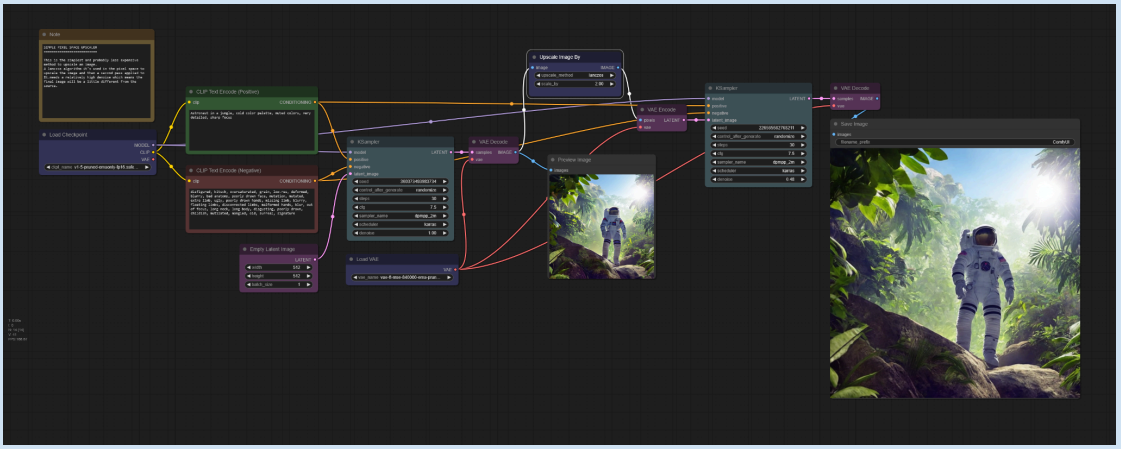
KSampler

Denoises latent images through diffusion.

model	Checkpoint used for inference.
positive, negative	Positive and negative conditioning. Guides the model towards desired features in the output, enhancing specific attributes as defined by the user. **Example positive prompt:** ● Astronaut in a jungle, cold color palette, muted colors, very detailed, sharp focus **Example negative prompt:** ● *disfigured, kitsch, oversaturated, grain, low-res, deformed, blurry, bad anatomy, poorly drawn face, mutation, mutated, extra limb, ugly, poorly drawn hands, missing limb, blurry, floating limbs, disconnected limbs, malformed hands, blur, out of focus, long neck, long body, disgusting, poorly drawn, childish, mutilated, mangled, old, surreal, signature*

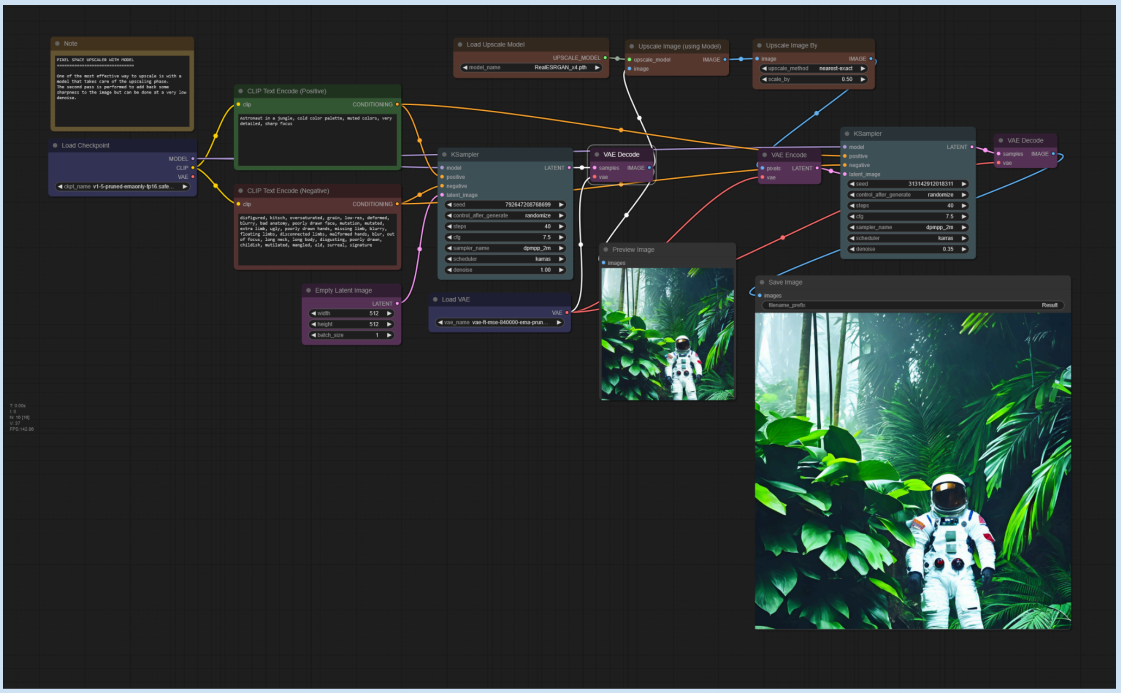
seed, control_after_generate	Random seed. Set to a specific seed (i.e., set **control_after_generate = Fixed**) to compare results when tweaking settings.
steps	Number of denoising steps. Higher values can give higher results with diminishing returns. **Typical setting: 20-40**
cfg	CFG (Classifier-Free Guidance) Scale. Controls how strongly the model adheres to the conditioning prompts. Higher values enforce stricter adherence to the prompts, but excessively high values may negatively impact image quality. **Typical setting: 3.5-12**
sampler	Specifies the algorithm used to iteratively refine the latent image during the denoising process. Each sampler employs a unique method to predict and remove noise, guiding the image toward the desired outcome. Understanding Stable Diffusion Samplers **Typical settings:** • **DDIM + Karras** • **Euler + Karras** • **dpmpp_2m_sde + Normal**
scheduler	The scheduler defines the sequence and progression of noise levels (often referred to as sigma values) throughout the denoising steps. It controls the trajectory of noise removal and impacts how an image evolves from a noisy latent representation into a coherent output. Understanding Stable Diffusion Samplers **Typical settings:** • **DDIM + Karras** • **Euler + Karras** • **dpmpp_2m_sde + Normal**

