# **Reinforcement Learning for Multimodal Reasoning in Multimodal Large Language Models**

## **1. Trang bìa**

**Đề tài:** Reinforcement Learning for Multimodal Reasoning in Multimodal Large Language Models  
**Mô hình sử dụng:** Qwen-VL-2B-Instruct  
**Sinh viên thực hiện:** Phạm Thế Anh  
**Lớp:** AI17D.DS  
**Giáo viên hướng dẫn:** Nguyễn An Khương  
**Thời gian:** Tháng 7/2025

## **2. Tóm tắt**

Dự án nhằm khai thác khả năng suy luận đa phương thức (multimodal reasoning) của các MLLM (Multimodal Large Language Models), đặc biệt là Qwen-VL-2B-Instruct, thông qua Reinforcement Learning (RL) có đối sánh (DPO - Direct Preference Optimization). Dự án thiết lập một pipeline huấn luyện với Unsloth + DPO, kết hợp dữ liệu ảnh-văn bản tự tạo. Kết quả thể hiện sự cải thiện về tính nhất quán và hợp lý trong câu trả lời của mô hình sau huấn luyện.

## **3. Giới thiệu**

### **3.1 Bối cảnh**

* Multimodal LLMs (MLLMs) là xu hướng then chốt trong AI, nhưng còn chưa tối ưu hoá về suy luận logic.
* Suy luận đa bước (multi-step reasoning) khi đồng thời xử lý hình ảnh + văn bản là thách thức.
* RLHF được xem là hướng tiếp cận tự nhiên nhưng tốn kém.
* DPO giải quyết điểm yếu: không cần reward model, training nhanh, ổn định hơn.

### **3.2 Mục tiêu**

* Fine-tune Qwen-VL-2B-Instruct với Reinforcement Learning.
* Tối ưu hóa khả năng suy luận trên đầu vào đa modal.
* Đánh giá hiệu quả huấn luyện DPO với tính nhất quán và logic.

## **4. Cơ sở lý thuyết**

### **a. DPO (Direct Preference Optimization)**

**DPO** là một phương pháp tối ưu hóa trực tiếp dựa trên **preference (sở thích)** mà **không cần reward model** như PPO hay Reinforcement Learning truyền thống.

#### **🔍 Cơ chế hoạt động:**

* Dựa trên cặp dữ liệu (chosen, rejected) — ví dụ, hai phản hồi khác nhau cho cùng một prompt.
* Loss function:  
   LDPO=−log⁡σ(β(rθ(x,ychosen)−rθ(x,yrejected)))\mathcal{L}\_{\text{DPO}} = -\log \sigma(\beta (r\_\theta(x, y\_{\text{chosen}}) - r\_\theta(x, y\_{\text{rejected}})))LDPO​=−logσ(β(rθ​(x,ychosen​)−rθ​(x,yrejected​)))  
   Trong đó:  
  + σ\sigmaσ: sigmoid
  + rθr\_\thetarθ​: logit score từ model (không cần reward model riêng)
  + β\betaβ: hệ số nhiệt độ (temperature), kiểm soát độ nhạy.

#### **✅ Ưu điểm:**

* Không cần huấn luyện reward model.
* Dễ huấn luyện, dễ scale.
* Có thể kết hợp với LoRA và 4-bit để tiết kiệm tài nguyên.

#### **🧠 Áp dụng cho Qwen-VL:**

* Qwen-VL xử lý **cả ảnh và văn bản**.

Bạn cần dataset có format:  
  
  
  
{

"prompt": "Câu hỏi + hình ảnh",

"chosen": "Câu trả lời tốt",

"rejected": "Câu trả lời kém hơn"

}

* Ảnh được encode qua ViT + Q-Former -> rồi nối với prompt.

### **🖼️ b. Qwen-VL (e.g., Qwen-VL-2B-Instruct)**

#### **Kiến trúc:**

* **ViT image encoder** (Vision Transformer): trích xuất đặc trưng ảnh.
* **Q-Former**: module chuyển feature ảnh thành token có thể xử lý như văn bản.
* **Transformer decoder-only (2B)**: tiếp nhận các token ảnh + text.

#### **Dòng Instruct:**

* Đã được fine-tune để **trả lời các prompt dạng instruction**.
* Phù hợp cho các tác vụ như: **VQA, captioning, OCR, visual reasoning**.

### **⚡ c. Unsloth**

[Unsloth](https://github.com/unslothai/unsloth) là thư viện được tối ưu hóa để huấn luyện **mô hình LLM/MLLM nhanh và tiết kiệm tài nguyên**.

