

# ĐƯA QUẢNG BÌNH RA THẾ GIỚI QUA LĂNG KÍNH AI

TÊN NHÓM: 9H53





**KHỞI ĐỘNG**

# TẠI SAO LÀ “PHONG NHA”?

Tên gọi mang trong mình vẻ đẹp, gió trời và hơi thở của di sản.



## 1. PHONG

Có nghĩa là “**gió**”.

Thể hiện sự tự do, khoáng đạt và sức sống mạnh mẽ của thiên nhiên Quảng Bình.

GIÓ Ở QUẢNG BÌNH  
MÁT QUÁ TRỜI!



## 2. NHA

Có nghĩa là “**răng**”.

Những chiếc nhũ đá trong hang dài và nhọn như hàm răng.

NHÌN GIỐNG  
HÀM RĂNG  
THỂ NHỎ



## PHONG NHA = GIÓ QUUA KỂ RĂNG

Phong Nha là nơi gió thổi qua những hang động kỳ vĩ, tạo nên bức tranh thiên nhiên hùng vĩ và thơ mộng.

“Phong Nha = nơi gió kể chuyện qua những hang động triệu năm.”



## FUN FACT:

Phong Nha – Kẻ Bàng được UNESCO công nhận là Di sản thiên nhiên thế giới 2 lần: năm 2003 và 2015!



Hệ thống hang động  
lớn nhất Đông Nam Á



Giá trị địa chất – địa  
mạo độc đáo toàn cầu



Báu vật vô giá  
của thiên nhiên



STUDENTS

HOME

ABOUT

CONTENT

OTHERS

# GIỚI THIỆU THÀNH VIÊN



PHẠM ĐỨC MINH



NGUYỄN THÀNH LUÂN



NGUYỄN THANH AN BÌNH



NGUYỄN KHÁNH TOÀN



NGUYỄN HỒ MÃN TUỆ



MENTORS

HOME

ABOUT

CONTENT

OTHERS

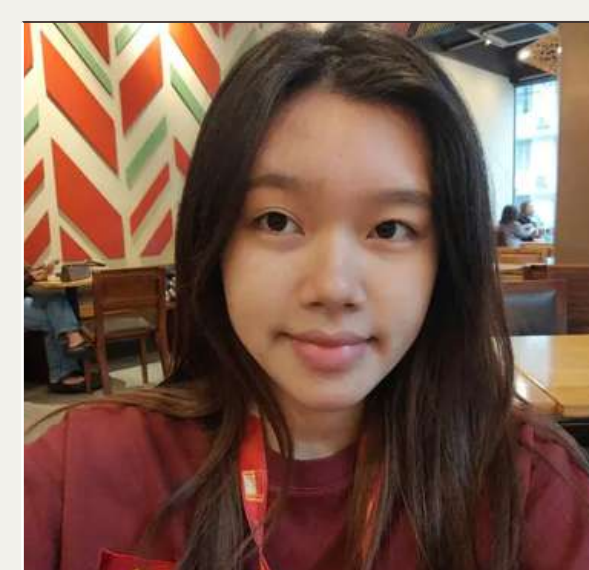
# GIỚI THIỆU THÀNH VIÊN



*ĐẶNG CAO CƯỜNG*



*QUANG PHẠM*



*NGUYỄN THỊ HÀ CHI*

# GIỚI THIỆU THÀNH VIÊN



*NGUYỄN PHÚ VINH*



*NGUYỄN PHƯƠNG TRANG*



*DUY NGUYỄN*



1. ĐỘNG LỰC CỦA ĐỀ TÀI & VẤN ĐỀ CỦA MODEL GỐC
2. TEXT-2-IMAGE MODEL, FLUX, LORA LÀ GÌ?
3. QUY TRÌNH THỰC HIỆN
4. THÀNH QUẢ & SẢN PHẨM CUỐI CÙNG
5. MỤC TIÊU TƯƠNG LAI & KẾT LUẬN

# LIST OF CONTENTS



# 1. ĐỘNG LỰC CỦA ĐỀ TÀI & VẤN ĐỀ CỦA MODEL GỐC





# CHI PHÍ XÂY DỰNG BỘ POSTER DU LỊCH



Ước tính khi thiết kế thủ công cho toàn bộ các địa danh du lịch tại **Quảng Bình**

## 300+

### Địa danh du lịch

Hang động • Biển Núi Sông Danh thắng



## 3.000 +

### Thời gian thiết kế

300 địa danh × 10 giờ/poster



KHOẢNG **600 triệu VNĐ**

### Chi phí thiết kế

Ước tính 2 triệu VNĐ/poster



## 300+ Poster

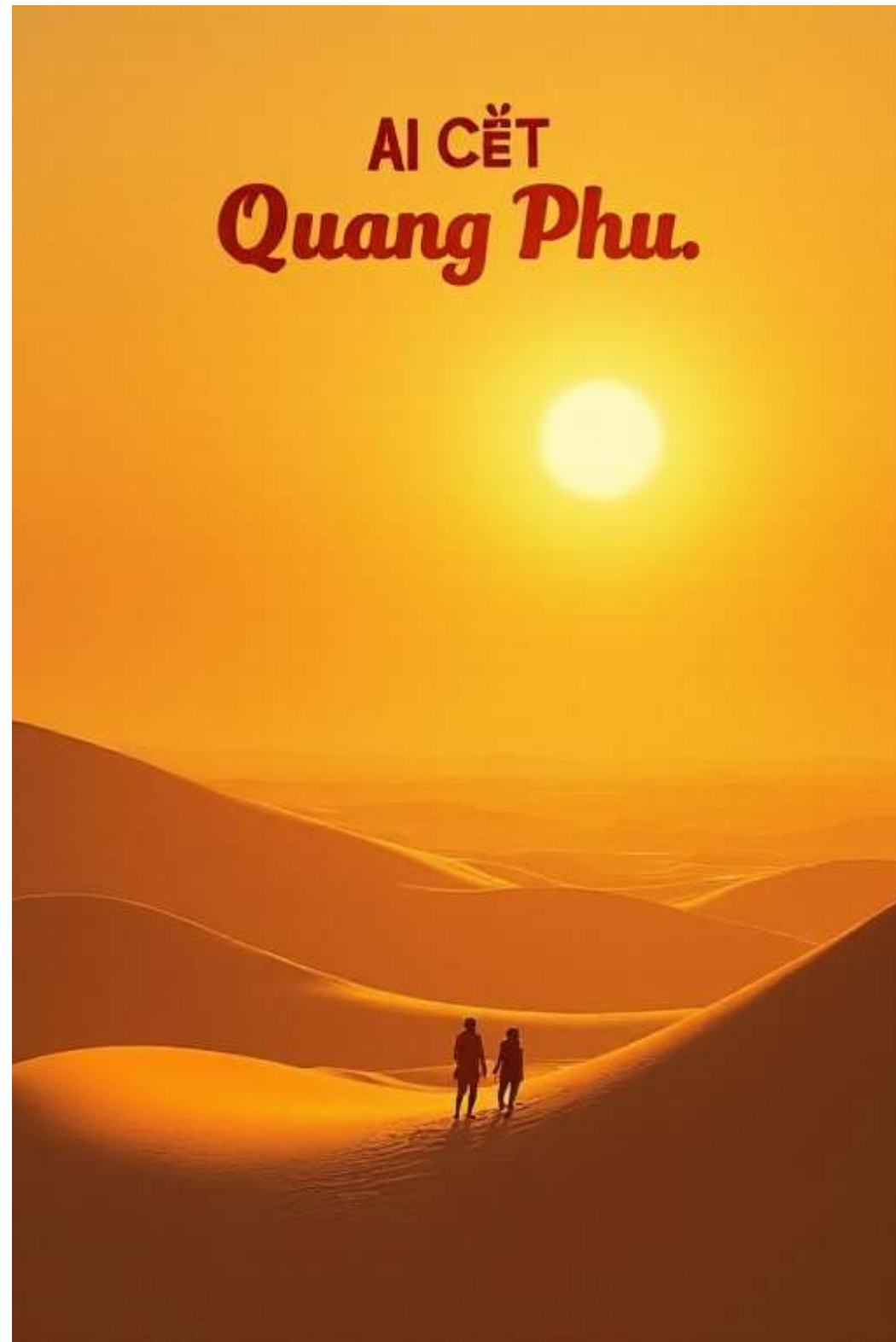
### Khối lượng công việc

Mỗi địa danh cần 1 poster



## TẠO POSTER BẰNG FLUX.1-DEV?

**PROMPT:** "Travel poster layout, Quang Phu Sand Dunes, vast golden sand, beautiful warm sunset, glowing light, tiny people walking. Bold modern typography text "ĐỒI CÁT QUANG PHÚ". Cinematic lighting, photorealistic landscape, graphic design, magazine cover style, vibrant colors. --ar 3:4"



**PROMPT:** "Cinematic travel poster layout, inside Phong Nha Cave, majestic stalactites, underground river, small wooden boat, mystical glowing lights. Bold elegant typography text "ĐỘNG PHONG NHA". Highly detailed, stunning nature photography, graphic design elements, movie poster style. --ar 3:4"





## CÁC KHÓ KHĂN VỀ KỸ THUẬT CỦA CÁC MODEL TEXT2IMAGE HIỆN NAY

### KHÔNG THỂ SINH RA ẢNH PHONG NHA CHÍNH XÁC

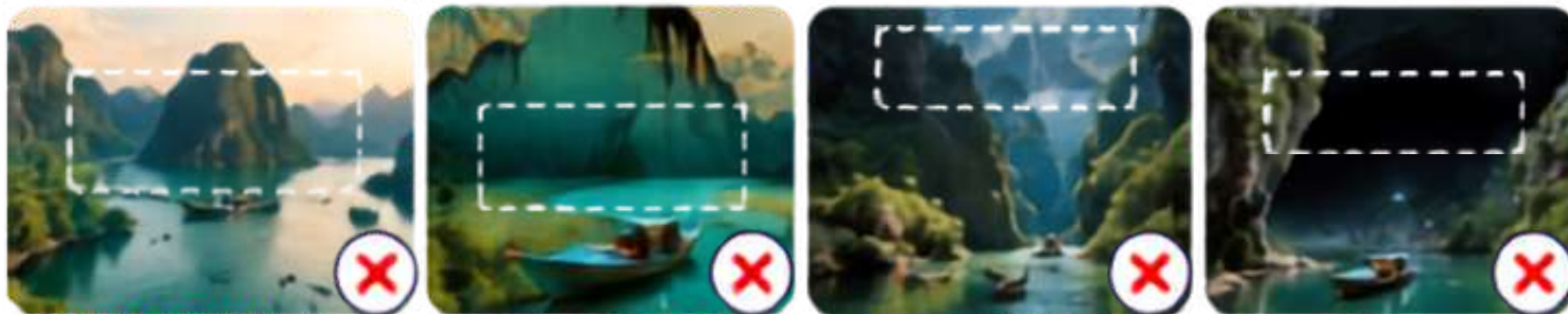
**Prompt:** Phong Nha cave, Quang Bình, Vietnam



➡ Ảnh sinh ra sai đặc trưng địa danh, cảnh quan không đúng Phong Nha

### KHÔNG THỂ SINH RA TEXT

**Prompt:** Poster du lịch Phong Nha



➡ Không tạo được chữ hoặc text bị méo, sai nội dung, không rõ ràng

## NHÓM HƯỚNG TỚI GIẢI PHÁP

XÂY DỰNG MỘT MODEL AI  
TEXT TO IMAGE + LAYOUT GENERATION  
CHUYÊN BIỆT CHO POSTER DU LỊCH QUẢNG BÌNH



SINH ẢNH  
PHONG NHA  
CHÍNH XÁC



TẠO POSTER  
CÓ TEXT RÕ RÀNG,  
BỐ CỤC HỢP LÝ



TAO RA NHỮNG POSTER DU LỊCH CHUYÊN NGHIỆP,  
ĐÚNG BẢN SẮC QUẢNG BÌNH



# QUY TRÌNH DỰ ÁN HUẤN LUYỆN MODEL AI TEXT TO IMAGE TẠO POSTER DU LỊCH QUẢNG BÌNH

[HOME](#)[ABOUT](#)[CONTENT](#)[OTHERS](#)

Quy trình gồm 2 bước chính: Image Generation và Layout Generation

**1**

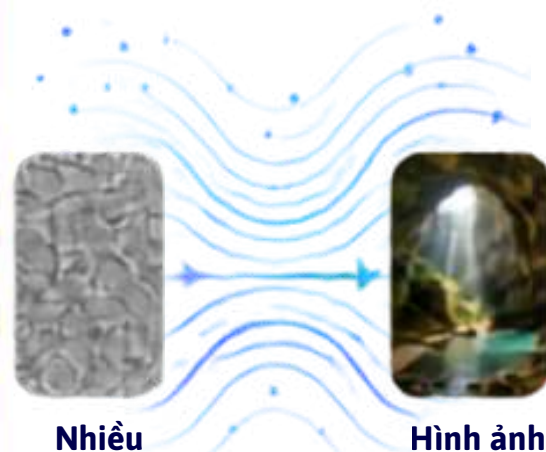
## IMAGE GENERATION

### MÔ TẢ VĂN BẢN



PROMPT: "pn\_cave, inside cave, grand underground river chamber, tall illuminated stalagmites and rock columns lit by warm golden spotlights, dark calm water reflecting artificial light, small tour boat floating in background"

### FLOW MATCHING (TEXT-TO-IMAGE MODEL)



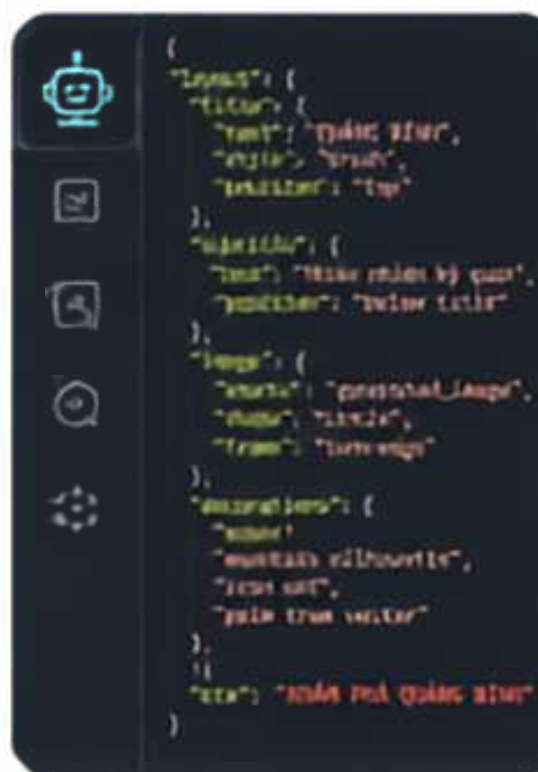
Flow Matching

### HÌNH ẢNH DU LỊCH ĐƯỢC TẠO

**2**

## LAYOUT GENERATION

### CODING AGENT



AI coding agent sắp xếp các thành phần và bố cục.