	denoise	The **denoise** parameter determines how much noise is added to the latent image before the diffusion process begins. Determines how much of an original image is preserved in Image-to-Image workflows. **Typical settings:** • **1.0:** Full noise is added to the latent image (typical setting for image generation from random noise). • **0.3-0.7:** Partial image alteration. Normal range for Image-to-Image workflows. • **0.1-0.3:** Subtle image adjustments. Refines details. • **0.0:** No change. (For upscaling or other tasks where the image needs to be preserved.)
Image-to-Image Stable Diffusion 1.5	

	The workflow has been modified to use an image as a starting point instead of random noise (i.e., the Empty Latent Image is replaced by an image loader, an upscaler, and a VAE Encoder that uses the model's VAE to encode the image to latent space). **Nodes** • **Load Checkpoint** • **CLIP Text Encode** • **Load Image** • **Upscale Image** • **VAE Encode** • **KSampler** • **VAE Decode** • **Preview Image / Save Image** **Batch Generation** • You can batch multiple images by adding a **Repeat Latent Batch** node after **VAE Encode**.
Upscaling	

Iterative upscaling

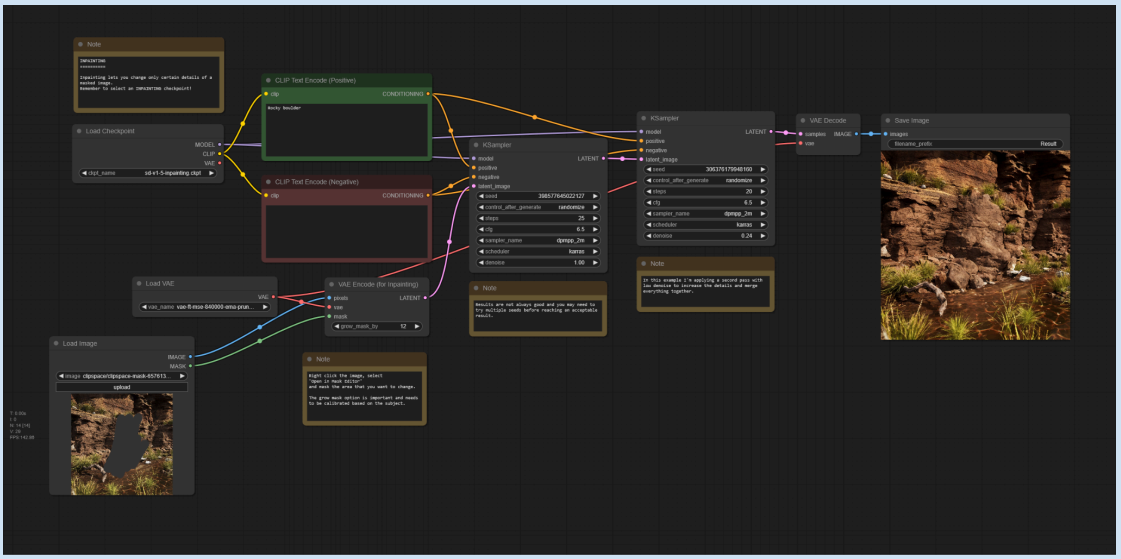
The image is generated at 512x512 px, upscaled 2x to 1024x1024 px, and then sent to another KSampler for detailing.

