

# Керування генерацією зображень: SDXL, DALL·E, ControlNet, LoRA

**Нейромережеві технології і системи**

Лектор: Лящинський П.Б.

# Stable Diffusion v1.5: революція у доступності

У серпні 2022 року Stability AI випустила Stable Diffusion v1.5 — першу відкриту high-quality text-to-image модель, що працює на пристрої користувача.

## 512

### Роздільність

Пікселів базової генерації  
(може масштабуватися  
вище)

## 860M

### Параметрів

U-Net містить 860  
мільйонів параметрів

## 8GB

### VRAM мінімум

Можливість запуску на GPU  
(RTX 3060+)

## 2.3B

### Зображень датасету

LAION-5B: масштабний  
датасет пар зображення-  
текст

Stable Diffusion checkpoint

protogenX34OfficialR\_1.ckpt [60fe2f34]

txt2img

img2img

Extras

PNG Info

Checkpoint Merger

Train

Tokenizer

Settings

Extensions

green sapling rowing out of ground, mud, dirt, grass, high quality, photorealistic, sharp focus, depth of field

Negative prompt (press Ctrl+Enter or Alt+Enter to generate)

26/75

Generate

Style 1

None

Style 2

None

Sampling method

Euler a

Sampling steps

20

Restore faces

Tiling

Hires. fix

Width

512

Height

512

Batch count

4

Batch size

1

CFG Scale

12

Seed

1441787169

Script

None

green sapling rowing out of ground, mud, dirt, grass, high quality, photorealistic, sharp focus, depth of field

Steps: 20, Sampler: Euler a, CFG scale: 12, Seed: 1441787169, Size: 512x512, Model hash: 60fe2f34, Model: protogenX34OfficialR\_1

Time taken: 8.62s Torch active/reserved: 3699/4702 MiB, Sys VRAM: 7020/24576 MiB (28.56%)

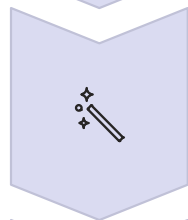
# Latent Diffusion: ключова інновація

Stable Diffusion використовує Latent Diffusion Models (Rombach et al., 2022) — diffusion у стисненому латентному просторі замість pixel space.



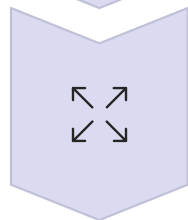
## Encoder

VAE encoder стискає зображення  $512 \times 512 \times 3$  у латент  $64 \times 64 \times 4$  (8x compression).



## Diffusion

U-Net працює у латентному просторі, значно швидше та ефективніше.



## Decoder

VAE decoder розпаковує латент назад у pixel space  $512 \times 512 \times 3$ .

Це зменшує обчислювальну складність у 64 рази порівняно з pixel-space diffusion, роблячи inference практичним на споживчому обладнанні.

# Text-to-Image Pipeline

## Архітектура процесу

Text-to-Image pipeline складається з декількох ключових компонентів, що працюють послідовно для перетворення текстового опису в зображення.

**Текстовий енкодер** (зазвичай CLIP або T5) аналізує вхідний prompt і перетворює його на високорівневі embedding-вектори, що містять семантичне значення опису.



### Текстовий prompt

Вхідний опис бажаного зображення



### Енкодер

Перетворення тексту в embeddings



### Diffusion Model

Генерація латентного простору



### Декодер

Фінальне зображення

# Компоненти Text-to-Image

## CLIP Text Encoder

Перетворює текст у 768-вимірний вектор, що захоплює семантичне значення prompt. Модель навчена на мільйонах пар текст-зображення.

- Токенізація до 77 токенів
- Контекстуальні embeddings
- Кросс-attention з U-Net

## U-Net Diffusion

Нейронна мережа, що поступово знижує шум у латентному просторі, керуючись текстовими embeddings через cross-attention механізми.

- Ітеративний denoising процес
- Attention layers для контролю
- Timestep conditioning

## VAE Decoder

Декодує компактний латентний вектор назад у повнорозмірне зображення високої роздільної здатності з правильними кольорами та деталями.

