PortraitMaster FLUX1 — Introductory Note

Thank you for downloading and purchasing PortraitMaster.

This documentation is not intended as a beginner's tutorial on ComfyUI, nor is it a general guide to diffusion pipelines. It assumes a working familiarity with the interface and modular logic of ComfyUI, and focuses instead on the architectural philosophy, advanced composition logic, and engineering of the PortraitMaster FLUX1 Workflow Suite.

Most nodes used in this workflow are already widely known and documented. However, this suite rethinks how these components interact, fusing them into a consistent portrait rendering system through rigorous experimentation and daily iteration.

What PortraitMaster is — and What It Isn't

This document is not a step-by-step course on how ComfyUI or FLUX1 works. It is a deep technical layer built on top of them, structured for advanced users who seek surgical control over character expression, identity consistency, micro-detailing, and professional image output.

At the heart of the workflow is FLUX1-Dev, a diffusion UNet trained in float8 formats — specifically fp8_e4m3fn and e5m2. These formats allow compressed representation of tensors with ultra-fast inference while preserving reconstruction fidelity in portrait-oriented tasks. The result is faster computation, especially when paired with hardware that supports native 8-bit matrix operations.

Complementing the UNet is the dual-text encoder system: T5-XXL and CLIP-L.

- T5-XXL, a massive transformer from Google, handles descriptive and complex linguistic conditioning with broad generalisation capability.
- CLIP-L anchors prompts with visual precision, enabling refined semantic targeting through image-prompt intersection.

Redux, in this context, is a style-delta modifier trained to inject nuanced visual signatures (skin tone, light diffusion, texture harmonisation) without overpowering the base character structure. When used with LoRA adapters, Redux enables hybridisation between prompt-driven intent and dataset-level realism.

LoRA adapters, stacked via the PowerLoader module, surgically alter visual expression — from micro-skin fidelity to makeup, tone, or even era-specific gloss. Paired with SigCLIP Vision Patch14/384, the system allows bi-directional conditioning: text-to-image and image-to-latent loops, with high alignment in the visual attention maps.

And at the latent decode level, we employ a purpose-trained VAE (autoencoder) optimised for facial reconstruction. This VAE decodes high-density latents with controlled chromatic and tonal recovery — critical for avoiding the plastic overtones of weaker decoders.

Diffusion Isn't Magic — It's Architecture

Let's be clear: generative AI is not a button that makes art.
Diffusion models are layered systems of inference, where information is sculpted from noise by recursive evaluation over hundreds of steps. These systems rely on conditioners — which can be image embeddings, CLIP token maps, expression vectors — to guide how noise collapses into form.

Three major types of conditioning exist in this workflow:

- Textual (CLIP, T5): semantic intent, camera logic, visual modifiers.
- Visual (CLIP-Vision, Redux): embedding of identity, lighting, and mood.
- Latent (LoRA, Sigma reweighting): structural anchors that alter the behavior of the denoiser or patch grid.

A well-designed flow must orchestrate these signals in order, with balance, and with an understanding that no single conditioner owns the final output. Instead, they negotiate — often with unexpected results — and this is where real mastery lies.

We don't just stack components: we shape the temporal and spatial relationships between them. Sigma samplers, Detail Daemon overlays, and Lying Sigma rebalancers are tools that reshape the latent field itself — where realism is not generated, but emerges from careful interplay.

Why PortraitMaster Exists — Beyond Nodes

This workflow suite is the product of hundreds of thousands of tests, parameter variations, controlled A/B runs, and — most importantly — visual judgment born from years of professional image production. It doesn't come from documentation alone. It comes from failure, trial, the exploration of community experiments, Patreon test branches, and beta models that crash half the time.

PortraitMaster is not a technical exercise. It is the result of a passion to create tools that elevate AI image generation from novelty to craft — where every toggle, every blend, every upscaling curve matters.

You won’t find “plug and play” perfection here. What you will find is a suite of mechanisms — meticulously designed — that give you professional control, if you are willing to invest time and curiosity.

📁 Inside the ZIP — What You’ll Find

Inside the downloaded archive, you'll receive:

- The full .json ComfyUI workflows — ready to be loaded without manual node assembly. These include all node groups and parameter presets for Expression, Core, and Lite variants.
- Sample files for portrait generation — including layered templates, expression editor images, and mock reference portraits that allow immediate testing of character-based generation.
- Comprehensive documentation for every step — with detailed descriptions of each node group, input requirement, recommended configurations, and advanced tuning strategies to maximise image quality and control.

You will also find a second document:
“Character Consistency Engineering — PortraitMaster Method”.
This goes beyond generation. It is a meticulous guide on how to maintain photographic coherence across dozens or hundreds of AI renders, keeping identity, pose, gaze, and lighting stable while iterating clothing, style, or mood. This is a long and demanding process — not automated, not mass-produced — but necessary if you're building serious content pipelines or AI-based casting portfolios.

This Is Not a Shortcut. It's a Discipline.

If you believe generative AI is an “instant photoshoot” tool, you will be disappointed.

This is not MidJourney. It’s not a filter.
This is not a one-click solution.

This is a platform for creators — built to reward time, attention, and taste.
Hours of experimentation will yield control.
Repetition will yield consistency.
And from there, true creative direction becomes possible.

Welcome to PortraitMaster.

PortraitMaster FLUX1 Workflow Suite — Technical Documentation

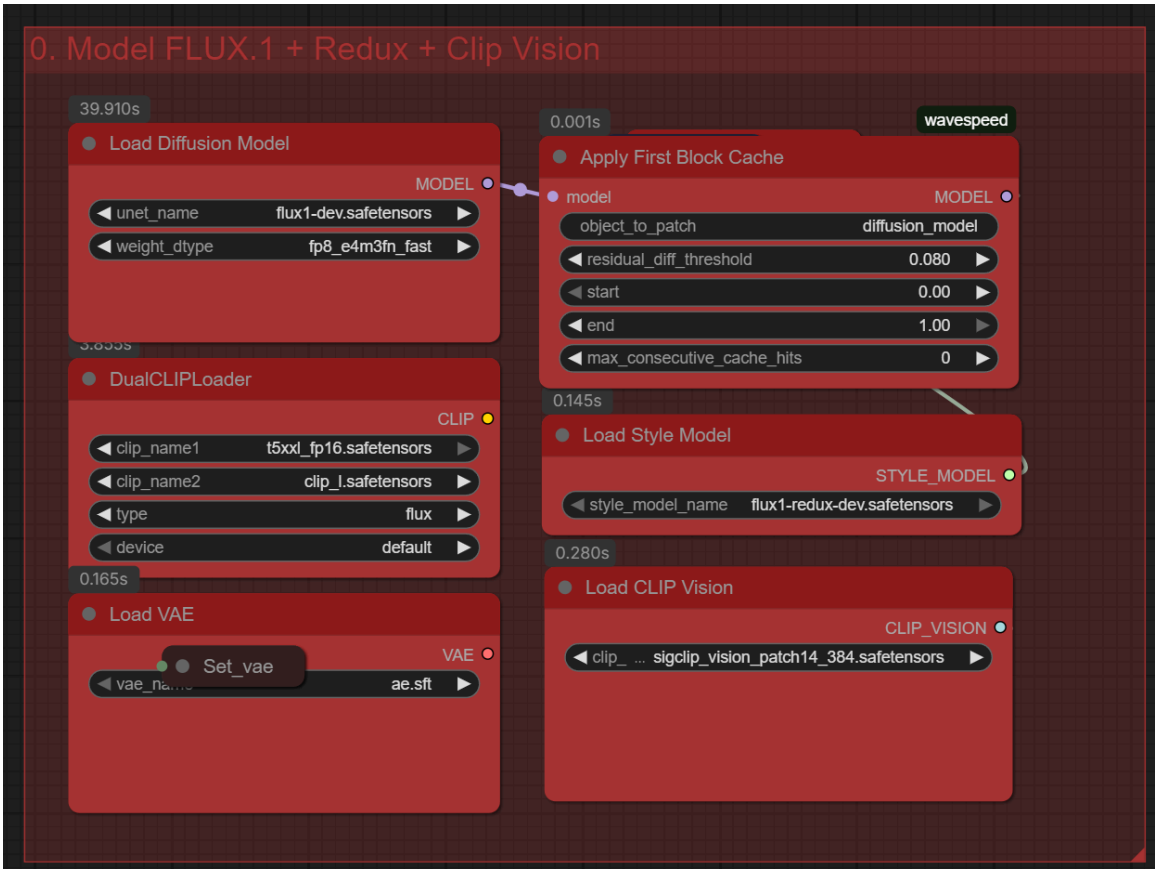
Variants: Expression • Core • Lite

1 Introduction & Scope

This document provides a technical overview of the “PortraitMaster FLUX1” workflow family for ComfyUI. The three variants — Expression, Core, and Lite — share an identical graph topology; performance differences lie primarily in the default parameters, active sub-modules, and optional post-processing chains suited to various GPU budgets.

Each workflow is modular, enabling rapid substitution of models, schedulers, and conditioning sources without restructuring the graph. The documentation is organised per functional group; every group header in the ComfyUI canvas maps to a chapter in this file.

