Open-Source GenAI on OpenStack

https://github.com/mmartial/OpenInfra25-GenAI

Julian Pistorius
Research Software Engineer
Indiana University

Mike Lowe
Lead Systems Programmer
Indiana University

Martial Michel, Ph.D.
Chief Technologist, Vice President AI & Data Sciences
Infotrend Inc.

INDIANA UNIVERSITY

Infotrend

Julian Pistorius

- Co-founder and maintainer of Exosphere -- a researcher-friendly interface to OpenStack that bridges the gap between complex cloud tech and scientists
- Works with Indiana University's OpenStack / Jetstream2 research cloud ecosystem —- making cloud compute more accessible and reliable for academia

Mike Lowe

- Serves as the Lead Systems Programmer for the Jetstream2 cloud infrastructure project at Indiana University
- Co-author of the "Jetstream2: Accelerating cloud computing via Jetstream" -- how OpenStack is used under the hood in Jetstream2

Martial Michel

mmartial 🔗 gkr.one/blg

- Building bridges between "what's possible" and "what we actually ship": 25+ years in distributed systems; from MPI data serialization to modern (and containerized) AI/ML platforms.
- Co-author of NIST SP 500-332 + co-chair of IEEE 2302-2021 on Cloud Federation, plus co-chair of OpenStack's Scientific SIG.

- Jetstream2 provides on-demand **OpenStack**-based interactive cloud computing for researchers and educators.
- Unlike traditional HPC clusters, it offers a virtualized cloud environment with VMs, storage, software stacks and a user-friendly **Exosphere** interface.
- Distributed across five sites — Indiana University (primary), Arizona State University, Cornell University, University of Hawaii, and Texas Advanced Computing Center.

- Jetstream2 official site https://jetstream-cloud.org/
- Get Started page https://jetstream-cloud.org/get-started/
- Jetstream2: Democratizing Cloud Computing for U.S. Research https://www.socallinuxexpo.org/scale/22x/presentations/sponsor-talk-jetstream2-democratizing-cloud-computing-us-research/

Our OpenStack Instance: `g5.xl`

20 VCPUs
240 GB RAM
300 GB SSD
NVIDIA H100 GPU with 80 GB VRAM

---

https://docs.jetstream-cloud.org/general/instance-flavors/#full-gpu

Infotrend's CoreAI

https://github.com/Infotrend-Inc/CoreAI

Build Docker images for ML/CV projects

- CUDA or CPU builds
- TensorFlow
- PyTorch
- OpenCV
- Jupyter Lab
- Ubuntu based

Run as a non-root `coreai` user.

Same `FROM`, multiple applications

Infotrend's CoreAI -- Demo Projects

https://github.com/Infotrend-Inc/CoreAI-DemoProjects

Domain	Project Name
CV	CLIP (Contrastive Language-Image Pre-training) Model Implementation
CV	Fashion MNIST Classification
CV	Fast Neural Style Transfer
DS	Home Credit Default Risk Recognition
LLM	AI Agent with Web Search and LiteLLM
LLM	Fine Tuning LLaMa using QLoRA
LLM (CV)	Flux1Schnell Image Generation
LLM (+ CV)	Gemma3 LLM + VLM (Image Understanding)
LLM	RAG Pipeline
ML	Brain Tumor Segmentation
Multimedia	Video Transcription
NLP	NLP with Disaster Tweets

Pre-requisite

https://github.com/mmartial/OpenInfra25-GenAI/tree/main/CoreAI

- Docker
  - Docker Compose
- a Tailscale account
  - containers will be used in a Tailscale network (`tailnet`) and have no exposed ports
- NVIDIA Container Toolkit
  - an NVIDIA GPU with at least 24GB of VRAM
- a PyTorch enhanced container
  - Infotrend's CoreAI