- Розпакування з латентного простору
- Відновлення деталей
- Фінальна обробка кольорів

# SDXL: Stable Diffusion XL

У липні 2023 Stability AI випустила SDXL — значне покращення архітектури та якості генерації.

## Архітектурні покращення

- Більший U-Net: 2.6B параметрів (3× більше за SD1.5)
- Роздільність 1024×1024 нативно
- Два text encoders: OpenCLIP ViT-G + CLIP ViT-L
- Refined refiner model для post-processing
- Покращений VAE

## Покращення якості

SDXL демонструє значно кращу деталізацію, особливо для рук, тексту в зображеннях, та складних композицій. Кращий prompt following та композиція.

# Архітектура SDXL: деталі

## Dual Text Encoders

OpenCLIP ViT-G/14 (більший, 1.4B params) + CLIP ViT-L/14. Конкатенація ембеддінгів для кращого семантичного представлення.

## Base + Refiner

Base model генерує за 30-50 кроків. Refiner model додає деталізацію за останні 10-20 кроків.

## Conditioning

Додаткові умови для кращої якості та композиції.

## Timestep Conditioning

Покращені часові ембеддінги.

SDXL потребує 16GB+ VRAM для навчання, але інференс можливий на 12GB через різні оптимізаційні техніки.

# Текстові ембедінги: CLIP

CLIP (Contrastive Language-Image Pre-training, OpenAI 2021) є базовою моделлю для text conditioning у diffusion моделях.

## Архітектура

Dual-encoder: Image Encoder (Vision Transformer) та Text Encoder (Transformer). Навчається через contrastive loss на 400M+ пар зображення-текст.

Виводить text embeddings розмірності 768 (ViT-L) або 1280 (ViT-G), що використовуються як conditioning для U-Net через cross-attention.

## Prompt Engineering

CLIP добре розуміє різні стилі, об'єкти, дії та композиції.  
Ефективні prompts:

- Специфічні деталі: "oil painting", "digital art"
- Художники: "in style of Van Gogh"
- Якість keywords: "highly detailed", "8k"
- Lighting: "volumetric lighting", "golden hour"

# T5 та OpenCLIP ембединги



## CLIP (OpenAI)

Vision-Language encoder, навчений на парах image-text. Обмежена довжина контексту (77 tokens), але відмінна взаємодія між зображеннями та текстом.



## T5 (Google)

Text-to-Text Transfer Transformer. Краще розуміння мови, більший контекст (512 tokens), кращий для складних промптів.



## OpenCLIP

Open-source імплементація CLIP, навчена на LAION датасеті. Використовується у SDXL. Більші моделі (ViT-G) для кращої якості.

Комбінація різних текстових енкодерів (як у SDXL) дозволяє використовувати переваги кожного: CLIP для візуальної бази, T5 для нюансів мови.

# Концепція ControlNet

ControlNet революціонував можливості керування генерацією зображень, дозволяючи точно контролювати структурні аспекти результату через додаткові умовні входи.

## Основна ідея

Замість покладання виключно на текстовий опис, ControlNet приймає додаткові зображення-умови (контури, карти глибини, пози) і використовує їх для прямого керування просторовою структурою генерованого контенту.

### Геометричний контроль

Точне визначення форм, контурів і меж об'єктів у сцені

### Структурна відповідність

Збереження архітектури та композиції референсного зображення

### Гнучка стилізація

Застосування нового стилю при збереженні структури

# Типи Control-моделей



## Canny Edge Detection

Контроль на основі контурів і країв об'єктів. Ідеальний для перетворення ескізів у повноцінні зображення зі збереженням базової геометрії та композиції.



## Depth Map

Використання карт глибини для контролю просторової структури сцени. Дозволяє точно відтворювати тривимірну геометрію та перспективу референсного зображення.



## OpenPose

Контроль пози людських фігур через скелетні keypoints. Незамінний для генерації персонажів у специфічних позах зі збереженням анатомічної правильності.