2 Group 0 — Model FLUX 1 + Redux + CLIP Vision



2.1 Objectives

- **Load the base diffusion backbone (FLUX 1)**
- Apply first-block caching for inference acceleration
- Attach paired text encoders (T5-XXL + CLIP-L)
- Inject the Redux style modifier
- Load a dedicated VAE for high-fidelity reconstruction
- Register CLIP-Vision embeddings for downstream guidance

2.2 Node Map (High-Level Flow)

The group initialises the computational context. Execution begins with "Load Diffusion Model", producing a UNet handle consumed by "Apply First Block Cache" to patch early residuals. Text conditioning is injected via "DualCLIPLoader", while stylistic deltas are merged through "Load Style Model". The VAE and CLIP-Vision loaders expose latent-space reconstruction and image-level embedding streams respectively.

2.3 Node-Level Specification

Node	Role	Key Parameters	Artifact(s)
Load Diffusion Model	Loads FLUX1 UNet weights	unet_name = flux1-dev.safetensors weight_dtype = fp8_e4m3fn_fast	flux1-dev.safetensors
Apply First Block Cache	Injects residual-diff caching (WaveSpeed)	residual_diff_threshold = 0.08 start=0 • end=1 max_cache_hits=0	—
DualCLIPLoader	Text encoders (T5-XXL & CLIP-L)	clip_name1 = t5xxl_fp16.safetensors clip_name2 = clip_l.safetensors type = flux	t5xxl_fp16.safetensors; clip_l.safetensors
Load Style Model	Applies Redux style delta	style_model_name = flux1-redux-dev.safetensors	flux1-redux-dev.safetensors
Load VAE	Latent auto-encoder	vae_name = ae.sft	ae.sft
Load CLIP Vision	Vision-space encoder	clip_vision_name = sigclip_vision_patch14_384.safetensors	sigclip_vision_patch14_384.safetensors

2.4 Prerequisites & Installation

Environment — ComfyUI ≥ 0.3.15 (Python 3.10, CUDA 11.8).

Clone the required custom-node repositories into "/custom_nodes":

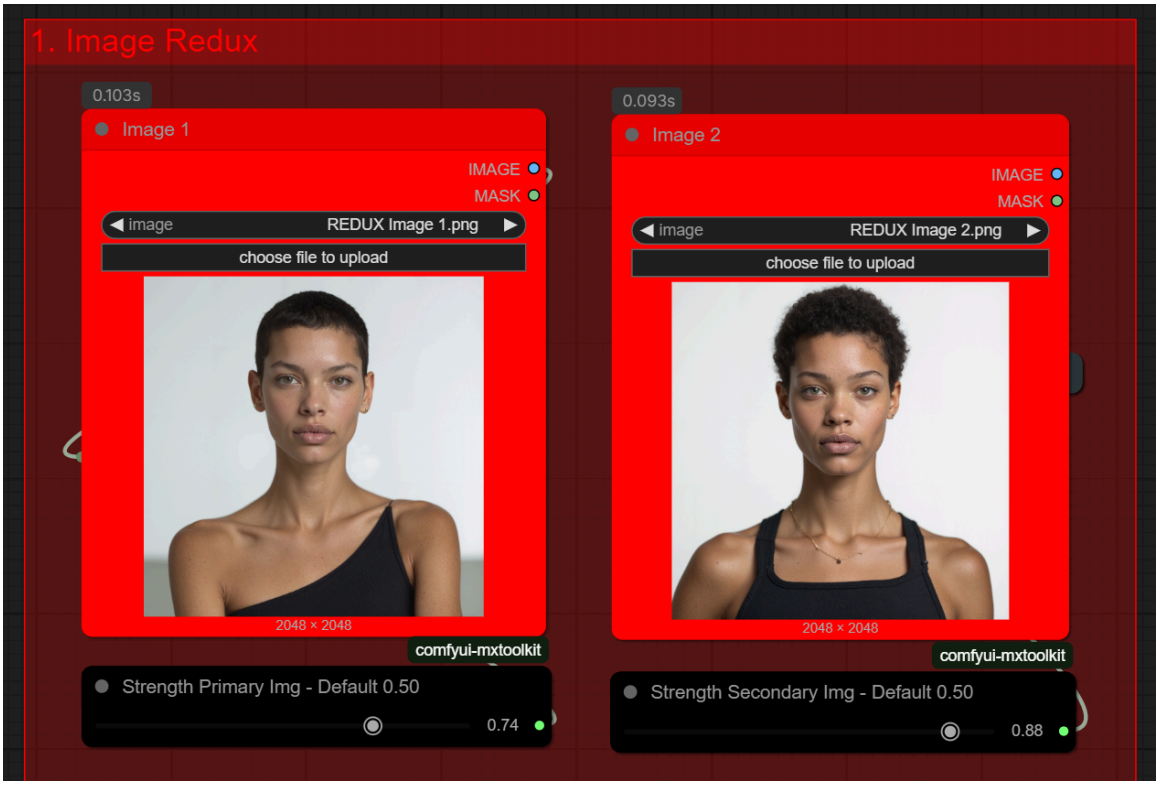
- git clone https://github.com/wavespeed-comfy/comfy_fbcache ./custom_nodes/WaveSpeed-FBCache
- git clone https://github.com/aria-comfy/dual-clip-loader ./custom_nodes/DualCLIPLoader

Place model files in the corresponding directories:

File	Target Folder
flux1-dev.safetensors	models/checkpoints
flux1-redux-dev.safetensors	models/style_models
ae.sft	models/vae
t5xxl_fp16.safetensors	models/clip
clip_l.safetensors	models/clip
sigclip_vision_patch14_384.safetensors	models/clip_vision

After copying, restart ComfyUI to index the new assets. No additional requirements.txt entries are needed beyond those supplied by the cloned nodes.

3 Group 1 — Image Redux & Conditioning



3.1 Objectives

This group injects visual conditioning into the diffusion stream by fusing two reference portraits through the Redux pipeline. The dual-image approach enables controlled identity blending, style transfer, and dataset diversification while maintaining topology consistency for subsequent latent operations.

3.2 Asset Package

The Gumroad ZIP includes starter resources:

• Mockup Face.psd — a layered template with safe-zone guides for head placement.

• REDUX Image 1.png & REDUX Image 2.png — sample frontal portraits for quick validation.

Placeholders may be substituted with original, rights-cleared photography. Stock libraries (paid or free) are acceptable if their licence permits AI derivative works. For experimentation, Pinterest offers ample test material; ensure attribution is not required for final output.

3.3 Image Selection Guidelines

• Framing: shoulders-up, neutral lighting, minimal occlusion.

• Pose Consistency: align gaze direction and head tilt across both images to minimise warping.

• Diversity: combine heterogeneous demographics (e.g., Asian × Caucasian) to explore

hybrid facial traits.
• Resolution: ≥ 2048 px square recommended for full latent bandwidth.
• Legal: secure copyright or shoot proprietary reference to guarantee unrestricted commercial usage.

3.4 Node-Level Specification

Node	Role	Key Parameters	Input/Output
Image 1	Primary reference portrait	Strength slider (1–100 → 0.01–1.00)	IMAGE
Image 2	Secondary reference portrait	Strength slider (1–100 → 0.01–1.00)	IMAGE
CLIP Vision Encode ×2	Embeds each image in vision latent space	Encoder: sigclip_patch14_384	CONDITIONING
Get_conditioning	Retrieves CLIP embeddings from cache	—	CONDITIONING
Set_reduxconditionin	Injects fused conditioning into downstream pipeline	blend_mode = additive weighting = dynamic	CONDITIONING

3.5 Latent Noise Injection — Parameter Reference

This control panel, courtesy of comfyui-mxtoolkit, operates **after** the initial encode phase. Values outside the suggested ranges are legal but increase the likelihood of artifacts.

• Latent Upscale (1.0 – 2.0) — default 1.25
1.0 preserves native scale; 1.25–1.5 enhances micro-detail; >1.6 may clip highlights.

• Flux Guidance (0.0 – 2.5) — default 1.90
Boosts structural rigidity. 1.8–2.0 recommended for training-grade portraits.