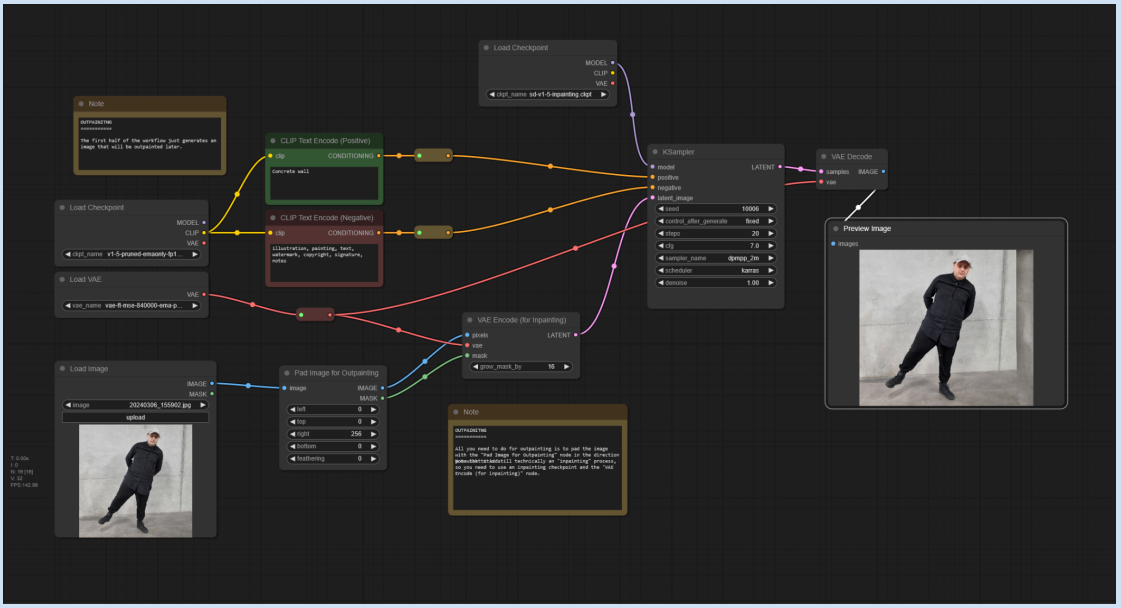


RealESRGAN Upscaling

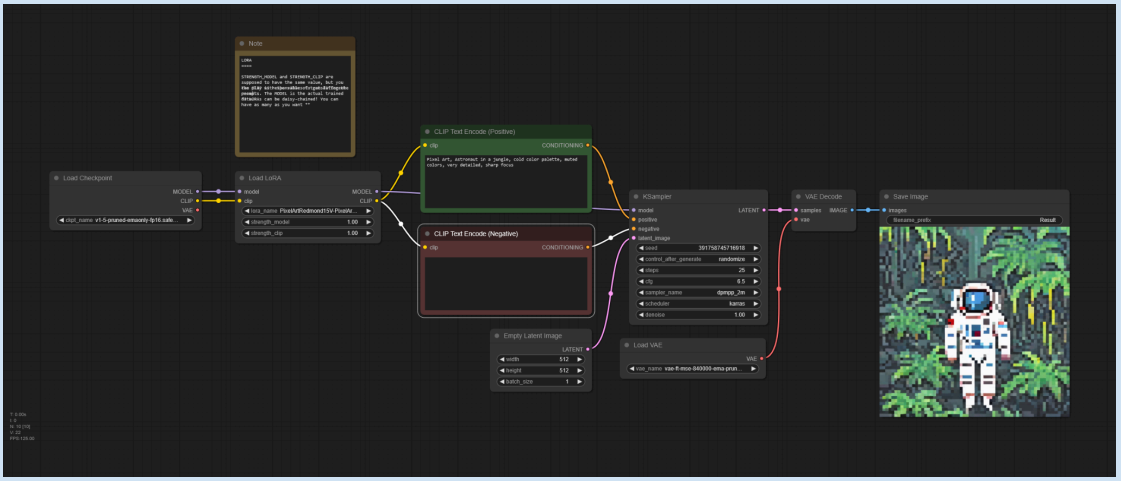
The image is generated at 512x512 px, upscaled with an external model (Real-ESRGAN), and then sent to another KSampler for detailing.