## Scribble

Робота з грубими начерками та схематичними малюнками. Перетворює прості каракулі у детальні зображення, інтерпретуючи намір художника.



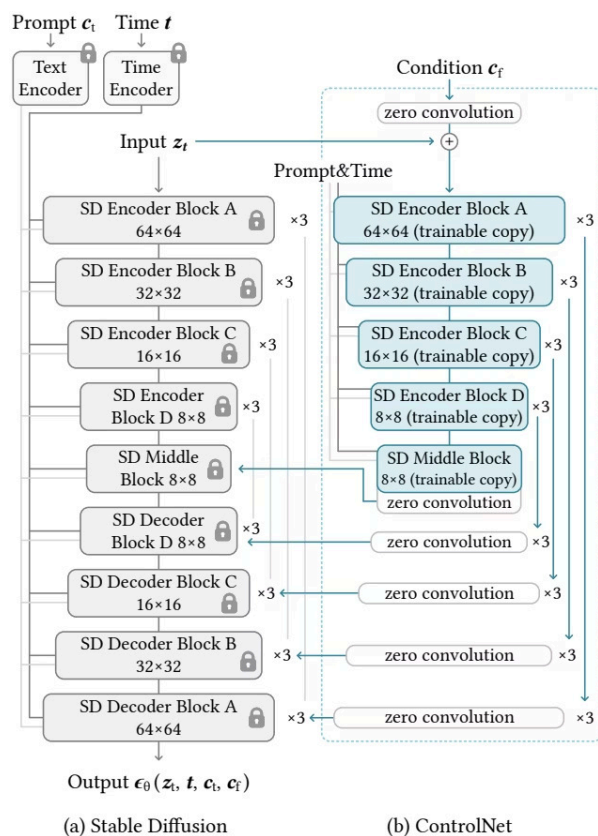
## Segmentation Map

Контроль через семантичну сегментацію, де різні кольори представляють різні класи об'єктів. Точне визначення, що і де має бути в сцені.



## Normal Map

Використання карт нормалей для детального контролю орієнтації поверхонь та мікрогеометрії об'єктів у тривимірному просторі.



ControlNet працює як додатковий "диригент" для основної моделі Stable Diffusion (її архітектури U-Net), допомагаючи їй створювати зображення, які відповідають конкретним вказівкам. На схемі:

- **Заблоковані сірі блоки** — це незмінна частина моделі Stable Diffusion (наприклад, версії 1.5 або 2.1), яка генерує зображення. Її функціонал залишається без змін.
- **Навчувані сині блоки та білі нульові шари згортки** — це компоненти ControlNet. Вони додаються до U-Net моделі, щоб вона могла приймати додаткові вказівки. Сині блоки навчаються інтерпретувати ваші дані (наприклад, контури або пози), а білі шари згортки забезпечують плавне вбудовування цих даних, починаючи з "нульового" впливу, щоб не зіпсувати оригінальну модель Stable Diffusion.

# Zero-Convolution трюк

## Інноваційний підхід до ініціалізації

ControlNet використовує "нульову конволюцію" — шар з вагами, ініціалізованими нулями. На початку тренування це означає, що ControlNet не впливає на оригінальну модель.

### Переваги цього підходу:

- Збереження якості базової моделі на старті
- Поступове навчання контрольних механізмів
- Стабільність процесу тренування
- Можливість швидкого fine-tuning

Поступово під час тренування ваги zero-convolution шарів навчаються, і ControlNet починає ефективно модулювати генерацію згідно з умовним входом.



### Технічна деталь

Zero-initialized 1×1 конволюції діють як "лінії зв'язку" між копією U-Net та оригіналом. Вони починають з нульового впливу і поступово навчаються передавати потрібні сигнали.