### LAYOUT CODE (CẤU TRÚC POSTER)

KHU VỰC TIÊU ĐỀ

KHU VỰC PHỤ ĐỀ

KHU VỰC HÌNH ẢNH

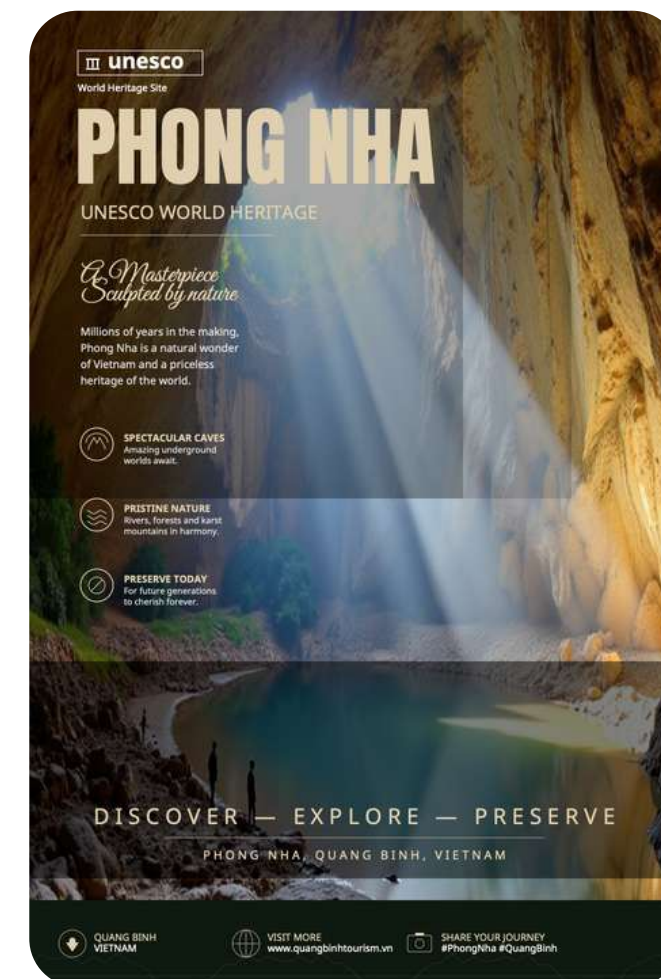
YẾU TỐ TRANG TRÍ



KHU VỰC CTA / THÔNG TIN



### POSTER HOÀN CHÍNH





## 2. TEXT-2-IMAGE MODEL, FLUX, LORA LÀ GÌ?



# TEXT-TO-IMAGE MODEL

INPUT PROMPT

“

*“an astronaut  
riding a horse”*

AI MODEL

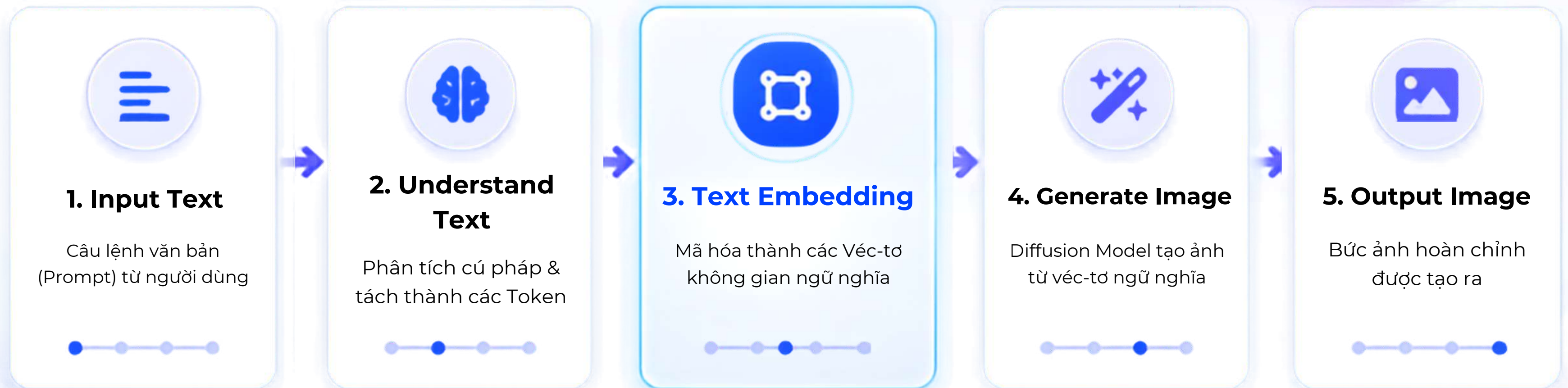
TEXT-TO-IMAGE  
MODEL  
AI ENGINE

GENERATED IMAGE





# TEXT-TO-IMAGE MODEL



**Điểm mấu chốt:** Máy tính không thể trực tiếp “vẽ từ chữ viết. Bắt buộc phải có bước chuyển đổi văn bản thành **Text Embedding** (các con số biểu diễn ý nghĩa) để mô hình sinh ảnh hiểu được nội dung.

# TEXT-TO-IMAGE MODEL

## Prompt Văn Bản Đầu Vào

Ví dụ câu lệnh: “A white sand dune at night”

## Tokenization (Tách Token)

Cắt câu thành từng mảnh nhỏ: “A”, “hat”, “night”

## Text Encoder (như CLIP / T5)

Mạng Nơ-ron xử lý mối tương quan giữa các token trong ngữ cảnh toàn cầu.

```
// Đầu ra: Semantic Embedding Vector
```

```
Input: “A white sand dune at night”
```

```
Output Vector:
```

```
0.12, -0.45, 0.83, 0.29,
```

```
-0.01, 0.04, 0.87, -0.18,
```

```
... (hàng trăm chiều dữ  
liệu)
```

```
// Mỗi con số mã hóa 1 đặc trưng ngữ nghĩa
```

```
// (màu sắc, hành động, không gian, thời gian)
```

# TEXT-TO-IMAGE MODEL

“

## Prompt Văn Bản Đầu Vào

Ví dụ câu lệnh: “A white sand dune at night”



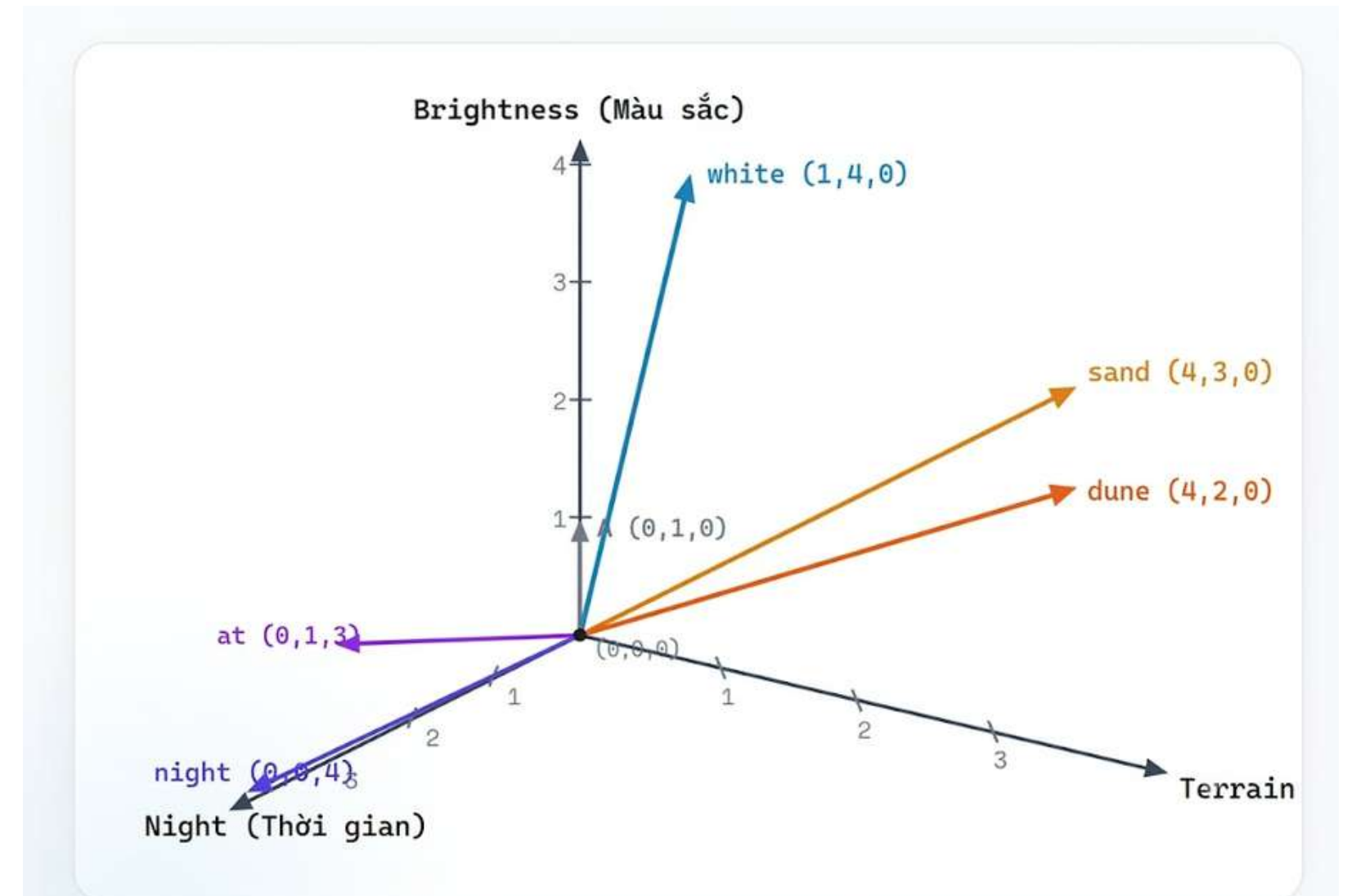
## Tokenization (Tách Token)

Cắt câu thành từng mảnh nhỏ:  
[“A”, “white”, “sand”, “dune”, “at”, “night”]



## Text Encoder (như CLIP / T5)

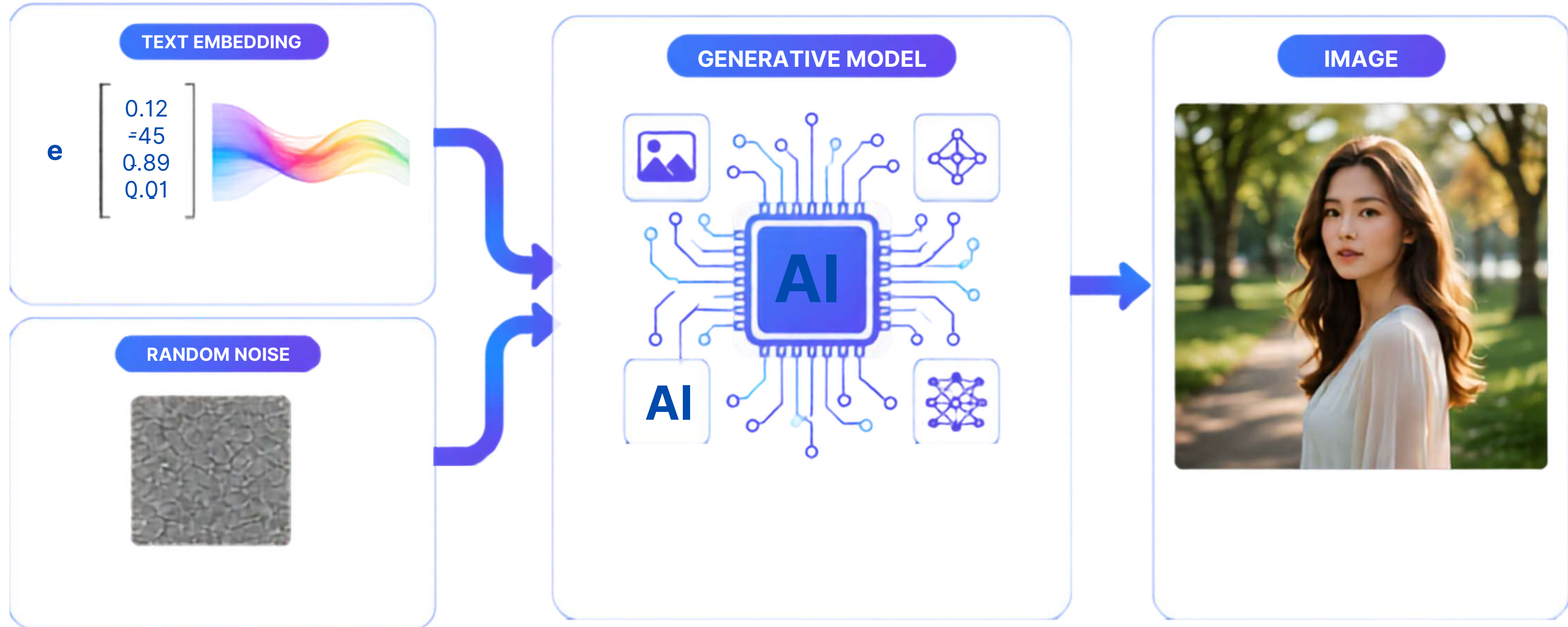
Mạng Nơ-ron xử lý mối tương quan giữa các token trong ngữ cảnh toàn câu.



Mỗi từ/token được biểu diễn thành một **vector** trong không gian đa chiều.  
Mối quan hệ ngữ nghĩa giữa các từ được mô hình hóa thông qua **hướng** và **khoảng cách** giữa các vector.



