

Workflow SwarmUI/ComfyUI pour images et vidéos en silhouettes réalistes (ombres chinoises)

1. Mise à jour de SwarmUI et support des modèles vidéo récents

Pour commencer, assurez-vous d'utiliser la **dernière version de StableSwarmUI** (SwarmUI) incluant la mise à jour de ComfyUI. SwarmUI s'appuie sur ComfyUI en back-end ([A Stable Diffusion 3 Tutorial With Amazing SwarmUI SD Web UI...](#)), ce qui permet de bénéficier des nouveautés de ComfyUI. Les récents modèles de génération vidéo (text-to-video et image-to-video) sont désormais pris en charge via des *custom nodes*. En particulier, le modèle ouvert d'Alibaba souvent associé à **Qwen** – connu sous le nom de **Wan 2.1** – est intégré et recommandé dans SwarmUI pour la vidéo. Ce modèle Wan 2.1 est actuellement **le plus performant** en génération vidéo, surpassant nettement les alternatives ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)). SwarmUI supporte officiellement Wan 2.1 pour des workflows texte→vidéo ou image→vidéo grâce à des nodes ComfyUI dédiés ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)).

Côté installation, il faudra : **(a)** mettre à jour SwarmUI/ComfyUI, **(b)** installer les extensions de nœuds vidéo, et **(c)** télécharger les poids des modèles vidéo. Via le **ComfyUI Manager**, installez par exemple l'extension **LTX Video** (pour le modèle LTX-V0.9, ~2 milliards de paramètres) et les wrappers pour Wan 2.1. Ensuite, téléchargez les fichiers du modèle Wan 2.1 depuis HuggingFace (versions repackagées pour ComfyUI) et placez-les dans le répertoire adapté (voir section 4) ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)). *NB*: Certains modèles vidéo open source supplémentaires existent (p. ex. **Hunyuan Video** de Tencent, ~12 Mds de paramètres, ou **Genmo Mochi**), mais d'après la documentation de SwarmUI le modèle **Wan 2.1 surpasse actuellement tous les autres** en qualité ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)). Hunyuan offre une qualité décente mais inférieure ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)), tandis que LTX Video est très rapide mais sa qualité d'image est jugée inférieure (« *fast but ugly* ») ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)). Pour une utilisation optimale, privilégiez donc Wan 2.1 (petite version) afin d'obtenir les meilleurs résultats vidéo. SwarmUI étant modulable, ces modèles vidéo peuvent être utilisés soit via l'onglet *Generate* (texte→vidéo) soit via des workflows ComfyUI personnalisés (image→vidéo avec contrôle d'images clés).

2. Checkpoints SDXL et LoRA pour un style silhouette réaliste

Pour générer des **images fixes** au style photoréaliste « ombre chinoise », il est recommandé d'utiliser **Stable Diffusion XL (SDXL)** comme base, éventuellement une variante fine-tunée pour le réalisme. Un excellent choix de checkpoint SDXL est par exemple **Juggernaut XL** – un modèle SDXL ayant fait ses preuves pour la photoréalité et la fidélité aux prompts ([The 11 Best SDXL Models on Civitai](#)). Ce modèle merge est connu pour produire des images spectaculaires et détaillées en restant polyvalent (portraits, paysages, etc.) ([The 11 Best SDXL Models on Civitai](#)), ce qui en fait une base robuste. À défaut, le checkpoint SDXL 1.0 officiel (de Stability AI) couplé à son *Refiner* peut aussi fonctionner pour affiner les détails – mais tenez compte de la VRAM disponible (voir section 4).