# Інтеграція ControlNet у SDXL Pipeline

## Підготовка conditioning image

Обробка референсного зображення відповідним препроцесором (Canny, Depth, Pose) для створення control map

## Завантаження ControlNet weights

Підключення натренованих ваг конкретного типу ControlNet (зазвичай 1-2 GB)

## Комбінування з text prompt

Текстовий опис працює разом з візуальними умовами для повного контролю

## Налаштування strength

Регулювання сили впливу ControlNet (0.5-1.5) відносно text guidance

## Guided denoising

Diffusion процес з одночасним урахуванням тексту та візуальних умов

# HuggingFace Diffusers: практичний інструментарій

HuggingFace Diffusers — це state-of-the-art бібліотека для diffusion models у PyTorch. Надає простий API для inference та training.

```
from diffusers import StableDiffusionPipeline
import torch

# Завантаження моделі
pipe = StableDiffusionPipeline.from_pretrained(
    "stabilityai/stable-diffusion-xl-base-1.0",
    torch_dtype=torch.float16
).to("cuda")

# Генерація
prompt = "A serene landscape with mountains at sunset"
image = pipe(
    prompt=prompt,
    num_inference_steps=50,
    guidance_scale=7.5
).images[0]

image.save("output.png")
```

Diffusers підтримує всі major scheduler-и (DDPM, DDIM, Euler, DPM-Solver++), ControlNet, LoRA fine-tuning, та inpainting/outpainting.

# Image-to-Image: варіації та трансформації

Image-to-Image pipeline дозволяє трансформувати існуючі зображення, зберігаючи певні аспекти оригіналу та змінюючи інші відповідно до нового запиту.

## Процес трансформації

1. **Кодування оригіналу** у латентний простір через VAE encoder
2. **Додавання контрольованого шуму** до латентного представлення
3. **Denoising процес** з урахуванням нового prompt
4. **Декодування** модифікованого латентного вектора

## Параметр strength

Визначає ступінь трансформації:

- **0.0-0.3:** мінімальні зміни, збереження структури
- **0.4-0.6:** помірні модифікації
- **0.7-1.0:** радикальні перетворення

# Фіксований латентний простір

**Ініціалізація**  
Вихідне зображення кодується у  
фіксований латентний вектор

**Варіації**  
Різні результати зі збереженням  
базової композиції



**Часткове зашумлення**  
Додавання контрольованої  
кількості гаусового шуму

**Guided denoising**  
Очищення з урахуванням нового  
prompt

Фіксація латентного простору забезпечує стабільність композиції та дозволяє створювати контрольовані варіації вихідного зображення, змінюючи лише окремі аспекти згідно з новим описом.

# Inpainting: заповнення областей

## Технологія локальної генерації

Inpainting дозволяє замінювати або модифікувати конкретні області зображення, залишаючи решту незмінною.

### Принцип роботи:

1. Створення маски для цільової області
2. Кодування всього зображення
3. Деноїзинг тільки в межах маски
4. Згладжування країв з оригіналом

#### Видалення об'єктів

Інтелектуальне заповнення простору після видалення небажаних елементів

#### Заміна деталей

Зміна конкретних елементів з збереженням контексту

#### Реставрація

Відновлення пошкоджених або неповних зображень

# Outpainting: розширення меж

Outpainting розширює зображення за його початкові межі, генеруючи контекстуально відповідний контент, що природно продовжує оригінальну сцену.

## Аналіз країв

Модель вивчає граничні пікселі та розуміє контекст сцени

## Екстраполяція

Передбачення та генерація продовження на основі наявної інформації

## Безшовна інтеграція

Плавний перехід між оригінальним та згенерованим контентом

# Основи формулювання промптів

Prompt engineering — це критично важлива навичка для отримання якісних результатів від генеративних моделей. Правильно сформульований prompt містить чітку структуру та специфічні елементи.