#### **💡 Tính năng nổi bật:**

* **Tối ưu cho T4 / A100 / Colab**.
* Tích hợp dễ dàng với Hugging Face (SFTTrainer, DPOTrainer).
* Hỗ trợ:  
  + **LoRA** (Parameter-efficient tuning).
  + **QLoRA (4-bit quantization)**.
  + **Fast tokenizer / Fast model wrapper** (giảm overhead tính toán).

#### **🧪 Với Vision-Language:**

* Hỗ trợ mô hình như **Qwen-VL-2B** (có phần vision encoder).
* Cung cấp UnslothVisionDataCollator để xử lý batch chứa ảnh + text.

### **🧩 d. LoRA + 4-bit (QLoRA)**

#### **📌 LoRA (Low-Rank Adaptation):**

* Chỉ fine-tune **một vài layer nhỏ** (matrices A, B) chèn vào các linear layer.
* Giảm số lượng tham số cập nhật xuống rất nhỏ (~1%-5%).
* Rất tiết kiệm bộ nhớ và thời gian.

#### **📌 4-bit Quantization (QLoRA):**

* Thay vì lưu trọng số float32 → float16 → giờ thành **int4**.
* Kết hợp **nf4 quantization** + **double quantization** để giữ hiệu năng.
* Có thể load **mô hình 7B** trên **T4 15GB VRAM**.

#### **📦 Tích hợp:**

* Thư viện bitsandbytes + transformers hỗ trợ QLoRA.
* Dễ dùng với load\_in\_4bit=True, bnb\_config=....

### **5. Phương pháp**

#### **5.1 Dữ liệu gốc**

* **Nguồn**: [Traffic Detection Project – Kaggle](https://www.kaggle.com/datasets/yusufberksardoan/traffic-detection-project)
* **Số lượng ban đầu**:  
  + Train: 5805 ảnh
  + Validation: 549 ảnh
  + Test: 279 ảnh
* **Sau khi lọc**:  
  + Train: 852 ảnh
  + Validation: 85 ảnh
  + Test: 50 ảnh
* **Lý do lọc dữ liệu**:  
  + Loại bỏ ảnh mờ, nhãn sai.
  + Đảm bảo tính đa dạng về loại phương tiện, thời tiết, bối cảnh,...

#### **5.2 Tạo dữ liệu huấn luyện (Gemini API)**

* Viết các template prompt yêu cầu suy luận, ví dụ:  
  + “Có bao nhiêu phương tiện đang di chuyển?”
  + “Thời điểm trong ngày khi chụp ảnh là gì?”
* Sử dụng Gemini API để sinh nhiều câu trả lời cho mỗi prompt.  
  + Gán “chosen” là câu trả lời hợp lý nhất.
  + Gán “rejected” là câu trả lời ít logic hơn.
* Đánh giá chất lượng câu trả lời bằng semi-automatic labeling.

#### **5.3 Chuẩn hóa dữ liệu sang định dạng DPO**

* Dạng .jsonl, mỗi mẫu gồm:

{

"prompt": "Tình trạng giao thông hiện tại trong ảnh là gì? <image>",

"chosen": "Tình trạng giao thông đông đúc với nhiều xe đang di chuyển gần nhau.",

"rejected": "Trong ảnh không có xe nào, chỉ có người đi bộ."

}

#### **5.4 Cấu hình huấn luyện**

* **Mô hình**: Qwen-VL-2B-Instruct (load bằng FastLanguageModel)
* **Huấn luyện**:  
  + Trainer: DPOTrainer
  + Epochs: 3
  + Batch size: 2
  + Learning rate: 1e-5
  + GPU: Tesla T4