Getting started

https://github.com/mmartial/OpenInfra25-GenAI/blob/main/CoreAI/compose.yaml.example

```yaml
services:
  oi25-coreai-cpo:
    image: infotrend/coreai:25b01-cpo-12.6.3_2.6.0_4.11.0
    command: /run_jupyter.sh
    environment:
      - WANTED_UID=<WANTED_UID> # replace this to your user ID `id -u`
      - WANTED_GID=<WANTED_GID> # replace this to your group ID `id -g`

  oi25-coreai-ollama:
    image: ollama/ollama:latest
    command: serve

  tailscale-oi25-coreai:
    image: tailscale/tailscale:latest
    hostname: tailscale-oi25-coreai
    environment:
      - TS_AUTHKEY=tskey-auth- # TODO: Add your TS_AUTHKEY here
```

No ports exposed: access through `tailscale-oi25-coreai`'s Tailscale IP

Ollama

https://github.com/mmartial/OpenInfra25-GenAI/tree/main/CoreAI/iti/01-ollama

Serve local Large Language Models (LLMs)

- REST https://github.com/ollama/ollama/blob/main/docs/api.md
- Python's OpenAI API https://pypi.org/project/openai/
- Python's Ollama's API https://pypi.org/project/ollama/

- Ollama https://github.com/ollama/ollama

Search Agent

https://github.com/mmartial/OpenInfra25-GenAI/tree/main/CoreAI/iti/03-SearchAgent

tool-using LLM: web search with local LLM

- 🧰 Define tools: web search + fetch/summarize helper
- 🔁 Agent loop: plan → choose tool → execute → observe
- 📥 Retrieve & summarize: gather top hits, extract key snippets
- 🧠 Synthesize: use context to produce a grounded answer

Retrieval Augmented Generation (RAG)

https://github.com/mmartial/OpenInfra25-GenAI/tree/main/CoreAI/iti/02-RAG_Pipeline

Extract key snippets from documents and use them as context for LLM Q&A

- 📄 Ingest: small document set
- ✂️ Chunk & embed: break text into vectors
- 🧭 Index: store in vector database
- 🤖 LLM Q&A: retrieve top chunks as context

- Ragbits https://ragbits.deepsense.ai/
- Docling https://docling-project.github.io/docling/
- Chromadb https://www.trychroma.com/

Gemma 3

https://github.com/mmartial/OpenInfra25-GenAI/tree/main/CoreAI/iti/04-Gemma3

Instruction-tuned Vision Language Model (VLM) + LLM

- 🧩 Initialize tokenizer & model `pipeline`
- 💬 Prompting (text): prompt payload (system/user) to answer a question
- 🖼️ Multimodal: image understanding using interpretive analysis prompt

> *"Describe the image. what is the field of expertise needed, explain the idea behind the meaning of the image?"*

- Gemma-3-4b-it https://huggingface.co/google/gemma-3-4b-it

FLUX.1[Schnell]

https://github.com/mmartial/OpenInfra25-GenAI/tree/main/CoreAI/iti/05-Flux1

12B text-to-image diffusion model, distilled for 1-4 steps fast inference

- 📦 Install & import libraries (`diffusers`, `transformers`)
- 🌀 Load FLUX.1 [schnell] (with `torch.bfloat16` precision)
- 🎨 Run inference: prompts + seeds → generate images

ComfyUI

https://github.com/comfyanonymous/ComfyUI

- 🧩 Modular & Visual — Graph-based interface for building AI image generation workflows (without coding).
- 🔗 Node-Driven Architecture — Everything is built from nodes — each performing a specific task in the pipeline.
- 🧠 Flexible Workflows — Users connect visual nodes (prompts, samplers, models, processors) to create complex pipelines.
- 🧰 Customizable — Supports `custom nodes`, enabling fine-tuned control at every stage.
- 🪄 Pipeline Flow — Prompt → Encode → Model → Noise → Sampler → Decode → Output.

ComfyUI-Nvidia-Docker

https://github.com/mmartial/comfyui-nvidia-docker

- 🐳 ComfyUI in a container with NVIDIA GPU support, bundling CUDA, drivers, and all dependencies.
- 🔒 Clean File Permissions — Supports `comfy` user UID/GID mapping to match host users and avoid permission issues on shared volumes.
- 🧭 Easy Management — Integrates ComfyUI-Manager for smooth updates and node handling.
- 🧩 Flexible Configuration — Enables `user scripts`, custom launch args, and isolated data through separate `run` & `basedir` folders.
- 🔐 Security Controls — Provides configurable ComfyUI `security levels` to match your environment.