En complément du checkpoint principal, il faut intégrer un **LoRA** spécialisé pour le style *silhouette*. Un LoRA est un petit modèle qui oriente le rendu vers un style particulier sans changer le checkpoint de base. Sur HuggingFace, on trouve par exemple « *SDXL LoRA – silhouette* » (par NTC AI) conçu pour cet effet ([ntc-ai/SDXL-LoRA-slider.silhouette · Hugging Face](#)). Ce LoRA s'active avec le mot-clé « *silhouette* » dans le prompt ([ntc-ai/SDXL-LoRA-slider.silhouette · Hugging Face](#)) et va forcer le modèle à produire des sujets en contre-jour, sans détails faciaux, avec un contour lumineux (rim light). En utilisant un checkpoint photoréaliste couplé à ce LoRA, on obtient le mélange idéal : **décors réalistes + sujets en ombre chinoise**. N'hésitez pas à expérimenter plusieurs LoRAs ou merges disponibles sur CivitAI/HuggingFace – par exemple des LoRAs de lighting/cinématographie – mais celui cité est un bon point de départ spécifiquement pour les silhouettes. Enfin, peaufinez le *prompt* pour renforcer le style : décrivez la scène avec des termes de lumière appropriés (« *contre-jour, lumière dorée de fin d'après-midi, silhouette noire détaillée uniquement par les contours lumineux, ombre projetée nette au sol* »). Ces mots-clés (backlight, rim lighting, etc.) aideront Stable Diffusion à comprendre l'effet recherché ([3 ways to control lighting in Stable Diffusion - Stable Diffusion Art](#)) ([3 ways to control lighting in Stable Diffusion - Stable Diffusion Art](#)). En négatif, ajoutez des termes pour exclure les visages visibles ou les artefacts (« *pas de visage, pas de détails intérieurs du sujet, pas de bruit* »), afin de bien obtenir des personnages entièrement dans l'ombre.

3. Alternatives open source à DALL·E 3 pour image-to-image

DALL·E 3 étant un service propriétaire, il convient de le remplacer par une solution open source auto-hébergeable. La principale alternative est d'utiliser **Stable Diffusion** (notamment SDXL) avec des outils de contrôle avancés. SDXL offre déjà une qualité d'image élevée et, couplé à des modules comme **ControlNet**, il permet de faire de l'**image-to-image** en imposant une structure ou un style spécifique ([Top Open-Source AI Alternatives For Beginners | Restackio](#)). Par exemple, on peut fournir à SDXL une image de référence (un dessin de silhouette issu du storyboard) et, via ControlNet, contraindre la génération pour qu'elle suive précisément les contours tout en produisant un rendu réaliste. Cette approche offre un contrôle du style équivalent à DALL·E 3, tout en restant open source.

En pratique, vous pouvez déployer **ComfyUI** ou **Automatic1111** en local, qui exposent des API HTTP pour générer des images à la demande. Dans votre cas, ComfyUI est déjà utilisé via SwarmUI – il suffit donc d’exploiter son API (comme le fait le script Python fourni) pour automatiser des requêtes image→image. Une autre option émergente est l’outil **Ollama** – connu pour gérer des modèles IA localement via des appels simples. Ollama se concentre surtout sur les modèles de langage, mais il commence à intégrer certains modèles de diffusion d’images d’après la communauté, ce qui pourrait à terme fournir une interface unifiée pour SDXL via API. Quoi qu’il en soit, l’écosystème Stable Diffusion reste la solution open source la plus viable : elle bénéficie d’une forte communauté et de nombreuses améliorations. D’ailleurs, les modèles open source progressent rapidement – par exemple Stability AI évoque un futur *Stable Diffusion 3* open-source qui, même s’il n’égale pas encore DALL·E 3, servira de base à de nouvelles innovations collaboratives ([Top Open-Source Ai Alternatives For Beginners | Restackio](#)). En résumé, **Stable Diffusion XL + ControlNet/LoRA** constitue aujourd’hui l’alternative libre la plus efficace à DALL·E 3 pour de la génération d’images réalistes avec contrôle du style, le tout accessible via une API locale.