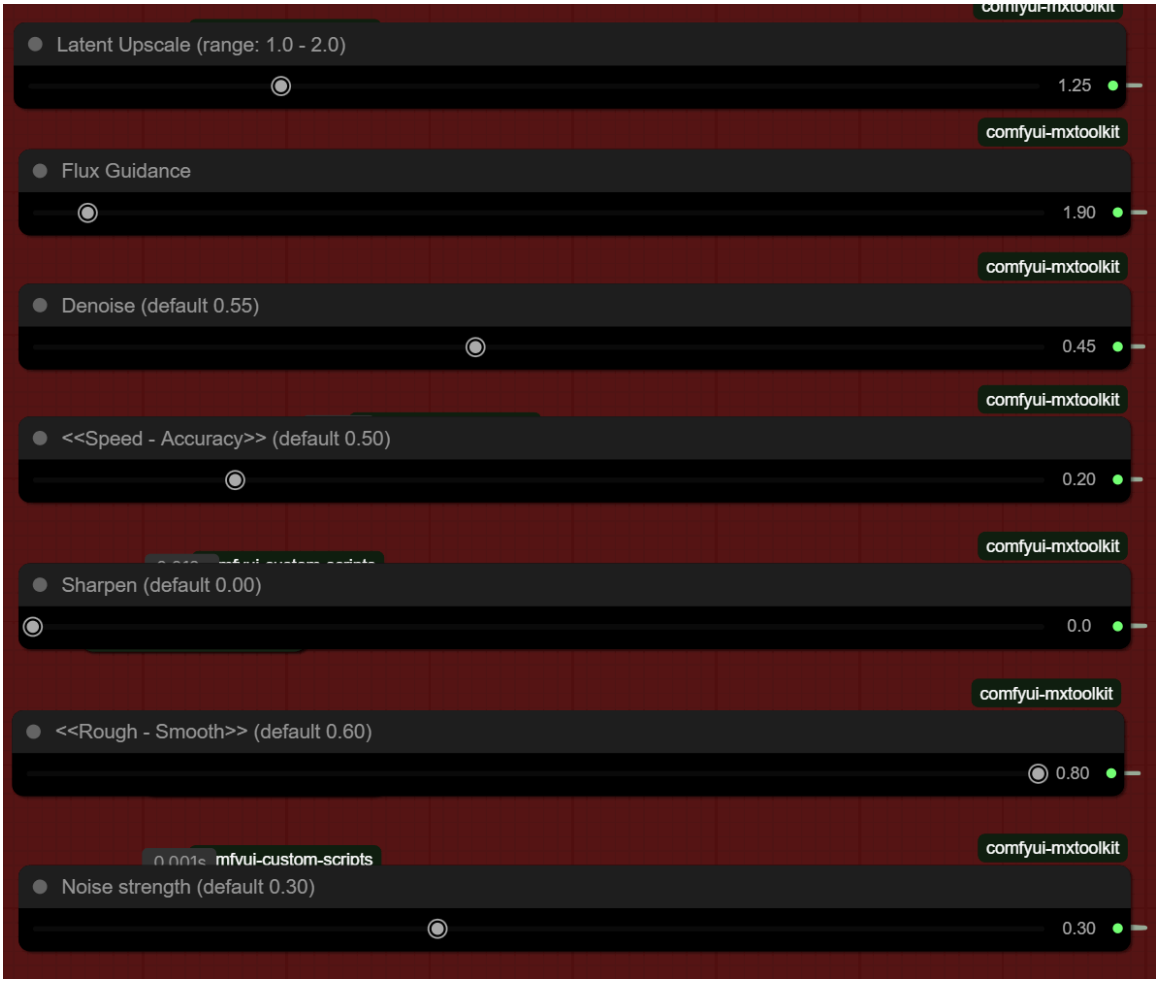
• Denoise (0.0 – 1.0) — default 0.45
0.3–0.4 keeps texture; 0.5–0.6 balances; >0.7 risks identity drift.

• Speed ↔ Accuracy (0.0 – 1.0) — default 1.00
Full quality path is advised for final renders; 0.6–0.8 suffices for previews.

• Sharpen (0.0 – 0.5) — default 0.00
0.1–0.2 gently accents irises and lip lines.

• Rough ↔ Smooth (0.0 – 1.0) — default 0.80
0.65–0.75 yields a glossy editorial finish.

• Noise Strength (0.0 – 0.8) — default 0.30
0.3–0.4 is ideal for LoRA dataset generation; >0.5 introduces stylisation.



3.6 Practical Workflow

1. Drop your two aligned portraits into **Image 1** and **Image 2**.
2. Adjust each strength slider (1 – 100). A 70/30 split favours Image 1 while retaining traits from Image 2.
3. Iterate Latent Noise Injection parameters in small steps, exporting test frames.
4. Once satisfied, lock parameters and proceed to the sampling group (next chapter).

4 Group 2 — Size / LoRA / Prompt

4.1 Objectives

This block defines render resolution, selectively injects specialised LoRA weights, and anchors the text-prompt that steers the CLIP pipeline. Combined, these three nodes set the aesthetic foundation before diffusion sampling begins.

4.2 Resolution Guidelines

Flux1 inherits latent patching from the SD-XL architecture: internal tiles are 128 px. For portrait work, **square frames** (1024^2–1536^2) deliver maximal detail without fragmenting facial proportions. The *Basic Image Size* node auto-rounds odd values to the nearest 128 px divisor; setting `width_override = height_override = 1024` resolves to an internal latent of 1152 × 896 (scale ≈ 1.29) — the most efficient point on consumer GPUs.

4.3 Node Breakdown

Node	Role	Key Parameters	Comments
Basic Image Size	Sets latent W/H & batch	width_override, height_override, resolution multiplier	Prefer 1024 × 1024; avoid ratios <1:1 when training LoRA
Power LoRA Loader	Attaches multiple LoRA adapters	Strength sliders per slot (0–1)	Supports up to 4 adapters; enable *Toggle All* for quick A/B
CLIP Text Encode (Prompt)	Tokenises master prompt	context_clip = sigclip_patch14_384	Feed with GPT-generated prompt

4.4 Recommended LoRA Library

LoRA	Purpose	Trigger Words	Link
Flux Skin Texture	Enhances pores & micro-imperfections while retaining natural look	flux skin texture	https://civitai.com/models/651043/flux-skin-texture
Photorealistic Skin (No Plastic)	Removes plastic sheen; preserves organic micro-contrast	photorealistic skin	https://civitai.com/models/1157318/photorealistic-skin-no-plastic-flux
Skin Tone Glamour	Editorial colour balance for soft-lit glamour shots	glamour skin tone	https://civitai.com/models/562884/skin-tone-glamour-photography-style-human-skin-color-xl-f1d-sd15-pony-illu
Female Face Macro	Macro close-ups with eyelash & reflection fidelity	macro female portrait	https://civitai.com/models/1019792/female-face-portraits-detailed-skin-closeup-macro-flux

Luscious Lips	Boosts lip volume & symmetry for beauty visuals	luscious lips	https://civitai.com/models/951276/luscious-lips-and-detailed-faces
ESC Makeup	Adds photo-real makeup accents without over-processing	ESC makeup	https://civitai.com/models/1060990/esc-makeup

4.5 Prompt Assistant — Portrait Flux.1 GPT

Use the dedicated GPT at https://chatgpt.com/g/g-67e2fa234e388191bec4e06357fc1275-portrait-flux-1 to autogenerate structured prompts. Drop a reference image into the chat, request a character description, then copy the result directly into the **CLIP Text Encode** node. The GPT can post-edit specific attributes (e.g., "blue eyes, damaged skin, blonde hair") on demand.

4.6 Micro-Sampler Controls

Detail Demon Sampler — injects a micro-detail emphasis layer during sampling.

• 0.00 = off
• 0.03–0.07 = subtle grain
• 0.08–0.12 = pronounced pores / hair strands
>0.15 = risk of artificial speckling
Recommended: **0.05–0.10** for 1024^2 renders.

Advanced Lying Sigma Sampler — rebalances sigma deviation in the latent noise field.

• 0.00 = neutral
• -0.02 to -0.06 = reduces haloing / texture bleed
• < -0.08 = stylised cinematic, lower realism
Recommended: **-0.05 to -0.06** for photoreal faces.

Tip: tune in concert with *Noise Strength* and *Smoothness* controls to maximise skin clarity.

4 Group 2 — Size / LoRA / Prompt (Expanded)

4.1 Module Scope & Objectives

Group 2 is the *pre-sampling* staging area. Here we: (1) declare the target canvas size that drives the UNet's convolution grid, (2) stack one or more LoRA adapters to surgically alter visual style or anatomy, and (3) feed a token-dense prompt into CLIP to anchor semantic intent. Any change made in this group propagates downstream, influencing how latent noise is interpreted by Flux1.

4.2 Resolution Strategy — Getting the Most out of Flux1

Flux1's UNet derives from SD-XL, meaning it operates natively on 128-pixel patches. Internally, latent dimensions are multiplied by **0.130 ×** to generate token maps for the text encoder. Keeping height and width divisible by 128 avoids padding artefacts and guarantees that attention maps align with facial landmarks.

• **1024 × 1024 px** — ideal for single-GPU cards (12 GB+). Gives eye-level catch-lights and pore fidelity with minimal VRAM fragmentation.
• **1280 × 1280 px** — pushes skin micro-detail further, but requires ~17 GB VRAM. LoRAs set above **0.8** may introduce checkerboarding at this size.
• **1536 × 1536 px** — cinematic head-and-shoulders crops; needs multi-GPU pipeline or Intel Arc 20 GB. Use *latent upscaling* in Group 3 instead if you lack headroom.

4.2.1 Basic Image Size Node — Field Reference

Field	Type	Guideline	Effect on Pipeline
width_override	int	≥ 1024 & ÷128	Expands latent grid horizontally
height_override	int	Same as width for symmetry	Adjusts vertical grid
resolution	float	1.0 – 2.0	Scales canvas after override values
batch_size	int	1 – 4 (GPU bound)	Parallel image generation

Tip: When preparing training data, lock **width_override = height_override** to maintain aspect ratio across the entire dataset; inconsistent framing can degrade LoRA convergence.