Stable Diffusion terminology

Component	Role	Analogy
🧠 Model	Core generator turning text into images	🎨 Painter
📖 CLIP	Encodes text into concepts the model understands	🗣️ Translator
🧩 LoRA	Adds new styles or knowledge to the base model	🧰 Custom brush
🔄 Sampler	Algorithm that refines noise into the image	🔄 Painting technique
🌌 Latent	Compressed internal image representation	📐 Blueprint
🖼️ VAE	Converts latent image to actual pixels	🖨️ Printer

https://en.wikipedia.org/wiki/Stable_Diffusion

Pre-requisite

https://github.com/mmartial/OpenInfra25-GenAI/tree/main/ComfyUI

- Docker
  - Docker Compose
- a Tailscale account
  - containers will be used in a Tailscale network (`tailnet`) and have no exposed ports
- NVIDIA Container Toolkit
  - an NVIDIA GPU with at least 24GB of VRAM
- a ComfyUI environment
  - container: mmartial/comfyui-nvidia-docker

Getting started

https://github.com/mmartial/OpenInfra25-GenAI/blob/main/ComfyUI/compose.yaml.example

```yaml
services:
  oi25-comfyui-nvidia:
    image: mmartial/comfyui-nvidia-docker:latest
    volumes:
      - ./run:/comfy/mnt
      - ./basedir:/basedir
    environment:
      - WANTED_UID=<WANTED_UID> # TODO: replace this to your user ID `id -u`
      - WANTED_GID=<WANTED_GID> # TODO: replace this to your group ID `id -g`
      - BASE_DIRECTORY=/basedir
      - SECURITY_LEVEL=normal

  tailscale-oi25-comfyui:
    image: tailscale/tailscale:latest
    hostname: tailscale-oi25-comfyui
    environment:
      - TS_AUTHKEY=tskey-auth- # TODO: Add your TS_AUTHKEY here
```

No ports exposed: access through `tailscale-oi25-comfyui`'s Tailscale IP

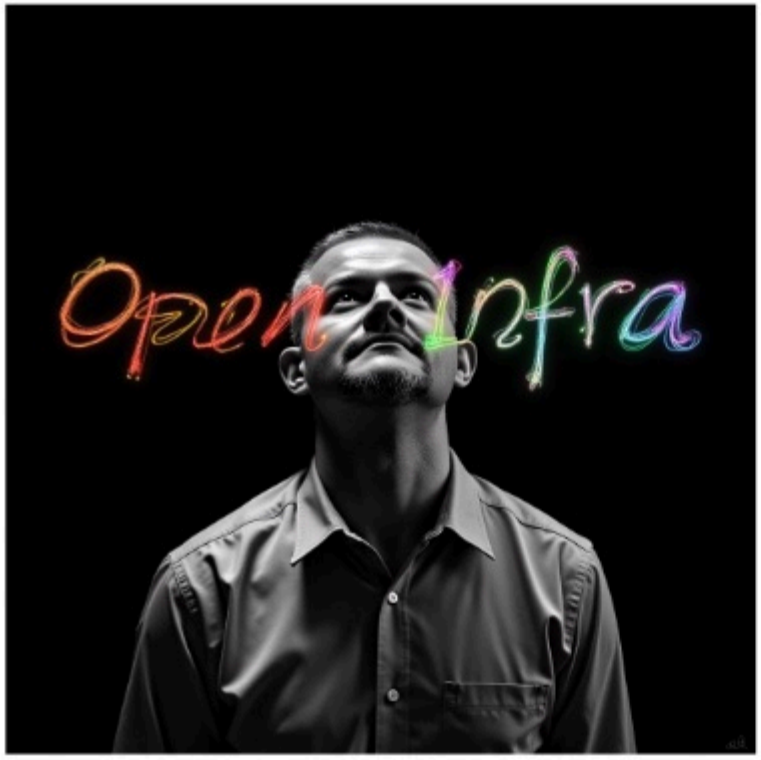
Flux.1 "tok man" Low Rank Adaptation (LoRA)