## Ключові компоненти ефективного prompt:

- **Об'єкт або сцена:** що потрібно зобразити
- **Стиль та техніка:** художній напрямок, манера виконання
- **Освітлення та атмосфера:** настрій, час доби
- **Деталі та якість:** специфікація рівня деталізації
- **Композиція:** ракурс, розташування об'єктів



### Приклад структурованого prompt

"A majestic lion, digital painting, dramatic sunset lighting, highly detailed fur texture, close-up portrait, cinematic composition"

# Positive Prompt: що включити



## Головний об'єкт

Чіткий опис того, що має бути центром уваги: персонаж, предмет, сцена або абстрактна концепція



## Художній стиль

Вказівка на манеру виконання: реалізм, аніме, імпресіонізм, 3D-рендер, олійний живопис тощо



## Освітлення

Тип і характер світла: золота година, драматичне освітлення, м'яке студійне світло, neon glow



## Якісні маркери

Ключові слова для підвищення якості: highly detailed, 8k resolution, professional, masterpiece

Positive prompt визначає всі бажані характеристики зображення. Модель намагатиметься максимально включити всі зазначені елементи в фінальний результат.

# Negative Prompt: що виключити

Negative prompt — це потужний інструмент для уникнення небажаних елементів, артефактів і характеристик у згенерованому зображенні.

## Типові артефакти

blurry, low quality, distorted, deformed, ugly, bad anatomy, extra limbs, missing fingers

## Небажані стилі

cartoon, anime (якщо потрібен реалізм), sketch, draft, unfinished

## Технічні дефекти

watermark, text, signature, jpeg artifacts, duplicate, cropped, out of frame

## Композиційні проблеми

poorly drawn, bad composition, cluttered, chaotic, unbalanced

# Ваги tokenів і акцентування

Багато інтерфейсів дозволяють керувати важливістю окремих частин prompt через спеціальний синтаксис вагів.

## Синтаксис Automatic111

(keyword:1.3) — посилення на 30%

(keyword:0.7) — послаблення на 30%

((keyword)) — еквівалент 1.21x

## Синтаксис ComfyUI

(keyword:1.5) — множник ваги

Значення більше 1 посилюють,  
менше 1 послаблюють вплив

## DALL·E підхід

Використовує природну мову для акцентів: "strongly", "slightly", "very"

## Практичний приклад

"A (red:1.4) rose, with (delicate petals:1.2), in a garden, (blurred background:0.8)"

Червоний колір посилено, ніжні пелюстки помірно акцентовані, фон розмитий ослаблено

# Зміна стилю через промпти

## **Класичний олійний живопис**

"oil painting, renaissance style, chiaroscuro lighting, classical composition"

## **Кіберпанк естетика**

"cyberpunk, neon lights, futuristic, digital art, blade runner aesthetic"

## **Акварельна техніка**

"watercolor painting, soft colors, gentle brushstrokes, artistic, flowing"

## **Мінімалістичний дизайн**

"minimalist, geometric shapes, clean lines, modern, simple composition"

# Контроль деталізації та пропорцій

Правильне формулювання prompt дозволяє точно керувати рівнем деталізації різних елементів та їх відносними розмірами в композиції.

1

## Рівень деталізації

**Високий:** "extremely detailed, intricate, 8k, hyperrealistic, sharp focus"

**Середній:** "detailed, clear, professional quality"

**Стилізований:** "stylized, artistic interpretation, simplified"

2

## Контроль композиції

**Ракурс:** "close-up portrait, wide angle, bird's eye view, from below"

**Кадрування:** "full body shot, upper body, face only, environmental shot"

3

## Відносні розміри

**Акценти:** "(large:1.3) central object, small background details"

**Пропорції:** "properly proportioned, anatomically correct, realistic scale"

# Prompt Templates і структури

## Базовий template для портретів

[Суб'єкт], [Стиль],  
[Освітлення], [Якість],  
[Деталі], [Атмосфера]

Negative: [Артефакти],  
[Небажані стилі]

### Приклад використання

"A young woman with curly hair, digital painting, soft golden hour lighting, highly detailed, trending on artstation, serene atmosphere"

Negative: "blurry, distorted, bad anatomy, low quality"

## Template для пейзажів

[Локація], [Час доби],  
[Погода], [Стиль],  
[Якість]

Negative: [Дефекти]

### Приклад

"Mountain landscape, sunset, dramatic clouds, photorealistic, 8k resolution, national geographic quality"

Negative: "cartoon, unrealistic, oversaturated, jpeg artifacts"

# Fine-tuning під стиль користувача

Персоналізація моделей через навчання на специфічних стилях дозволяє створювати унікальні результати, що відповідають конкретним вимогам користувача.