4.3 Power LoRA Loader — Advanced Usage

The *Power LoRA Loader* allows **blended adaptation** by linearly combining up to four LoRA matrices in a single forward pass. Each slider multiplies the Δ-weights before injection into Flux1's attention blocks.

• `strength 0.0` — bypasses the adapter (identity mapping).
• `strength 0.05–0.15` — subtle stylistic cues; good for skin tone shifts.

• `strength 0.3–0.5` — dominant influence, suitable for macro skin LoRAs.
• `> 0.6` — overrides base model; stack cautiously to avoid texture clashes.

When **multiple LoRAs** are active, the effective weight is simply the sum of all strengths. Keep the total ≤ **1.0** to prevent tensor overflow and attention saturations.

4.3.1 Curated Skin-Focused LoRAs

Flux Skin Texture (https://civitai.com/models/651043/flux-skin-texture)
Micro-bump map enhancer. Adds realistic pores and sebaceous texture without amplifying colour noise. Best in the 0.06–0.12 range.
Trigger: “flux skin texture”

Photorealistic Skin (No Plastic)
(https://civitai.com/models/1157318/photorealistic-skin-no-plastic-flux)
Targets over-smoothed regions by reinstating mid-frequency detail. Combine with *Detail Demon* 0.05 for magazine-grade renders.
Trigger: “photorealistic skin”

Skin Tone Glamour
(https://civitai.com/models/562884/skin-tone-glamour-photography-style-human-skin-color-xl-f1d-sd15-pony-illu)
Imbues subtle melanin variation and warmer undertones. Use 0.04–0.08 for beauty adverts.
Trigger: “glamour skin tone”

Female Face Macro
(https://civitai.com/models/1019792/female-face-portraits-detailed-skin-closeup-macro-flux)
Precision macro detail for eyelashes, fine wrinkles, and specular sweat highlights. Requires 1024^2 or higher canvas.
Trigger: “macro female portrait”

Luscious Lips (https://civitai.com/models/951276/luscious-lips-and-detailed-faces)
Accentuates lip geometry and moist specularity. Stack with *ESC Makeup* for fashion close-ups.
Trigger: “luscious lips”

ESC Makeup (https://civitai.com/models/1060990/esc-makeup)
Applies natural makeup shading—eyeshadow, blush, and subtle eyeliner—without crossing into CGI gloss.

Trigger: “ESC makeup”

4.4 Prompt Engineering & GPT Assistant

A well-structured prompt reduces diffusion entropy. The **CLIP Text Encode** node leverages Sig-CLIP ViT-H/14 384 to embed up to **75 tokens**. Surplus tokens are truncated; therefore favour concrete nouns and photographic modifiers over verbose prose.

Workflow with *Portrait Flux.1 GPT*:
1. Upload a reference selfie or stock portrait into the GPT chat.
2. Ask: “Generate a detailed portrait prompt (English, photographic)”.
3. Iterate: “Change to blue eyes, damaged skin, blonde hair”.
4. Copy the final prompt block → paste into the **Prompt** textbox.

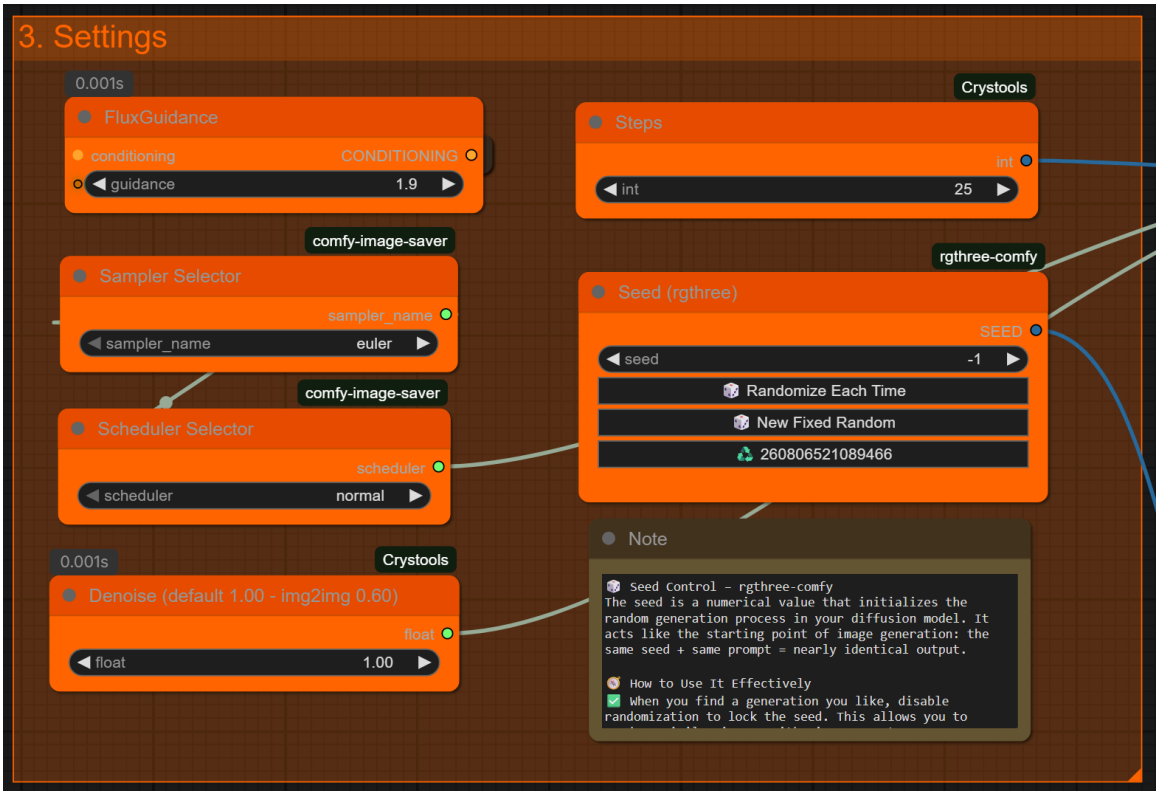
Best practice: Prepend *Trigger Words* for active LoRAs at the front of the prompt to guarantee token proximity (e.g., “flux skin texture, photorealistic skin, ” …).

4.5 Micro-Sampler Diagnostics

Control	Range	Primary Effect	Common Pitfalls	Pro Tip
Detail Demon	0.00 – 0.15	Adds high-frequency detail into normal maps; boosts skin tactility, eyebrow fibres.	Values >0.12 manifest grain in low-light backdrops.	Cascade with *Sharpen 0.1* for razor-sharp irises.
Advanced Lying Sigma	0.00 → −0.12	Re-weights sigma decay, suppressing halo/fringe artefacts and uneven gloss patches.	< −0.08 darkens mid-tones; can flatten dynamic range.	Pair with *Noise Strength 0.25* for filmic grain.

Advanced tuning sequence:
• Start with *Detail Demon 0.05* and *Lying Sigma -0.04*.
• Render a test batch; measure SSIM vs. reference shot.
• Increment *Detail Demon* by 0.02 until over-sharpening appears, then step back.
• Adjust *Lying Sigma* in −0.01 steps to squash halo rings around high-contrast edges.

5 Group 3 — Settings



5.1 Module Scope & Purpose

Group 3 consolidates **inference hyper-parameters**. Here we finalise guidance strength, sampler / scheduler pairing, denoise intensity, step count, and seed management. Tweaks in this stage do **not** alter graph topology; they refine how latent noise converges toward the prompt and Redux conditioning.

5.2 Node Overview

Node	Role	Critical Fields	Best-Practice Range
FluxGuidance	Applies global guidance multiplier	guidance (float)	1.6 – 2.1 for portrait fidelity
Steps	Total UNet iterations	int	20 – 28 steps (noise-to-image)
Sampler Selector	Chooses sampling algorithm	sampler_name	Euler, Euler-a, DPM++ SDE Karras
Scheduler Selector	Sets noise-variance schedule	scheduler	Beta, Karras, Polyex

Denoise	Img2Img noise strength	float	0.45 – 0.60 (portraits) 1.00 (txt2img)
Seed	Initialises RNG for reproducibility	seed (int)	-1 =random; fixed = repeatable

5.3 Seed Control — Theory & Practice

A **seed** initializes the pseudo-random number generator that lays down the latent noise grid. Keeping *prompt, LoRA stack, and seed* identical yields nearly deterministic output, enabling micro-iteration on other parameters.

• **Randomize Each Time** — sets seed = −1; perfect for broad exploration.
• **New Fixed Random** — draws a fresh seed then locks it; ideal when you land on a promising composition.
• **Use Last Queued Seed** — reuses the previous job's seed, supporting iterative prompt refinement.

Flux1 quirks: certain seeds bias toward ultra-close crops, elongated neck geometry, or forehead clipping. If you encounter these artefacts:
1. Nudge *seed* ±50.
2. Lower *denoise* by ~0.05.
3. Optionally decrease *FluxGuidance* to 1.6.