# TEXT-TO-IMAGE MODEL





# FLUX-1 DEV MODEL



Nền tảng mạnh mẽ cho việc tạo ảnh chất lượng cao từ văn bản



## High-Quality Generation

Tạo ra hình ảnh đạt chất lượng nghệ thuật vượt trội với độ chi tiết cao, màu sắc tự nhiên, xử lý ánh sáng và kết cấu (texture) chân thực.



Chi tiết  
cao



Màu sắc tự  
nhiên



Kết cấu chân  
thực



## Strong Prompt Following

Khả năng hiểu và tuân thủ chặt chẽ mô tả đầu vào, render chính xác chữ viết (text) và bố cục chi tiết trong ảnh.



Hiểu ngữ cảnh  
phức tạp



Bố cục chi  
tiết chữ viết



## Fine-Tuning Capability

Linh hoạt trong việc huấn luyện và tùy biến mô hình (Lora, ControlNet), cho phép thích ứng hoàn hảo với nhận diện thương hiệu của dự án.



FLUX-1 Dev Model mang đến sự cân bằng hoàn hảo giữa **chất lượng**, **độ chính xác** và **khả năng tùy biến**, đáp ứng mọi nhu cầu sáng tạo hình ảnh từ đơn giản đến chuyên nghiệp.



# FINE-TUNING

Tinh chỉnh mô hình để thích ứng tốt hơn với dữ liệu và nhiệm vụ cụ thể.



## ĐỊNH NGHĨA

Quá trình huấn luyện tiếp mô hình đã học sẵn (Pre-trained Model) trên một tập dữ liệu hẹp và chuyên biệt.



## MỤC TIÊU

Giúp mô hình hiểu cảnh đặc thù, thuật ngữ chuyên ngành hoặc làm chủ nhiệm vụ cụ thể.

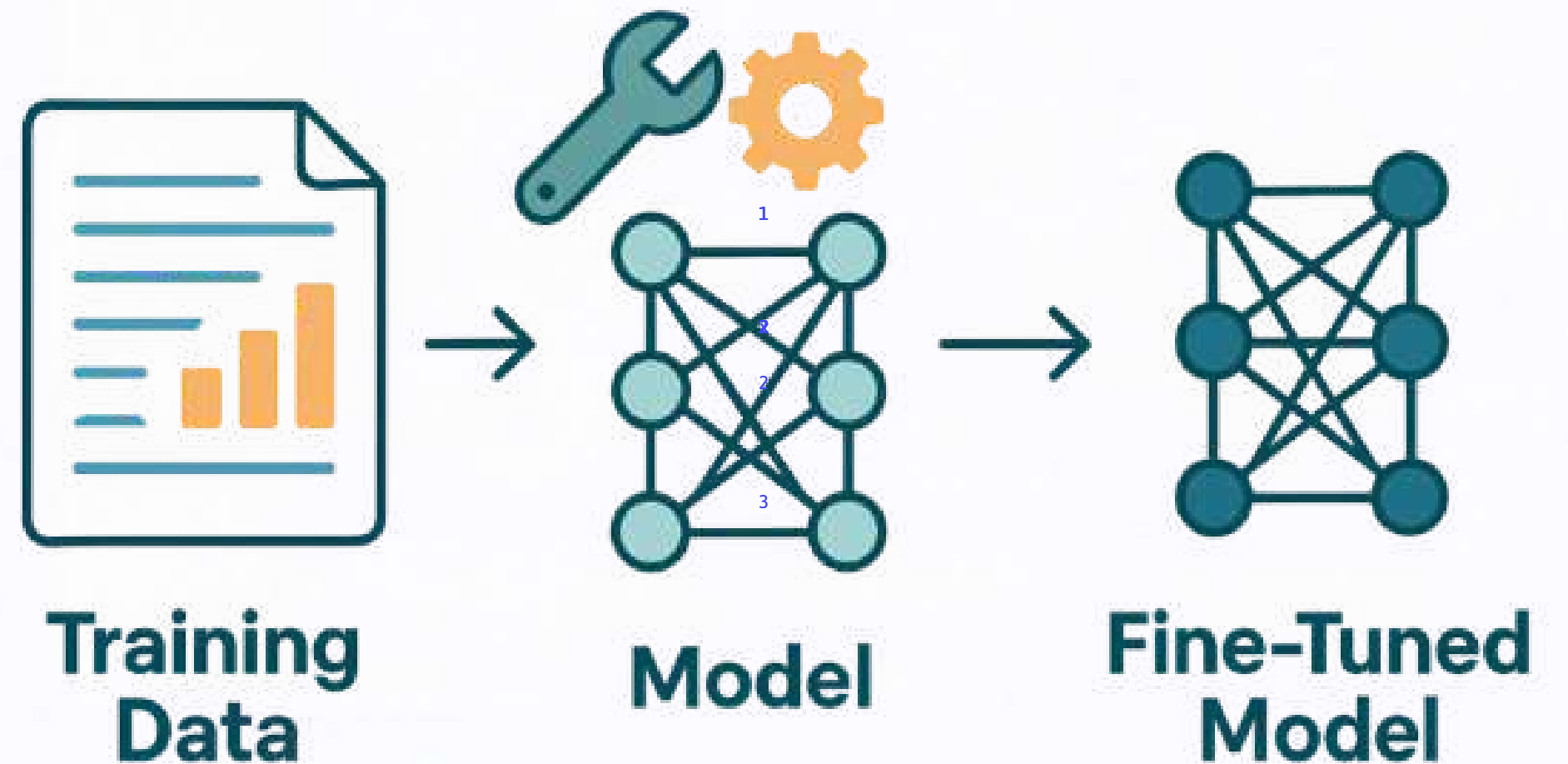


## KHÁC BIỆT

Không bắt đầu từ con số 0 mà kế thừa toàn bộ trí thức nền tảng sẵn có.



## NEURAL NETWORK ARCHITECTURE



# THỰC TRẠNG KHỔNG LỒ CỦA FLUX

Để đạt hiệu suất cao và chất lượng ổn định, mô hình FLUX đòi hỏi nguồn lực tính toán và dữ liệu khổng lồ.



## 12B

THAM SỐ MÔ HÌNH  
(PARAMETERS)

Quy mô cực lớn giúp mô hình hiểu sâu và sinh ảnh chi tiết hơn.



## ~100+ GB

VRAM YÊU CẦU  
KHI FULL FINE-TUNING

Đòi hỏi VRAM cực lớn để huấn luyện toàn bộ mô hình.



## Bất khả thi

CHO CÁC GPU  
THÔNG THƯỜNG HIỆN NAY

Vượt quá khả năng của hầu hết GPU phổ thông trên thị trường.



## KẾT LUẬN

Fine-Tuning FLUX là một bài toán tài nguyên cực lớn.  
Cần chiến lược tối ưu và hạ tầng mạnh mẽ để triển khai hiệu quả.



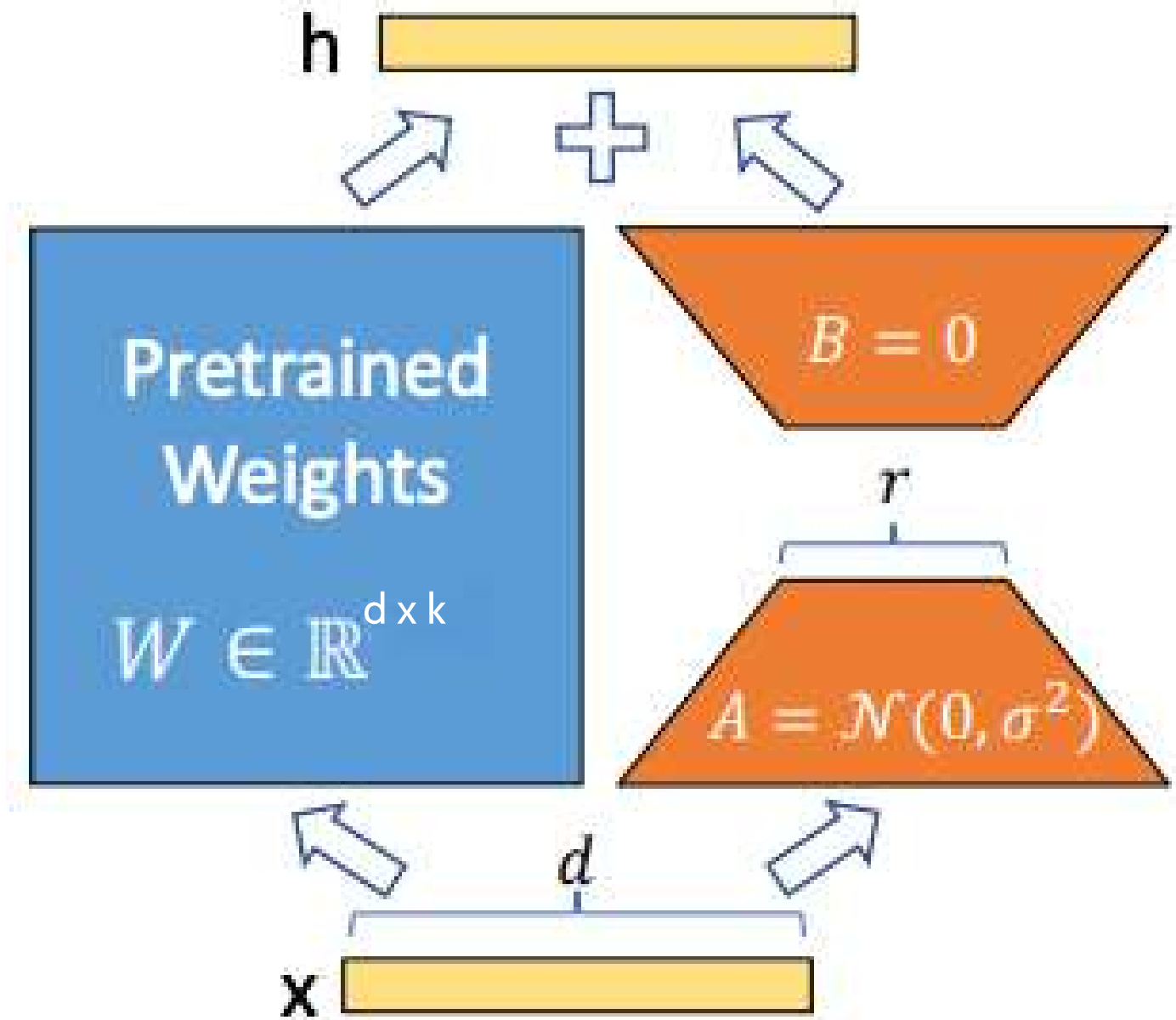


✦

LORA FINE-TUNING

✦

LoRA (Low-Rank Adaptation) là kỹ thuật fine-tuning hiệu quả, chỉ cập nhật một lượng tham số nhỏ thông qua các **ma trận hạng thấp**, giúp tiết kiệm bộ nhớ và tăng tốc quá trình huấn luyện.





### 3. QUY TRÌNH THỰC HIỆN



# QUÁ TRÌNH TẠO DATASET



## MỤC TIÊU

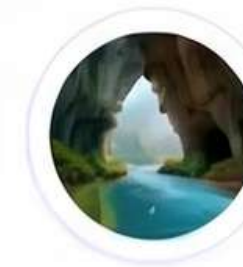
Xây dựng dataset chất lượng cao phục vụ việc fine-tune mô hình LoRA (Flux.1).[dev] cho chủ đề đồi cát Quang Phú và động Phong Nha.

Tìm **36** bức ảnh về đồi cát **Quang Phú** và động **Phong Nha**



**16 ảnh**

Đồi cát Quang Phú



**20 ảnh**

Động Phong Nha



## QUY TRÌNH THỰC HIỆN



### 01. Tìm kiếm

Tìm kiếm ảnh từ các nguồn uy tín và đa dạng.



### 02. Lọc ảnh

Sàng theo các tiêu chí chất lượng và nội dung.



### 03. Tải về

Tải độ phân giải cao nhất có thể.



### 04. Tổ chức

Phân loại và đặt tên ảnh khoa học, dễ quản lý.



Kết quả: 36 ảnh chất lượng cao • 16 ảnh đồi cát Quang Phú • 20 ảnh động Phong Nha



## TIÊU CHÍ CHỌN ẢNH



**Rõ ràng, sắc nét**

Ảnh có độ phân giải cao, không mờ



**Chưa qua chỉnh màu quá nhiều**

Màu sắc tự nhiên, trung thực.



**Ít hoặc không có yếu tố con người**

Ưu tiên ảnh không có người hoặc rất ít.



**Không có yếu tố gây nhiễu**

Không có xe cộ, tàu thuyền, công trình nhân tạo rõ rệt.



**Không có logo, watermark hoặc chữ**

## ✕ Ví dụ ảnh đạt yêu cầu



## ✕ Ví dụ ảnh không đạt yêu cầu





# QUY TRÌNH GẮN TAG CAPTION



## 01 SỬ DỤNG TRIGGER WORD

Luôn bắt đầu caption bằng một "trigger word" — một cụm từ độc nhất, chưa tồn tại (ví dụ: **qp\_dunes**, **pn\_cave**) để AI học cách gắn riêng cụm từ này với đặc trưng của địa danh, không bị nhầm với các đôi cát khác nó đã biết trên internet.

### VÍ DỤ:

"**qp\_dunes**, quang ph sand dunes,  
a wide golden sand dune under blue sky, vietnam"



Trigger word giúp mô hình gắn cụm từ này với hình ảnh của địa danh, đảm bảo tính nhận diện và nhất quán khi tạo ảnh.

## 02 MÔ TẢ CHI TIẾT NỘI DUNG ẢNH

Mô tả đúng những gì nhìn thấy trong ảnh: góc chụp, ánh sáng, thời tiết, các vật thể xuất hiện... bằng tiếng Anh.



### Góc chụp

(góc rộng, góc thấp, toàn cảnh, cận cảnh...)



### Ánh sáng

(bình minh, hoàng hôn, ban ngày, ánh sáng mạnh, ánh sáng mềm...)



### Thời tiết

(trời trong, nhiều mây, nắng, có mây...)



### Bối cảnh & vật thể

(đồi cát, hàng cây mặt nước, không có người, không có xe cộ...)

## ĐA DẠNG HOÁ GÓC CHỤP/THỜI ĐIỂM

Đa dạng hóa góc chụp/thời điểm trong ảnh để mô hình học được nhiều biến thể, tránh học "vết" một góc duy nhất.



