What is better for ollama and ComfyUI on a windows 11 pc, one rtx3060 or two 1050ti

CLAUDE

For running Ollama and ComfyUI on Windows 11, **one RTX 3060 is significantly better** than two GTX 1050 Ti cards. Here's why:

RTX 3060 advantages:

- 12GB VRAM (the 12GB variant) vs 4GB per 1050 Ti - this is crucial since both Ollama (for LLMs) and ComfyUI (for Stable Diffusion) are extremely VRAM-hungry
- Much faster memory bandwidth (360 GB/s vs 112 GB/s per 1050 Ti)
- Tensor cores for AI acceleration, which the 1050 Ti lacks entirely
- Better CUDA core performance (3584 cores vs 768 per 1050 Ti)
- Single GPU means no multi-GPU complications

Why dual 1050 Ti cards fall short:

- Combined you'd only have 8GB total VRAM, but it doesn't pool - each card has its own 4GB, which is too little for modern LLMs or high-quality image generation
- Neither Ollama nor ComfyUI efficiently utilize multiple weaker GPUs - they generally prefer one strong GPU
- Multi-GPU setups on Windows can be finicky with driver overhead and compatibility issues
- Much older architecture (Pascal vs Ampere) without AI-focused hardware

For context, Ollama needs 4-8GB+ VRAM for smaller models (7B-13B parameters), while larger models need much more. ComfyUI workflows often require 8-12GB+ for decent resolution and quality. The RTX 3060's 12GB gives you comfortable headroom for both applications, while the 1050 Ti cards would constantly struggle with VRAM limitations.

Two gtx1080ti

Now we have a **much closer comparison**. Two GTX 1080 Ti cards vs one RTX 3060 is actually interesting, and the answer depends on your priorities:

Two GTX 1080 Ti advantages:

- 22GB total VRAM (11GB per card) vs 12GB on the RTX 3060
- Much more raw compute power combined (2× 3584 CUDA cores = 7168 total vs 3584)
- Higher memory bandwidth per card (484 GB/s per 1080 Ti vs 360 GB/s on 3060)
- Better for certain parallel workloads

RTX 3060 advantages:

- Tensor cores for AI acceleration (1080 Ti has none)
- Better software support - newer architecture means better optimization in newer applications
- Single GPU simplicity - no multi-GPU headaches
- Lower power consumption (~170W vs 2× 250W)
- All VRAM accessible to one application at once

For your specific use case:

Ollama strongly prefers a single GPU with maximum VRAM. It doesn't efficiently use multiple GPUs. The RTX 3060's 12GB in one card is better than having two separate 11GB pools. The Tensor cores also help with inference speed.

ComfyUI can potentially use multiple GPUs, but it's not seamless - most workflows run on one GPU. The extra VRAM of dual 1080 Ti cards could help with very large/complex workflows, but you'd need to manually configure multi-GPU support.

My recommendation: RTX 3060 for simplicity and better AI-specific performance, unless you're specifically running ComfyUI workflows that absolutely require more than 12GB VRAM. The Tensor cores and single-GPU design will give you a smoother, more reliable experience with both applications.

I have one pc with a rtx3080 another with two 1080ti 11g. Best split of ollama and comfyUI

Great setup! Here's the **optimal split**:

RTX 3080 → ComfyUI

Dual 1080 Ti → Ollama

Reasoning:

RTX 3080 for ComfyUI:

- Tensor cores dramatically accelerate Stable Diffusion inference (often 2-3x faster than 1080 Ti)
- 10GB VRAM is sufficient for most ComfyUI workflows at reasonable resolutions
- Newer architecture with better FP16 performance, which image generation heavily uses
- ComfyUI benefits most from the AI-specific hardware acceleration

Dual 1080 Ti for Ollama:

- Ollama can actually utilize multiple GPUs when loading larger models - it will split model layers across both cards

- 22GB combined VRAM means you can run much larger models (30B-70B parameter models become feasible)
- Tensor cores matter less for LLM inference compared to raw VRAM capacity
- The 1080 Ti's strong compute power is still excellent for text generation

Practical benefits:

- You can run large language models (like Llama 70B quantized) on the dual 1080 Ti setup while simultaneously generating images on the RTX 3080
- Separating the workloads across two PCs means no resource contention
- The RTX 3080 will generate images significantly faster than the 1080 Ti cards would

If you find ComfyUI workflows hitting VRAM limits on the 3080, you could swap them, but for typical use cases this split maximizes the strengths of each system.