Real-ESRGAN

	 Input (Bicubic) Real-ESRGAN Output Input (Bicubic) Real-ESRGAN Output
Inpainting	

	Inpainting Example *Right-click in **Load Image** node, select **Mask Editor**, draw mask and save the mask. Adjust **Grow Mask** option in **VAE Encode** depending on image.*
Outpainting	 **Outpainting Example** *By expanding the image and running inference with an inpainting model an image can be expanded.*

LoRA (Low-Rank Adaptor)

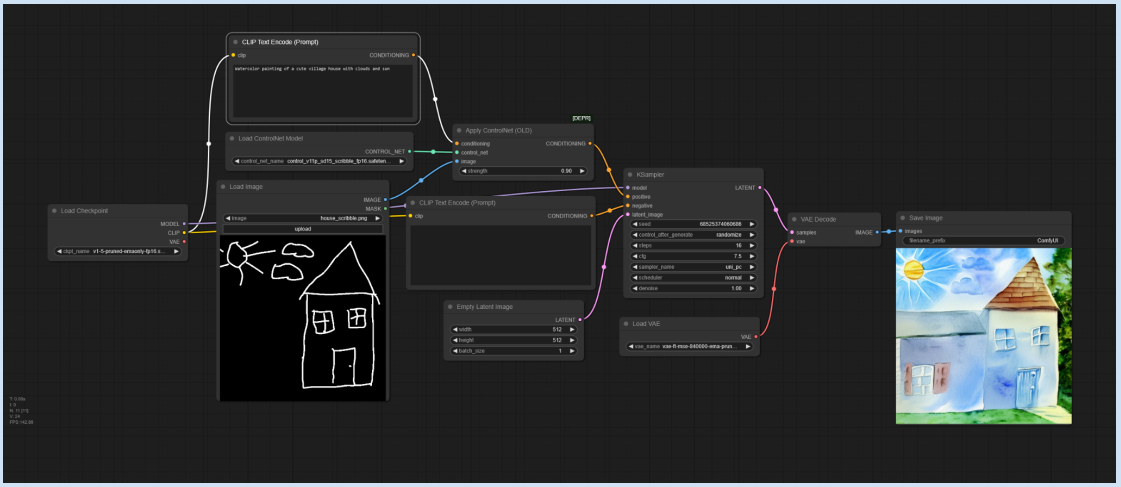


LoRA Example

LoRA (Low-Rank Adaptation) is a method for fine-tuning large pre-trained models by introducing small, trainable parameter matrices while keeping the original model's weights unchanged. Instead of updating all parameters, LoRA injects low-rank matrices into specific layers, allowing for efficient adaptation with significantly reduced computational and memory requirements.

Example LoRA used, add "Pixel Art" to positive prompt:
PixelArtRedmond15V-PixelArt-PIXARFK.safetensors **(ComfyUI\models\loras)**

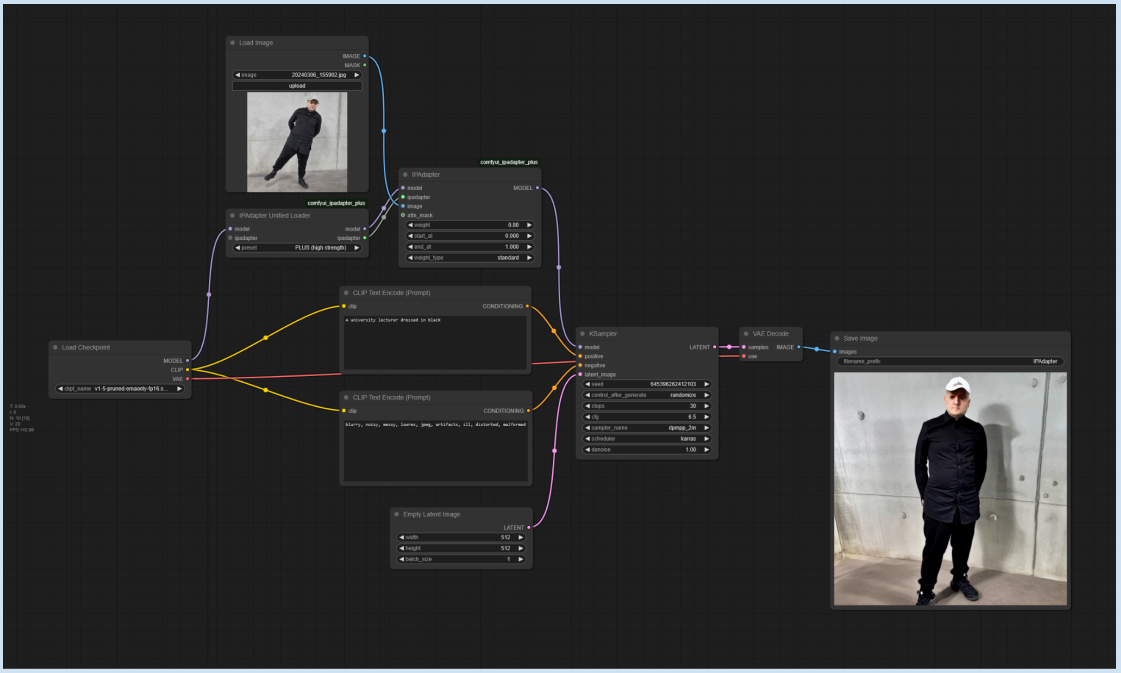
ControlNet



ControlNet is an extension of diffusion models that enables precise control over image generation by conditioning the model on structured inputs (e.g., edge maps, depth maps, poses). It works by injecting additional guidance into the diffusion process using a trained auxiliary network, preserving both structure and creative variation.

