

自然语言处理导论

MiniGPT4-Finetuning 项目报告



Author: 夏子渊 金裕涵 林滨 程韬 潘越

Date: 2025-06-20

2025 春夏 Semester

Table of Contents

1. Abstract	3
2. Project Introduction	3
3. Technical Details	3
3.1. 理论知识	3
3.1.1. 指令微调	3
3.1.2. 视觉语言模型	4
3.1.3. 评估指标	4
3.2. 技术细节	5
3.2.1. 转换数据集	5
3.2.2. 实验流程	6
3.2.3. 评估方法与解码策略	6
4. Experiment Results	7
4.1. 实验流程	7
4.1.1. Conda 环境	7
4.1.2. 预训练模型获取	7
4.1.3. 数据准备	7
4.1.4. 指令微调	7
4.1.5. 评估	9
4.2. 评估结果与分析	10
4.2.1. 模型性能持续提升	10
4.2.2. 训练动态揭示关键拐点	10
4.2.3. 局限性与后续工作	11
附录	12
References	12

1. Abstract

MiniGPT4-Finetuning 项目旨在基于视觉语言模型 MiniGPT-4[1]，对 **Flickr30k**[2] 数据集开展指令微调，以增强图像-文本理解与生成能力。本文档总结了项目背景、环境配置、模型下载、微调流程、评估结果与未来工作等内容，为后续复现与迭代提供参考。

2. Project Introduction

MiniGPT-4 通过结合视觉编码器与 Vicuna 语言模型，实现了图像到文本的高质量对齐。为了进一步提升其在 Image Captioning 任务上的理解与表现，我们开展了面向 Flickr30k 数据集的指令微调。

实验环境:

- 操作系统: Ubuntu 18.04 LTS
- 显卡: NVIDIA GeForce RTX 3090 \times 7
- Python: 3.10.13
- CUDA: 11.7
- PyTorch: 2.0.1

3. Technical Details

3.1. 理论知识

3.1.1. 指令微调

指令微调 (Instruction Tuning) 是指在大规模预训练语言模型的基础上，利用带有明确指令 (instruction) 的数据对模型进行再训练，使其能够更好地理解和执行各种自然语言指令。其核心思想是通过多样化的指令-响应对，提升模型的泛化能力和任务适应能力。

具体来说，假设我们有一个预训练模型 f_θ ，输入为图像 I 和文本指令 x ，输出为文本 y 。指令微调的目标是最小化如下损失函数：

$$\mathcal{L}(\theta) = \mathbb{E}_{(I,x,y) \sim \mathcal{D}} [-\log P_\theta(y|I, x)]$$

其中 \mathcal{D} 表示带有指令的训练数据集。

例如，给定一张图片 I 及指令 x ：

- 指令 x ：“请详细描述这张图片的内容。”
- 期望输出 y ：“一只小狗在草地上奔跑，背景有蓝天和白云。”

通过指令微调，模型不仅学习到图片与文本的对应关系，还能理解“详细描述”这类指令的语义，从而在遇到不同指令时做出相应的生成。

3.1.2. 视觉语言模型

视觉语言模型 (Vision-Language Model, VLM) 是一类能够同时处理图像和文本信息的深度学习模型。其核心思想是将图像通过视觉编码器 (如 CLIP、ViT 等) 提取为一组高维特征向量, 这些特征向量通常被称为 image token。具体来说, 输入的原始图像首先经过视觉编码器, 被分割成若干 patch, 每个 patch 经过编码后生成一个 token, 所有 token 共同表征整张图片的语义信息。

随后, 这些 image token 会被送入与文本模型 (如大语言模型 LLM) 对齐的投影层, 将视觉特征映射到与文本 token 相同的向量空间。这样, 模型就可以将图像 token 与文本 token 拼接在一起, 输入到统一的 Transformer 或多模态架构中, 实现图像与文本的联合建模。通过这种方式, 视觉语言模型不仅能够理解图片内容, 还能根据输入的文本指令生成相应的文本描述, 实现图像到文本的生成任务 (如 Image Captioning、视觉问答等)。

以 MiniGPT-4 为例, 其流程为: 图像输入后, 首先由视觉编码器生成 image token, 经过线性投影层后与文本 token 拼接, 最终输入到 Vicuna 语言模型中进行推理和生成。这种设计使得模型能够充分融合视觉和语言信息, 提升多模态理解与生成能力。

3.1.3. 评估指标

3.1.3.1. BLEU

BLEU (Bilingual Evaluation Understudy) 是一种广泛使用的机器翻译评估指标。它通过比较生成文本与参考文本之间的 n-gram 重叠程度来计算分数。BLEU 分数范围从 0 到 1, 1 表示完全匹配。主要特点:

