零样本组合图像检索

Santhosh Kakarla
George Mason University
skakarl3@gmu.edu

Gautama Shastry Bulusu Venkata
George Mason University
sbulusuv@gmu.edu

June 10, 2025

ABSTRACT

组合图像检索（CIR）使用户能够通过将参考图像与文本指令结合，实现细粒度的视觉编辑——例如“将裙子变蓝”或“去除条纹”。使用预训练多模态编码器的零样本 CIR 方法通常分别编码文本和图像，在 FashionIQ 等基准测试中达到 20-25% 的 Recall@10。为提高性能，我们使用基于 BLIP2 的微调策略，通过 Q-Former 将图像和文本融合为统一表示，将 Recall@10 提高到 45.6%（衬衫），40.1%（裙子），和 50.4%（顶部 T 恤），平均 Recall@50 提升至 67.6%。在此基础上，我们利用 CLIP 的单独文本和视觉编码器实现检索 DPO：每个文本提示通过 CLIP transformer 编码，正样本和通过 FAISS 挖掘出的困难负样本通过 CLIP 的图像编码器编码，直接偏好优化损失在相似度分数之间施加边界。尽管广泛调整了 DPO 缩放因子、索引参数和采样策略，由于（1）缺乏联合的图像-文本融合，（2）边界损失与 top-K 度量不对齐，（3）负样本质量差，以及（4）视觉和 transformer 层冻结，该方法在 Recall@10 仅达到 0.02%——远低于零样本和提示调优的基线。这些发现强调了有效的基于偏好的 CIR 需要真正的多模态融合、排名对齐的目标和精细的负样本采样。

1 简介

组合图像检索（CIR）使用户能够通过指定参考图像以及文本修改来发出语义丰富的视觉搜索查询——例如“把裙子变蓝”、“去掉花卉图案”或“在衬衫上加条纹”。零样本组合图像检索（ZSCIR）方法通过将图像和文本独立输入到预训练的多模态编码器（例如，CLIP）中，并计算两个模态之间的余弦相似度来处理这些查询。该流程不需要任务特定的训练数据，使其具有高度灵活性；但是，它难以解释和结合视觉和语言上的细微差别。在 FashionIQ 基准测试上，该测试包含真实世界的产品图像和人工标注的编辑指令，ZSCIR 仅能达到约 20-25 % 的 Recall@10。如此有限的性能限制了 CIR 在关键应用中的效用——比如电子商务个性化，在此客户希望获得精确的产品变体；数字设计原型制作，需要对视觉编辑进行细粒度控制；以及内容策划或审核，其中语义驱动的过滤是必不可少的。

为了解决这些限制，我们在 BLIP2 框架内采用了一种微调策略，引入了一个轻量级的 Q-Former 模块，该模块学会将视觉和文本输入融合为统一的表示。在微调过程中，Q-Former 同时使用参考图像的嵌入和编辑指令的标记嵌入，生成单一的、动态的提示向量。然后，通过对比分检损失将这个融合后的提示与候选图像进行比较，使其更接近目标图像并远离负样本。在不需要成对的编辑图像示例的情况下，这种方法使得通用的多模态编码器适应于 CIR 任务，提高了对细微属性变化和对象操作的灵敏度。据实验证明，微调后的 BLIP2 在 FashionIQ 上实现了衬衫编辑的 Recall@10 为 45.6 %，连衣裙编辑为 40.1 %，以及上衣编辑为 50.4 %，平均 Recall@50 为 67.6 %。