Càng đa dạng, mô hình càng hiểu sâu về đề danh và tạo ra hình ảnh chính xác, linh hoạt hơn.



# ✦ QUY TRÌNH GẮN LABEL CAPTION ✦



**LABEL:** *qp\_dunes*, quang phu sand dunes, wide shot of a massive golden sand dune ridge with fine ripple textures, sharp lighting contrast dividing the slope, distant low hills and buildings under a bright blue sky



**LABEL:** *qp\_dunes*, quang phu sand dunes, wide landscape of expansive golden sand dunes in vietnam, rolling sand slopes scattered with small low-growing green shrubs and vegetation patches, bright daylight under a deep blue sky with a few small white clouds



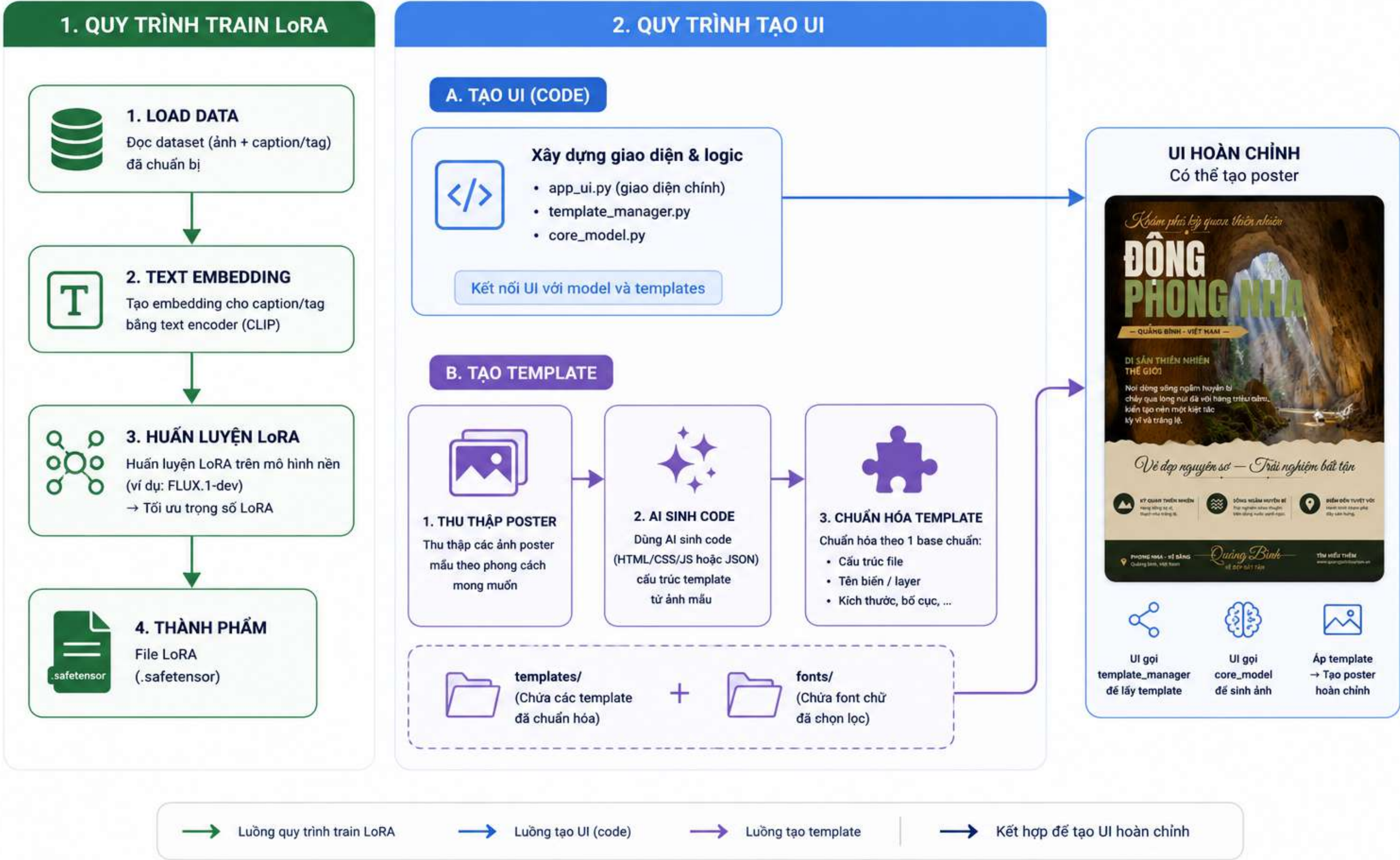
**LABEL:** *pn\_cave*, inside cave, spacious dry cavern interior, wooden walkway path with floor lights, massive hanging stalactites and towering stalagmite pillars, warm spotlight illumination highlighting rock textures



**LABEL:** *pn\_cave*, outside cave, yellow tourist boat on vibrant green emerald river, towering grey limestone cliff wall covered in lush green tropical foliage, dark cave mouth at water level, bright daytime lighting



# QUY TRÌNH THỰC TẾ SAU KHI CHUẨN BỊ DATA





## 4. THÀNH QUẢ & SẢN PHẨM CUỐI CÙNG



**PROMPT:** "qp\_dunes, quang phu sand dunes, wide landscape of expansive golden sand dunes in vietnam, rolling sand slopes scattered with small low-growing green shrubs and vegetation patches, bright daylight under a deep blue sky with a few small white clouds"

**BEFORE**



Hình sinh ra trước khi  
áp dụng LoRA.



**AFTER**



Hình sinh ra sau khi  
áp dụng LoRA.

**PROMPT:** "qp\_dunes, quang phu sand dunes with people"

**BEFORE**



Hình sinh ra trước khi  
áp dụng LoRA.



**AFTER**



Hình sinh ra sau khi  
áp dụng LoRA.

**PROMPT:** "pn\_cave, inside cave, grand underground river chamber, tall illuminated stalagmites and rock columns lit by warm golden spotlights, dark calm water reflecting artificial light, small tour boat floating in background"

**BEFORE**



Hình sinh ra trước khi  
áp dụng LoRA.



**AFTER**



Hình sinh ra sau khi  
áp dụng LoRA.

# QUY TRÌNH TẠO POSTER TỰ ĐỘNG VỚI GIAO DIỆN (UI) TỐI ƯU

## 1 CẤU TRÚC HỆ THỐNG



1 file giao diện (UI) +  
quản lý model & ảnh +  
quản lý template & thư mục chứa  
các mẫu template có sẵn.

## 2 CHẾ ĐỘ TEMPLATE



Template có cấu trúc cố định  
và vùng nội dung linh hoạt  
để tự động điền dữ liệu.

## 3 QUY TRÌNH HOẠT ĐỘNG



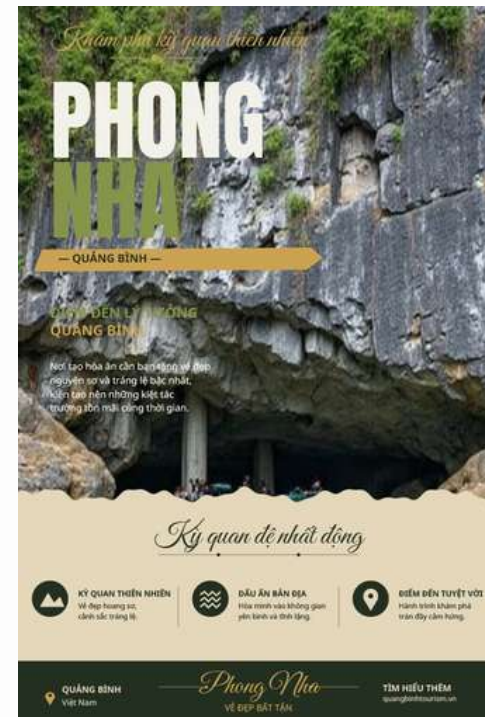
Người dùng chỉ cần thao tác đơn giản,  
hệ thống tiện lợi,  
render chữ và xuất file PDF chuyên nghiệp.

## 4 ƯU ĐIỂM ĐỘT PHÁ

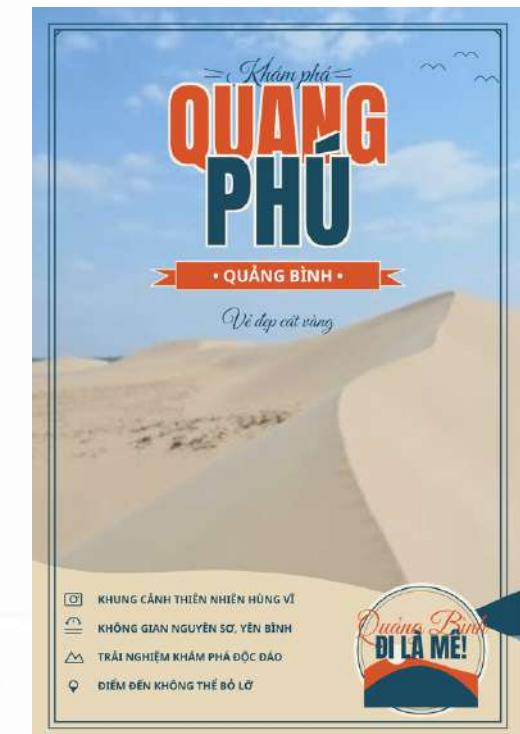


Khắc phục khuyết điểm của Flux (text, render  
chữ), cho phép tùy biến, chọn font, bố cục  
và chỉnh sửa linh hoạt.

PROMPT: "pn\_cave, phong nha cave, outside cave".

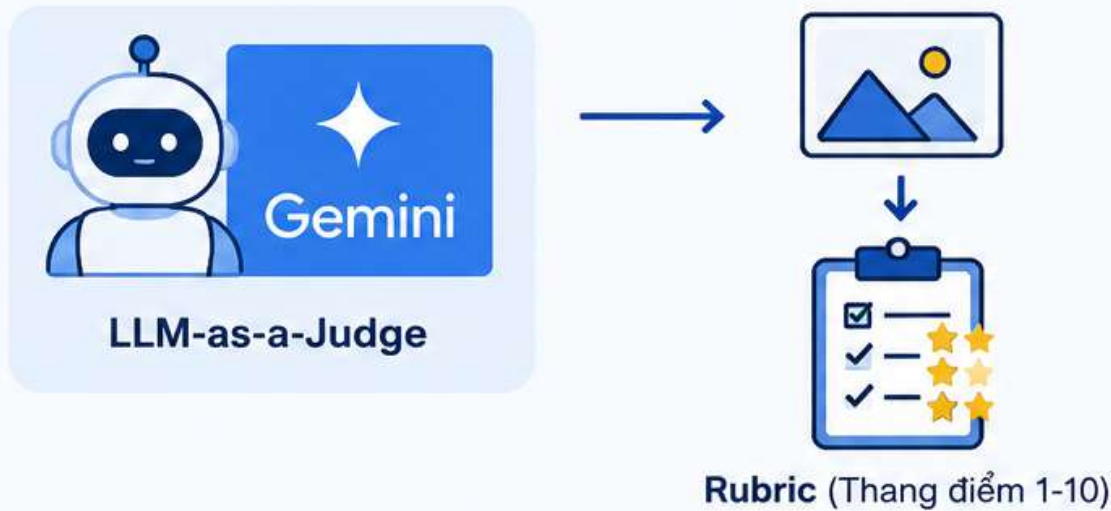


PROMPT: "cp\_dunes, quang phủ sand dunes".



# SO SÁNH, ĐÁNH GIÁ KẾT QUẢ

## PHƯƠNG PHÁP ĐÁNH GIÁ: LLM-AS-A-JUDGE



Sử dụng **Gemini** đóng vai trò như một chuyên gia đánh giá độc lập (**LLM-as-a-Judge**) để chấm điểm tự động các bức ảnh được sinh ra.

Với khả năng đa thức (**Multimodal**), Gemini phân tích trực tiếp hình ảnh theo hệ tiêu chuẩn định sẵn (**Rubric**) trên **thang điểm 10**, giúp đảm bảo tính nhất quán, khách quan và giảm thiểu chi phí đánh giá thủ công.

## CÔNG THỨC TÍNH ĐIỂM

### 1. Điểm đánh giá trung bình cho từng tiêu chí

Điểm đánh giá trung bình cho từng tiêu chí  $c$  trên tập mẫu  $N$  ảnh thử nghiệm được xác định bởi:

$$S_c = \frac{1}{N} \sum_{i=1}^N S_{c,i}$$

Trong đó:

- $S_c$ : Điểm trung bình của tiêu chí  $c$ .
- $S_{c,i} \in [1, 10]$ : Điểm do Gemini chấm cho ảnh thứ  $i$  theo tiêu chí  $c$ .
- $N$ : Tổng số lượt/mẫu ảnh được thử nghiệm (ở đây  $N = 8$ ).

### 2. Chênh lệch hiệu năng giữa hai mô hình ( $\Delta$ )

$$\Delta_c = S_c^{\text{LoRA}} - S_c^{\text{Base}}$$

Trong đó:

- $\Delta_c$ : Chênh lệch điểm trung bình của tiêu chí  $c$ .
- $S_c^{\text{LoRA}}$ : Điểm trung bình của mô hình LoRA.
- $S_c^{\text{Base}}$ : Điểm trung bình của mô hình Base.