- 计算 n-gram 精确率
- 使用简短惩罚因子
- 支持多个参考文本
- 对词序敏感

3.1.3.2. CIDEr

CIDEr (Consensus-based Image Description Evaluation) 是专门为图像描述任务设计的评估指标。它的主要特点包括:

- 使用 TF-IDF 加权来强调重要词汇
- 考虑 n-gram 的共识程度
- 对罕见但准确的描述给予更高权重
- 分数范围从 0 到 10, 越高越好

3.1.3.3. ROUGE-L

ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence) 是一种基于最长公共子序列的评估指标。其特点:

- 不要求严格连续匹配

- 考虑词序信息
- 计算召回率和精确率
- 对句子结构敏感
- 分数范围从 0 到 1

这些指标能够从不同角度（n-gram 匹配、语义共识、结构相似性）衡量模型生成描述的质量。

3.2. 技术细节

3.2.1. 转换数据集

本项目使用 Flickr30k 数据集进行模型的指令微调。Flickr30k 是一个广泛使用的图像描述数据集，包含约 3 万张从 Flickr 收集的图片，每张图片通常附有 5 条由人工标注的英文描述。

为了适配模型的训练需求，我们对原始 Flickr30k 数据集的标注文件进行了预处理。具体步骤如下，由 `prepare_flickr30k.py` 脚本实现：

1. **输入：**脚本读取位于 `flickr30k/flickr_annotations_30k.csv` 的原始标注文件。
2. **核心处理逻辑：**
 - 脚本遍历 CSV 文件中的每一行。
 - 对于每张图片，它会解析 `raw` 字段中包含的 JSON 格式的多条人工标注描述。
 - **关键步骤：**为了给模型提供更丰富的上下文信息，脚本将一张图片对应的所有（通常是 5 条）描述文本**拼接成一个单一的、更长的字符串，作为该图片的“基准详细描述”**（`grounded_caption`）。
 - 在 处 理 过 程 中 ， 脚 本 会 检 查 对 应 的 图 像 文 件 是 否 存 在 于 `flickr30k/flickr30k-images/` 目录下，若图像文件缺失，则跳过该条标注。
3. **输出：**预处理完成后，脚本会生成一个名为 `flickr30k/flickr30k_grounded_detail.json` 的 JSON 文件。该文件包含了所有有效图片的 `image_id` 及其对应的拼接后的 `grounded_caption`。该 JSON 文件将直接作为模型微调的输入。
例如，JSON 文件中的一个条目格式如下：

```
{
  "image_id": "1000092795",
  "grounded_caption": "A man in a blue shirt is standing on a ladder cleaning a window. A man on a ladder cleans the window of a tall building. A man on a ladder outside a building cleaning windows. A man on a ladder washes windows on a building. A man on a scaffold outside a building washes a window."
}
```

该预处理脚本使用了 `tqdm` 库来可视化处理进度。

3.2.2. 实验流程

实验的整体流程遵循标准的深度学习模型训练范式，具体步骤由 `train.py` 脚本执行，并通过 `finetuning.py` 中的指导命令进行调用：

1. 环境与参数初始化：

- 根据 `run_cfg` 配置初始化分布式环境（如果使用多 GPU，本项目主要针对单 GPU 微调连接层）。
- 使用 `setup_seeds(cfg)` 设置全局随机种子以确保实验的可复现性。
- 使用 `setup_logger()` 配置日志记录。
- 打印配置文件 `cfg` 内容。

2. 任务、数据集与模型构建：

- 通过 `tasks.setup_task(cfg)` 设置具体的微调任务。
- 调用 `task.build_datasets(cfg)` 加载预处理后的 Flickr30k 数据集（即 `flickr30k_grounded_detail.json`）。
- 通过 `task.build_model(cfg)` 构建 MiniGPT-4 模型，此步骤会加载指定的预训练模型权重（`ckpt`）和 LLM（`llama_model`）。

3. 模型并行化：

- 如果配置为分布式训练，模型会被包装在 `torch.nn.parallel.DistributedDataParallel` 中。

4. 实验跟踪（Wandb）：

- 如果 `cfg.run_cfg.wandb_log` 为 `true`，则初始化 wandb，并使用 `wandb.watch(model)` 监控模型梯度和参数。

5. 训练器初始化与执行：

- 通过 `get_runner_class(cfg)` 获取指定的训练器类（通常为 `runner_base`）。
- 初始化训练器实例：

```
runner = RunnerClass(cfg=cfg, job_id=job_id, task=task, model=model, datasets=datasets)
```
- 调用 `runner.train()` 开始模型的训练过程。训练日志和模型检查点会保存在 `output_dir` 指定的目录中。