## **6. Triển khai**

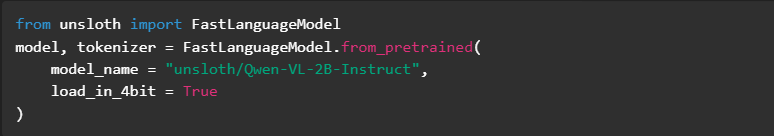
### 

### **Cài đặt:**

****

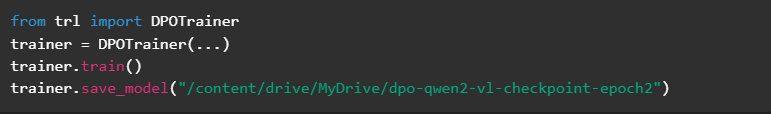
### 

### **Load model:**

****

### 

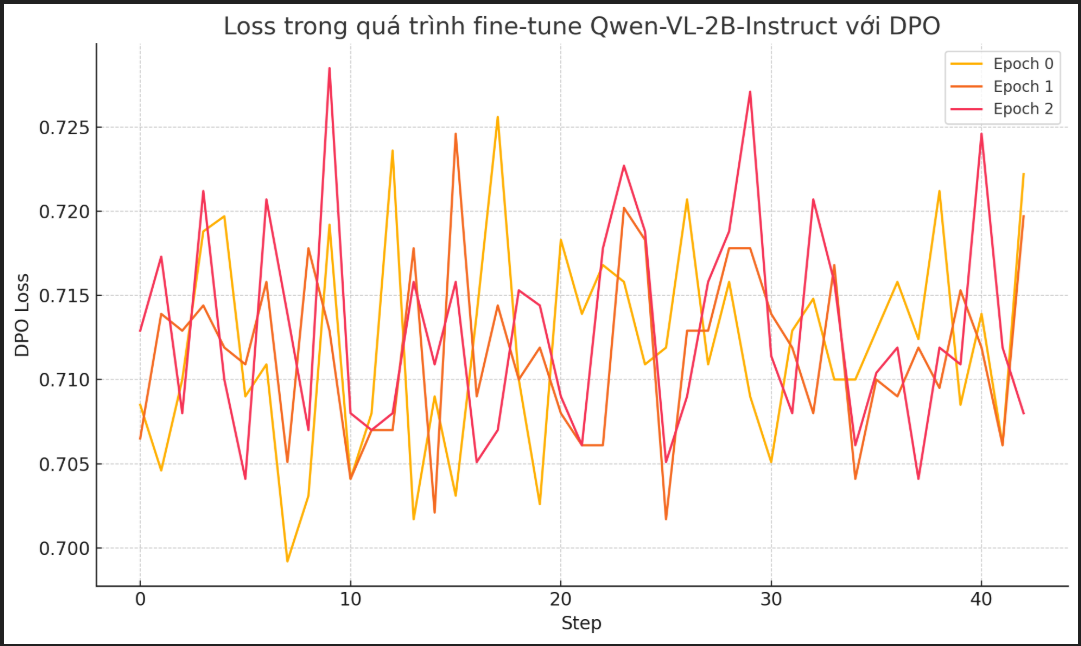
### **Huấn luyện:**

****

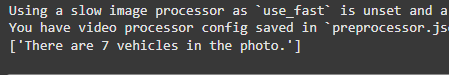
## 

## 

## **7. Kết quả**



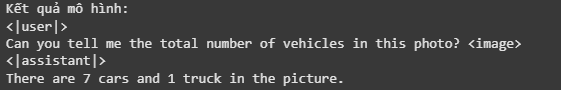
## **Trước fine-tune ( mô hình gốc) :**



### 

### 

### **Sau fine-tune:**

****

## **8. Kết luận**

### **Đạt được:**

* **Fine-tune nhị gọn bằng DPO**
* **Cải thiện khả năng suy luận trên multimodal**
* **Dễ triển khai với GPU nhỏ**

### **Hạn chế:**

* **Dataset chỉ 1000 ảnh**
* **Cần GPU T4 trở lên**

### **Hướng phát triển:**

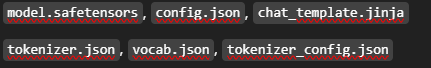
* **Sử dụng PPO, RLHF**
* **Sinh data preference tự động**
* **DPO + LoRA multi-adapter**

## **9. Phụ lục**

### **Cấu hình Colab:**

* **GPU: Tesla T4**
* **RAM: 15GB**

### **File quan trọng:**

****

### **Code viết thủ công:**

* **Tự viết DPO collator, trainer, runner script**

## **Tài liệu tham khảo**

**Unsloth – Thư viện huấn luyện LoRA hiệu suất cao:  
 →** [**https://github.com/unslothai/unsloth**](https://github.com/unslothai/unsloth)

**Qwen-VL-2B-Instruct – Mô hình đa phương thức open-source của Alibaba Cloud:  
 →** [**https://huggingface.co/Qwen/Qwen-VL-2B-Instruct**](https://huggingface.co/Qwen/Qwen-VL-2B-Instruct)

**DPO Paper (Direct Preference Optimization) – Arxiv 2023:  
 →** [**https://arxiv.org/abs/2305.18290**](https://arxiv.org/abs/2305.18290)

**Transformers Documentation by Hugging Face:  
 → https://huggingface.co/docs/transformers/index**

**PEFT (Parameter-Efficient Fine-Tuning) – LoRA & adapters:  
 →** [**https://github.com/huggingface/peft**](https://github.com/huggingface/peft)

**BitsAndBytes (4-bit Quantization):  
 →** [**https://github.com/TimDettmers/bitsandbytes**](https://github.com/TimDettmers/bitsandbytes)