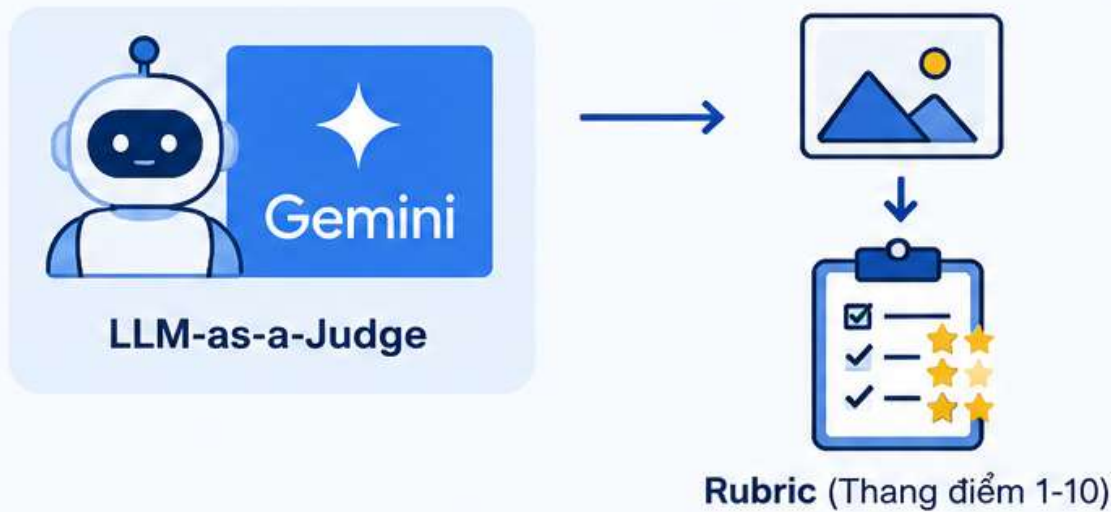
- $\Delta_c > 0$ : LoRA vượt trội hơn Base
- $\Delta_c = 0$ : Hiệu năng tương đương
- $\Delta_c < 0$ : LoRA kém hơn Base



**Mục tiêu:** Đánh giá khách quan, nhất quán và tiết kiệm chi phí thông qua LLM-as-a-Judge (Gemini) trên thang điểm 10.

# SO SÁNH, ĐÁNH GIÁ KẾT QUẢ

## PHƯƠNG PHÁP ĐÁNH GIÁ: LLM-AS-A-JUDGE



Sử dụng **Gemini** đóng vai trò như một chuyên gia đánh giá độc lập (**LLM-as-a-Judge**) để chấm điểm tự động các bức ảnh được sinh ra.

Với khả năng đa thức (**Multimodal**), Gemini phân tích trực tiếp hình ảnh theo hệ tiêu chuẩn định sẵn (**Rubric**) trên **thang điểm 10**, giúp đảm bảo tính nhất quán, khách quan và giảm thiểu chi phí đánh giá thủ công.

## CÔNG THỨC TÍNH ĐIỂM

### 1. Điểm đánh giá trung bình cho từng tiêu chí

Điểm đánh giá trung bình cho từng tiêu chí  $c$  trên tập mẫu  $N$  ảnh thử nghiệm được xác định bởi:

$$S_c = \frac{1}{N} \sum_{i=1}^N S_{c,i}$$

Trong đó:

- $S_c$ : Điểm trung bình của tiêu chí  $c$ .
- $S_{c,i} \in [1, 10]$ : Điểm do Gemini chấm cho ảnh thứ  $i$  theo tiêu chí  $c$ .
- $N$ : Tổng số lượt/mẫu ảnh được thử nghiệm (ở đây  $N = 8$ ).

### 2. Chênh lệch hiệu năng giữa hai mô hình ( $\Delta$ )

$$\Delta_c = S_c^{\text{LoRA}} - S_c^{\text{Base}}$$

Trong đó:

- $\Delta_c$ : Chênh lệch điểm trung bình của tiêu chí  $c$ .
- $S_c^{\text{LoRA}}$ : Điểm trung bình của mô hình LoRA.
- $S_c^{\text{Base}}$ : Điểm trung bình của mô hình Base.



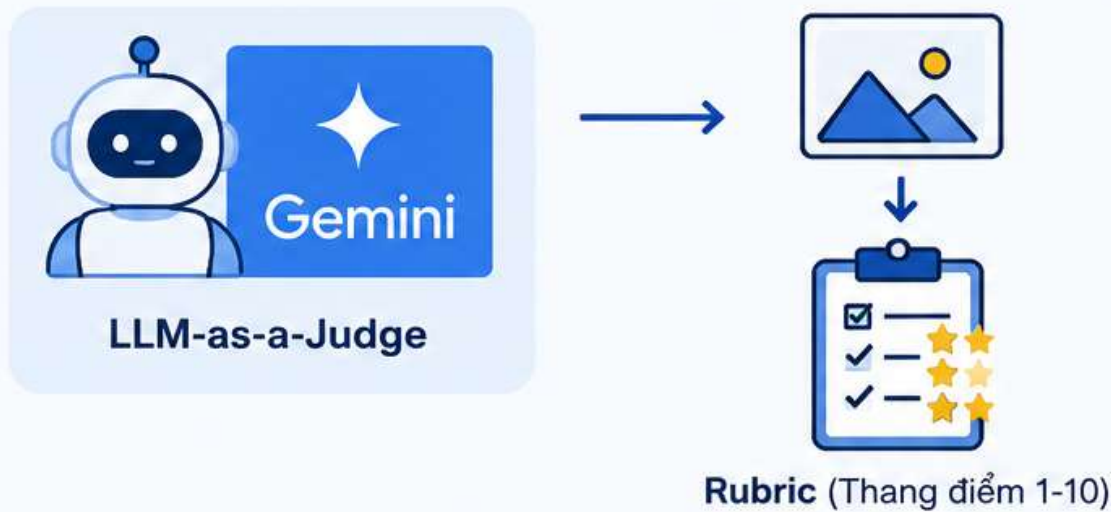
- $\Delta_c > 0$ : LoRA vượt trội hơn Base
- $\Delta_c = 0$ : Hiệu năng tương đương
- $\Delta_c < 0$ : LoRA kém hơn Base



**Mục tiêu:** Đánh giá khách quan, nhất quán và tiết kiệm chi phí thông qua LLM-as-a-Judge (Gemini) trên thang điểm 10.

# SO SÁNH, ĐÁNH GIÁ KẾT QUẢ

## PHƯƠNG PHÁP ĐÁNH GIÁ: LLM-AS-A-JUDGE



Sử dụng **Gemini** đóng vai trò như một chuyên gia đánh giá độc lập (**LLM-as-a-Judge**) để chấm điểm tự động các bức ảnh được sinh ra.

Với khả năng đa thức (**Multimodal**), Gemini phân tích trực tiếp hình ảnh theo hệ tiêu chuẩn định sẵn (**Rubric**) trên **thang điểm 10**, giúp đảm bảo tính nhất quán, khách quan và giảm thiểu chi phí đánh giá thủ công.

## CÔNG THỨC TÍNH ĐIỂM

### 1. Điểm đánh giá trung bình cho từng tiêu chí

Điểm đánh giá trung bình cho từng tiêu chí  $c$  trên tập mẫu  $N$  ảnh thử nghiệm được xác định bởi:

$$S_c = \frac{1}{N} \sum_{i=1}^N S_{c,i}$$

Trong đó:

- $S_c$ : Điểm trung bình của tiêu chí  $c$ .
- $S_{c,i} \in [1, 10]$ : Điểm do Gemini chấm cho ảnh thứ  $i$  theo tiêu chí  $c$ .
- $N$ : Tổng số lượt/mẫu ảnh được thử nghiệm (ở đây  $N = 8$ ).

### 2. Chênh lệch hiệu năng giữa hai mô hình ( $\Delta$ )

$$\Delta_c = S_c^{\text{LoRA}} - S_c^{\text{Base}}$$

Trong đó:

- $\Delta_c$ : Chênh lệch điểm trung bình của tiêu chí  $c$ .
- $S_c^{\text{LoRA}}$ : Điểm trung bình của mô hình LoRA.
- $S_c^{\text{Base}}$ : Điểm trung bình của mô hình Base.



- $\Delta_c > 0$ : LoRA vượt trội hơn Base
- $\Delta_c = 0$ : Hiệu năng tương đương
- $\Delta_c < 0$ : LoRA kém hơn Base



**Mục tiêu:** Đánh giá khách quan, nhất quán và tiết kiệm chi phí thông qua LLM-as-a-Judge (Gemini) trên thang điểm 10.

# SO SÁNH, ĐÁNH GIÁ KẾT QUẢ

BẢNG 1: ĐỒI CÁT QUANG PHÚ

| Tiêu chí đánh giá            | Base Model (TB) | LoRA Model (TB) | Chênh lệch ( $\Delta$ ) |
|------------------------------|-----------------|-----------------|-------------------------|
| 1. Subject & Domain Fidelity | 7.82            | 8.94            | +1.12                   |
| 2. Prompt Alignment          | 8.68            | 8.52            | -0.16                   |
| 3. Aesthetic & Composition   | 8.36            | 8.71            | +0.35                   |
| 4. Artifacts Control         | 8.24            | 8.48            | +0.24                   |
| AVERAGE SCORE                | 8.28            | 8.66            | +0.38                   |

BẢNG 2: ĐỘNG PHONG NHA

| Tiêu chí đánh giá            | Base Model (TB) | LoRA Model (TB) | Chênh lệch ( $\Delta$ ) |
|------------------------------|-----------------|-----------------|-------------------------|
| 1. Subject & Domain Fidelity | 7.38            | 8.32            | +0.94                   |
| 2. Prompt Alignment          | 8.22            | 8.11            | -0.11                   |
| 3. Aesthetic & Composition   | 8.16            | 8.02            | -0.14                   |
| 4. Artifacts Control         | 7.94            | 7.86            | -0.08                   |
| AVERAGE SCORE                | 7.93            | 8.08            | +0.15                   |



## 5. MỤC TIÊU TƯƠNG LAI & KẾT LUẬN



# ĐỊNH HƯỚNG PHÁT TRIỂN & MỞ RỘNG

Hướng tới hệ thống AI tạo poster du lịch Quảng Bình thông minh, tự động, đa dạng và dễ tiếp cận.

## TRAIN TRÊN NHIỀU ĐỊA DANH KHÁC

Mở rộng dữ liệu và huấn luyện trên nhiều địa danh nổi bật là Quảng Bình.



Đa dạng hòa dữ liệu giúp model hiểu sâu bởi cảnh, phong cảnh và đặc trưng từng nơi.

## KẾT HỢP VỚI AI THÀNH 1 AGENT

Chỉ cần nhập prompt, AI agent sẽ động sinh poster đẹp mà không cần chọn template thủ công.



Tạo nhiều template phức tạp hơn



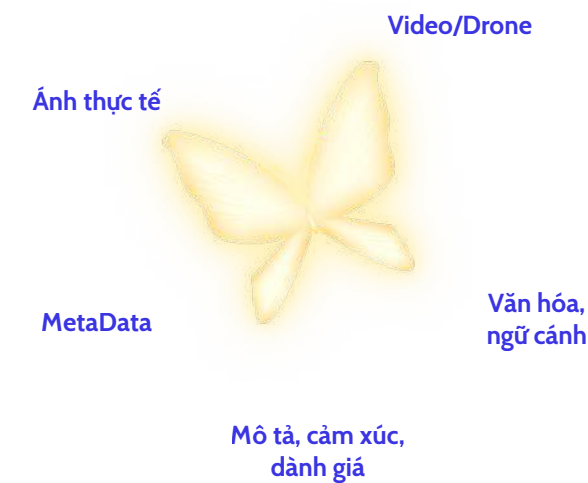
Bố cục linh hoạt, sáng tạo hơn font phù hợp



Tự động chọn cấu trúc, màu sắc, font phù hợp

## THU THẬP NHIỀU DATA THỰC TẾ HƠN

Thu thập nhiều dữ liệu thể đa dạng, chất lượng cao và huấn luyện trên tập data rộng hơn.



## ĐÓNG GÓI LẠI – TIỆN LỢI HƠN, DỄ TIẾP CẬN

Đóng gói thành sản phẩm hoàn chỉnh, dễ dùng cho mọi đối tượng.



Nền tảng web Nhanh chóng & dễ dùng Phù hợp với đối tượng Ổn định, bảo mật



# KẾT LUẬN



## 01. Làm chủ công nghệ cốt lõi

Nắm vững cơ chế hoạt động của mô hình Text-to-Image (Flux 1) và các kỹ thuật chính (LoRA, qLoRA).



## 02. Khắc phục điểm yếu sinh ảnh địa danh

Ứng dụng LoRA fine-tune mô hình thành công, giải quyết triệt để hạn chế của Flux gốc để tạo ra hình ảnh các địa danh sát với thực tế.



## 03. Giải quyết bài toán tạo Poster và Văn bản

Xây dựng cơ chế Template tự động, khắc phục hoàn toàn lỗi render chữ (text) lớn của AI, xuất ra PDF chuyên nghiệp.



## 04. Đóng gói hệ thống hoàn chỉnh

Tích hợp quy trình xử lý phức tạp thành một hệ thống duy nhất với giao diện (UI) đơn giản, trực quan và dễ tiếp cận.



## 05. Thành phẩm mang tính ứng dụng cao

Tạo ra được quy trình tự động hóa sản xuất poster hoàn chỉnh, hiện đại, đáp ứng trực tiếp câu thiết kế thực tế.



HOME

ABOUT

CONTENT

OTHERS

GOOD  
JOB!

XIN LỬA

# THANK YOU

HA

HA

HA

KHỦNG  
QUÁ

THANK YOU FOR LISTENING!❤️

PROJECT 7

ĐỜI BỐ  
BỐ QUẢN

SEE YOU!

GOODBYE  
EVERYONE!👋

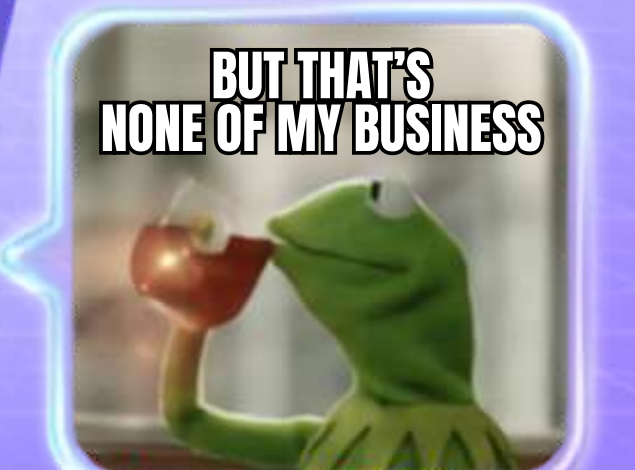
QNA TIME

# WHO ASKED?

Because great minds ask questions...  
and we answer (maybe).

PROJECT 7

?



## 1. PRETRAINING STAGE

Millions of image-text pairs



Learned from millions of image-text pairs.

## 2. PRETRAINED WEIGHTS

Zoom into the model's internal parameters



General knowledge is preserved.

## 3. WHY FREEZE THEM?

The frozen weights already enable a wide range of capabilities.



We freeze these weights because they already contain **valuable general knowledge**.



Only a small **LoRA adapter** will learn the new tourism style.