3.2.3. 评估方法与解码策略

模型性能的评估通过 `evaluate.sh` 脚本统一调度。该脚本首先评估原始预训练模型（`prerained_minigpt4_7b.pth`）作为基线，随后迭代评估微调过程中保存的各个模型检查点（epoch 0 至 epoch 4）。

评估的核心命令格式如下（以预训练模型为例）：

```
proxy torchrun --master-port 3456 --nproc_per_node 1 eval_scripts/eval_flickr30k.py \
  --cfg-path eval_configs/minigpt4_flickr_eval.yaml \
  --ckpt path/to/model_checkpoint.pth \
  --save_path path/to/eval_output
```

其中：

- `eval_scripts/eval_flickr30k.py`: 执行评估逻辑, 加载模型, 生成描述并计算 BLEU、CIDEr、ROUGE-L 等指标。
- `eval_configs/minigpt4_flickr_eval.yaml`: 定义评估所需的配置, 包括数据集路径和可能的解码参数。
- `--ckpt`: 指定当前评估使用的模型权重。
- `--save_path`: 存储评估结果和生成的描述。

解码策略(如束搜索宽度 `num_beams`、温度系数 `temperature`)对生成结果质量有重要影响。这些参数在上述 `minigpt4_flickr_eval.yaml` 配置文件中设定, 或存在默认值。为保证评估的公平性和可比性, 预训练和各微调检查点均采用统一的解码设置进行评估。

4. Experiment Results

4.1. 实验流程

4.1.1. Conda 环境

```
conda env create -f environment.yml
conda activate minigptv
```

4.1.2. 预训练模型获取

下载 Vicuna-7B 语言模型:

```
git clone https://huggingface.co/Vision-CAIR/vicuna-7b
cd vicuna-7b && git lfs pull
```

下载 MiniGPT-4 模型权重[3], 并配置 `train_configs/minigpt4_flickr_finetune.yaml`。

4.1.3. 数据准备

执行脚本生成注解文件:

```
python prepare_flickr30k.py
```

4.1.4. 指令微调

使用单卡 GPU 训练线性映射层:

```
torchrun --nproc-per-node 1 train.py --cfg-path train_configs/
minigpt4_flickr_finetune.yaml
```

训练过程部分如图所示:

```
(minipy) ~ - ssh@74-finetuning-gpu(nvidia) x linux torchrun --nnproc-per-node 7 train.py --cfg-path train_configs/minipid_flickr_finetune.yaml\nW0608 11:08:51.958560 140636613841856 torch/distributed/run.py:757]\nW0608 11:08:51.958560 140636613841856 torch/distributed/run.py:757] =====\nW0608 11:08:51.958560 140636613841856 torch/distributed/run.py:757] Setting OMP_NUM_THREADS environment variable for each process to be 1 in default, to avoid your system being overloaded,\nplease further tune the variable for optimal performance in your application as needed.\nW0608 11:08:51.958560 140636613841856 torch/distributed/run.py:757] =====\nWelcome to bitsandbytes. For bug reports, please submit your error trace to: https://github.com/TimDetters/bitsandbytes/issues\n=====BUG REPORT=====\nWelcome to bitsandbytes. For bug reports, please submit your error trace to: https://github.com/TimDetters/bitsandbytes/issues\n=====BUG REPORT=====\nWelcome to bitsandbytes. For bug reports, please submit your error trace to: https://github.com/TimDetters/bitsandbytes/issues\n=====BUG REPORT=====\nWelcome to bitsandbytes. For bug reports, please submit your error trace to: https://github.com/TimDetters/bitsandbytes/issues\n=====BUG REPORT=====\nWelcome to bitsandbytes. For bug reports, please submit your error trace to: https://github.com/TimDetters/bitsandbytes/issues\n=====BUG REPORT=====\nWelcome to bitsandbytes. For bug reports, please submit your error trace to: https://github.com/TimDetters/bitsandbytes/issues\n=====BUG REPORT=====\nWelcome to bitsandbytes. For bug reports, please submit your error trace to: https://github.com/TimDetters/bitsandbytes/issues\n=====BUG REPORT=====\nrs/bitsandbytes/issues\n=====BUG REPORT=====\nWelcome to bitsandbytes. For bug reports, please submit your error trace to: https://github.com/TimDetters/bitsandbytes/issues\n=====BUG REPORT=====\nWelcome to bitsandbytes. For bug reports, please submit your error trace to: https://github.com/TimDetters/bitsandbytes/issues\n=====BUG REPORT=====\nTrue\nTrue\nTrue\nTrue\nTrue\nTrue\nTrue\ndistributed init [rank 0, world 7]: env//\ndistributed init [rank 3, world 7]: env//\ndistributed init [rank 6, world 7]: env//\ndistributed init [rank 5, world 7]: env//\ndistributed init [rank 4, world 7]: env//\ndistributed init [rank 3, world 7]: env//\ndistributed init [rank 2, world 7]: env//\nTrue\ndistributed init [rank 7, world 7]: env//\n2025-06-08 11:09:06.774 [INFO]
```

```

===== Running Parameters =====
2025-06-08 11:09:06,775 [INFO] {
  "amp": true,
  "device": "cuda",
  "dist_backend": "nccl",
  "dist_url": "tcp://",
  "distributed": true,
  "evaluate": false,
  "gpus": 0,
  "init_lr": 3e-05,
  "iters_per_epoch": 500,
  "job_name": "minigt4_flickr_finetune",
  "lr_sched": "linear_warmup_cosine_lr",
  "max_epoch": 5,
  "min_lr": 1e-05,
  "num_workers": 4,
  "output_dir": "output/minigt4_flickr_finetune",
  "rank": 0,
  "resumeckpt_path": null,
  "seed": 42,
  "task": "image_text_pretrain",
  "train_splits": [
    "train"
  ],
  "wandb_log": true,
  "warmup_lr": 1e-06,
  "warmup_steps": 200,
  "weight_decay": 0.05,
  "world_size": 7
}
2025-06-08 11:09:06,775 [INFO]
===== Dataset Attributes =====
2025-06-08 11:09:06,775 [INFO]
===== flickr30k_grouped_detail =====
2025-06-08 11:09:06,776 [INFO] {
  "batch_size": 2,
  "build_info": {
    "env_path": "/mnt/data4/home/zyuan/MiniGP4-Finetuning/flickr30k/flickr30k_grouped_detail.json",
    "image_path": "/mnt/data4/home/zyuan/MiniGP4-Finetuning/flickr30k/flickr30k-images"
  },
  "data_type": "images",
  "text_processor": {
    "train": {
      "name": "blip_caption"
    }
  },
  "vis_processor": {
    "train": {
      "image_size": 224,
      "name": "blip_image_train"
    }
  }
}
2025-06-08 11:09:06,776 [INFO]