## Збір референсного датасету

10-50 зображень бажаного стилю з описами. Якість та різноманітність важливіші за кількість.

## Підготовка та анотування

Анотування кожного зображення детальними описами, що фіксують ключові стилістичні особливості.

## Тренування LoRA або DreamBooth

Fine-tuning базової моделі на підготовленому датасеті з оптимальними гіперпараметрами.

## Створення prompt guidelines

Документування оптимальних пром프트ів і негативних пром프트ів для активації навченого стилю.

# Мотивація для LoRA

Fine-tuning великих diffusion моделей традиційним способом вимагає величезних обчислювальних ресурсів та часу.

Перетренування всіх вагів Stable Diffusion (кілька мільярдів параметрів) для кожного нового стилю непрактично.

## Проблеми традиційного fine-tuning:

- **Обчислювальна вартість:** потрібні потужні GPU і багато часу
- **Пам'ять:** зберігання повних копій моделі для кожного стилю
- **Катастрофічне забування:** втрата загальних здібностей
- **Складність поширення:** важко ділитися великими моделями



## LoRA рішення

Замість модифікації всіх ваг, LoRA додає невеликі low-rank матриці, які адаптують поведінку моделі. Результат: файли 2-200 MB замість 2-7 GB.

# Математична основа LoRA

LoRA базується на гіпотезі, що адаптація до нового завдання має низьку "внутрішню розмірність" — тобто потребує модифікації лише невеликої частини простору параметрів.

## Стандартний підхід

При fine-tuning оновлюються всі ваги матриці  $W$ :

$$W' = W + \Delta W$$

де  $\Delta W$  має ті ж розміри, що й  $W$  (наприклад,  $4096 \times 4096$ )

## LoRA підхід

$\Delta W$  розкладається на дві маленькі матриці:

$$\Delta W = BA$$

де  $B$  має розмір  $(4096 \times r)$  і  $A$  має розмір  $(r \times 4096)$ ,  
 $r \ll 4096$

Типово  $r=4, 8, 16, 32$

Це радикально зменшує кількість параметрів, що тренуються: замість 16М параметрів ( $4096^2$ ) ми тренуємо лише  $2 \times 4096 \times r$  параметрів, що при  $r=8$  дає лише ~65К параметрів — зменшення в 250 разів!

# Тренування власного LoRA

01

## Підготовка датасету

Зберіть 10-100 зображень бажаного стилю або об'єкта. Більше — краще, але навіть 10-20 якісних зображень можуть дати гарні результати.

03

## Налаштування параметрів

Виберіть rank (зазвичай 8-32), learning rate ( $1e-4$  —  $1e-5$ ), кількість epochs (10-50), batch size згідно з доступною VRAM.

05

## Тестування та оцінка

Генеруйте тестові зображення з різними prompts та alpha значеннями. Оцініть якість стилізації та можливе overfitting.

02

## Caption/Tagging

Створіть текстові описи для кожного зображення. Можна використати BLIP2 для автоматичного caption або вручну додати теги.

04

## Запуск тренування

Використайте autotrain, SFT, або інші LoRA тренери. Процес займе 30 хв — 3 години залежно від датасету та hardware.

