

GLOSARIO DE TÉRMINOS BÁSICOS

En este glosario, explicaremos algunos de los términos que aparecen comúnmente en interfaces de generación de imágenes basadas en modelos de difusión, como la que se muestra en la imagen.

- **CLIP TEXT ENCODE (PROMPT):** CLIP (Contrastive Language–Image Pre-training) es un modelo desarrollado por OpenAI que entiende la relación entre texto e imágenes. "CLIP Text Encode" se refiere al proceso de convertir el texto de tu "prompt" (la descripción de la imagen que quieres generar) en una representación numérica (un vector o *embedding*) que el modelo de difusión puede entender y utilizar para guiar el proceso de generación de la imagen. Hay dos entradas de prompt, una para el "positive prompt" (lo que quieres ver) y otra para el "negative prompt" (lo que *no* quieres ver).
- **CONDITIONING:** Este término se refiere a la información que se utiliza para guiar el proceso de generación del modelo de difusión. En el contexto de la generación de imágenes a partir de texto, el "conditioning" principal proviene de los *embeddings* generados por el "CLIP Text Encode" de tus prompts positivo y negativo. Otros elementos, como máscaras o información de control (como en ControlNet), también pueden ser parte del conditioning.
- **LOAD LORA:** LoRA (Low-Rank Adaptation) es una técnica para adaptar un modelo de lenguaje grande (o en este caso, un modelo de difusión) a una tarea o estilo específico con un entrenamiento mucho más eficiente que el *fine-tuning* completo. "Load LoRA" se refiere a cargar un archivo LoRA pre-entrenado que modifica el comportamiento del modelo base para generar imágenes con características particulares (por ejemplo, un estilo artístico específico, un personaje, etc.).
- **STRENGTH_MODEL:** En el contexto de cargar un LoRA, *strength_model* controla cuánto impacto tiene el LoRA cargado en el modelo base. Un valor de 0.0 significa que el LoRA no tiene ningún efecto, mientras que un valor de 1.0 aplica el efecto completo del LoRA. Valores intermedios mezclan el efecto del LoRA con el comportamiento original del modelo base.
- **STRENGTH_CLIP:** Similar a *strength_model*, *strength_clip* en el contexto de LoRA controla cuánto impacto tiene el LoRA en la parte del modelo relacionada con el entendimiento del texto (la parte CLIP). Ajustar este valor puede influir en cómo el LoRA responde a tus prompts.
- **MODEL:** Se refiere al modelo base de difusión que se está utilizando para generar la imagen (por ejemplo, Stable Diffusion 1.5, Stable Diffusion XL, etc.).
- **CLIP:** Se refiere a la parte del modelo que procesa y entiende el texto del prompt.
- **VAE ENCODE / VAE DECODE:** VAE (Variational Autoencoder) es una componente

de los modelos de difusión.

- **VAE ENCODE:** Convierte una imagen del espacio de píxeles (la imagen visible) a un espacio latente de menor dimensión. Esto se usa a menudo para trabajar con imágenes existentes (por ejemplo, para *inpainting* o *outpainting*).
- **VAE DECODE:** Convierte una representación del espacio latente de vuelta al espacio de píxeles para obtener la imagen final visible. Esto es el paso final después de que el KSampler ha terminado de refinar el ruido en el espacio latente.
- **LATENT:** Se refiere al "espacio latente", que es una representación comprimida y de menor dimensión de la imagen. Los modelos de difusión operan principalmente en este espacio latente porque es computacionalmente más eficiente que trabajar directamente con imágenes de alta resolución en el espacio de píxeles. El proceso de generación implica refinar progresivamente el ruido en este espacio latente hasta que se parece a lo que describe el prompt.
- **KSAMPLER:** Es el componente central que realiza el proceso de difusión. Toma el ruido inicial en el espacio latente, el conditioning (de los prompts, etc.) y, a través de una serie de pasos, elimina iterativamente el ruido para generar la imagen final en el espacio latente.
- **SEED:** Es un número entero que inicializa el generador de números aleatorios. Usar la misma semilla con los mismos parámetros garantiza que obtendrás exactamente la misma imagen de salida. Cambiar la semilla generará una imagen diferente, incluso si todos los demás parámetros son idénticos.
- **CONTROL AFTER GENERATE:** Esta opción, a menudo encontrada en samplers, controla qué sucede con la semilla después de cada generación (por ejemplo, si se fija, si se incrementa aleatoriamente, etc.).
- **STEPS:** El número de pasos que el KSampler realiza para refinar la imagen latente. Más pasos generalmente resultan en una imagen más detallada y fiel al prompt, pero tardan más tiempo.
- **CFG (CLASSIFIER-FREE GUIDANCE):** Controla cuánto debe seguir el modelo el prompt. Un valor de cfg más alto hace que el modelo se adhiera más estrictamente al prompt, pero valores demasiado altos pueden llevar a artefactos o imágenes de menor calidad.
- **SAMPLER_NAME:** El algoritmo específico utilizado por el KSampler para realizar el proceso de eliminación de ruido. Diferentes samplers (como Euler, DPM++ 2M Karras, etc.) tienen diferentes métodos y pueden producir resultados ligeramente diferentes o tener diferentes características de velocidad y eficiencia.
- **SCHEDULER:** Acompaña al sampler y define cómo se ajusta el "ruido" en cada paso del proceso de muestreo.
- **DENOISE:** Este parámetro, generalmente entre 0.0 y 1.0, controla cuánto ruido se

elimina de la imagen latente inicial.

- Si empiezas con ruido aleatorio (generación de texto a imagen), denoise suele ser 1.0 para eliminar todo el ruido y obtener una imagen coherente.
- Si empiezas con una imagen existente en el espacio latente (por ejemplo, para *image-to-image* o *inpainting*), un valor de denoise menor que 1.0 significa que el sampler no eliminará *todo* el ruido, permitiendo que la imagen generada retenga algunas características de la imagen original. Un denoise de 0.0 no cambiaría la imagen original en absoluto.
- **SAVE IMAGE:** El paso final para guardar la imagen generada desde el espacio de píxeles a un archivo (como PNG o JPEG).
- **EMPTY LATENT SPACE IMAGE:** Representa el punto de partida para la generación de una imagen a partir de texto (text-to-image). Es esencialmente un tensor lleno de ruido aleatorio en el espacio latente. El KSampler toma este ruido y, guiado por los prompts y otros parámetros, lo transforma gradualmente en la imagen deseada.

En general, estos suelen ser los términos necesarios para poder comenzar a trabajar con ComfyUI. En caso de necesitar más información, siempre podéis recurrir a internet, ya que ComfyUI posee una amplia comunidad muy explicativa.