4. Arborescence propre des modèles sur la machine (Titan X 12 Go)

Pour garder une installation ordonnée, suivez la structure de dossiers de modèles préconisée par ComfyUI/SwarmUI. Dans le répertoire `stable-swarmui` (ou équivalent) vous devriez avoir un dossier **Models** (ou `models`). Organisez-y les sous-dossiers comme suit :

- **checkpoints/** – pour les modèles principaux de diffusion (checkpoints SDXL, mais aussi les modèles vidéo du type Wan 2.1 ou LTX, car SwarmUI les range également parmi les modèles de diffusion).
- **Lora/** – pour tous les fichiers LoRA (`.safetensors`) que vous utiliserez (ex: le LoRA silhouette).
- **ControlNet/** – pour les modèles ControlNet ou assimilés (ex: `control_canny.safetensors`, `control_depth.safetensors`, etc. si vous en utilisez pour guider la pose ou les contours).

Vous pouvez également avoir d’autres dossiers (`VAE/` pour d’éventuels VAE custom, `embeddings/` pour des embeddings de texte, etc.), mais les trois ci-dessus couvrent l’essentiel. En adoptant cette arborescence claire (`Models/checkpoints`, `Models/Lora`, `Models/ControlNet`), SwarmUI détectera automatiquement les modèles disponibles et vous pourrez facilement les sélectionner dans l’interface. Veillez à nommer les fichiers de façon explicite (par ex. `“sdxl_JuggernautXL.safetensors”`, `“lora_silhouette.safetensors”`, `“control_depth.safetensors”`).

Adaptation à la Titan X 12 Go – La NVIDIA Titan X Pascal dispose de 12 Go de VRAM, ce qui est suffisant pour SDXL en résolution standard, mais il faut éviter les modèles trop lourds. Par exemple, pour la vidéo Wan 2.1, utilisez la version *1.3B paramètres* plutôt que la version *14B* : la petite version tient sur ~10 Go et fonctionne même sur des « petites » cartes ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)), alors que la 14B exigerait près de 10× plus de VRAM (typiquement une RTX 3090/4090) ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)) – donc hors de portée de la Titan X. De même, restez en précision float16 ou int8 pour les modèles vidéo afin d'économiser la VRAM (SwarmUI peut charger Wan 2.1 en FP8 automatiquement) ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)). Si un modèle dépasse votre VRAM, sachez qu'il existe parfois des versions quantifiées **GGUF/ggml** plus légères ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)), mais au prix d'une génération plus lente. Étant donné que vous avez 128 Go de RAM système, vous pouvez aussi activer le *swap* CPU (« low VRAM mode ») dans ComfyUI pour les tâches dépassant légèrement le GPU – mais cela ralentira le processus. En résumé, privilégiez les modèles optimisés (SDXL base ~7 Go, Wan 1.3B, etc.) qui correspondent aux 12 Go de la Titan X, et tout fonctionnera de manière fluide.

5. Sources fiables pour télécharger les modèles

Il est **essentiel de télécharger les checkpoints et LoRA depuis des sources fiables** pour éviter les fichiers corrompus ou malveillants. Trois sources sont particulièrement recommandées : **CivitAI**, **Hugging Face** et **Ollama**.

- **CivitAI** : c'est la plateforme communautaire principale pour les modèles Stable Diffusion. Vous y trouverez de nombreux checkpoints SDXL photoréalistes (comme Juggernaut XL, DreamShaper XL, etc.) ainsi que des LoRA pour les styles spécifiques. Les utilisateurs y publient des exemples et notes, ce qui aide à évaluer la qualité. Par exemple, Juggernaut XL y est très populaire et bien noté pour sa qualité ([The 11 Best SDXL Models on Civitai](#)). Téléchargez toujours les versions vérifiées (le site indique souvent si le modèle a été scanné).
- **Hugging Face** : c'est la référence pour les dépôts officiels ou open source. Stability AI y héberge le modèle SDXL officiel. De plus, de nombreux modèles vidéo complexes comme Wan 2.1 ou Hunyuan Video sont disponibles via HuggingFace (souvent dans des dépôts officiels ou repackagés pour ComfyUI). Dans la documentation SwarmUI, des liens HuggingFace sont fournis pour Wan 2.1 ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)) – utilisez ces dépôts officiels pour obtenir les fichiers **.safetensors**. HuggingFace offre des checksums et est un hébergeur de confiance.
- **Ollama** : bien que surtout connu pour les LLM, Ollama propose un catalogue de modèles IA téléchargeables via leur outil en ligne de commande. S'ils offrent des modèles de diffusion d'images, ce sera une source à considérer. L'avantage est que les modèles Ollama sont packagés proprement et généralement éprouvés, ce qui peut simplifier l'intégration. Gardez toutefois en tête que la communauté SD se