I'm more concerned about running large flux models

Ah, **Flux changes everything**. For large Flux models:

Dual 1080 Ti (22GB) → ComfyUI with Flux

RTX 3080 (10GB) → Ollama

Why this is better for Flux:

Flux VRAM requirements are brutal:

- Flux Dev/Schnell in full precision: 24GB+ VRAM
- Flux with fp8 quantization: 12-16GB minimum
- Flux with LoRAs, ControlNets, or complex workflows: easily 16-20GB+
- The RTX 3080's 10GB will severely limit you - you'd be stuck with heavy quantization or low resolutions

Dual 1080 Ti advantages for Flux:

- 22GB total gives you breathing room for Flux Dev in fp8 with multiple LoRAs
- ComfyUI *can* use both GPUs for Flux - while not perfect, it can offload model components across cards
- You can run higher quality settings and larger batch sizes
- Future-proofs you for even larger models

RTX 3080 for Ollama:

- 10GB is plenty for most LLM use cases (7B-34B models run fine)
- Ollama is less VRAM-hungry than Flux

- If you need larger models, Ollama can offload to CPU RAM (slower but functional)

Bottom line: Flux is one of the most VRAM-demanding image models available. The 22GB on the dual 1080 Ti setup is essential for running it without constant VRAM bottlenecks. The RTX 3080's Tensor cores don't help enough to compensate for the VRAM deficit with Flux.

Is flux good for comfyUI, should I consider others?

Flux is **excellent** for ComfyUI, but whether it's the best choice depends on what you're creating. Let me break down the landscape:

Flux Strengths:

- **Best text rendering** of any open model - actually readable text in images
- Superior prompt adherence and understanding of complex instructions
- Photorealistic quality rivals Midjourney/DALL-E 3
- Great at hands, faces, and difficult details
- Active development and LoRA ecosystem growing fast

Flux Weaknesses:

- Massive VRAM requirements (your main constraint)
- Slower generation times than SD models
- Smaller LoRA/checkpoint library compared to SDXL
- Less community resources and workflows (still relatively new)

Strong Alternatives to Consider:

SDXL (Stable Diffusion XL):

- Much lower VRAM (6-8GB comfortable)
- Huge ecosystem of LoRAs, checkpoints, ControlNets
- Faster generation
- Can run multiple models/workflows easily with your 22GB
- Quality is very good, just not quite Flux-level for photorealism
- **Best for:** versatility, speed, anime/artistic styles

Pony Diffusion / Illustrious XL:

- SDXL-based, optimized for anime/illustration
- Massive style flexibility
- Same low VRAM benefits as SDXL
- **Best for:** anime, artistic, stylized work

SD 1.5:

- Even lower VRAM (4-6GB)
- Absolutely massive model/LoRA library
- Very fast
- Quality is lower than SDXL/Flux but good enough for many uses
- **Best for:** experimentation, learning, iteration speed

My Recommendation:

Start with SDXL on your dual 1080 Ti setup. Here's why:

1. Your 22GB means you could run **multiple SDXL workflows simultaneously** - incredible for productivity
2. The ecosystem is mature with tons of resources
3. You can always load Flux for specific projects that need it
4. Faster iteration = better learning and more output

Use Flux when you specifically need:

- Photorealistic images with text
- Maximum prompt adherence
- Professional-quality outputs where speed doesn't matter

With 22GB VRAM, you're not forced to choose - you can have both SDXL and Flux set up, switching based on the project. Many ComfyUI users keep multiple model types installed for different use cases.

What kind of images are you planning to generate? That would help narrow down the best starting point.

I have found the images better out of flux than the SD models. Photorealistic is importantly to me

Then **stick with Flux** - you've already identified what works best for your needs, and photorealism is exactly where Flux excels.