Maintain a spreadsheet of 'golden seeds' that reliably produce framed, well-proportioned faces. Reuse them across variant prompts to preserve identity.

5.4 Inference Hyper-Parameter Tuning

Steps vs. Sampler
• Euler / Euler-a — robust, forgiving; 24–28 steps for high-detail faces.
• DPM++ 2M Karras — cleaner gradients; 18–22 steps often suffice.
• Heun ++ — preserves edge contrast; combine with *Detail Demon ≥0.08*.

Scheduler pairing
• Beta — default; balanced sharpness vs. colour stability.
• Karras — smoother noise decay; reduces blotching on dark skin.
• Polyex — experimental; vivid mid-tones, occasionally posterises highlights.

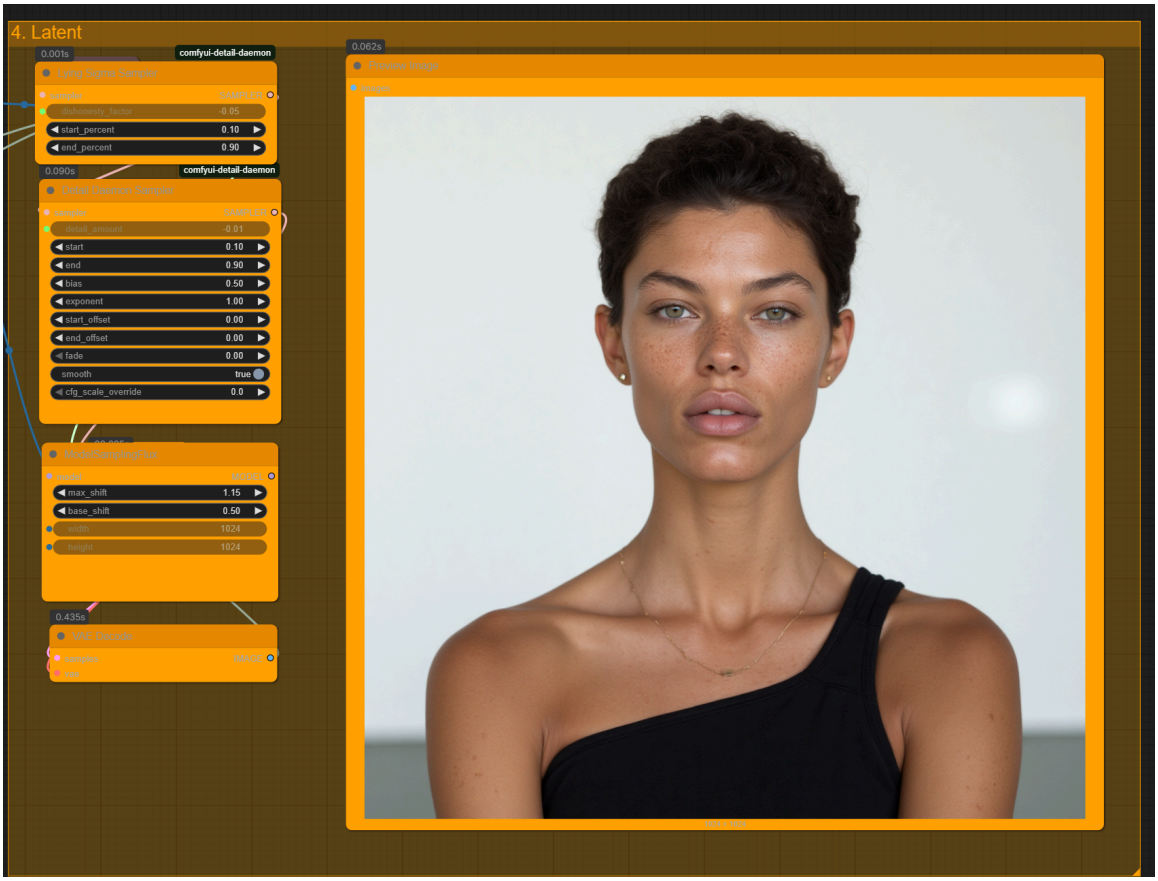
Denoise knob
0.45–0.55 retains identity for img2img Redux workflows.
≥ 0.9 effectively becomes text-to-image generation.

5.5 Practical Workflow

1. Start with **FluxGuidance 1.9**, **Euler**, **Beta**, **25 steps**, **denoise 1.0**, and seed = −1.

2. Render a 4‑image batch.
3. Choose the best frame → click **New Fixed Random** to lock its seed.
4. Modify prompt / LoRA strengths; keep seed fixed to audit isolated changes.
5. If noise artefacts appear, reduce *Steps* by 2 and raise *FluxGuidance* to 2.0 — this reins in over‑processing without sacrificing fidelity.

6 Group 4 — Latent



6.1 Module Scope

The Latent group applies **post-sampler refinements** *inside* the latent space prior to VAE decoding. It manipulates sigma trajectories and micro-detail masks rather than the RGB output, ensuring razor-sharp skin without introducing pixel-domain artefacts. Both samplers inherit their core amplitude parameters from *Group 2* to keep global control centralised.

6.2 Node Map & Field Reference

Node	Role	Key Fields (Group 4)	Inherited Fields (Group 2)
Lying Sigma Sampler	Re-weights sigma decay curve to suppress haloing and uneven specularity.	start_percent (0.10) end_percent (0.90)	dishonesty_factor (−0.05)
Detail Daemon Sampler	Injects high-frequency bump layer into latent;	start / end (0.10-0.90) bias (0.0) exponent (1.0)	detail_amount (0.01)

	boosts pores & eyelashes.		
ModelSamplingFlux	Shifts latent grid between UNet passes to mitigate tile seams.	max_shift (1.15) base_shift (0.50)	—
VAE Decode	Decodes refined latent into RGB image.	—	—

6.3 Parameter Guidance

Lying Sigma Sampler

• `dishonesty_factor` is set in Group 2; typical range −0.02 – −0.08.
• Use `start_percent`/`end_percent` to gate the correction window. 0.10-0.90 targets mid-diffusion where halos tend to appear.
• Narrowing the window (e.g., 0.25-0.75) preserves highlight roll-off on glossy skin.

Detail Daemon Sampler

• `detail_amount` is routed from Group 2. Values 0.05–0.10 suit 1024^2 canvases.
• `bias` skews detail toward highlights (>0) or shadows (<0). Keep 0.0 for even skin.
• `exponent` controls fall-off: 1.0 linear, >1 accentuates extremes.
• `smooth = true` avoids ringing on low-frequency areas (e.g. cheeks).

ModelSamplingFlux

Latent jitter combats visible tile edges and pattern repetition:
• `base_shift` = baseline pixel offset (0.50 ≈ half-pixel).
• `max_shift` scales additional random offset per step.
Values >1.5 may blur micro-geometry; keep ≤1.25 for portrait work.

