SDXL vs SD 1.5 vs Flux: Which Image Model Should You Run Locally?

February 11, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** Flux Dev produces the best images but needs 12GB+ VRAM (quantized). SDXL is the practical middle ground at 8GB with strong LoRA support. SD 1.5 still works on 4GB cards and has the biggest ecosystem, but its 512x512 native resolution and anatomy issues show their age. For most people with a 12GB+ GPU in 2026, start with Flux Dev in GGUF Q5 via ComfyUI. Drop to SDXL if you need a specific LoRA or ControlNet that only exists for that model.

📚 **More on this topic:** Stable Diffusion Locally · Flux Locally · ComfyUI vs A1111 vs Fooocus · Planning Tool

Three image models, three different eras. SD 1.5 launched in 2022 and still runs on potato GPUs. SDXL arrived mid-2023 with 4x the resolution. Flux dropped in 2024 and produces images that look like a different technology entirely.

The problem: they all run locally, they all have ecosystems, and picking the wrong one means downloading gigabytes of models you'll replace in a week. This guide compares them on the numbers that matter and tells you which to install for your GPU and your use case.

The Three Models at a Glance

	SD 1.5	SDXL	Flux Dev
Release	August 2022	July 2023	August 2024
Parameters	~860M	~3.5B (6.6B with refiner)	12B
Native resolution	512x512	1024x1024	1024x1024+
Minimum VRAM	4 GB	8 GB	12 GB (quantized)
Architecture	UNet	UNet (larger)	DiT (transformer)
Text rendering	Poor	Poor	Good (95%+ single-word accuracy)

	SD 1.5	SDXL	Flux Dev
License	CreativeML Open RAIL++	CreativeML Open RAIL++	Non-commercial (Dev) / Apache 2.0 (Schnell)
Status	Deprecated, huge ecosystem	Active, maturing	Active, growing fast

SD 1.5 is the Honda Civic of image gen. Cheap to run, parts everywhere, gets the job done. SDXL is the midrange sedan: better in every measurable way, still affordable. Flux is the sports car: noticeably better output, but you need the hardware to match.

VRAM Requirements

This is usually what decides the question for you.

Minimum VRAM to Generate Images

Model	Precision	VRAM Used	Minimum GPU
SD 1.5	FP16	~4 GB	GTX 1060 6GB, RTX 3050
SD 1.5 + ControlNet	FP16	~6-8 GB	RTX 3060, RTX 4060
SDXL	FP16	~7-8 GB	RTX 3060 8GB, RTX 4060
SDXL + Refiner	FP16	~12-18 GB	RTX 3060 12GB+ (sequential), 16GB+ (simultaneous)
Flux Dev	FP16	~24 GB	RTX 3090, RTX 4090
Flux Dev	FP8	~12-16 GB	RTX 3060 12GB, RTX 4070
Flux Dev	GGUF Q5	~8-10 GB	RTX 3060 8GB (tight)
Flux Dev	GGUF Q4 / NF4	~6-8 GB	RTX 4060, RTX 3060
Flux Dev	Nunchaku INT4	~4-8 GB	RTX 3060 (with CPU offload)
Flux Schnell	FP8	~6-8 GB	RTX 4060, RTX 3060

A few things jump out. SD 1.5 runs on almost anything with a discrete GPU. SDXL needs a modern 8GB card. Flux at full precision is a 24GB-only proposition, but quantized versions bring it within reach of 8-12GB cards with varying quality tradeoffs.

The Nunchaku/SVDQuant INT4 path deserves a mention: it pushes Flux down to 4-8GB VRAM with per-layer CPU offloading and claims 3x faster inference than standard NF4. If you have an 8GB card and want Flux, this is the path to try first.

The SDXL refiner trap: You'll see guides telling you to load both the base and refiner model. That needs 12-18GB. Most people skip the refiner entirely and get good results. Unless you're chasing the last 5% of detail, just run the base.

Generation Speed

Seconds per image on common GPUs. SD 1.5 at 512x512/20 steps, SDXL at 1024x1024/20 steps, Flux Dev at 1024x1024/20 steps.