concentre sur CivitAI/HuggingFace, donc pour les derniers modèles de génération d'images et vidéos, ces derniers resteront vos sources de premier choix.

En résumé, **CivitAI** pour tout ce qui est *fine-tunes* et LoRAs communautaires, et **HuggingFace** pour les modèles de base (SDXL) et modèles vidéo avancés – en priorisant les comptes officiels (Stability AI, ComfyUI, Alibaba, etc.) et les créateurs reconnus. Ces sources garantissent d'obtenir les bonnes versions compatibles avec votre pipeline SwarmUI.

6. Configuration prête à l'emploi dans SwarmUI

L'objectif est de préparer un **workflow clé-en-main** dans SwarmUI pour générer images et vidéos avec une cohérence artistique immédiate, c'est-à-dire sans avoir à affiner manuellement chaque paramètre à chaque fois. Pour cela, on va configurer un *workflow* ComfyUI pré-paramétré pour le style *ombre chinoise réaliste*.

Dans SwarmUI, rendez-vous dans l'onglet *Comfy Workflow* (ou *Simple*). Vous pouvez importer un fichier de workflow JSON prédéfini (par exemple `workflow_ombre_chinoise.json` s'il a été fourni). Ce workflow doit charger automatiquement le bon **checkpoint SDXL** (via un nœud *Checkpoint Loader*), appliquer le **LoRA silhouette** (nœud *Load LoRA* avec poids défini, e.g. 1.0 ou 1.5), et inclure éventuellement un **ControlNet** configuré. Concrètement, le workflow pourrait avoir une structure comme : *Input Image* (optionnel) → *ControlNet (canny/pose)* → *SDXL Text-to-Image* (ou *Image-to-Image*) avec LoRA → *SDXL Refiner* (optionnel) → *Sortie Image*; et de même une branche pour la vidéo : *Image Initiale* → *Wan 2.1 I2V* → *Sortie Vidéo*. L'idée est que toutes les composantes du style sont déjà en place.

Images fixes : si vous souhaitez générer une image unique, le workflow utilisera SDXL. Il suffit alors d'éditer le *prompt texte* pour décrire la scène souhaitée (en restant dans le ton silhouette), d'ajouter ou non une image de référence (par exemple un tracé de silhouette en entrée du ControlNet pour guider la pose), puis de lancer la génération. Grâce au LoRA et aux réglages prédéfinis (lumière dorée, etc.), **l'image produite aura directement le bon style** sans itérations supplémentaires. Vous pouvez bien sûr ajuster des paramètres si besoin (intensité du LoRA, échelle de guidance CFG, etc.), mais en principe le rendu devrait correspondre à l'esthétique *ombres chinoises cinématographiques* dès le premier essai.