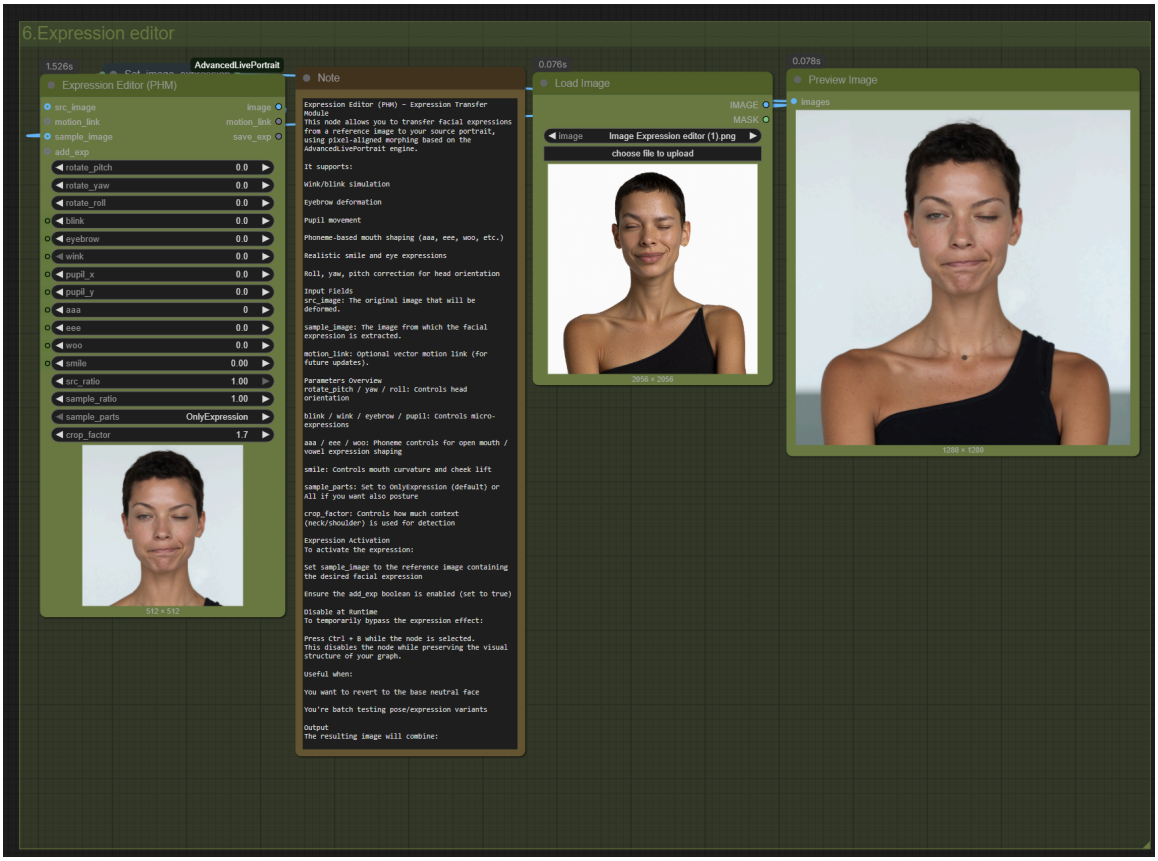
6.4 Workflow Tips

1. Dial in *Detail Daemon* and *Lying Sigma* strengths in Group 2 first; leave start/end bands at default 0.10-0.90 in Group 4.
2. If pore details look overly crisp, raise `bias` to 0.2 and set `smooth = true`.
3. Persistent stitching along the jawline? Increase `base_shift` to 0.60 and reduce `detail_amount` by 20 %.
4. Always inspect 200 % zoom before final export; latent-space tweaks can hide sub-pixel artefacts not obvious at native res.

6.5 Preview Image Widget

The on-canvas **Preview Image** node updates every render, providing an RGB check *before* the final Image Saver. Its border colour reflects the current seed state (orange = randomised, teal = locked). Use it to quick-scrub through parameter tweaks without polluting your output folder.

7 Group 6 — Expression Editor (PHM)



7.1 What Is the Expression Editor?

This node performs **expression transfer**: it morphs a generated face so that its micro-expressions match those of a reference photo. The engine is based on AdvancedLivePortrait's vector-alignment algorithm and operates entirely in the latent domain, so no post-processing blur is introduced.

The **sample_image** input is **disabled by default** (grey socket). When you press **Ctrl +B** while the node is selected, the toggle activates and the node will ingest the reference picture you supply (*Image Expression editor (1|2).png* included in the Gumroad pack) to drive pose and facial action units.

7.2 Field Reference

Field	Range	Effect
rotate_pitch / yaw / roll	−30 → +30	Head orientation
blink	0 → 1	Eyelid closure
wink	−1 → +1	Left / right wink

eyebrow	−1 → +1	Brow raise or frown
pupil_x / pupil_y	−1 → +1	Eye direction (x,y)
aaa / eee / ooo / woo	0 → 1	Phoneme-based mouth shapes
smile	0 → 1	Mouth corner lift
src_ratio	0 → 1	Blend ratio of source pose
sample_ratio	0 → 1	Blend ratio of sample pose
crop_factor	1.0 → 2.0	Context area for feature detection

Set **sample_parts** to **OnlyExpression** when you want to keep head pose from the base generation and copy only eyes + mouth + brows from the reference image.

7.3 Preset Library — 24 Ready-to-Use Expressions

Preset	Pitch	Yaw	Roll	Blink	Eyebrow	Wink	Pupil (x,y)	Mouth (aaa/eee/ooo/woo)	Smile	Notes
Eyes Closed + Gentle Smile	0	0	0	1.0	0.2	0.0	0,0	0/0/0/0	0.3	Serene expression
Big Smile Teeth	0	0	0	0.0	0.4	0.0	0,0	0.2/0/0.1/0	0.9	Cheerful, open mouth
Eyes Closed Laugh	2	0	0	1.0	0.6	0.0	0,0	0.3/0/0.4/0	1.0	Head tilt with laughter
Neutral Base	0	0	0	0	0	0	0,0	0/0/0/0	0	Default
Soft Smile	0	0	0	0	0.1	0	0,0	0/0/0/0	0.4	Subtle friendliness
Raised Eyebrow Skeptic	0	0	0	0	0.6	0	0,0	0/0/0/0	0.1	Curiosity
Side Wink Left	0	0	0	0	0.2	-1	0.3, 0	0/0/0/0.1	0.4	Playful
Side Wink Right	0	0	0	0	0.2	1	-0.3 ,0	0/0/0/0.1	0.4	Playful reverse
Surprised Eyes	0	0	0	0	-0.6	0	0,0	0/0.8/0/0	0.0	Eyes wide, mouth “eee”
O-Shape Mouth	0	0	0	0	0	0	0,0	0/0/1.0/0	0	Whistling

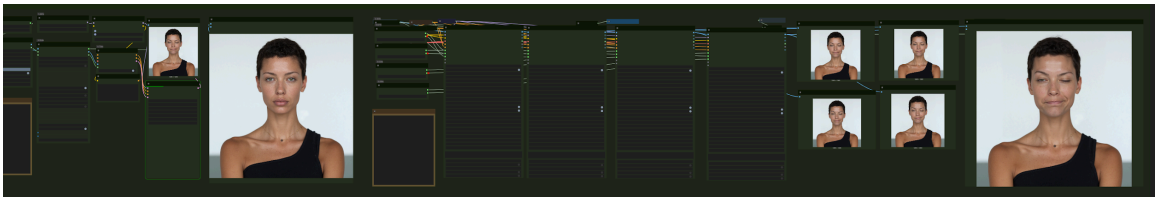
Shout “Woo”	0	0	0	0	-0.2	0	0,0	0/0/0/1.0	0	Excitement yell
Cheeky Smirk Left	0	-3	0	0	0.3	0	0.2,0	0/0/0/0	0.6	Asymmetric smile
Cheeky Smirk Right	0	3	0	0	0.3	0	-0.2,0	0/0/0/0	0.6	Asymmetric reverse
Eyes Up Left	-8	-15	0	0	0	0	-1,-1	0/0/0/0	0.0	Looking up left
Eyes Up Right	-8	15	0	0	0	0	1,-1	0/0/0/0	0.0	Looking up right
Eyes Down	8	0	0	0	0	0	0,1	0/0/0/0	0.0	Thoughtful
Head Tilt Left	0	0	-10	0	0.2	0	0,0	0/0/0/0	0.3	Cute tilt
Head Tilt Right	0	0	10	0	0.2	0	0,0	0/0/0/0	0.3	Cute tilt
Angry Frown	0	0	0	0.0	-1.0	0	0,0	0/0/0/0	0.0	Brow down
Pain Grimace	0	0	0	0.5	-0.8	0	0,0	0.6/0/0.3/0	0.0	Tension
Big Yawn	0	0	0	0.2	-0.5	0	0,0	1.0/0/0/0	0.0	Wide open mouth
Kiss Pout	0	0	0	0	0.1	0	0,0	0/1.0/0/0	0.2	Puckered lips
Side Glance Left	0	-15	0	0	0	0	-1,0	0/0/0/0	0.0	Eyes side left
Side Glance Right	0	15	0	0	0	0	1,0	0/0/0/0	0.0	Eyes side right

Copy these values into the Expression Editor or save them as preset files under */configs/phm_presets/*.json* for one-click access.

7.4 Workflow Example

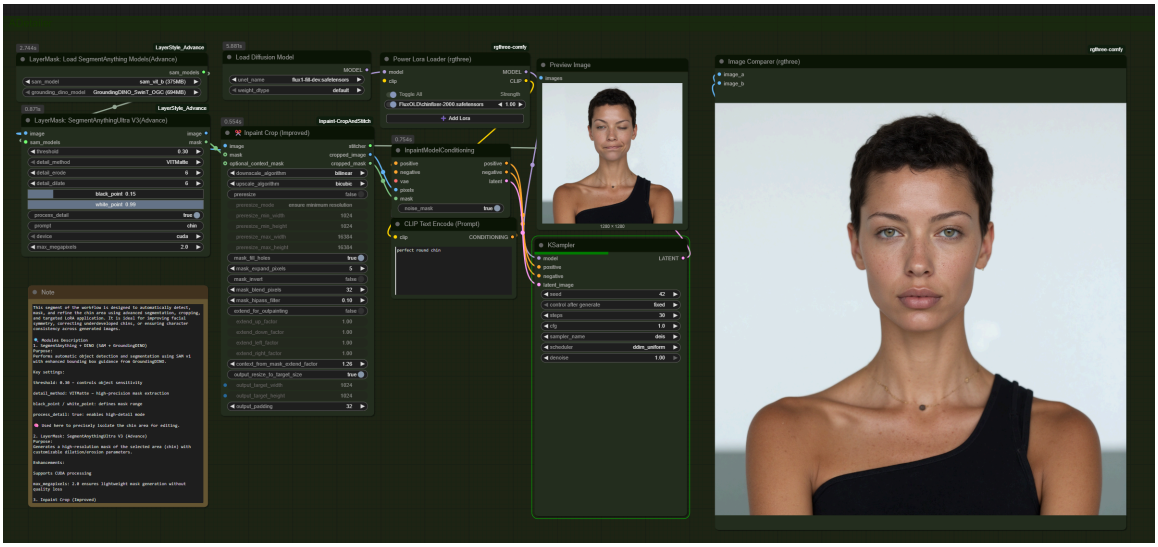
1. Generate a neutral portrait with *sample_image* disabled.
2. Select the Expression Editor node → press **Ctrl +B to enable sampling.**
3. Drop “Image Expression editor (1).png” into the loader.
4. Set `sample_parts = OnlyExpression` if you want to keep original head pose.
5. Render → check **Preview Image widget; adjust `smile` or `wink` sliders to taste.**
6. Deactivate the node (Ctrl +B again) to compare with the base neutral face.