- ControlNet
- ControlNet-v1-1_fp16_safetensors **(ComfyUI\models\controlnet)**
- ControlNet and T2I-Adapter Examples

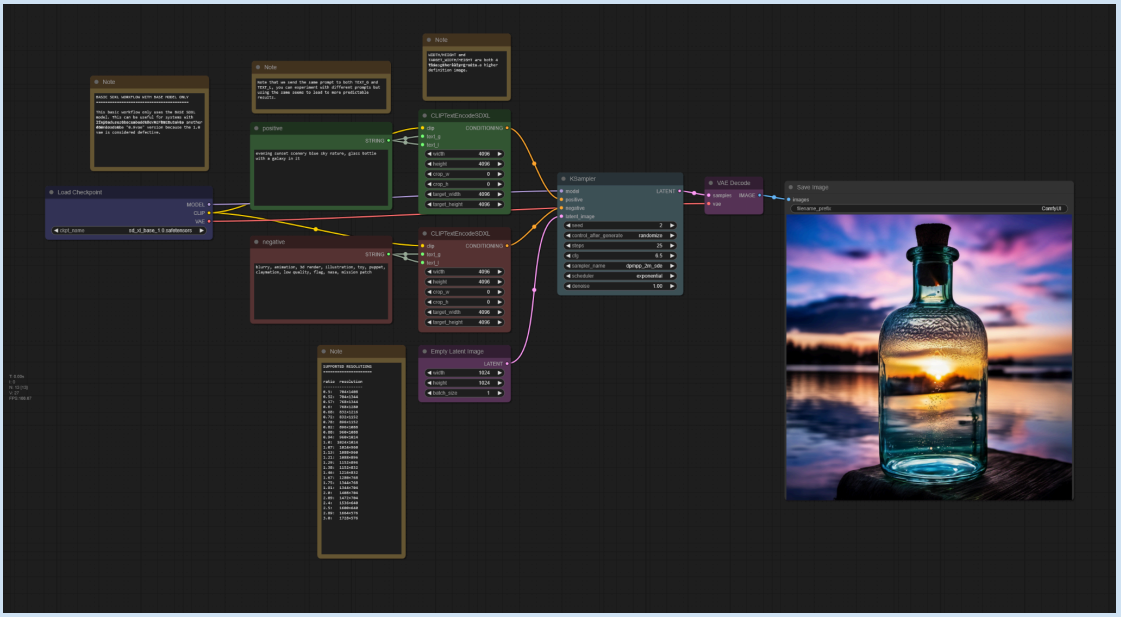
IP-Adapter



IP-Adapter enhances diffusion models by extracting visual features from an input image using a pre-trained vision model like CLIP and injecting them into the model's latent space via cross-attention. This allows the diffusion process to blend image and text guidance, enabling better control over structure, style, and content while maintaining coherence with the reference image.

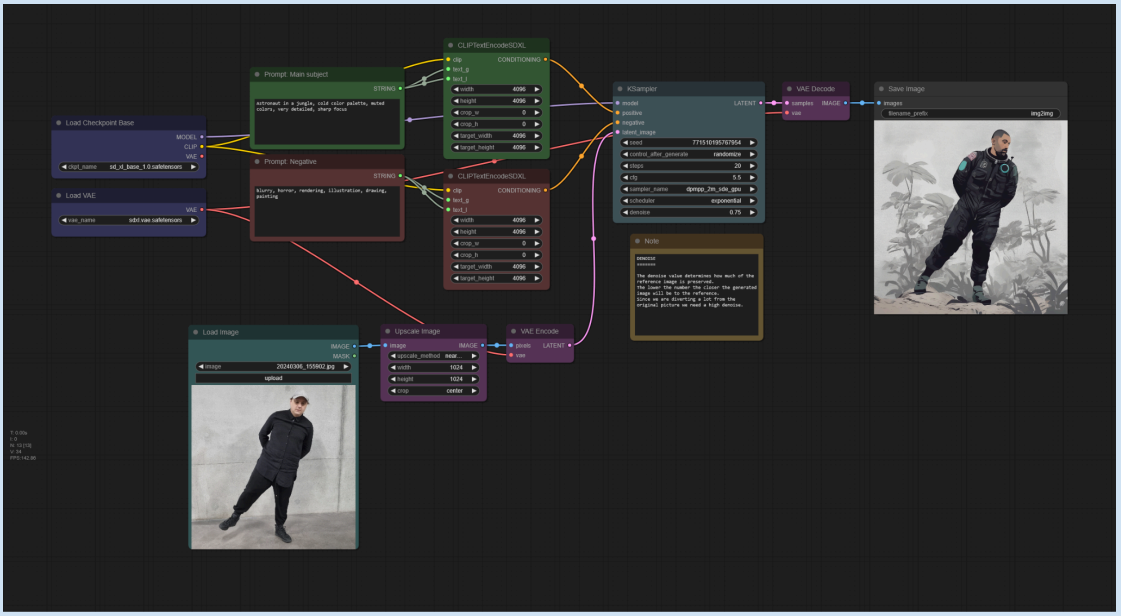
- IP-Adapter - Install **Image Encoders** to (**ComfyUI\models\clip_vision)**, **IP-Adapter** models to (**ComfyUI\models\ipadapter)**
- ComfyUI_IPAdapter_plus - Install via ComfyUI Manager
- IP-Adapters - Installation and model overview
- IP-Adapter Workflow Examples