在这些成果的基础上，我们研究了用于 CIR 的 Retrieval-DPO，这是一种检索增广的偏好优化框架，旨在将用户风格的排序信号直接注入检索模型。与提示调优不同，该方法处理的是独立的 CLIP 文本和视觉编码器：每个标题通过 CLIP 变压器编码为提示嵌入，而正样本和 FAISS 挖掘的困难负样本图像则由 CLIP 视觉编码器编码。我们使用 FAISS 索引在图库中为每个查询检索到最困难的负样本，并应用直接偏好优化损失——最大化正负相似度得分之间的对数几率边距。尽管进行了详尽的超参数调优（包括 DPO 缩放因子 β 、索引配置和负采样策略）和混合精度训练，Retrieval-DPO 仅产生了 0.02 % Recall@10。我们将这种明显的低表现归因于四个因素：（1）缺乏真正的多模态融合限制了模型学习跨模态关系的能力；（2）DPO 基于边距的目标没有直接优化顶端 K 排名指标；（3）FAISS 挖掘的负样本要么过于简单，要么过于模糊，削弱了偏好信号；以

及（4）冻结视觉和变压器层限制了表征的适应。这些发现揭示了简单偏好方法中的关键不足，并强调了未来的多模态偏好学习研究中对集成融合架构、与排名对齐的损失函数和精炼的负采样策略的需求。

2 相关工作

由 Radford 等人于 2021 年提出的对比语言-图像预训练（CLIP）介绍了一种双编码器框架，通过对比损失学习对齐的图像和文本表示，实现了在包括组合图像检索（CIR）在内的各种视觉语言任务中的有效零样本迁移。在 CLIP 的范式基础上，由 Li 等人于 2023 年提出的 BLIP2 使用轻量级的 Q-Former 来学习查询嵌入，将冻结的视觉编码器与大型语言模型结合，大大改善了检索和其他下游任务的多模态融合。由 Rafailov 等人于 2023 年提出的直接偏好优化（DPO）将人类反馈的强化学习重新构形成一种监督的、基于边界的损失，将模型输出与人类偏好对齐。检索-DPO 将该思想延展到检索场景，通过使用 FAISS 挖掘困难负样本，并将联合偏好损失应用于查询与图像嵌入之间的相似性得分。这些方法共同激发了我们对提示调优的 CIR 和基于偏好的检索优化的研究。

3 方法论

组合图像检索（CIR）结合参考图像和自然语言编辑指令，以检索反映指定修改的“目标”图像。传统的 CIR 方法需要在（参考，编辑，目标）三元组上进行完全监督的训练，收集这些数据是劳动密集型的。零样本 CIR（ZSCIR）通过利用像 CLIP 这样的模型来规避这一点，该模型通过对比预训练目标（InfoNCE，Rusak 等，2024 年）将图像和文本独立嵌入到共享的向量空间中。然后通过计算文本编码指令和图像嵌入之间的余弦相似性来执行检索。虽然 ZSCIR 不需要特定任务的数据，但它在细粒度编辑上存在困难——在 FashionIQ 上仅能达到 20-25 % Recall@10——因为它无法显式建模图像内容和文本修改之间的交互。

3.1 数据集准备

FashionIQ 基准 (Wu et al. 2019) 是我们的评估语料库。它包含三个服装类别——衬衫（约 17,000 对）、连衣裙（约 16,000 对）、和上衣 T 恤（约 18,000 对），每个类别都取自真实世界电子商务产品列表。每个参考-目标对都有 1 到 5 个由人类撰写的描述期望视觉修改的字幕（例如，“使连衣裙变红”，“给衬衫加条纹”，“去掉花卉图案”），整个数据集中共包含超过 50,000 个独特的编辑指令。

FashionIQ 通过 JSON 文件（`split.<category>.<split>.json`）提供官方的训练、验证和测试数据集划分，训练集约占 80 %，验证集约占 10 %，测试集约占 10 %。这些划分确保了参考图像或目标图像之间没有重叠。我们使用训练集进行模型微调，使用验证集进行超参数选择和提前停止（监控 Recall@10），测试集仅用于报告最终性能。