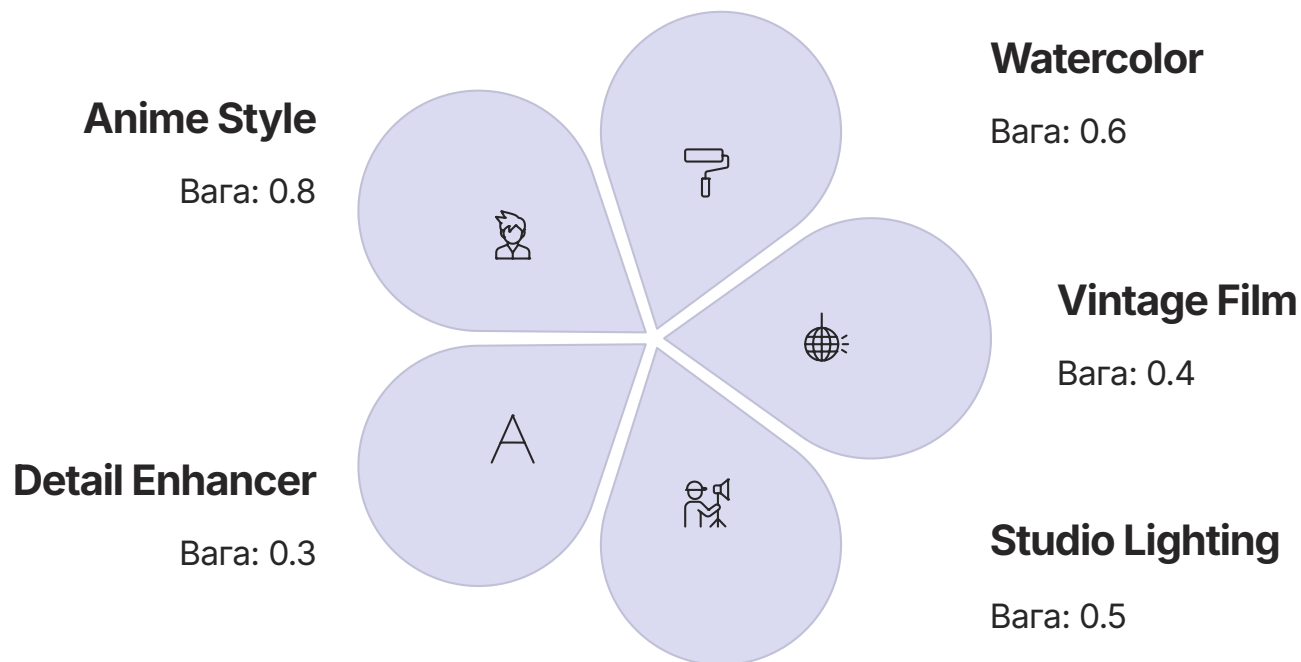
06

## Публікація та sharing

Завантажте LoRA на CivitAI, HuggingFace або інші платформи. Додайте приклади та рекомендовані промпти.

# Комбінування кількох LoRA

Потужна можливість LoRA — це змішування декількох адаптацій для створення унікальних комбінацій стилів та концепцій.



Експериментуйте з різними комбінаціями та вагами. Сума ваг може перевищувати 1.0, але зазвичай краще тримати загальну силу під контролем.

# CLIP як основа DALL·E

CLIP (Contrastive Language-Image Pre-training) — це фундаментальна модель, що навчена розуміти відповідність між текстом та зображеннями.



DALL·E використовує CLIP embeddings як цільовий простір, до якого prior network проектує текстові описи, а decoder перетворює на пікселі.

# Порівняння: використання та доступність

## DALL-E 3

- **Доступ:** лише через OpenAI API або ChatGPT Plus
- **Вартість:** платна модель використання
- **Кастомізація:** обмежена, без доступу до ваг
- **Конфіденційність:** дані передаються на сервери OpenAI
- **Переваги:** найкраще розуміння складних промптів, офіційна підтримка

## SDXL

- **Доступ:** повністю open-source, безкоштовний
- **Вартість:** лише hardware (власний GPU або cloud)
- **Кастомізація:** повний контроль, fine-tuning, LoRA, ControlNet
- **Конфіденційність:** повністю локальне виконання
- **Переваги:** екосистема розширень, спільнота, гнучкість

# Сильні сторони DALL·E



## Природне розуміння мови

DALL·E 3 надзвичайно добре інтерпретує складні, розмовні промпти з контекстом, метафорами та абстрактними концепціями.