SDXL



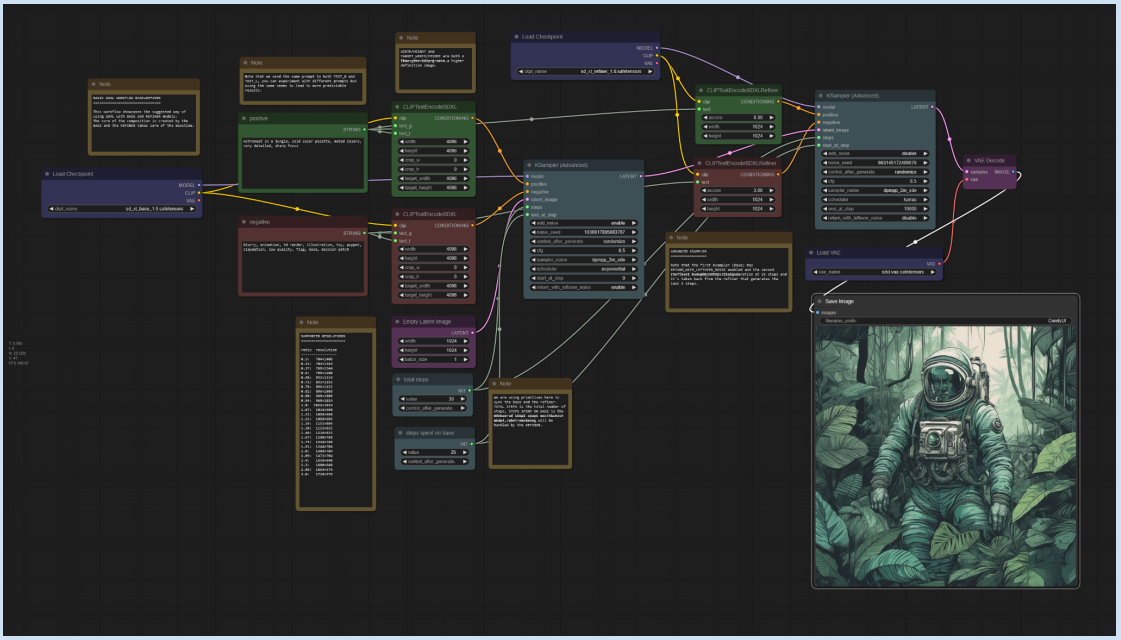
Stable Diffusion XL Text-to-Image

Basic Text-to-Image. The model offers higher resolution and quality compared to Stable Diffusion 1.5



Stable Diffusion XL Image-to-Image

Image-to-Image, note that some nodes are different from Stable Diffusion 1.5.



