

# DEEPEYES EVAL 流程介绍

基于强化学习 (RL) 的多模态代理项目

---

尹超

<https://github.com/Visual-Agent/DeepEyes>

# CONTENTS

---

1. DeepEyes 项目介绍
2. 评估流程介绍 (Evaluation)
3. 数据结构分析
  - 输入数据结构 (Input Data)
  - 输出数据结构 (Output Data)

# DEEPEYES 项目介绍

---

DeepEyes 构建在 **VeRL** (Volcano Engine Reinforcement Learning) 框架之上。

- 核心目标: 训练多模态代理 (Agent), 使其能够在推理链中直接整合视觉信息。
- 能力: 主动调用 `image_zoom_in_tool` 查看细节, 无需依赖冷启动或监督微调。

### 核心机制

端到端强化学习 (End-to-End RL): 利用结果奖励信号引导模型涌现出视觉搜索、比较等能力。激励模型“通过图像思考” (**Thinking with Images**)。

# 评估流程介绍 (EVALUATION)

---

DeepEyes 评估采用 **两步法 (Two-step Approach)**:

1. 推理 (Inference): 模型生成回答 (含 Thinking & Tool Call)。
2. 打分 (Scoring): 使用 LLM-as-a-judge (通常为 Qwen-2.5-72B)。

## 环境准备

先拉取官方 docker image 后，进入 workspace，再确保下载了相关模型和数据集（例如推理模型下载命令如下）：

```
HF_ENDPOINT=https://hf-mirror.com huggingface-cli download  
ChenShawn/DeepEyes-7B \  
--local-dir ~/models/DeepEyes-7B \  
--resume-download
```

评估脚本通过 OpenAI-compatible API (vLLM) 通信。需启动两个服务：

1. Target Model (被评估模型)

```
vllm serve /path/to/DeepEyes-7B-v1 \
--port 8000 \
--trust-remote-code \
--gpu-memory-utilization 0.9
```

2. Judge Model (打分模型)

```
vllm serve /path/to/Qwen-2.5-72B-Instruct \
--port 18901 \
--served-model-name "judge" \
--tensor-parallel-size 4
```

```
python eval/eval_vstar.py \
--model_name "DeepEyes-7B-v1" \
--api_key "EMPTY" \
--api_url "http://localhost:8000/v1" \
--vstar_bench_path "/path/to/vstar" \
--save_path "./results" \
--eval_model_name "default" \
--num_workers 8
```

### 参数解释:

- `model_name`: 输出文件夹名，用于区分实验。
- `api_key`: 鉴权密钥，本地一般用“EMPTY”。
- `api_url`: 被评测模型 API 地址。
- `vstar_bench_path`: V\* 数据集本地路径。
- `save_path`: 结果保存根目录。
- `eval_model_name`: 请求体中的 `model` 字段。
- `num_workers`: 并发推理线程数。

```
python eval/judge_result.py \
--model_name "DeepEyes-7B-v1" \
--api_key "EMPTY" \
--api_url "http://localhost:18901/v1" \
--save_path "./results" \
--eval_model_name "judge" \
--num_workers 8
```

#### 参数解释:

- `model_name`: 需与 Step 1 一致, 定位中间结果。
- `api_key`: 鉴权密钥。
- `api_url`: 打分模型 (Judge) API 地址。
- `save_path`: 结果读取目录。
- `eval_model_name`: Judge 模型在 vLLM 的服务名。
- `num_workers`: 并发打分线程数。

- 中间结果: `save_path/model_name/result_*.jsonl`
  - ▶ 包含: Prompts, Raw Model Outputs, Parsed Answers.
- 最终分数: `save_path/model_name/final_acc.json`
  - ▶ 生成于打分步骤之后。
  - ▶ 包含: 各类别准确率及总体准确率。

Usage Tip: 确保 `save_path` 和 `model_name` 在两步中完全一致，否则打分脚本无法找到推理结果。

# 数据结构分析

---

训练数据通常为 **Parquet** 格式，每条数据代表一个训练样本。

- `data_source`: 数据来源标识 (e.g., "rag\_v2-train").
- `prompt`: 对话历史 (Role: system, user).
- `env_name`: 关键字段。指定工具环境 (e.g., "visual\_toolbox\_v2").
- `reward_model`: 奖励计算配置 (e.g., ground\_truth).
- `images`: 图像数据 (路径或二进制)。

参考: [verl/workers/agent/envs/visual\\_agent/generate\\_trainset.py](#)

```
{  
  "data_source": "deepeyes-train",  
  "prompt": [  
    { "role": "system", "content": "You are a helpful assistant..." },  
    { "role": "user", "content": "What is the color of the flower?..." }  
],  
  "env_name": "visual_toolbox_v2",  
  "ability": "qa",  
  "reward_model": {  
    "style": "rule",  

```

结果保存为 JSONL 文件，每行一个 JSON 对象。

- `image`: 输入图像文件名
- `question`: 输入问题
- `answer`: 标准答案 (Ground Truth)
- `status`: 执行状态
- `pred_ans`: 模型预测并提取的简短答案
- `pred_output`: 完整的对话交互记录
  - 包含 思维链 `<think>`
  - 包含 工具调用 `<tool_call>`

参考: `results/deepeyes-7b/  
result_direct_attributes_deepeyes-7b.jsonl`

```
{  
    "image": "sa_4988.jpg",  
    "question": "What is the color of the flower?",  
    "answer": "The color of the flower is white.",  
    "pred_ans": "white",  
    "pred_output": [  
        {  
            "role": "assistant",  
            "content": "<think>\nI need to check the details of the flower center.\n</think>\n<tool_call>\n{ \"name\": \"image_zoom_in_tool\", \"arguments\": {...} }\n</tool_call>\n<answer>The color of the flower is white.</answer>"  
        }  
    ],  
    "status": "success"  
}
```