GPU	SD 1.5	SDXL	Flux Dev (FP16)	Flux Dev (FP8)	Flux Schnell (4 steps)
RTX 4090	~1 sec	~2-4 sec	~18 sec	~11-14 sec	~3-5 sec
RTX 3090	~2-3 sec	~6 sec	~40 sec	~26-30 sec	~6-10 sec
RTX 4070	~2-3 sec	~7 sec	—	~49 sec	~10-15 sec
RTX 4060	~4-5 sec	~16 sec	—	—	~20-30 sec
RTX 3060 12GB	~5-6 sec	~13 sec	~10+ min*	~400 sec*	~20-40 sec

*RTX 3060 running Flux Dev at FP16/FP8 involves heavy CPU offloading and is borderline unusable for iterating. Use GGUF Q4 instead (~2-3 minutes) or Nunchaku INT4 for better speed.

The speed gap is real. SD 1.5 at 512x512 is essentially instant on modern GPUs. SDXL at 1024x1024 is comfortable on anything 8GB+. Flux Dev is where patience enters the equation. On a 3090, 40 seconds per image is fine for final renders but painful for prompt iteration. That's where Flux Schnell comes in: 4 steps, 6-10 seconds on a 3090, good enough for drafting.

Optimization options that help:

- **xformers**: ~5-10% speed gain, mainly saves VRAM
- **TensorRT FP16**: ~1.5-2x faster than native PyTorch
- **TensorRT FP8**: Up to 2.4x faster (tested on Flux Dev, needs RTX 40/50 series)

Image Quality

This is where the generational differences actually show.

SD 1.5: Workable, but dated

Native output is 512x512. You almost always need hires fix or an upscaler to get usable images. Anatomy is unreliable: extra fingers, mangled hands, weird limb counts. Complex prompts frequently get ignored or misinterpreted. Negative prompts are mandatory to avoid the worst artifacts.

The quality ceiling is surprisingly high with the right checkpoint, LoRA stack, and prompt engineering. But getting there takes work. You're fighting the model, not collaborating with it.

SDXL: The solid middle

1024x1024 native resolution means no upscaling just to get something usable. Anatomy improved over SD 1.5 but still inconsistent: correct finger count about 45% of the time in complex hand poses. Prompt adherence is better, negative prompts still help. The quality jump from SD 1.5 to SDXL is immediately obvious.

Community checkpoints like Juggernaut XL and RealVisXL push photorealism further than the base model. If you need reliable output without babysitting every generation, SDXL with a good checkpoint is the practical choice.

Flux: Different league

Correct finger count 85% of the time. Natural hand positioning 90%. Single-word text rendering accuracy 95%+. Multi-word text 85-90%. Complex prompts with spatial relationships ("a red ball on top of a blue box to the left of a green cone") actually work.

Flux uses a different architecture (DiT transformer vs UNet) and a different training approach (flow matching vs diffusion). The result is images that feel like they understood the prompt instead of pattern-matching parts of it. Simple prompts produce excellent results without negative prompt engineering.

The tradeoff: no native negative prompt support. Flux uses flow matching without classifier-free guidance, so the standard negative prompt workflow doesn't apply. Workarounds exist (Dynamic Thresholding, Perpendicular Negative Guidance) but they're 2-3x slower. In practice, most Flux users don't bother with negatives because the baseline output quality is high enough.

LoRA and Checkpoint Ecosystem

LoRAs don't transfer between model families. An SD 1.5 LoRA won't work with SDXL. An SDXL LoRA won't work with Flux. This matters more than most people realize when choosing a model.

	SD 1.5	SDXL	Flux
LoRA availability	Largest. 3+ years of community work. Tens of thousands on CivitAI.	Second largest. Growing since mid-2023. Thousands available.	Smallest but growing fast. Training is more expensive.
Checkpoint/ merge variety	Massive. Hundreds of specialized merges for every style.	Strong. Juggernaut XL, RealVisXL, and many others.	Limited. Base model is already very good.
Anime/stylized	Dominant. This is where SD 1.5 still wins.	Good, catching up.	Growing, but less variety.
LoRA training VRAM	8 GB minimum, 12 GB comfortable	10-12 GB minimum (aggressive optimization), 16-24 GB comfortable	24 GB minimum (QLoRA, rank 4-8), 48 GB comfortable
Training cost (CivitAI)	500 Buzz	500 Buzz	2,000 Buzz (4x more)

If you need a LoRA for a niche anime style, a specific character, or a particular aesthetic, check CivitAI for your model family first. SD 1.5 almost certainly has it. SDXL probably has it. Flux might not yet.