Vidéos : pour générer une vidéo cohérente, deux approches sont possibles. La plus simple est d'utiliser le modèle Wan 2.1 en mode texte→vidéo avec le même prompt que pour l'image (en décrivant un mouvement ou une scène animée). Toutefois, pour une meilleure cohérence artistique, il est souvent utile de **partir d'une image clé**. Avec SwarmUI, vous pouvez d'abord générer une image fixe de la scène (comme storyboard visuel), puis la passer dans le module *Image2Video*. Le workflow prêt-à-l'emploi doit faciliter cela : par exemple, un nœud *Init Image* alimentera Wan 2.1 (I2V) pour créer la séquence animée à partir de cette image. Configurez le nombre de frames, la durée (par ex. 5 secondes à 15 fps = 75 frames), et lancez la génération vidéo. Wan 2.1 va alors produire un clip en conservant l'esthétique de l'image de départ (silhouettes en contre-jour, etc.) sur toute la séquence. Assurez-vous d'utiliser le *même prompt* que pour l'image initiale afin que le style et

l'environnement restent cohérents entre l'image et la vidéo. Avec cette configuration, vous pouvez manuellement générer différentes scènes de la série *Déclics* tout en gardant une unité artistique (même éclairage, mêmes rendus de silhouettes).

En somme, la configuration prête à l'emploi consiste à **pré-charger le bon modèle, le LoRA de style, et éventuellement un ControlNet**, de sorte que l'utilisateur n'ait plus qu'à saisir son idée (prompt et/ou dessin de silhouette) et à cliquer sur *Générer*. Cela garantit une **cohérence immédiate** dans le style visuel, d'une image à l'autre et d'une image à la vidéo. N'hésitez pas à sauvegarder plusieurs workflows pour différents cas (par ex. un workflow silhouette sans image de référence, un autre avec ControlNet pose, un pour les vidéos...) afin de passer de l'un à l'autre facilement selon les besoins de la production.

7. Bonus : modèles ou nœuds pour renforcer le style « ombre chinoise »

Pour aller encore plus loin dans le respect strict du style silhouette, quelques techniques et outils supplémentaires méritent d'être mentionnés. Tout d'abord, le **LoRA Silhouette** demeure l'outil central – il a été entraîné spécifiquement sur des images en ombres chinoises et donc oriente fortement la génération vers ce rendu caractéristique. En augmentant son poids (par exemple 1.5 ou 2.0 au lieu de 1.0), vous pouvez obtenir des silhouettes encore plus marquées. À l'extrême, un poids très élevé foncera totalement les sujets et ne laissera apparaître que les contours lumineux.

([ntc-ai/SDXL-LoRA-slider.silhouette · Hugging Face](#)) Exemple d'image générée avec un LoRA silhouette sur SDXL (sujet entièrement noir en contre-jour d'un coucher de soleil). Le LoRA spécialisé permet d'obtenir directement ce fort contraste et la lisibilité du contour, sans détails internes.

Ensuite, pensez à exploiter **ControlNet** pour verrouiller certains aspects visuels. Par exemple, en utilisant le modèle *ControlNet Canny*, vous pouvez prendre un dessin de silhouette (même sommaire) et en extraire les bords pour guider Stable Diffusion. Le modèle suivra alors exactement ces bords pour générer le sujet, assurant que la silhouette reste conforme au storyboard tout en remplissant l'intérieur de noir et en créant un fond réaliste. De même, un ControlNet *Depth* pourrait être détourné en fournissant une « carte de profondeur » binaire (sujet proche vs fond lointain) – cela obligerait le rendu à distinguer nettement le plan du sujet (à noircir) du plan du fond. Combinés, ces outils donnent un contrôle quasi-total : **dessinez la forme de l'ombre**, et l'IA la respecte à la lettre dans un style photoréaliste. SwarmUI facilite l'utilisation de plusieurs ControlNets si besoin, via des nœuds *ControlNetLoader* multiples branchés sur le modèle SDXL.