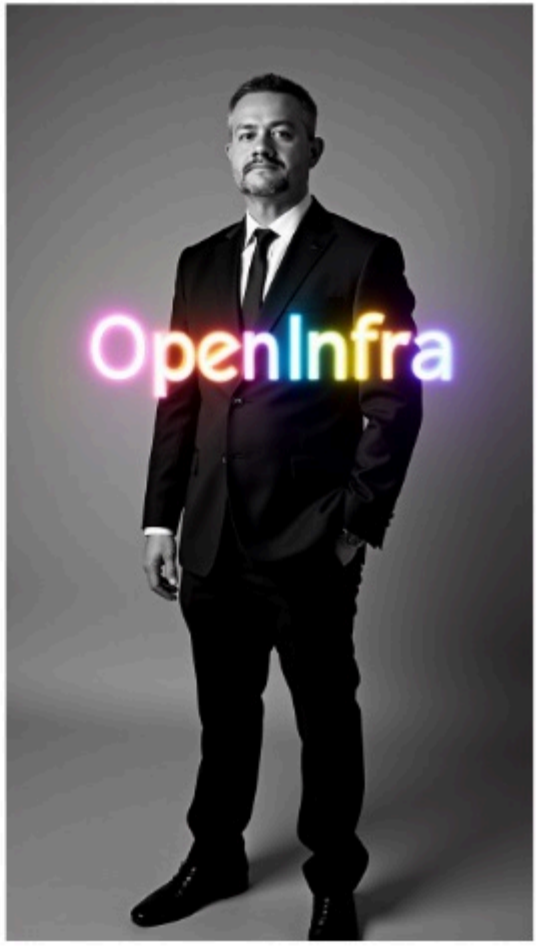
https://www.gkr.one/blg-20240818-flux-lora-training

FLUX.1 LoRA to generate new outputs reflecting the training subject

- 🪄 Training Details — LoRA trained for 4,000 steps using 25 input images.
- ⚡ Hardware Performance — On an NVIDIA RTX 4090, training completed in ≈140 minutes.
- 💾 Model Output — Final LoRA weight file `flux_dev-tok.safetensors` is ~200 MB.

- FLUX.1-dev https://huggingface.co/black-forest-labs/FLUX.1-dev
- FLUX.1 LoRA training https://www.gkr.one/blg-20240818-flux-lora-training
- ai-toolkit https://github.com/ostris/ai-toolkit