Millions of Image-Text Pairs



Pretraining (FLUX.1-dev)



Pretrained Weights



Freeze (Do not update)

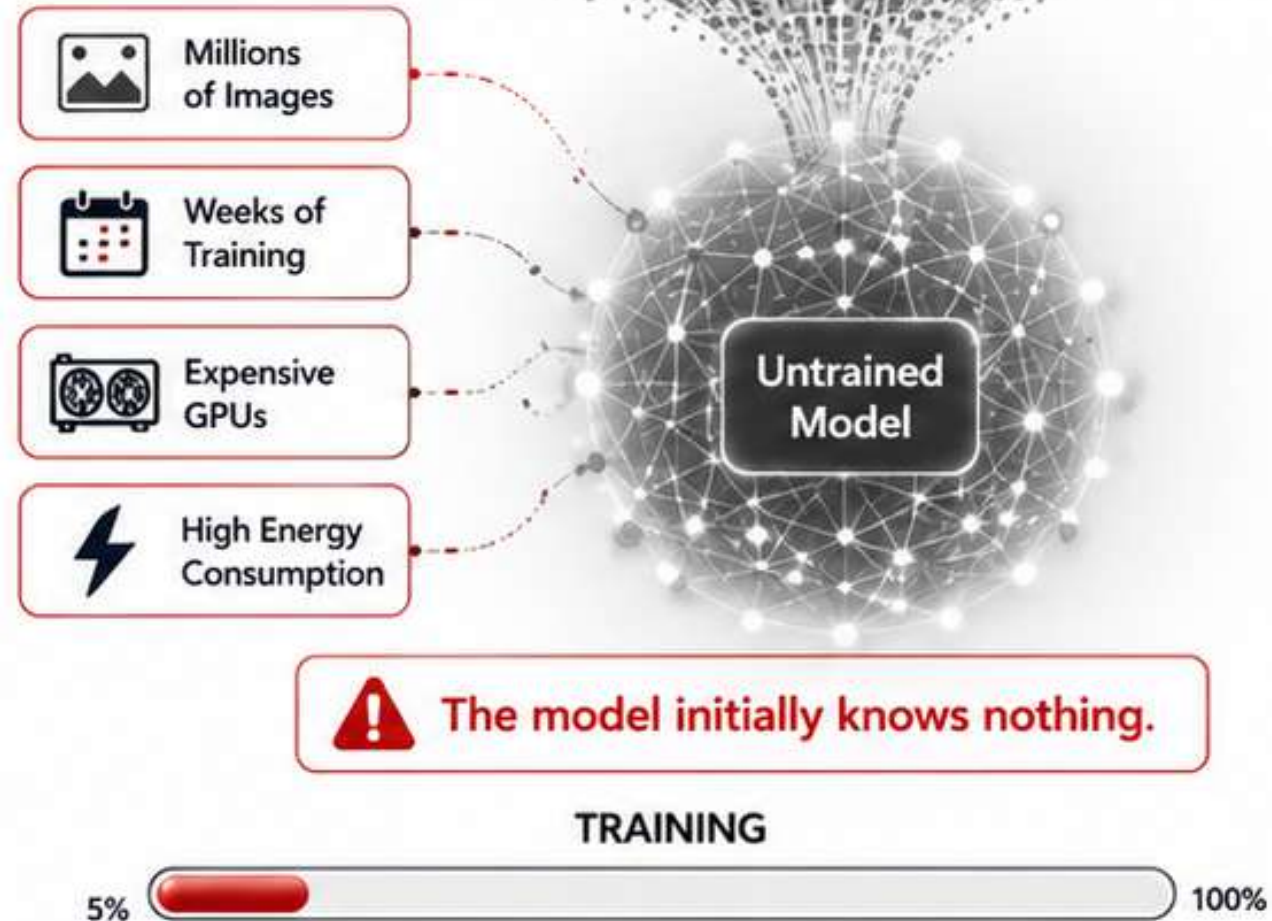


LoRA Learns New Style

We keep the **powerful general knowledge** and only learn **what is new**.

## TRAINING FROM SCRATCH

Millions of image-text pairs



# VS

### TRAINING FROM SCRATCH

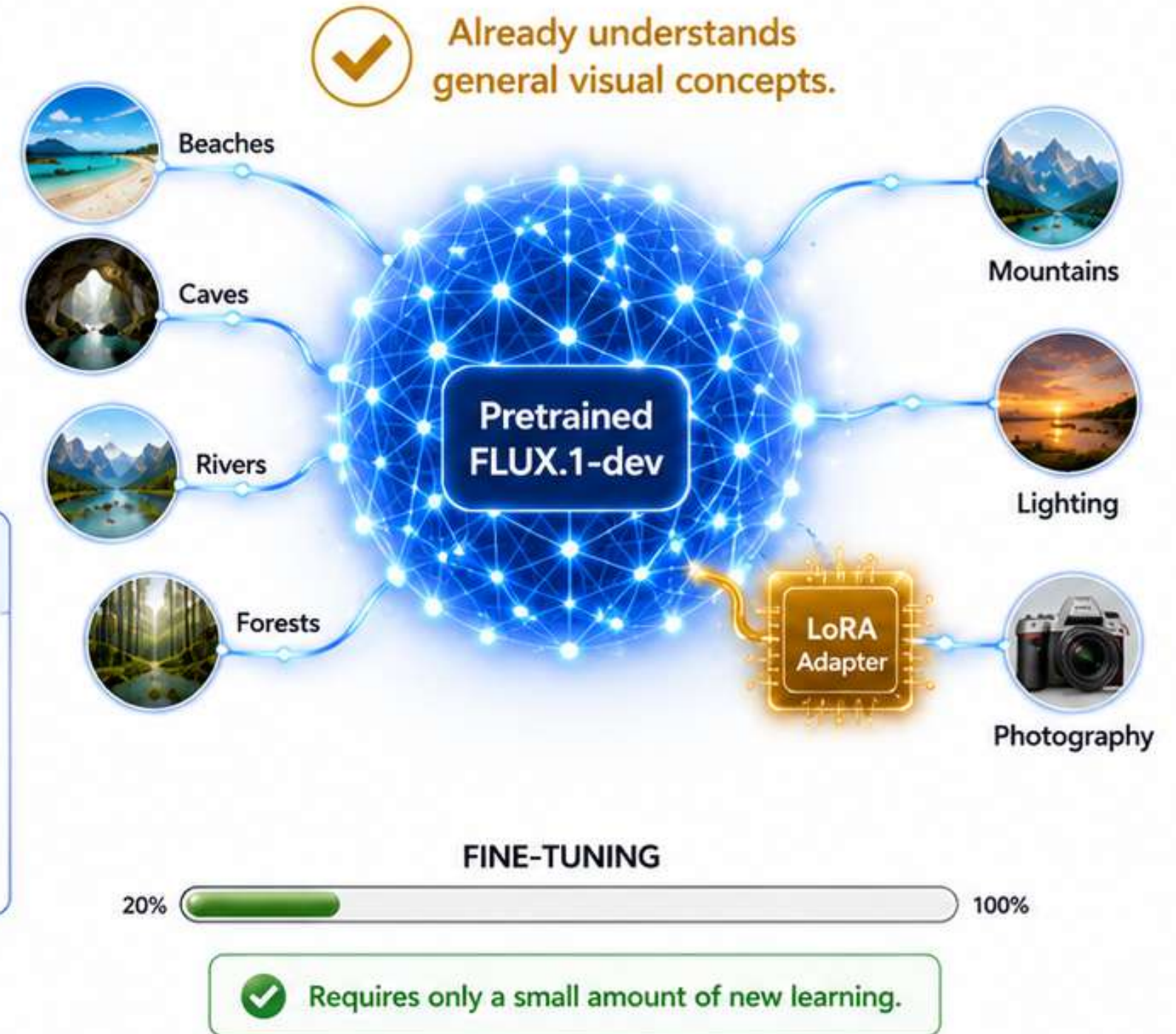
- ❌ Slow Training
- ❌ Expensive
- ❌ Millions of Images Required

### PRETRAINED MODEL + LoRA

- ✅ Fast Fine-Tuning
- ✅ Lower GPU Cost
- ✅ Small Dataset Required

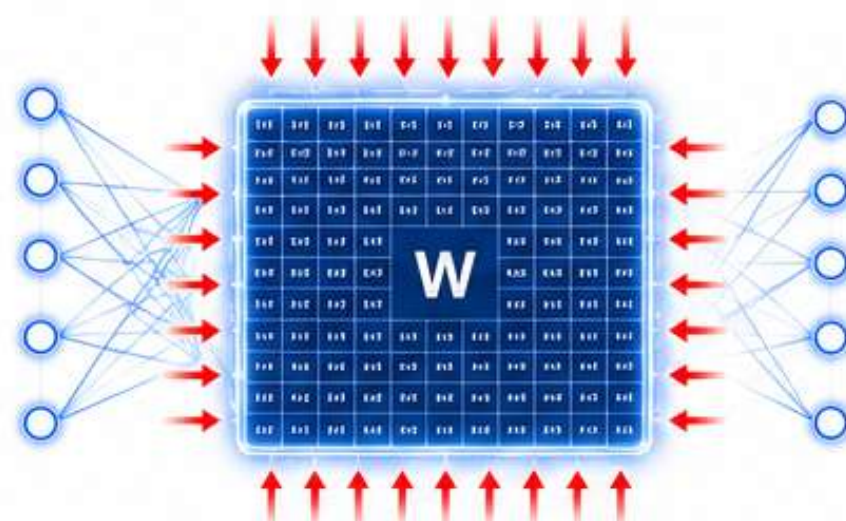
VS.

## USING A PRETRAINED MODEL



★ This is where **LoRA** becomes useful.

## 1. TRADITIONAL FINE-TUNING



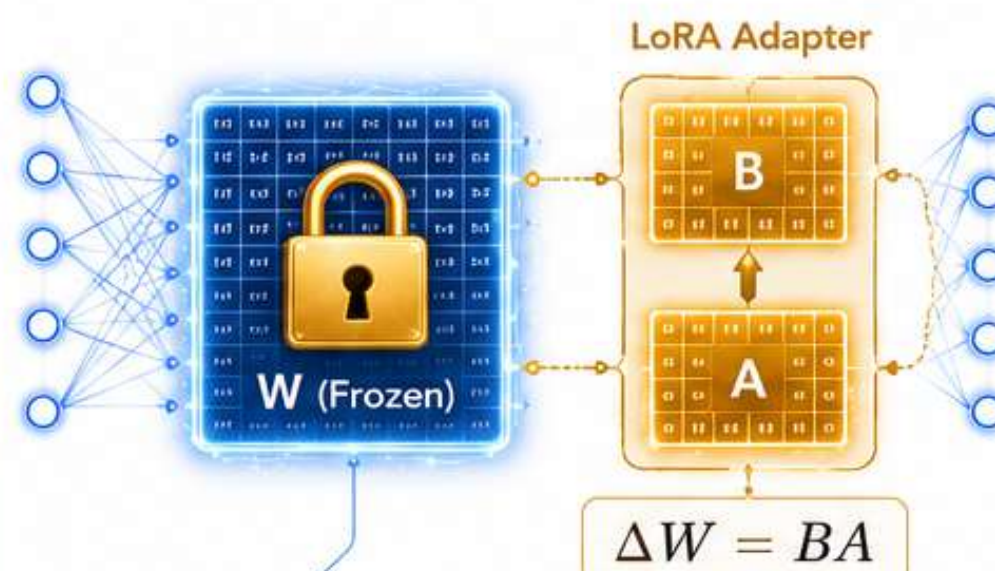
$$W \rightarrow W + \Delta W$$

⚠ Every weight is updated.



Updating billions of parameters requires significant computation and GPU memory.

## 2. LoRA (LOW-RANK ADAPTATION)



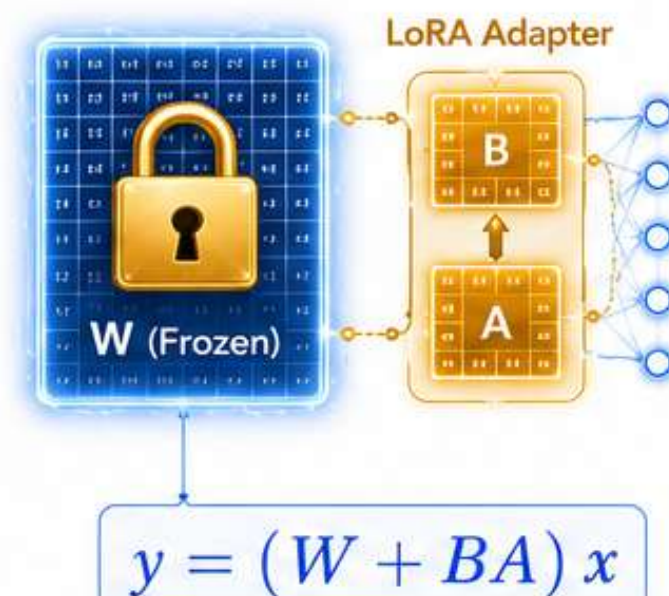
$W$  =  
Pretrained  
Knowledge

$A$  =  
Small  
Trainable Matrix

$B$  =  
Small  
Trainable Matrix

Only a lightweight adapter is learned.

## 3. FINAL MODEL (WITH LoRA)



The original knowledge **remains unchanged** while a lightweight adapter **learns** the new task.