```

```

name: Model Attributes #####
2825-06-08 11:09:06,777 [INFO] ("arcov", "mainloop")
{"ckpt": "/mnt/data/home/ziyuan/MiniGP4-Finetuning/prained_minigt4_7b.pth",
 "freeze_path_encoder": 0,
 "end_sys": "###",
 "freeze_optimizer": true,
 "freeze_vit": true,
 "image_size": 224,
 "llama_model": "/mnt/data/home/ziyuan/MiniGP4-Finetuning/vicuna-7b",
 "max_text_len": 580,
 "model_type": "pretrain_vicunaB",
 "num_query_tokens": 32,
 "prompt": "",
 "prompt_path": "prompts/alignment.txt",
 "prompt_tokenizer": "gpt4o",
 "use_grad_checkpoint": true,
 "use_grad_checkpoint": false,
 "vit_precision": "fp16"}

2825-06-08 11:09:06,777 [INFO] Building datasets...
2825-06-08 11:09:06,899 [INFO] Loading LLaMA
Loading checkpoint shards: 100% |#####| 2/2
08:27-08:08 12.62s/it
Loading checkpoint shards: 100% |#####| 2/2
2825-06-08 11:11:40,180 [INFO] Loading LLaMA Done
08:27-08:08 12.68s/it
Loading checkpoint shards: 100% |#####| 2/2
08:28-08:08 12.55s/it
Loading checkpoint shards: 100% |#####| 2/2
08:28-08:08 12.30s/it
Loading checkpoint shards: 100% |#####| 2/2
08:27-08:08 13.07s/it
Loading checkpoint shards: 100% |#####| 2/2
08:27-08:08 13.99s/it
Loading checkpoint shards: 100% |#####| 2/2
08:28-08:08 14.85s/it
2825-06-08 11:12:43,597 [INFO] freeze vision encoder
2825-06-08 11:12:43,597 [INFO] Loading ViT Done
Loading O-Farmer
2825-06-08 11:12:53,000 [INFO] freeze OFarmer
Load MiniGPT-4 Checkpoint: /mnt/data/home/ziyuan/MiniGP4-Finetuning/prained_minigt4_7b.pth
Loading O-Farmer Done
Osd
Prompt length: prompts
#####
Warning: -loader-prompts -img Describe this image in detail. ####Assistant:
Load MiniGPT-4 Checkpoint: /mnt/data/home/ziyuan/MiniGP4-Finetuning/prained_minigt4_7b.pth