OpenInfra


OpenInfra

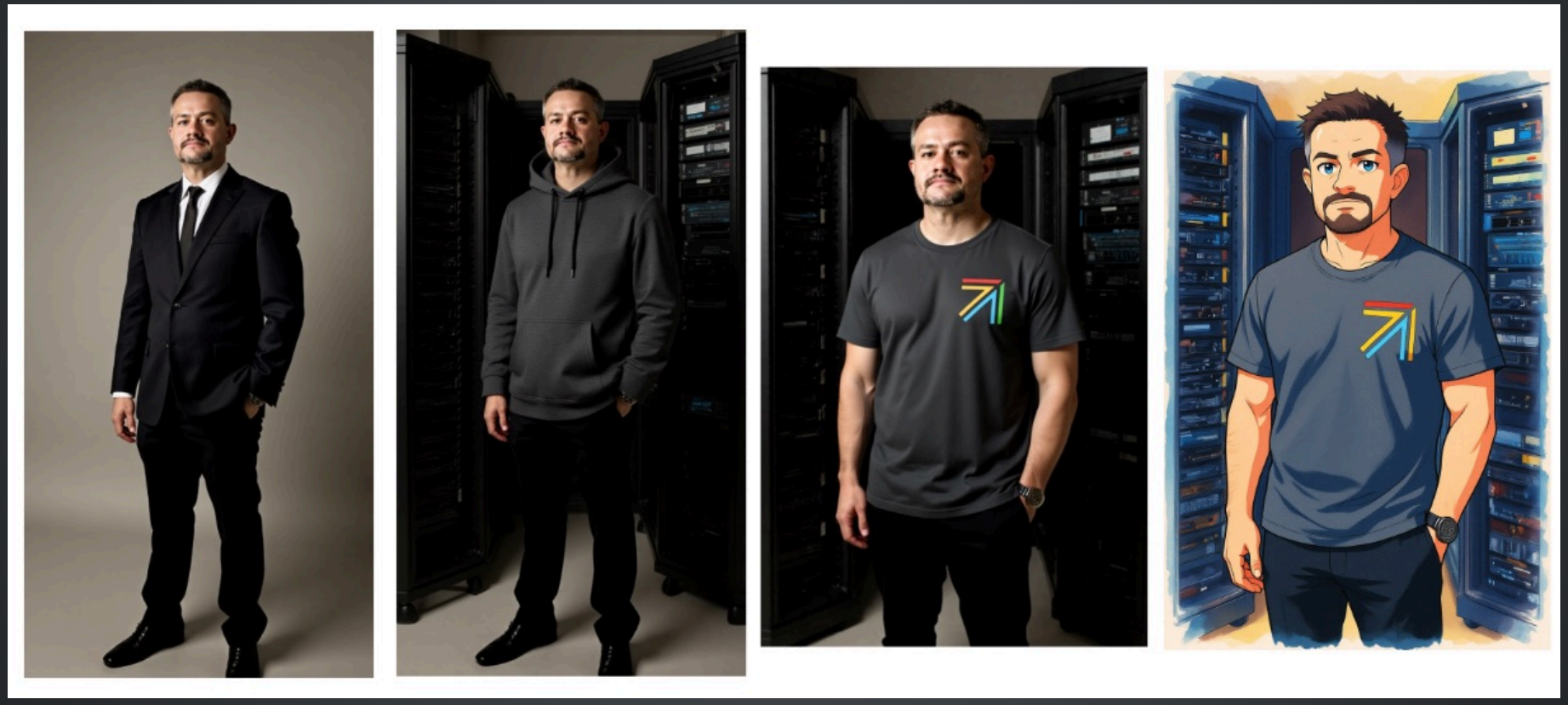
Flux Kontext

https://github.com/mmartial/OpenInfra25-GenAI/tree/main/ComfyUI/02-Flux_Kontext

- 🪄 In-Context Image Editing + Generation — Provide an image + text instruction, and FLUX Kontext interprets the scene and applies the requested edit.
- 🧭 Local & Global Edits — Make targeted changes (e.g. change hair color, remove or add objects) or transform entire scenes, styles, or layouts.
- 🧍 Character/Object Consistency — Successive edits preserve identity, style & composition, minimizing visual drift across multiple editing steps.

- FLUX.1 Kontext https://bfl.ai/models/flux-kontext
- ComfyUI Flux Kontext https://docs.comfy.org/tutorials/flux/flux-1-kontext-dev

1. text removal + re-color black and white source
2. background addition + outfit change
3. outfit change + add the OpenInfra logo on the outfit
4. style change