With your dual 1080 Ti setup (22GB VRAM), you're actually in a good position for Flux:

Optimizing Flux on Your Setup:

Use fp8 quantization:

- Flux Dev fp8: ~12-14GB VRAM
- Minimal quality loss compared to full precision
- Leaves headroom for LoRAs and workflow complexity

ComfyUI multi-GPU setup:

- Configure ComfyUI to utilize both 1080 Ti cards
- Model components can be split across GPUs
- Not as efficient as single-GPU but workable with 22GB total

Workflow optimization:

- Use efficient samplers (fewer steps when possible)
- Consider Flux Schnell for faster iterations, Dev for final renders
- Batch processing when you need multiple variations

LoRA strategy:

- Flux LoRAs are growing rapidly - photorealism, lighting, specific styles
- Much smaller VRAM footprint than full model training
- Stack 2-3 LoRAs for specific looks

The Reality:

If Flux is consistently giving you better photorealistic results than SD models, that's what matters. The VRAM requirement is real, but you have enough to work with it effectively. The quality difference you're seeing is significant - Flux genuinely is a generation ahead for photorealism and prompt adherence.

Your setup is viable. Just optimize your ComfyUI workflows for VRAM efficiency and you'll be fine. What resolution are you typically generating at?

Define those terms/usage Loras/fp8. Also, is there anything better than flux for photorealistic

Let me explain those terms and answer your question about alternatives:

LoRAs (Low-Rank Adaptation):

What they are:

- Small add-on files (typically 50-500MB) that modify a base model's behavior
- Think of them like "style filters" or "skill plugins" for your AI model
- Much smaller than full models (which are 5-20GB+)

Usage examples:

- A "cinematic lighting" LoRA makes images look more like movie scenes
- A "realistic skin texture" LoRA improves facial detail
- Subject-specific LoRAs (celebrities, objects, art styles)

- You can stack multiple LoRAs together (e.g., lighting + skin texture + camera style)

Why they matter:

- Let you customize Flux without retraining the entire model
- Use minimal extra VRAM (usually 500MB-1GB each)
- Growing library of community-created LoRAs for different effects

fp8 (8-bit Floating Point):

What it is:

- A compression technique that reduces model size and VRAM usage
- "Quantization" = converting the model's numbers from high precision to lower precision
- fp8 = 8-bit, fp16 = 16-bit, fp32 = 32-bit (full precision)

Practical impact:

- Flux Dev full precision: ~24GB VRAM
- Flux Dev fp8: ~12-14GB VRAM (almost 50% reduction)
- Quality loss is minimal - most people can't tell the difference
- Slightly faster generation too

How to use:

- Download fp8 versions of Flux models (available on Hugging Face, CivitAI)
- ComfyUI automatically handles them once loaded
- Just drop in place of the full precision model

Is There Anything Better Than Flux for Photorealism?

Short answer: Not really in the open-source world.

Current landscape:

Closed/Commercial (better, but not accessible):

- Midjourney v6 - arguably still king for artistic photorealism
- DALL-E 3 - excellent but you can't run locally
- These require subscriptions and have content restrictions

Open-source alternatives:

- **SDXL Realistic checkpoints** (like RealVisXL, JuggernautXL): Good, but Flux is genuinely better

- **SD 3.5**: Newer but mixed reviews, not clearly better than Flux
- **Kolors**: Chinese model, interesting but less documented, questionable for photorealism

Emerging options:

- **Flux variants** (Flux Redux, Flux Pro) - refinements of Flux itself
- Various fine-tunes of Flux for specific photorealistic styles

The reality: Flux Dev is currently the best **locally-runnable** model for photorealism. If you're already getting better results from Flux than SDXL, you're using the right tool. The gap is real.

The only way to get notably better photorealism would be:

1. Wait for Flux updates/improvements (actively developed)
2. Use Midjourney/DALL-E (not local, subscription-based)
3. Fine-tune Flux yourself for your specific needs (advanced)

Since you already have Flux working and prefer its output, you've found the sweet spot. Focus on optimizing your workflow and exploring LoRAs to push quality even further rather than switching models.

What kind of photorealistic images are you generating - portraits, landscapes, products?