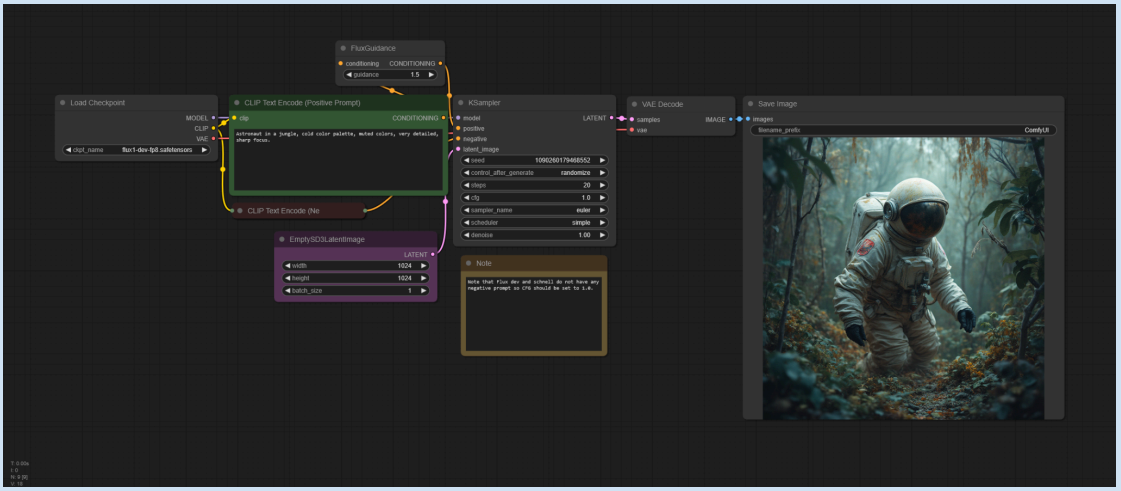
Stable Diffusion XL Text-to-Image with Refiner model

An additional pass with the SDXL refiner model has been added.

Stable Diffusion XL Examples

- SDXL Examples
- SDXL Turbo Examples

FLUX.dev



FLUX.dev Text-to-Image

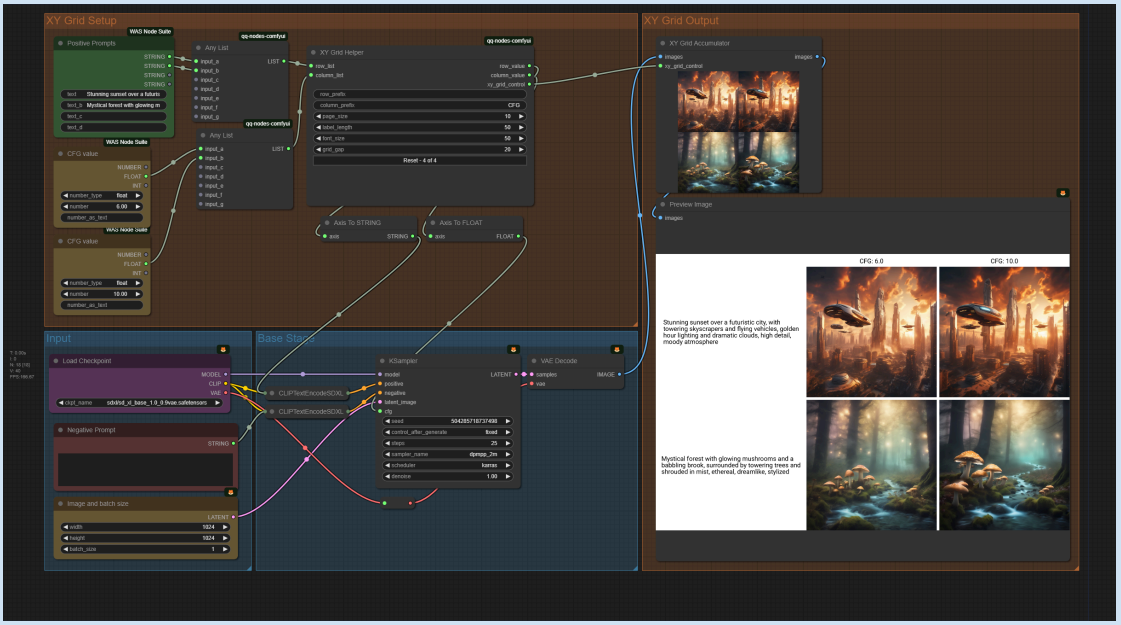
Note: FLUX.dev uses distilled guidance instead of traditional Classifier-Free Guidance (CFG) with negative prompts. This means the model is trained with guidance built-in, so no negative prompt is needed during inference.

Typical settings: Width/Height 1024/1024, Distilled CFG 1.0-3.5, steps 20, sampler/scheduler Euler/Simple

- FLUX.1-dev **(ComfyUI\models\checkpoints)**
- Flux Examples
- Home - FLUX 1.1 Pro - Black Forest Labs
- FLUX Tools - Black Forest Labs