Group 7 — Advanced Region Refinement



This block delivers two heavy-duty retouch stages: a *chin optimiser* crop-inpaint pass, and a multi-zone **Auto-Detailer** for face, eyes, hands, and hair.

8.1 Chin Optimiser — Inpaint Crop (Stitch)



SegmentAnything → **InpaintCropAndStitch** isolates the jawline, renders a dedicated 1024^2 fill, then stitches the patch back with alpha blending.

VRAM alert — the extra crop doubles memory. GPUs with **< 12 GB** may stall or swap; if that happens, select the nodes and press **M** to mute the block.

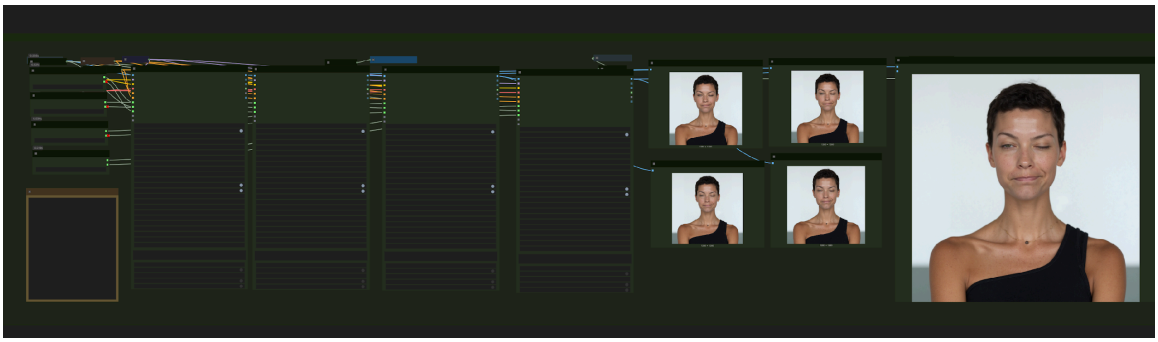
Key dials

Field	Typical	Comment
threshold	0.30	SA object sensitivity
detail_method	ViTMatte	High-precision matte
detail_dilate	6 px	Prevents halos

Field	Typical	Comment
bp/wp	0.15 / 0.99	Matte range
extend_for_upscaling	false	Keep 1:1 crop
mask_extend_factor	1.26	Seamless stitch

Tip → keep denoise = 1.0 inside the chin pass, then lower the global *Denoise* knob in Group 3 to preserve identity.

8.2 Auto-Detailer — Face / Skin / Eyes / Hands / Hair



Powered by **comfyui-impact-pack**: YOLOv8 bbox detectors + SAM segmentation feed inpaint masks to individual refiners (Occhi, faccia, capelli, etc.).

Model checklist

Model file	Purpose
sam_vit_b_01ec64.pth	SegmentAnything backbone
GroundingDINO_SwinT_COG.pth	Text-aware detector
bbox/hand_yolov8s.pt	Hand detector
bbox/Eyeful_v2-Paired.pt	Eye detector
bbox/face_yolov8m.pt	Face detector
segm/hair_yolov8n-seg_60.pt	Hair segmentation

Place them under models/impact/.

Recommended settings

guide_size 512, max_size 1024

steps 20–30, cfg 1.0–1.5, sampler Euler/DPM++

denoise 0.10 – 0.20 ← most critical

noise_mask + force_inpaint **enabled**

Thresholds: bbox_threshold 0.50, sam_threshold 0.93

bbox_crop_factor 2.5 – 3.0, feather 5–10 px

⚠️ *Raising denoise above 0.40 will rebuild the entire region and erase likeness.*

8.3 Quick workflow

Render the base portrait.

Enable Chin block if jaw symmetry needs a fix; monitor VRAM.

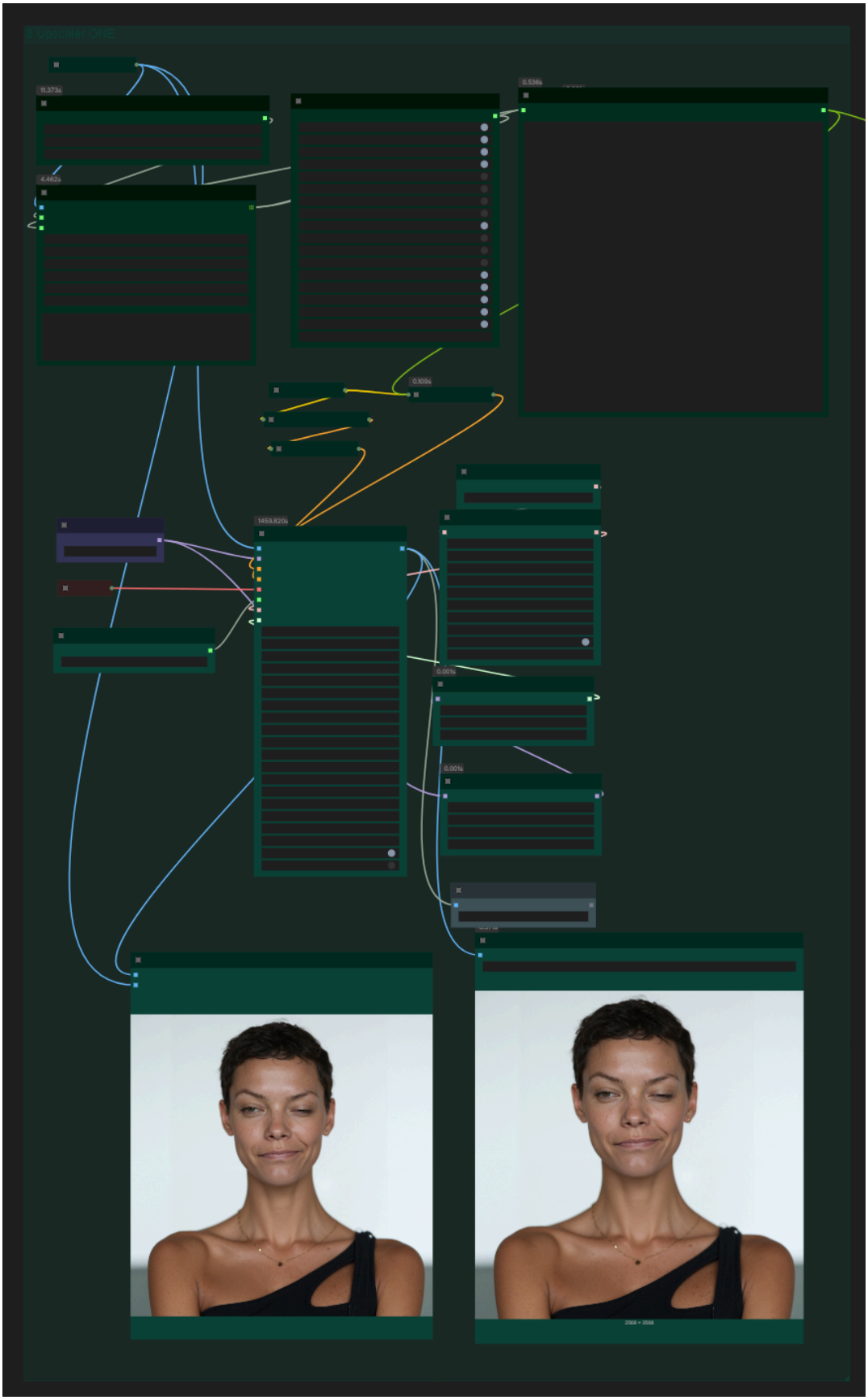
Tick the Auto-Detailer nodes you require (face, eyes, hands, hair).

Keep denoise ≈ 0.15 and render 20 steps.

Inspect results with **Image Comparer**; tweak thresholds.

Disable unused detectors to save time.

Group 8 — Upscaler ONE


Upscaler ONE

Two stages: **(1) JoyCaption image-caption LLM, (2) Ultimate SD Upscale latent enlarger**.
Stage 1 creates a neutral, token-dense description of the render (handy for archiving or second-pass prompting). Stage 2 lifts the portrait to 2×/4× while preserving skin fidelity.

9.2 Stage 1 — JoyCaption Beta One

Node	Purpose	Key fields
Load JoyCaption Beta One Model	Pulls fancyeast/llama-joycaption-beta-one-hf-llava (set quantization_mode = nf4 for lower VRAM).	—
JoyCaption Beta One	Generates the caption.	caption_type, caption_length, max_new_tokens, top_p, temperature, user_prompt
JoyCaption2 Extra Options	Fine toggles: • refer_character_name – inject LoRA trigger words. • include_camera_angle, specify_lighting_sources – richer vocabulary.	Boolean flags

Paste a template into **user_prompt** to steer the prose:

Generic Visual Description – highly neutral
Generic Portrait on White – studio vocabulary

Add the model/LoRA trigger inside character_name to fully anchor identity.