```

```
2025-06-08 11:14:36,584 [INFO] Start training
2025-06-08 11:14:42,182 [INFO] dataset_ratios not specified, datasets will be concatenated (map-style datasets) or chained (webdataset.DataPipeline).
2025-06-08 11:14:42,184 [INFO] Loaded 31614 records for train split from the dataset.
batch sizes (d2)
module.module.llama_proj.weight
module.module.llama_proj.bias
2025-06-08 11:14:42,218 [INFO] number of trainable parameters: 3149824
2025-06-08 11:14:42,218 [INFO] Start training epoch 0, 500 iters per inner epoch.
```

训练结果部分如图所示:

epoch0:	<pre> Train: data epoch: [0] [0/500] eta: 1:07:00 lr: 0.000001 loss: 1.8893 time: 0.8414 data: 0.0001 max mem: 17019 Train: data epoch: [0] [50/500] eta: 0:04:39 lr: 0.000008 loss: 2.8966 time: 0.4717 data: 0.0000 max mem: 18202 Train: data epoch: [0] [100/500] eta: 0:03:43 lr: 0.000015 loss: 1.6432 time: 0.4955 data: 0.0000 max mem: 18202 Train: data epoch: [0] [150/500] eta: 0:03:10 lr: 0.000023 loss: 1.2601 time: 0.5233 data: 0.0000 max mem: 18202 Train: data epoch: [0] [200/500] eta: 0:02:42 lr: 0.000030 loss: 1.0724 time: 0.5462 data: 0.0000 max mem: 18202 Train: data epoch: [0] [250/500] eta: 0:02:15 lr: 0.000038 loss: 2.8115 time: 0.5386 data: 0.0000 max mem: 18202 Train: data epoch: [0] [300/500] eta: 0:01:48 lr: 0.000029 loss: 1.8867 time: 0.5665 data: 0.0000 max mem: 18202 Train: data epoch: [0] [350/500] eta: 0:01:22 lr: 0.000029 loss: 2.4573 time: 0.5859 data: 0.0000 max mem: 18202 Train: data epoch: [0] [400/500] eta: 0:00:55 lr: 0.000029 loss: 2.2811 time: 0.5820 data: 0.0000 max mem: 18202 Train: data epoch: [0] [450/500] eta: 0:00:27 lr: 0.000028 loss: 1.9895 time: 0.6032 data: 0.0000 max mem: 18202 Train: data epoch: [0] [499/500] eta: 0:00:00 lr: 0.000028 loss: 1.3743 time: 0.6146 data: 0.0000 max mem: 18202 Train: data epoch: [0] Total time: 0:04:40 (0.5607 s / it) 2025-06-08 11:19:22,578 [INFO] Averaged stats: lr: 0.0000 loss: 1.7329 2025-06-08 11:19:22,583 [INFO] No validation splits found. 2025-06-08 11:19:22,628 [INFO] Saving checkpoint at epoch 0 to /mnt/data4/home/ziyuan/MiniGPT4-Finetuning/minigpt4/output/minigpt4_flickr_finetune/20250608110/checkpoint_0.pth. 2025-06-08 11:19:22,794 [INFO] Start training 2025-06-08 11:19:22,836 [INFO] Start training epoch 1, 500 iters per inner epoch. </pre>
epoch1:	<pre> Train: data epoch: [1] [0/500] eta: 0:04:23 lr: 0.000028 loss: 1.9601 time: 0.5261 data: 0.0000 max mem: 18202 Train: data epoch: [1] [50/500] eta: 0:04:13 lr: 0.000028 loss: 1.8204 time: 0.5758 data: 0.0000 max mem: 18203 Train: data epoch: [1] [100/500] eta: 0:03:47 lr: 0.000027 loss: 1.8685 time: 0.5789 data: 0.0000 max mem: 18203 Train: data epoch: [1] [150/500] eta: 0:03:19 lr: 0.000027 loss: 1.8785 time: 0.5775 data: 0.0000 max mem: 18786 Train: data epoch: [1] [200/500] eta: 0:02:51 lr: 0.000026 loss: 1.3415 time: 0.6040 data: 0.0000 max mem: 18786 Train: data epoch: [1] [250/500] eta: 0:02:22 lr: 0.000026 loss: 1.4390 time: 0.5709 data: 0.0000 max mem: 18786 Train: data epoch: [1] [300/500] eta: 0:01:54 lr: 0.000025 loss: 1.8319 time: 0.5839 data: 0.0000 max mem: 18786 