Input  
Prompt



Frozen  
Pretrained Model

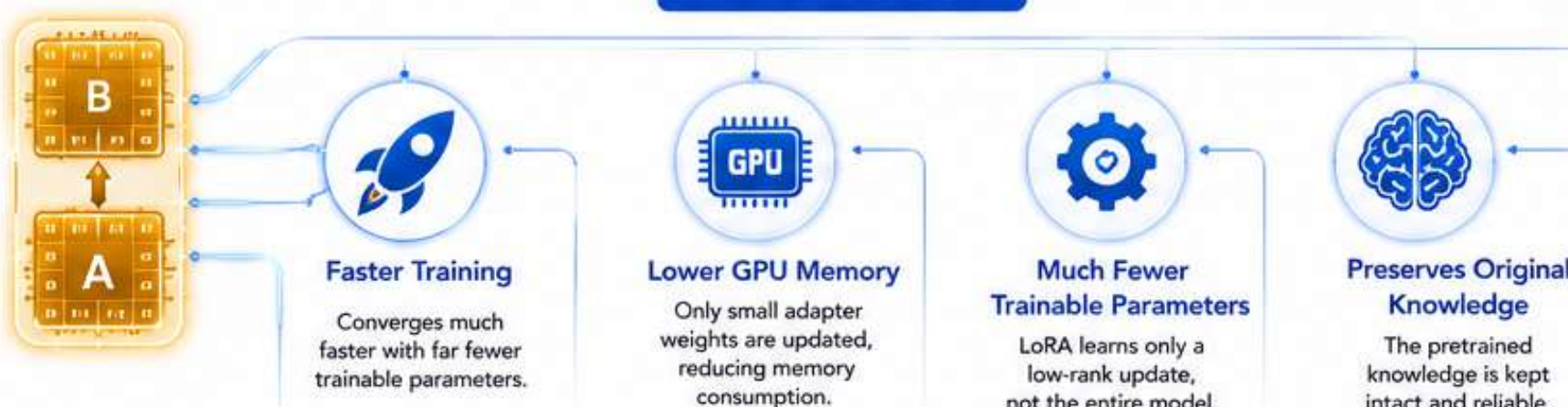


LoRA  
Adapter

Generated  
Quảng Bình  
Tourism Poster



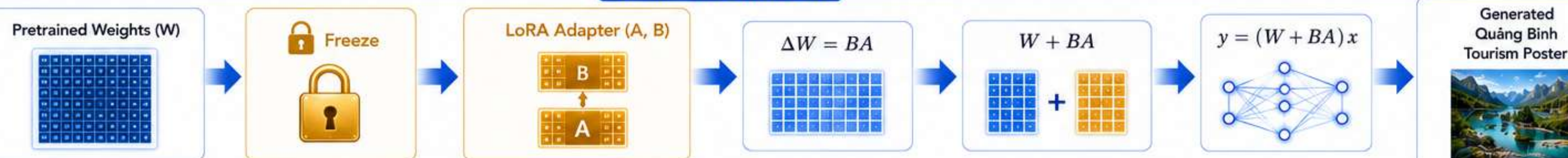
## 4. WHY LoRA?



## 5. COMPARISON

| Traditional Fine-Tuning              | LoRA                           |
|--------------------------------------|--------------------------------|
| Update all parameters                | Update only A and B            |
| High GPU memory                      | Low GPU memory                 |
| Slow training                        | Fast training                  |
| Risk of forgetting learned knowledge | Preserves pretrained knowledge |

## 6. LoRA WORKFLOW

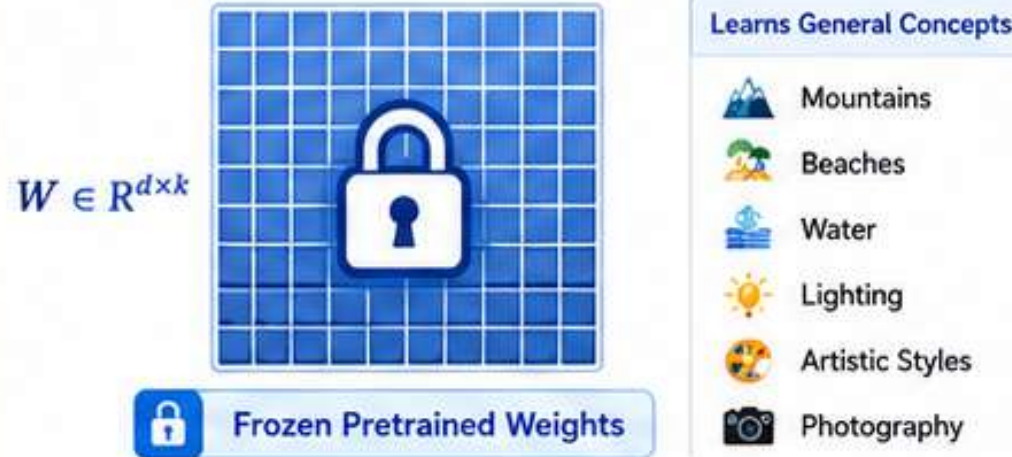


# Where Do the LoRA Matrices **A** and **B** Come From?

Understanding how LoRA creates and learns lightweight trainable parameters during fine-tuning.

## 1 The Pretrained Model

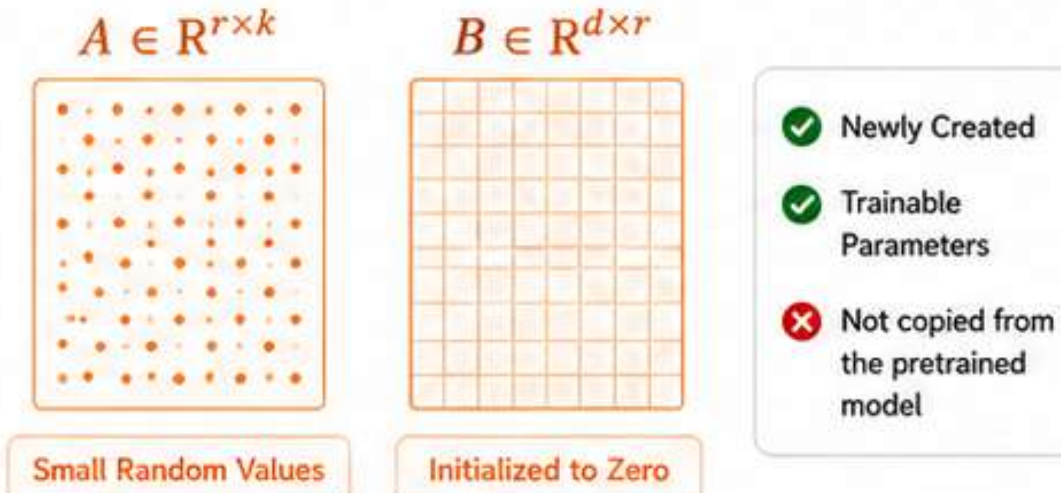
Flux.1-dev (Pretrained Model)



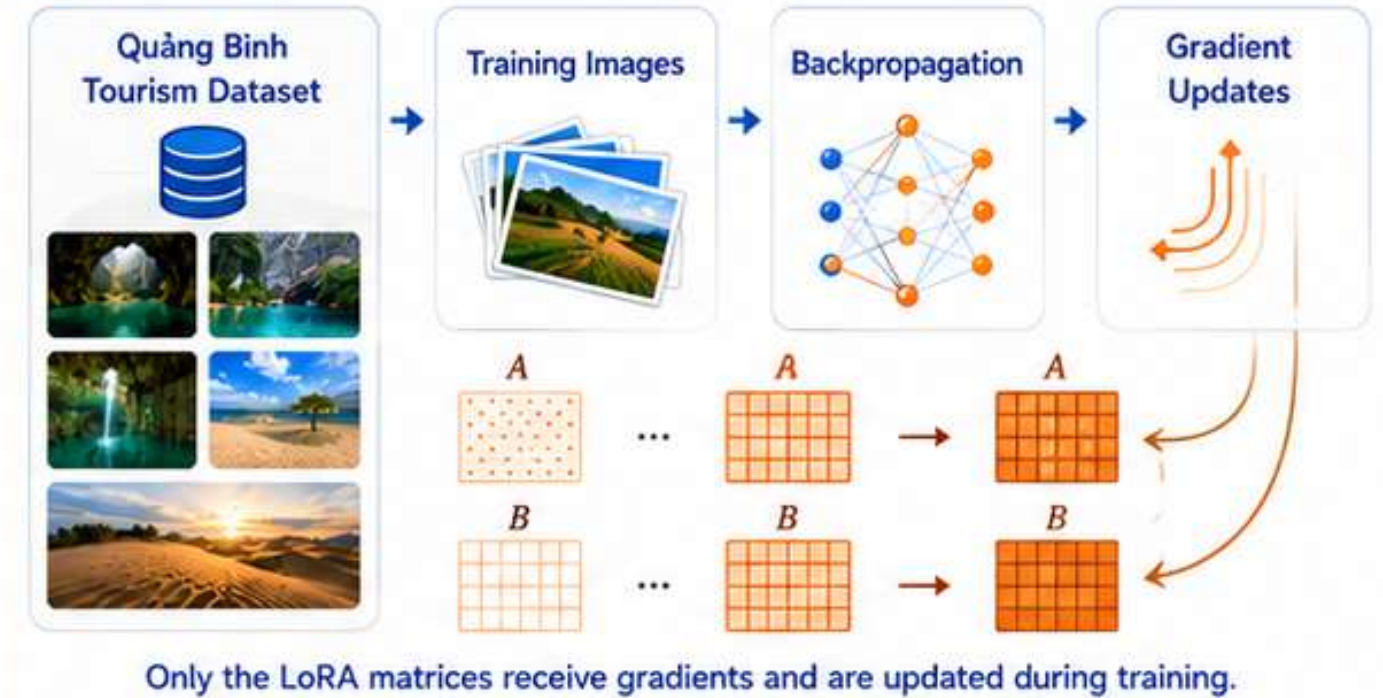
The pretrained model already contains general visual knowledge learned from millions of image-text pairs.

## 2 Creating LoRA

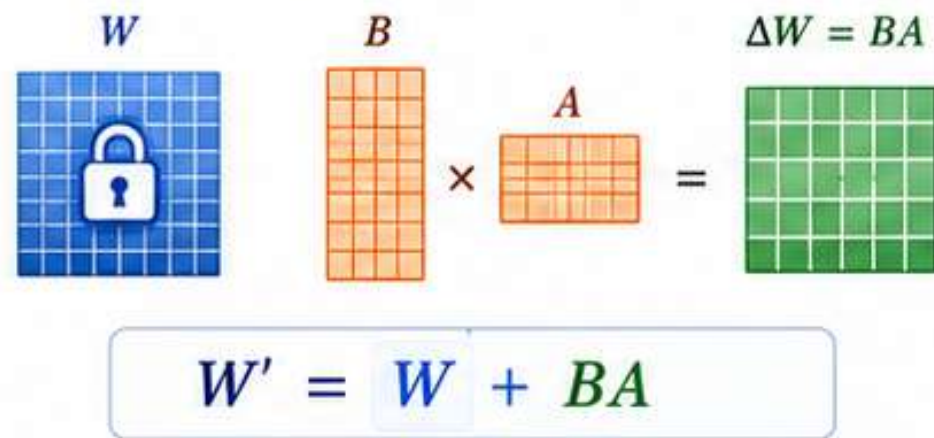
LoRA introduces new matrices **A** and **B** instead of modifying the pretrained weights.



## 3 Training Process

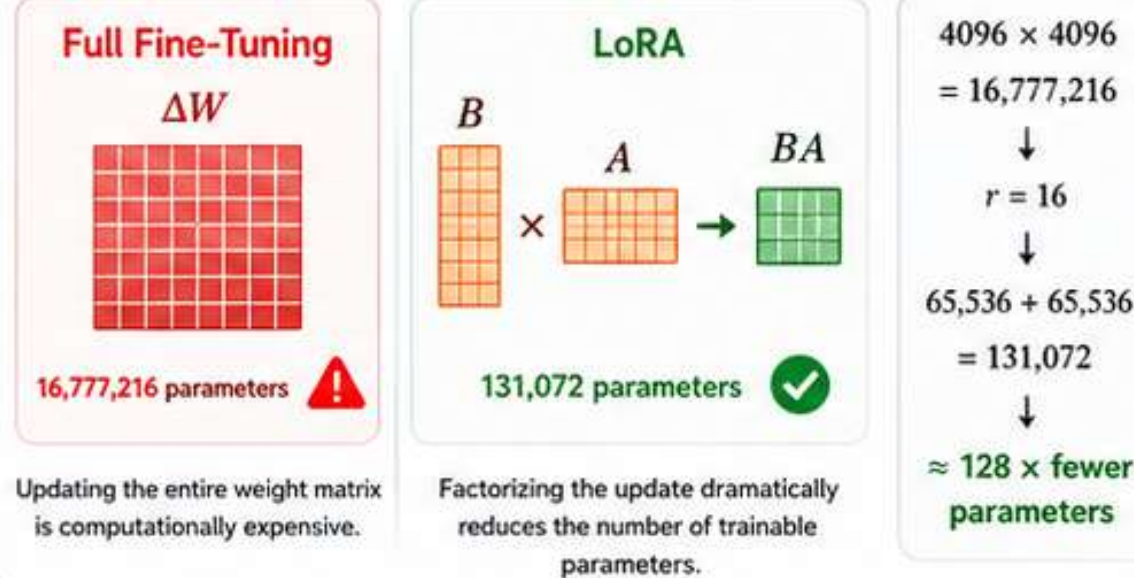


## 4 Low-Rank Update

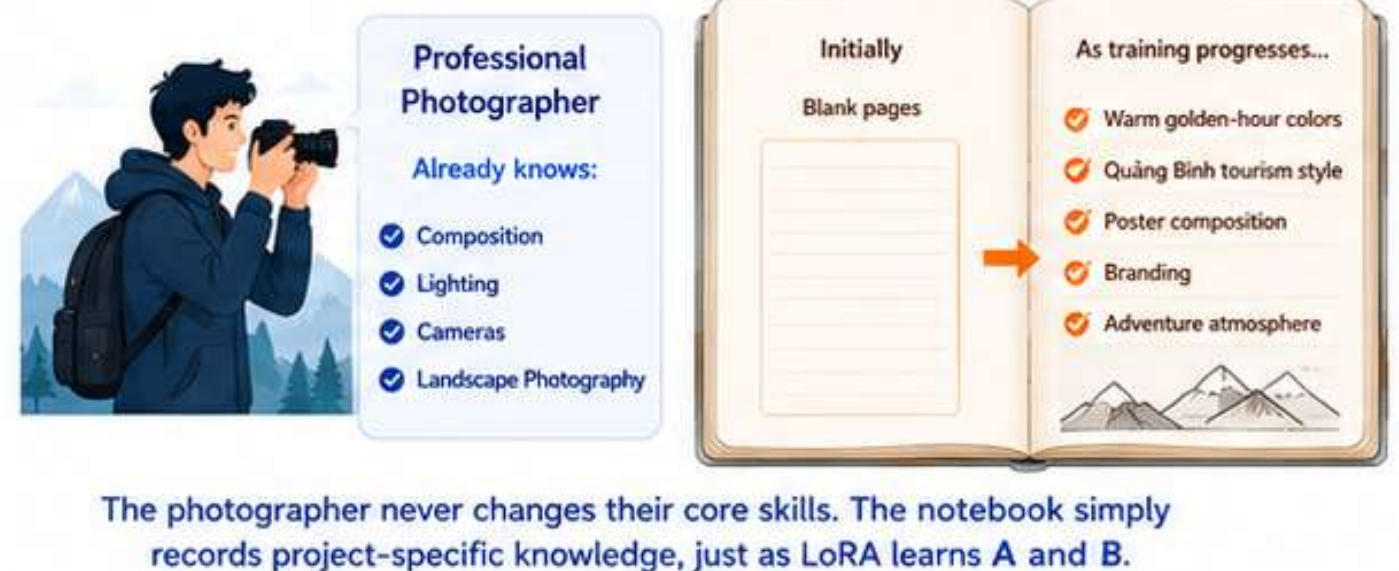


The product **BA** becomes a compact low-rank update that adapts the model without changing its original knowledge.