Flux LoRA training is also more expensive across the board. You need 24GB+ VRAM locally (vs 8GB for SD 1.5), and it costs 4x more on CivitAI's on-site trainer. The flip side: Flux LoRAs need fewer images (20-30 is often enough) and fewer training steps (500-1,500 vs 3,000-5,000 for SDXL).

ControlNet Support

ControlNet lets you guide image generation with reference images: edge maps, depth maps, poses, scribbles.

	SD 1.5	SDXL	Flux
Official ControlNet	Yes. 14 model types in v1.1.	No official models. Community-built.	Partial. BFL released Canny and Depth.
Types available	Canny, Depth, OpenPose, Scribble, Segmentation, Tile, LineArt, NormalBAE, HED, MLSD, Shuffle, IP2P, Inpaint, LineArt Anime	Most SD 1.5 types ported by community. Less standardized.	Canny, Depth, HED, Surface Normals, Union (multi-mode). InstantX, XLabs-AI, Jasperai.
Model sizes	Small (136 MB LoRA), Medium (723 MB), Large (1.45 GB)	Varies by author	12B per official model (same as base Flux)
Maturity	Mature, well-documented	Functional, some gaps	Catching up. Fewer options, larger downloads.

SD 1.5 has the most complete ControlNet ecosystem by a wide margin. If your workflow depends on specific ControlNet types (especially niche ones like Segmentation or Shuffle), check whether they exist for your target model before switching.

Flux Dev vs Flux Schnell

Both use the same 12B architecture. The differences matter.

	Flux Dev	Flux Schnell
Steps	20-30	1-4
Speed (RTX 3090)	~40 sec	~6-10 sec
Quality	Higher detail, better micro-textures	Good. Slightly less fine detail.
Text rendering	95%+ single-word	80-85% single-word
License	Non-commercial (commercial license available from BFL)	Apache 2.0 (fully open)
Best for	Final renders, portfolio work	Quick drafts, iteration, commercial projects

Use Schnell for drafting and iteration: test your prompt, get the composition right at 4 steps in seconds, then switch to Dev for the final generation. If you're building something commercial (selling prints, using images in a product), Schnell's Apache 2.0 license avoids the licensing question entirely.

What About SD 3.5?

Stability AI released SD 3.5 in late 2024 as the successor to SDXL. It exists in three sizes:

Variant	Parameters	VRAM	Notes
SD 3.5 Medium	2.5B	~3 GB	Surprisingly light. Runs on 12GB cards easily.
SD 3.5 Large	8B	~18 GB (11 GB with TensorRT FP8)	Mid-range quality.
SD 3.5 Large Turbo	8B (distilled)	~18 GB	Fewer steps, faster.

The honest take: the community largely skipped SD 3.5 in favor of Flux. SD 3.0 launched with quality issues and restrictive licensing, and even though 3.5 improved on both, the momentum had already shifted. Most active development on CivitAI and in ComfyUI workflows targets either SDXL or Flux.

SD 3.5 Medium is the one worth knowing about. At 2.5B parameters and ~3GB inference VRAM, it's the lightest modern image model and runs on GPUs that can't handle SDXL. If you have a 6GB card and SD 1.5 quality isn't cutting it, SD 3.5 Medium is an upgrade path.

One licensing note: Stability AI added an explicit content prohibition to SD 3.5 in July 2025, which caught users off guard. Free for commercial use under $1M revenue, but check the current terms.

What Fits Your GPU

8GB VRAM (RTX 4060, RTX 3060 8GB)

Model	How	Experience
SD 1.5	Native FP16	Fast, full quality. Best experience at this tier.
SDXL	Native FP16 (tight)	Works but close to the limit. Short context. Disable refiner.
Flux Dev	GGUF Q4 or Nunchaku INT4	Possible but slow (2-5 min/image). Noticeable quality loss vs FP16.

Model	How	Experience
Flux Schnell	FP8 / GGUF Q4	Workable at 4 steps. ~20-30 sec/image.