Train: data epoch: [1] [350/500] eta: 0:01:26 lr: 0.000025 loss: 1.4437 time: 0.5807 data: 0.0000 max mem: 18786 Train: data epoch: [1] [400/500] eta: 0:00:57 lr: 0.000024 loss: 1.7856 time: 0.5833 data: 0.0000 max mem: 18786 Train: data epoch: [1] [450/500] eta: 0:00:29 lr: 0.000024 loss: 1.4488 time: 0.5738 data: 0.0000 max mem: 18786 Train: data epoch: [1] [499/500] eta: 0:00:00 lr: 0.000023 loss: 1.1011 time: 0.5920 data: 0.0000 max mem: 18786 Train: data epoch: [1] Total time: 0:04:51 (0.5823 s / it) 2025-06-08 11:24:13,098 [INFO] Averaged stats: lr: 0.0000 loss: 1.6198 2025-06-08 11:24:14,003 [INFO] No validation splits found. 2025-06-08 11:24:14,048 [INFO] Saving checkpoint at epoch 1 to /mnt/data4/home/ziyuan/MiniGPT4-Finetuning/minigpt4/output/minigpt4_flickr_finetune/20250608110/checkpoint_1.pth. 2025-06-08 11:24:14,210 [INFO] Start training 2025-06-08 11:24:14,252 [INFO] Start training epoch 2, 500 iters per inner epoch. </pre>
epoch2:	<pre> Train: data epoch: [2] [0/500] eta: 0:04:40 lr: 0.000023 loss: 1.7006 time: 0.5603 data: 0.0000 max mem: 18786 Train: data epoch: [2] [50/500] eta: 0:04:32 lr: 0.000022 loss: 1.5678 time: 0.6237 data: 0.0000 max mem: 18786 Train: data epoch: [2] [100/500] eta: 0:04:00 lr: 0.000022 loss: 1.7640 time: 0.5963 data: 0.0000 max mem: 18786 Train: data epoch: [2] [150/500] eta: 0:03:30 lr: 0.000021 loss: 1.6700 time: 0.5938 data: 0.0000 max mem: 18786 Train: data epoch: [2] [200/500] eta: 0:03:00 lr: 0.000021 loss: 1.8516 time: 0.6045 data: 0.0000 max mem: 18786 Train: data epoch: [2] [250/500] eta: 0:02:30 lr: 0.000020 loss: 1.7562 time: 0.5997 data: 0.0000 max mem: 18786 Train: data epoch: [2] [300/500] eta: 0:02:00 lr: 0.000019 loss: 1.7730 time: 0.5969 data: 0.0000 max mem: 18786 Train: data epoch: [2] [350/500] eta: 0:01:30 lr: 0.000019 loss: 1.3857 time: 0.6159 data: 0.0000 max mem: 18786 Train: data epoch: [2] [400/500] eta: 0:01:00 lr: 0.000018 loss: 1.1799 time: 0.6024 data: 0.0000 max mem: 18786 Train: data epoch: [2] [450/500] eta: 0:00:30 lr: 0.000018 loss: 1.3858 time: 0.6197 data: 0.0000 max mem: 18786 Train: data epoch: [2] [499/500] eta: 0:00:00 lr: 0.000017 loss: 1.5360 time: 0.6271 data: 0.0000 max mem: 18786 Train: data epoch: [2] Total time: 0:05:03 (0.6074 s / it) 2025-06-08 11:29:17,942 [INFO] Averaged stats: lr: 0.0000 loss: 1.6039 2025-06-08 11:29:17,948 [INFO] No validation splits found. 2025-06-08 11:29:18,005 [INFO] Saving checkpoint at epoch 2 to /mnt/data4/home/ziyuan/MiniGPT4-Finetuning/minigpt4/output/minigpt4_flickr_finetune/20250608110/checkpoint_2.pth. 2025-06-08 11:29:18,166 [INFO] Start training 2025-06-08 11:29:18,222 [INFO] Start training epoch 3, 500 iters per inner epoch. </pre>