Enfin, n'oubliez pas le **travail de prompt** : certains mots-clés sont vos alliés pour ce style. Nous avons mentionné *backlight* (contre-jour) et *rim lighting* (lumière de bordure) qui sont cruciaux ([3 ways to control lighting in Stable Diffusion - Stable Diffusion Art](#)) ([3 ways to control lighting in Stable Diffusion - Stable Diffusion Art](#)). Vous pouvez aussi jouer avec des termes comme “*silhouette photography*”, “*golden hour silhouette*”, “*sunset shadows*” dans le prompt – ces expressions déclenchent souvent des biais dans le modèle vers des images très contrastées où le sujet est noir sur fond lumineux. Inversement, utilisez abondamment

le **negative prompt** pour bannir toute lumière frontale ou détail non désiré (par ex : “*no fill light, no facial detail, no interior detail, only outline*”). Il existe également des **extensions de contrôle de l'exposition** (par ex. des scripts de *denoising offset* ou LoRA d'éclairage HDR) qui peuvent aider à obtenir des noirs plus profonds sans perdre le contraste du pourtour – toutefois, dans la plupart des cas, un simple mot-clé “*dimly lit*” ou “*dark silhouette*” suffit à indiquer au modèle de ne pas éclairer le sujet ([3 ways to control lighting in Stable Diffusion - Stable Diffusion Art](#)).

En résumé, pour un style *ombre chinoise* irréprochable : combinez un bon checkpoint réaliste, un LoRA entraîné sur les silhouettes, un prompt judicieux (lumière de dos, etc.), et éventuellement un ControlNet pour figer la forme du sujet. Avec cette panoplie, Stable Diffusion respectera scrupuleusement l'esthétique *silhouette chinoise réaliste*, que ce soit pour des images isolées ou des séquences animées, et ce avec une intervention manuelle minimale. Bonne création pour *Déclics* !

Sources : SwarmUI/ComfyUI docs sur les modèles vidéo ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)) ([SwarmUI/docs/Video Model Support.md at master · mcmonkeyprojects/SwarmUI · GitHub](#)); recommandations de checkpoints SDXL ([The 11 Best SDXL Models on Civitai](#)); HuggingFace (LoRA silhouette) ([ntc-ai/SDXL-LoRA-slider.silhouette · Hugging Face](#)); Stable Diffusion lighting tips ([3 ways to control lighting in Stable Diffusion - Stable Diffusion Art](#)) ([3 ways to control lighting in Stable Diffusion - Stable Diffusion Art](#)); Restack (Open-source vs DALL-E 3) ([Top Open-Source Ai Alternatives For Beginners | Restackio](#)) ([Top Open-Source Ai Alternatives For Beginners | Restackio](#)); et autres documents cités.

Expliquant l'intégration de ComfyUI

Le processus pour intégrer un workflow dans ComfyUI est assez direct. D'abord, placez le fichier JSON dans le répertoire approprié. Ensuite, ouvrez ComfyUI et chargez le workflow depuis l'interface. Assurez-vous que les modèles sont dans les bons dossiers, puis configurez les variables d'environnement nécessaires. Si vous travaillez dans un environnement Python, installez toutes les dépendances avant de lancer ComfyUI. Enfin, ouvrez le workflow, réglez les variables des espaces réservés et lancez-le. Cette approche doit être assez claire et pratique.

Réflexion durant quelques secondes

Voici les étapes **précises** pour intégrer et exécuter le **workflow JSON** dans ComfyUI, en partant du principe que ComfyUI est déjà installé sur ta machine WSL/GPU :

1. Préparer ton environnement ComfyUI

1. **Vérifie que ComfyUI est installé** (si tu as cloné le repo SwarmUI, c'est déjà fait).

Positionne-toi dans le dossier racine de ComfyUI :

```
cd ~/swarmui/ComfyUI
```

2.

Active ton venv (si tu utilises un environnement virtuel) :

```
source venv/bin/activate
```

3.

Installe-le si besoin :

```
pip install -r requirements.txt
```

4.