9.3 Stage 2 — Ultimate SD Upscale

Dial	Recommended	Note
upscale_by	**2.0×**	2048^2 from 1024^2
steps	30 (8× needs 40)	—
denoise	**0.20–0.50**	0.20 = faithful, 0.50 = creative

Dial	Recommended	Note
tile_width / height	1280	Keep multiple of 64
mask_blur	64 px	Soft seams

Upscale model zoo

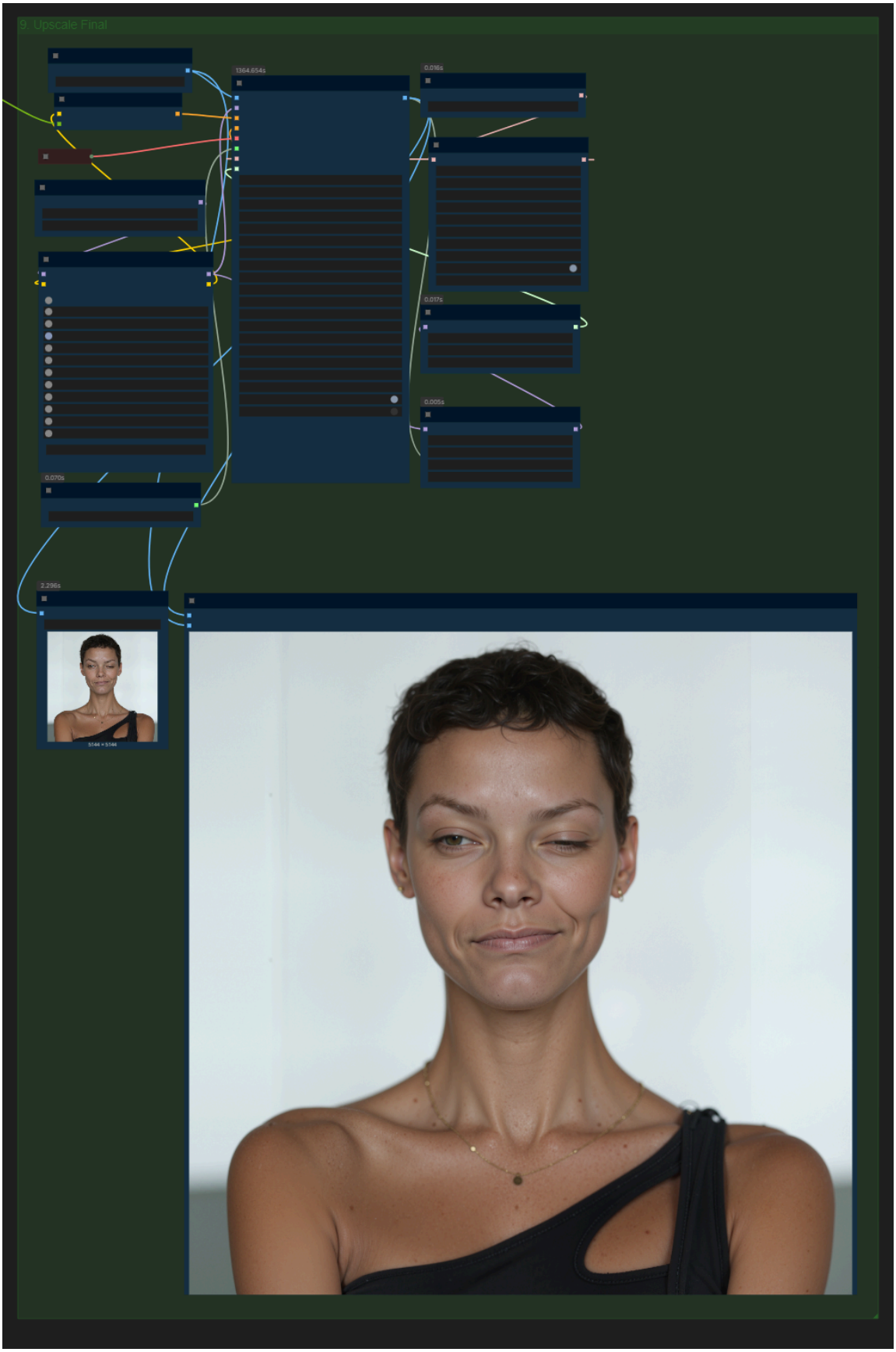
Model file	Use case	Source
4xFaceUpsharpDAT.pth	Balanced face sharpener	HuggingFace “FaceUpsharpDAT”
8x_NMKD-Faces_1.6M_G.pth	Aggressive 8× for tight crops	Civitai #96008
4xNomos8kSCHAT-L.pth	Mild sharpening, 8 K texture	GitHub Nomos Upscalers

Place models under **models/upscale_models/** and restart ComfyUI.

9.4 Quick workflow

1. Caption the base render → copy text.
2. (Optional) Feed caption into a second **CLIP Text Encode** for refined pass.
3. Load *4xFaceUpsharpDAT*, upscale_by 2, denoise 0.25, steps 30.
4. Compare results; if soft, swap to *8x_NMKD-Faces* and push denoise 0.35.

Group 9 — Upscale Final


9. Upscale Final
1364.654s
0.016s
0.017s
0.005s
0.070s
2.296s
5144 × 5144

The last block re-upsamples the portrait **and** lets you re-inject a bespoke FLUX-LoRA stack so you can polish skin, fabrics, or colour at the final resolution.

Purpose & context

Ultimate SD Upscale enlarges the latent to 2 × (or 4 × / 8 × with heavier models) and runs a short diffusion pass.
By loading **flux1-dev.safetensors** plus any LoRAs you want to strengthen, you confine stylistic pushes to the high-res canvas—leaving the earlier render clean and fast.

Node cheat-sheet

Node	Role	Key controls	Notes
Load FLUX original Model	Base UNet	flux1-dev.safetensors	Swap to *Flux1-Redux* for softer tonal roll-off
FLUX LoRA's Loader	Per-LoRA strength	sliders 0-1	Keep total ≤ 1.0
Load Upscale Model	ESRGAN kernel	e.g. *4xFaceUpsharpDAT.pth*	Put in *models/upscale_models*
Ultimate SD Upscale	Latent up-res	upscale_by, denoise, seam_fix_mode, tiling	See dials below
KSamplerSelect	Sampler	euler / dpm++	Match base render
Detail Daemon	Micro-detail	detail_amount 0.30	Raise to 0.35 for 8×
BasicScheduler	Sigma curve	steps 15, denoise 0.27	Overrides Upscale sigmas
ModelSamplingFlux	Tile jitter	max_shift 1.15	Hides grid artefacts

---

Critical dials

- **denoise 0.25 – 0.35**
  0.25 keeps likeness • *0.35* adds creative sheen

- steps 30 for 4× (40 for 8×)
- seam_fix_mode
  None → fastest • *Halftile* → softer edges • *Halftile + intersections* → no grid, slight blur
- Increase mask_blur to 64 px before switching seam-fix; often that's enough.

Recommended upscale kernels

File	Use case
4xFaceUpsharpDAT	Balanced, pore-sharp portraits
8x_NMKD-Faces_1.6M_G	Extreme zoom-ins, heavier VRAM
4xNomos8kSCHAT-L	Soft magazine gloss

Download from HuggingFace / Civitai / Nomos GitHub and drop into **models/upscale_models/**, then restart ComfyUI.

Quick workflow

1. Decide what to amplify (e.g., *RealisticSkin* 0.75, *VintageFilm* 0.3).
2. upscale_by 2, denoise 0.28, steps 30, seam-fix *None*.
3. Render and inspect at 200 %.
4. If grid shows → set seam-fix to *Halftile + intersections* or raise mask_blur.
5. For extra style, push LoRA strengths and bump denoise to 0.33.

⚠️ **Performance note** – running Upscaler ONE *and* Upscale Final doubles VRAM and time. On 12 GB GPUs lower tile_width/height to 1024 and enable tiled_decode if you hit OOM.

Group 10 — LayerStyle Color Pipeline (Exposure → Grain → Contrast → LUT)

This last chain is pure, non-destructive color work. Nothing happens in latent space any more — each node touches the final RGB frame, so you can toggle them on/off without re-running diffusion.

Node	Function	Practical dial-in
LayerColor : Exposure	Global lift / drop (–1 → +1 EV)	Leave at 0 unless your base render is obviously under / over
LayerFilter : Add Grain	Analogue noise overlay	grain_power 0.20 • grain_scale 0.40 • grain_sat 0.75
LayerColor : Brightness Contrast V2	Fine tone balance	brightness 1.00 • contrast 1.10 • saturation 0.95
LayerColor : LUT Apply	Drops a *.cube* LUT for full film-style color	Fuji_Astia.cube • color_space linear • strength 50 %

Every block is wrapped in a **Fast Bypasser** — flip the toggle if you want quick A/B comparisons.