epoch3:	<pre> Train: data epoch: [3] [0/500] eta: 0:04:47 lr: 0.000017 loss: 1.4256 time: 0.5756 data: 0.0000 max mem: 18786 Train: data epoch: [3] [50/500] eta: 0:04:38 lr: 0.000016 loss: 1.7858 time: 0.6249 data: 0.0000 max mem: 18786 Train: data epoch: [3] [100/500] eta: 0:04:11 lr: 0.000016 loss: 1.5871 time: 0.6141 data: 0.0000 max mem: 18786 Train: data epoch: [3] [150/500] eta: 0:03:41 lr: 0.000015 loss: 1.4189 time: 0.6203 data: 0.0000 max mem: 18786 Train: data epoch: [3] [200/500] eta: 0:03:10 lr: 0.000015 loss: 1.6425 time: 0.6242 data: 0.0000 max mem: 18786 Train: data epoch: [3] [250/500] eta: 0:02:39 lr: 0.000014 loss: 1.6643 time: 0.6480 data: 0.0000 max mem: 18786 Train: data epoch: [3] [300/500] eta: 0:02:07 lr: 0.000014 loss: 1.6156 time: 0.6514 data: 0.0000 max mem: 18786 Train: data epoch: [3] [350/500] eta: 0:01:36 lr: 0.000013 loss: 2.1249 time: 0.6852 data: 0.0000 max mem: 18786 Train: data epoch: [3] [400/500] eta: 0:01:04 lr: 0.000013 loss: 1.4653 time: 0.6445 data: 0.0000 max mem: 18786 Train: data epoch: [3] [450/500] eta: 0:00:32 lr: 0.000012 loss: 1.4642 time: 0.6355 data: 0.0000 max mem: 18786 Train: data epoch: [3] [499/500] eta: 0:00:00 lr: 0.000012 loss: 1.5794 time: 0.6457 data: 0.0000 max mem: 18786 Train: data epoch: [3] Total time: 0:05:23 (0.6472 s / it) 2025-06-08 11:34:41,802 [INFO] Averaged stats: lr: 0.0000 loss: 1.6006 2025-06-08 11:34:41,807 [INFO] No validation splits found. 2025-06-08 11:34:41,852 [INFO] Saving checkpoint at epoch 3 to /mnt/data4/home/ziyuan/MiniGPT4-Finetuning/minigpt4/output/minigpt4_flickr_finetune/20250608110/checkpoint_3.pth. 2025-06-08 11:34:42,011 [INFO] Start training 2025-06-08 11:34:42,054 [INFO] Start training epoch 4, 500 iters per inner epoch. </pre>
epoch4:	<pre> Train: data epoch: [4] [0/500] eta: 0:05:18 lr: 0.000012 loss: 1.4270 time: 0.6366 data: 0.0000 max mem: 18786 Train: data epoch: [4] [50/500] eta: 0:04:42 lr: 0.000012 loss: 1.3480 time: 0.6205 data: 0.0000 max mem: 18786 Train: data epoch: [4] [100/500] eta: 0:04:13 lr: 0.000011 loss: 1.7157 time: 0.6420 data: 0.0000 max mem: 18786 Train: data epoch: [4] [150/500] eta: 0:03:39 lr: 0.000011 loss: 1.7502 time: 0.6232 data: 0.0000 max mem: 18786 Train: data epoch: [4] [200/500] eta: 0:03:05 lr: 0.000011 loss: 1.6249 time: 0.5932 data: 0.0000 max mem: 18786 Train: data epoch: [4] [250/500] eta: 0:02:36 lr: 0.000010 loss: 1.6335 time: 0.5904 data: 0.0000 max mem: 18786 Train: data epoch: [4] [300/500] eta: 0:02:04 lr: 0.000010 loss: 1.9784 time: 0.6186 data: 0.0000 max mem: 18786 Train: data epoch: [4] [350/500] eta: 0:01:32 lr: 0.000010 loss: 1.6016 time: 0.5814 data: 0.0000 max mem: 18786 Train: data epoch: [4] [400/500] eta: 0:01:00 lr: 0.000010 loss: 1.0132 time: 0.5709 data: 0.0000 max mem: 18786 Train: data epoch: [4] [450/500] eta: 0:00:30 lr: 0.000010 loss: 1.4456 time: 0.5597 data: 0.0000 max mem: 18786 Train: data epoch: [4] [499/500] eta: 0:00:00 lr: 0.000010 loss: 1.6659 time: 0.5690 data: 0.0000 max mem: 18786 Train: data epoch: [4] Total time: 0:05:00 (0.6001 s / it) 2025-06-08 11:39:42,115 [INFO] Averaged stats: lr: 0.0000 loss: 1.5761 2025-06-08 11:39:42,121 [INFO] No validation splits found. 2025-06-08 11:39:42,166 [INFO] Saving checkpoint at epoch 4 to /mnt/data4/home/ziyuan/MiniGPT4-Finetuning/minigpt4/output/minigpt4_flickr_finetune/20250608110/checkpoint_4.pth. 2025-06-08 11:39:42,344 [INFO] No validation splits found. 2025-06-08 11:39:42,345 [INFO] Training time 0:25:03 </pre>