## 5 Why Two Matrices?



## 6 Photographer Analogy



### Key Takeaway

The matrices **A** and **B** are **not extracted** from the pretrained model. They are **newly created trainable parameters** introduced by LoRA.



During fine-tuning, only **A** and **B** are optimized through backpropagation.



The pretrained weights **W** remain **frozen** and keep their original knowledge.



Their product ( $\Delta W = BA$ ) becomes a lightweight update that efficiently adapts the model to a new task.

# How LoRA Controls Learning: Rank (r) and Scaling Factor (α)

Balancing model capacity and stable adaptation during fine-tuning

## 1 Low-Rank Decomposition

$$\Delta W = BA$$

Frozen Pretrained  
Weight Matrix  
 $W \in \mathbb{R}^{d \times k}$



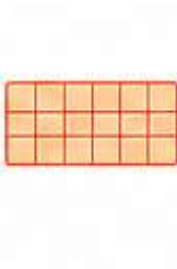
=

$B \in \mathbb{R}^{d \times r}$   
( $d \times r$ )



×

$A \in \mathbb{R}^{r \times k}$   
( $r \times k$ )



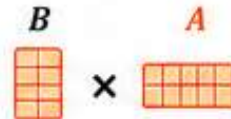
Frozen (Not Updated)

Trainable (Updated)

The rank (r) determines the size of the trainable LoRA adapter.

## 2 What Does Rank (r) Control?

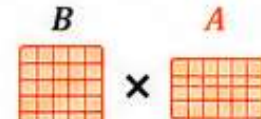
Low Rank  
 $r = 4$



- ◇ Few trainable parameters
- ⚡ Fast training
- 💾 Low GPU memory
- 🔒 Limited learning capacity

Fast

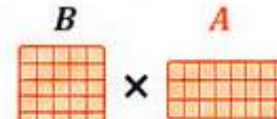
Medium Rank  
 $r = 16$



- ⚖️ Balanced adapter size
- 🕒 Good learning ability
- ⊕ Moderate memory
- 🌟 Best trade-off

Balanced

High Rank  
 $r = 64$



- 📁 More parameters
- 💻 Higher GPU memory
- 🏠 Can learn complex styles
- ⌚ Longer training time

High Capacity

Lower Capacity Higher Capacity

## 4 Why Scale the Update?

Small α

- Tiny update
- Slow learning
- Weak adaptation



Moderate α  
(Recommended)

- ✓ Balanced update
- ✓ Stable optimization
- ✓ Good generalization



Large α

- ✓ Very strong update
- ✓ Risk of unstable training
- ✓ Possible overfitting



α controls how strongly the learned update modifies the pretrained model.

## 5 Gradient Flow

Flux.1-dev  
(Pretrained Model)



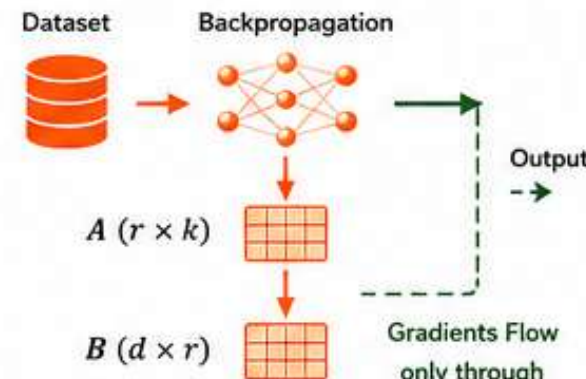
Frozen Weights (W)

No Gradient



Backpropagation updates only the LoRA parameters (A and B), while the scaling factor adjusts the magnitude of the learned update.

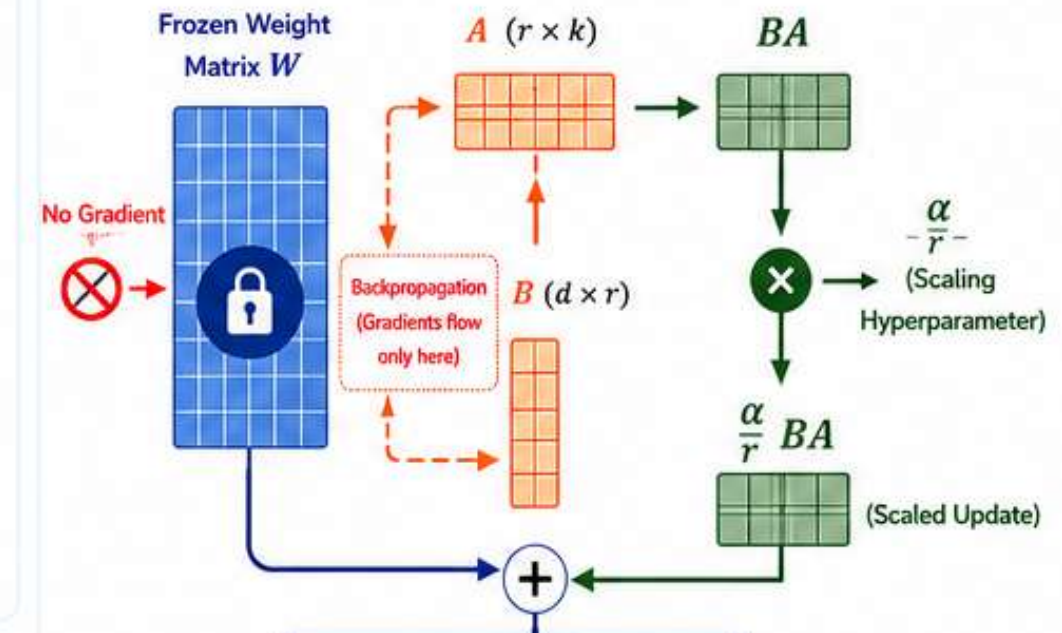
Training (Backpropagation)



Gradients Flow  
only through  
A and B

## 3 Scaling Factor (α)

$$W' = W + \frac{\alpha}{r} BA$$



$$W' = W + \frac{\alpha}{r} BA$$

(Updated Model)



The pretrained weight matrix  $W$  remains frozen. During training, only the LoRA matrices  $A$  and  $B$  receive gradient updates. The scaling factor  $\alpha$  adjusts the strength of the learned update before it is added to  $W$ .



Key  
Takeaways



Rank (r) determines the capacity of the LoRA adapter by controlling the number of trainable parameters.



Higher rank allows learning more complex patterns but increases computational cost and memory usage.



The scaling factor (α) adjusts the strength of the learned update before it is added to the frozen pretrained weights.



Together,  $r$  and  $\alpha$  balance efficiency, stability, and adaptation quality for the best fine-tuning performance.

The LoRA Update Equation

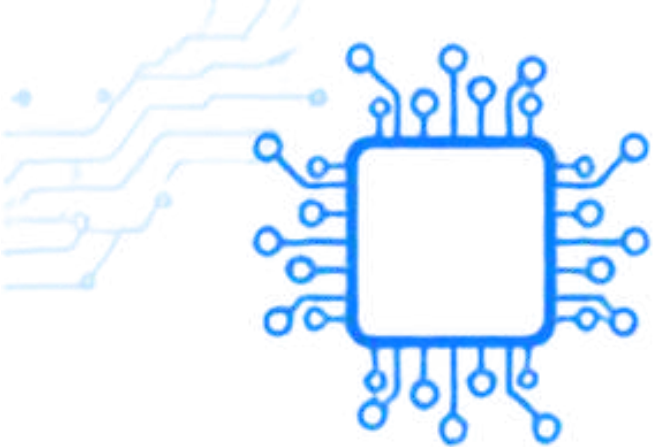
$$W' = W + \frac{\alpha}{r} BA$$

r : Low-rank  
dimension

α : Scaling  
factor

BA : Learned  
adaptation

W : Frozen  
weights

[HOME](#)[ABOUT](#)[CONTENT](#)[OTHERS](#)

# PHƯƠNG PHÁP



## 1. FLOW MATCHING



Huấn luyện mô hình sinh ảnh học vector biến đổi từ nhiều thành ảnh.



Giảm sai khác giữa hướng đi chuyên dự đoán và mục tiêu.



Tối ưu hàm loss để mô hình học chính xác quá trình sinh ảnh.



Dùng prompt với flux để tạo nhiều ngẫu nhiên và flow matching ra ảnh hoàn chỉnh.



Lựa chọn flow matching phù hợp với kiến trúc flux và hỗ trợ sinh hiệu quả.



Kết quả ảnh hoàn chỉnh, chất lượng cao và ổn định.

## 2. LORA TRAINING



Dùng LORA tính chính flux theo dữ liệu du lịch Quảng Bình.



Chỉ cập nhật phần nhỏ tham số thay vì toàn bộ mô hình.



Ưu điểm: giảm thời gian training, giảm bộ nhớ và tài nguyên GPU.



Giúp mô hình học đặc trưng về địa danh, cảnh quan và văn hóa Quảng Bình.



Tăng khả năng sinh chân thực, mang đậm bản sắc du lịch Quảng Bình.



# Trọng số trong mạng neuron



## 1. TRỌNG SỐ LÀ GÌ?



Trọng số quyết định lượng nội dung được truyền



💡 Thay đổi trọng số sẽ thay đổi cách model hiểu dữ liệu đầu vào

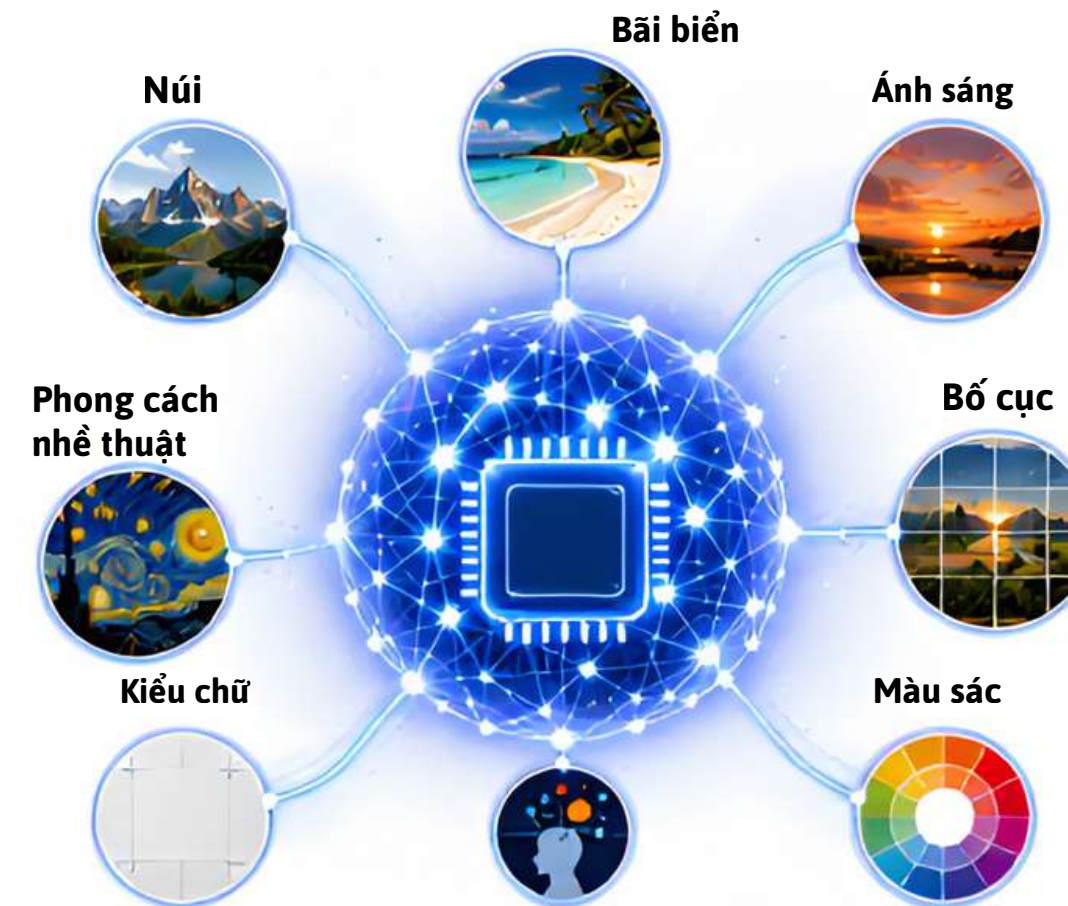
## 2. MA TRẬN TRỌNG SỐ

### Hàng tỷ trọng số

Các mô hình AI hiện đại (đặc biệt là mô hình khách tán – Diffusion Models) chứa hàng tỷ tham số số.



## 3. TRỌNG SỐ LƯU TRỮ TRI THỨC



Trọng số chính là “bộ nhớ” của mạng nơ-ron.

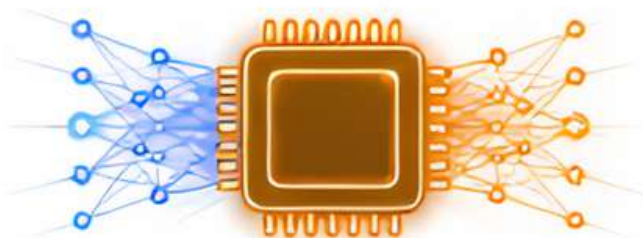
Ảnh/Văn bản đầu vào



Biển lúc hoàng hôn



Trọng số (Bộ nhớ)



Mô hình diễn giải



Kết quả được tạo ra