At 8GB, SD 1.5 and SDXL are the comfortable options. Flux is technically possible with aggressive quantization but the experience is rough for iterating. If you're generating final images from a known-good prompt, Flux Q4 is fine. For exploring and experimenting, stick with SDXL.

12GB VRAM (RTX 3060 12GB, RTX 4070)

Model	How	Experience
SD 1.5	Native FP16	Overkill. Runs great with ControlNet and LoRAs stacked.
SDXL	Native FP16	Comfortable. Room for LoRAs and ControlNet.
Flux Dev	FP8 or GGUF Q5	Good quality, manageable speed. The sweet spot for Flux.
Flux Schnell	FP8	Fast, good quality. ~20-40 sec/image.

12GB is where the choice gets interesting. You can run all three families without offloading. Flux Dev at FP8 or GGUF Q5 produces near-full-quality images. This is the minimum GPU I'd recommend for someone who wants to primarily use Flux.

24GB VRAM (RTX 3090, RTX 4090)

Model	How	Experience
SD 1.5	Native FP16	Near instant. Under 3 seconds/image.
SDXL	Native FP16 + Refiner	Full pipeline. 4-6 seconds/image.
Flux Dev	FP16 full precision	No compromises. Best possible output. ~18-40 sec/image.
Flux Schnell	FP16	3-10 seconds/image. Fastest Flux experience.

No compromises at 24GB. Run Flux Dev at full FP16 for maximum quality. Keep SDXL around for workflows where you need specific LoRAs or ControlNet types that Flux doesn't support yet. Speed is the only tradeoff: Flux Dev at 18-40 seconds per image vs SDXL at 4-6 seconds.

Choose Your Model

Use case	Best model	Why
Photorealism	Flux Dev	Best anatomy, prompt adherence, natural lighting
Anime/illustration	SD 1.5 (with checkpoint)	Biggest LoRA library for anime styles. Nothing else comes close.
Text in images	Flux Dev	95%+ accuracy on single words. Only reliable option.
ControlNet workflows	SD 1.5	14 official types, most complete ecosystem
Quick drafts/iteration	Flux Schnell or SDXL	Schnell at 4 steps, SDXL at 20 steps. Both fast enough.
Commercial use	Flux Schnell (Apache 2.0) or SDXL	Flux Dev requires a commercial license from BFL.
Low VRAM (4-6 GB)	SD 1.5	Only model that runs natively. SD 3.5 Medium as alternative.
Training your own LoRAs	SD 1.5 (8GB) or SDXL (12GB)	Flux LoRA training needs 24GB+.
Maximum quality, any hardware	Flux Dev (FP16, 24GB)	Best output of any open model, period.

The Bottom Line

SD 1.5 isn't dead, but it's the fallback option now. Use it for anime workflows with specific LoRAs, for ControlNet-heavy pipelines, or when you're stuck on a 4-6GB card.

SDXL is the safe pick. Runs on 8GB, generates at 1024x1024, has a mature ecosystem, and produces good images without fiddling. If you're not sure what to pick, start here.

Flux is where image generation is going. If you have 12GB+, start with Flux Dev in GGUF Q5 through ComfyUI. The quality difference over SDXL is obvious from the first image. Use Schnell for drafts, Dev for finals.

```
# SDXL (8GB+ VRAM) — install ComfyUI, download from CivitAI:
# Juggernaut XL or RealVisXL checkpoint

# Flux (12GB+ VRAM) — install ComfyUI, then:
# Download flux1-dev-Q5_K_S.gguf from city96/FLUX.1-dev-gguf on HuggingFace
# Download clip_l.safetensors and t5xxl_fp8_e4m3fn.safetensors
```

Related Guides

- Stable Diffusion Locally: Getting Started
- Flux Locally: Complete Guide
- ComfyUI vs Automatic1111 vs Fooocus
- What Can You Run on 8GB VRAM?
- ControlNet Guide for Beginners
- Local AI Planning Tool — VRAM Calculator

Sources: Tom's Hardware SD Benchmarks, ComfyUI GPU Benchmarks, Stable Diffusion Art SDXL vs Flux, Nunchaku/SVDQuant, BFL Flux.2 Blog, CivitAI LoRA Training Guide