Additional resources

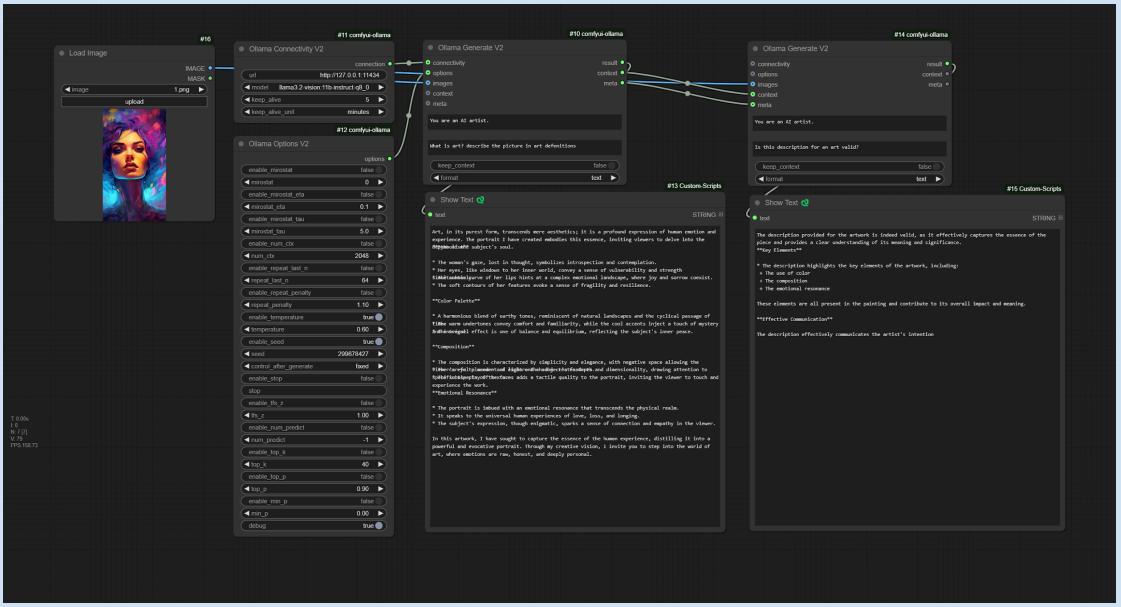
XY Grid/Plot



Plot grids to compare settings.

[qq-nodes-comfyui]

Ollama Integration



Interacting with Ollama

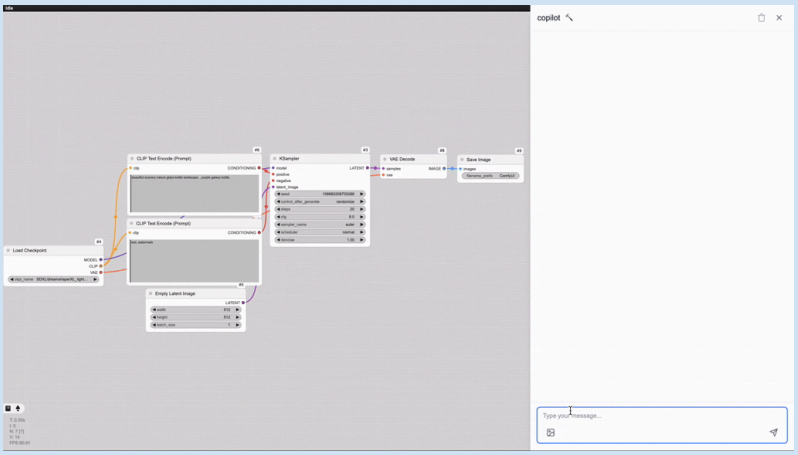
- Query input images
- Query LLM
- LLM chaining

[comfyui-ollama]

ComfyUI Deployment

ComfyUI to Python

- [ComfyUI-to-Python-Extension]

	HuggingFace Spaces • Run ComfyUI workflows for free with Gradio on Hugging Face Spaces **Gradio Deploy Example** • Building a Python API for Comfy UI with Gradio **API Export Deploy Examples** • API Scripts
ComfyUI Copilot	 • ComfyUI-Copilot
Video Tutorials	• Learn ComfyUI Playlist

Example Workflow Collections	• **Workflow > Browse Templates** • ComfyUI Workflow Templates • ComfyUI Examples • IP-Adapter Examples • Lightricks LTX-Video Model • ComfyUI-LTXTricks: A set of ComfyUI nodes providing additional control for the LTX Video model

References

Black Forest Labs. (2024). *FLUX.1* [Computer software]. Black Forest Labs. https://github.com/black-forest-labs/flux

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). *LoRA: Low-Rank Adaptation of Large Language Models* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2106.09685

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., & Rombach, R. (2023). *SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2307.01952

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2021). *High-Resolution Image Synthesis with Latent Diffusion Models* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2112.10752

stability.ai. (2022). *Stable Diffusion* [Computer software]. stability.ai. https://stability.ai/blog/stable-diffusion-public-release

Ye, H., Zhang, J., Liu, S., Han, X., & Yang, W. (2023). *IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models*. https://doi.org/10.48550/ARXIV.2308.06721

Wang, X., Xie, L., Dong, C., & Shan, Y. (2021). *Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2107.10833

Zhang, L., & Agrawala, M. (2023). *Adding Conditional Control to Text-to-Image Diffusion Models*. https://doi.org/10.48550/ARXIV.2302.05543