2. Placer les modèles

Crée cette arborescence **depuis la racine ComfyUI** :

```
ComfyUI/
├── models/
│   ├── checkpoints/
│   │   ├── sdxl_base_1.0.safetensors
│   │   ├── sdxl_refiner_1.0.safetensors
│   │   ├── juggernaut_xl.safetensors
│   │   └── RealisticVisionXL.safetensors
│   ├── lora/
│   │   ├── silhouette_photographic_style_lora.safetensors
│   │   └── rimlight_lora.safetensors
│   └── controlnet/
│       ├── diffusion_xl_depth.safetensors
│       ├── control_sd_xl_canny.safetensors
│       └── diffusion_xl_openpose.safetensors
```

3. Placer le workflow JSON

Copie le fichier `Windsurf_ComfyUI_Workflow_Template.json` dans un dossier accessible, par exemple :

```
ComfyUI/workflows/Windsurf_Template.json
```


- 1.
 2. Tu peux créer ce dossier `workflows/` si tu préfères organiser tes workflows.
-

4. Lancer ComfyUI

Depuis le terminal (toujours dans le venv) :

```
python main.py
```

- 1.
 2. Par défaut ComfyUI ouvre une page web sur `http://localhost:7800` (ou 7801).
-

5. Importer le workflow

1. Dans l'interface web, clique en haut à gauche sur **File** → **Load workflow**.
 2. Navigue jusqu'à `ComfyUI/workflows/Windsurf_Template.json` et ouvre-le.
 3. Le graphe des nœuds apparaît maintenant dans la zone centrale.
-

6. Configurer les « placeholders »

Le workflow utilise des variables dynamiques que tu dois renseigner avant de lancer la génération :

- `${storyboard_image_path}` : chemin absolu vers ton image de storyboard (ex. `/home/charly/Declics/S02/SQ0010-0010.png`).
- `${prompt_text}` : ton prompt principal (le texte issu de la fiche technique, adapté à la scène).
- `${negative_prompt}` : ton negative prompt.
- `${output_path}` : chemin complet vers le fichier de sortie (ex. `/home/charly/Declics/S02/E202_SQ0010-0010_Concept_v0001.jpg`).

- **`${seed}`** (optionnel) : si tu veux reproduire exactement la même image, passe un nombre fixe.

Pour remplir :

1. Clique sur le nœud **LoadImage** → onglet **Params** → remplace la valeur de **path** par ton storyboard.
 2. Clique sur **Prompt** → colle tes textes dans **positive** et **negative**.
 3. Clique sur **SaveImage** → dans **path**, colle ton nom de fichier complet avec la nomenclature Windsurf.
-

7. Ajuster les paramètres GPU

1. Dans le menu latéral **Settings** :
 - **CUDA Device** : choisis **TITAN X Pascal**.
 - Coche **Use low VRAM mode** si tu veux activer le swap CPU automatique.
 2. Vérifie que la résolution du sampler est bien **1920×1080** (ou ton fallback en 768×432 si tu passes par le upscaler).
-

8. Exécuter le workflow

1. Clique sur le nœud **KSampler** (ou sur un bouton **Run** en bas, selon ta version).
 2. Tu peux lancer soit un unique plan, soit un batch si tu as plusieurs placeholders.
 3. Vérifie la console pour suivre l'avancement et détecter d'éventuels messages d'erreur.
-

9. Récupérer tes images

Les fichiers seront générés automatiquement au chemin `${output_path}` que tu as spécifié.

- Si tu utilises la méthode **placeholder**, ils écrasent directement les fichiers vides créés par Windsurf.
 - Sinon, tu peux parcourir le dossier de sortie et renommer ou valider manuellement.
-

Bonnes pratiques

- **Sauvegarde ton workflow** après avoir rempli un plan : **File** → **Save workflow**, pour conserver tes réglages.
- Si tu changes de style lumineux, modifie simplement le texte du prompt ou crée plusieurs workflows (Golden, Blue, Neutre).
- Pour réutiliser le même storyboard sur plusieurs variantes, duplique le workflow et change uniquement le prompt et le `output_path`.