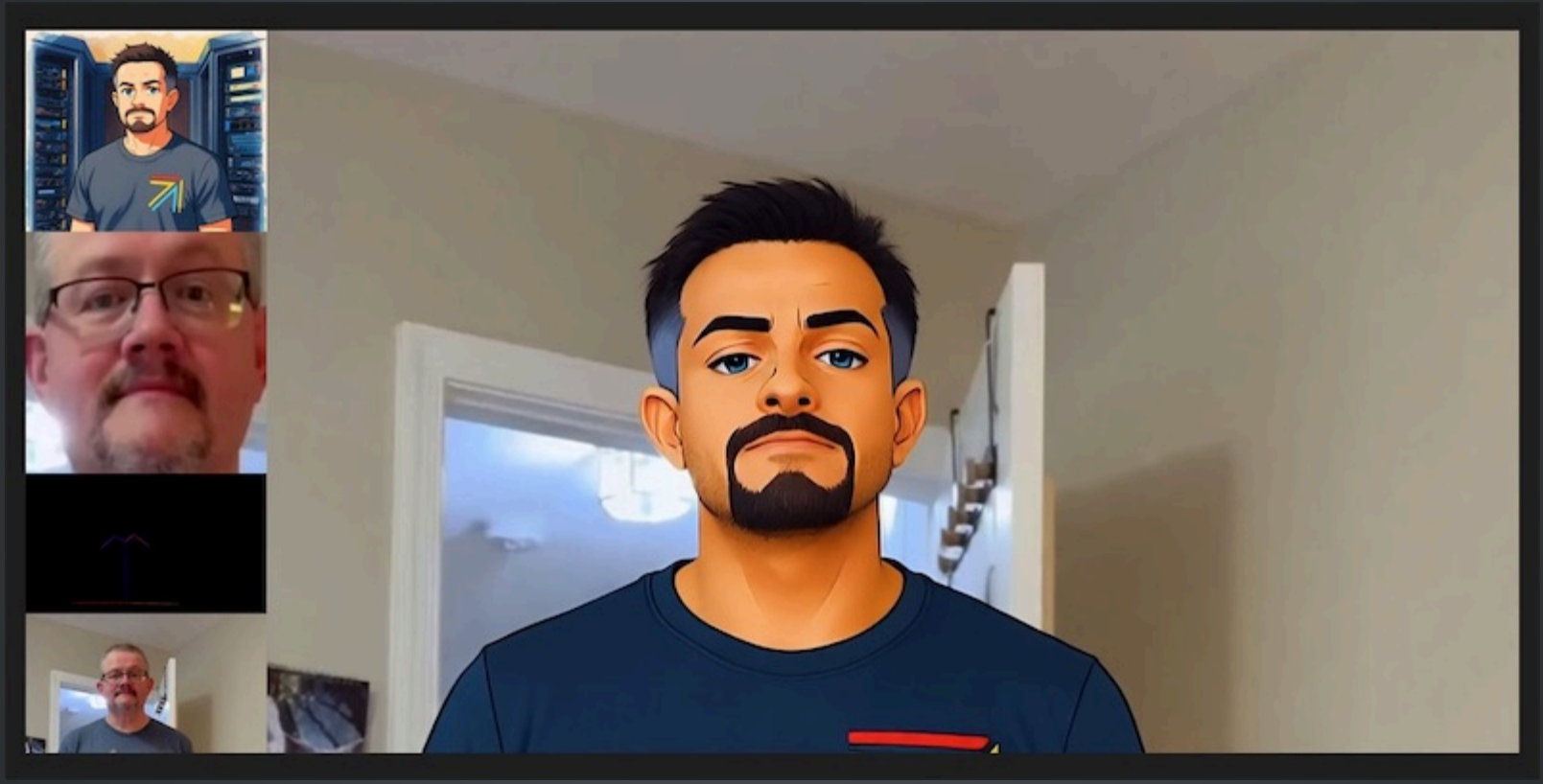


WAN 2.2 Animate

https://github.com/mmartial/OpenInfra25-GenAI/tree/main/ComfyUI/03-WAN2.2_Animate

- 🤖 Core Tools — WAN 2.2 Animate + Kijai's `custom node ComfyUI-WanAnimatePreprocess` (+ embedded workflow).
- 🧍 Preprocessing Pipeline — Detects people in frames, estimates body / hand / face keypoints, extracts aligned face crops, and formats results for WanAnimate.
- 🎬 Result — Transforms a single character image + reference video into a high-fidelity animated or replacement video.
- ⚠️ GPU — Minimum 32 GB VRAM (even with `fp8` weights).

- WAN 2.2 Animate https://huggingface.co/Wan-AI/Wan2.2-Animate-14B
- Kijai's `ComfyUI-WanAnimatePreprocess` https://github.com/kijai/ComfyUI-WanAnimatePreprocess



Thank you

Open-Source GenAI on OpenStack

https://github.com/mmartial/OpenInfra25-GenAI