4.1.5. 评估

运行如下脚本在多模型间对比性能：

```
bash evaluate.sh
```

4.2. 评估结果与分析

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	CIDEr	ROUGE-L
pretrained	0.3175	0.2371	0.1843	0.1529	0.4361	0.5758
epoch0	0.3427	0.2723	0.2204	0.2002	0.4502	0.6101
epoch1	0.3608	0.2822	0.2503	0.2320	0.4753	0.6469
epoch2	0.3728	0.3021	0.2720	0.2456	0.4829	0.6602
epoch3	0.3878	0.3314	0.2978	0.2688	0.4907	0.6701
epoch4	0.3912	0.3375	0.3005	0.2807	0.5115	0.6803

表 1 模型评估结果统计

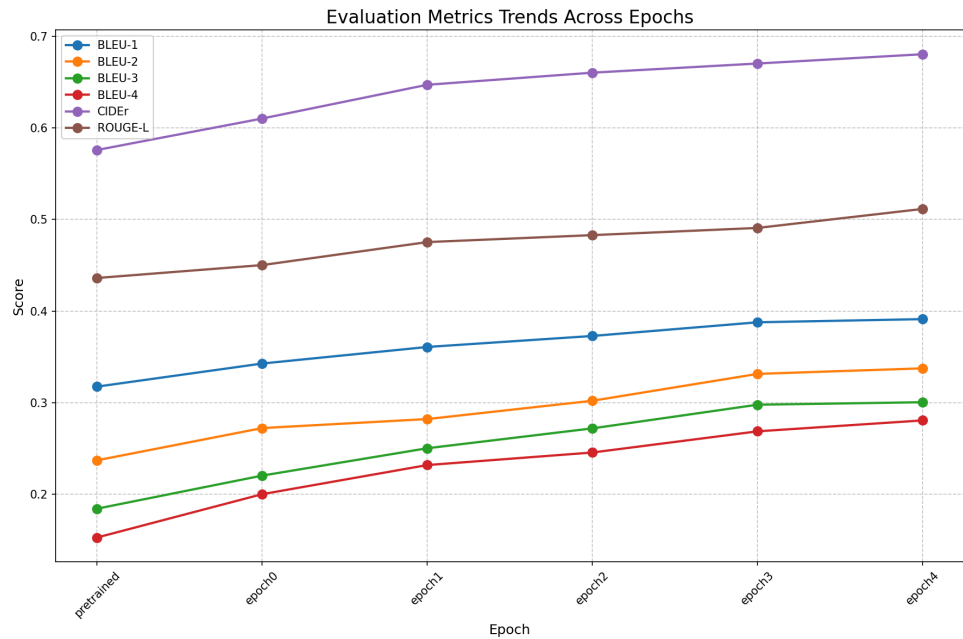


图 1 模型评估结果折线图

4.2.1. 模型性能持续提升

随着训练轮次 (Epoch) 的增加, 所有评价指标均呈现稳定上升趋势, 表明模型通过迭代学习有效捕捉了文本生成任务的核心规律。其中:

- 语义与连贯性优化显著: CIDEr 和 ROUGE-L 的增速远超其他指标 (详见图表斜率), 说明模型在生成内容的语义相关性、上下文连贯性上提升最为突出, 逐渐接近人类语言表达模式。
- 局部一致性稳步改进: BLEU 系列指标增长平缓但持续 (BLEU-1 至 BLEU-4 增幅约 50%-80%), 反映模型在局部词汇匹配和短语结构的准确性上逐步完善。

4.2.2. 训练动态揭示关键拐点

- 早期快速收敛: pretrained 至 epoch1 阶段所有指标快速跃升, 验证预训练权重提供了高质量初始化。

- 中后期差异化优化：epoch2 后 CIDEr 与 ROUGE-L 仍保持陡峭上升，而 BLEU 系列进入平缓增长期，表明模型后期更侧重于语义整体性而非局部词序精确度，符合文本生成任务的本质目标。
- 持续训练价值：截至 epoch4，各曲线仍未出现平台期，建议扩展训练轮次（如至 epoch6）以挖掘性能潜力。
- 重点优化方向：可针对性设计长依赖文本和抽象语义的增强训练模块（如注意力机制改进），进一步发挥 CIDEr 与 ROUGE-L 的优势。

4.2.3. 局限性与后续工作

- 评估维度补充：需增加人工评价或 SPICE 等细粒度指标，验证模型在视觉语义对齐上的表现（若为多模态任务）。
- 泛化能力检验：当前结果基于单一数据集，需在跨领域数据上验证鲁棒性。
- 探索更大的语言模型后端。
- 优化推理速度与显存占用。

附录

夏子渊	20%
金裕涵	20%
林滨	20%
程韬	20%
潘越	20%

表 2 贡献度百分比

References

- [1] D. Zhu, J. Chen, X. Shen, X. Li, 和 M. Elhoseiny, 《MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models》. [在线]. 载于: <https://arxiv.org/abs/2304.10592>
- [2] P. Young, A. Lai, M. Hodosh, 和 J. Hockenmaier, 《From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions》, *Transactions of the Association for Computational Linguistics*, 卷 2, 页 67–78, 2014.
- [3] D. Zhu, J. Chen, X. Shen, X. Li, 和 M. Elhoseiny, 《GitHub - Vision-CAIR/MiniGPT-4: Open-sourced codes for MiniGPT-4 and MiniGPT-v2 — github.com》. [在线]. 载于: <https://github.com/Vision-CAIR/MiniGPT-4>