## Генерація тексту в зображеннях

Значно краще, ніж SDXL, вміє генерувати читабельний текст у правильному написанні та розташуванні.



## Консистентність

Висока стабільність результатів, менше артефактів та дивних композицій. Краща анатомія людських фігур.



## Слідування інструкціям

Точно виконує складні мультичастинні промпти з специфічними деталями про розташування, кольори, стилі.

# Сильні сторони SDXL



## Прецизійний контроль

ControlNet, LoRA, і інші розширення дають безпрецедентний контроль над структурою, стилем та деталями генерації.



## Повна кастомізація

Можливість fine-tune під будь-які потреби, створення власних LoRA, інтеграція з іншими інструментами.



## Спільнота та екосистема

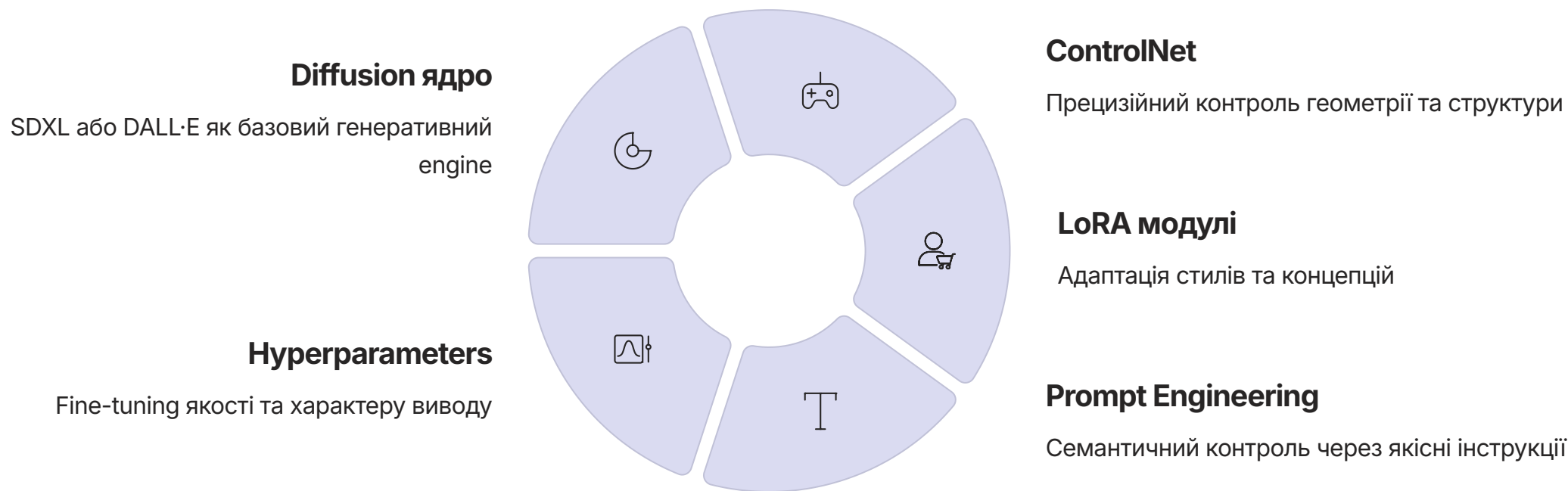
Величезна спільнота, тисячі готових LoRA, моделей, розширень. Постійні оновлення та покращення.



## Приватність і власність

Повністю локальне виконання, без передачі даних, повні права на згенеровані зображення, без обмежень використання.

# Підсумок: Сучасна генерація як контрольована система



Сучасна генерація зображень — це не чорна скринька, а складна контрольована система з множинними рівнями керування. Diffusion моделі забезпечують високоякісне ядро, ControlNet і подібні механізми дають структурний контроль, LoRA адаптує стилістику, а якісний prompt engineering є ключем до отримання саме того результату, який потрібен. Розуміння цих інструментів та вміння їх комбінувати відкриває практично необмежені творчі можливості.