所有图像都进行了统一的预处理：以 RGB 格式加载，使用双三次插值调整为 224×224 大小，中心裁剪，转换为 PyTorch 张量，并使用 CLIP 的统计数据（mean = [0.481, 0.457, 0.408]; std = [0.268, 0.261, 0.276]）进行标准化。文本编辑指令以 `<|image|>` 为前缀表明多模态输入，并被标记化为固定长度（CLIP 为 77 个标记；针对 BLIP2 的基于 Q-Former 的提示调优为 32 个可学习查询），根据需要进行截断或填充。这种一致的预处理流程确保了在零样本、提示调优和偏好优化的检索方法之间的公平比较。

3.2 使用 BLIP2 的 QFormer 进行微调

为了有效地融合视觉背景和文本编辑以实现合成图像检索，在 BLIP2 架构的冻结图像和语言组件之间引入了一个小型融合模块，称为 Q-Former。该过程首先通过训练好的视觉变换器对每个参考图像进行编码，输出捕捉局部特征（如颜色、纹理和形状）的补丁嵌入网格。这些补丁嵌入本身不能满足用户期望的修改，因此 Q-Former 使用一组固定的可学习查询向量来跨整个网格进行关注，生成一组紧凑的融合令牌，突出显示与即将进行的编辑最相关的图像区域。

接下来，这些融合的视觉标记通过一个可训练的线性层投射到与语言模型标记相同的嵌入空间。同时，以特殊图像标记为前缀的文本编辑指令由冻结的语言变换器的输入层进行标记化并嵌入。通过将投射后的视觉查询与文本标记嵌入拼接成一个单一序列，模型创造出一种真正的多模态输入。然后，该组合序列由冻结的语言骨干网络处理，其自注意层进一步整合跨模态的信息。在指定的分类位置（通常是第一个标记）的隐藏状态被提取，作为最终的多模态提示嵌入，封装了参考图像的视觉线索和请求编辑的语义。

训练这个融合头利用了一种对比检索目标。在每个批次中，提示嵌入与对应的目标图像嵌入配对——也由冻结的视觉编码器产生——并与批次中的所有其他图像进行对比。InfoNCE 损失鼓励每个提示最大化与真实目标的余弦相似度，同时最小化与干扰图像的相似度，使用一个温度超参数来调整负样本的难度。关键在于，仅更新 Q-Former 及其投影参数；主要的视觉和语言骨干保持冻结状态，保留其通用能力，并显著减少可训练参数的数量。

优化过程采用 AdamW 算法，通常使用 1e-4 的学习率和适度的权重衰减。大约 128 个三元组的小批量在稳定的梯度估计和 GPU 内存限制之间取得平衡。训练一般在 5 到 10 次迭代中收敛，提早停止则取决于保留的验证集上的性能——具体来说，是在组合检索任务上的 Recall@10。该有针对性的提示微调策略使得大规模的、冻结的 BLIP2 模型能够适应组合图像检索的需求，在无需广泛配对数据的情况下，相对于零样本基线取得显著改进。

训练使用 InfoNCE 对比损失：

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(\langle z_i, z_i^+\rangle/\tau)}{\sum_{j=1}^{B}\exp(\langle z_i, z_j^-\rangle/\tau)},$$

，其中 z_i 是第 i 个提示嵌入，z_i^+ 是匹配的目标图像嵌入，z_j^- 是第 j 个负样本嵌入，τ 是温度超参数。只有 Q-Former 及其投影层被更新（AdamW，学习率 1×10^{-4}，权重衰减 1×10^{-2}）。包含 128 个三元组的迷你批次和混合精度训练确保在 5-10 个周期内收敛，并基于验证集的 Recall@10 进行早停。

3.3 检索增强直接偏好优化（检索 DPO）

检索 DPO 框架首先在整个图像库上构建一个高效的最近邻索引。来自训练和验证部分的所有图像都被预处理——调整大小为 224×224，中心裁剪，转换为张量并标准化——然后通过 CLIP 的冻结视觉编码器进行编码。它们的 1,024 维特征向量被收集到一个 NumPy 数组中，进行 L2 标准化，并被添加到一个 `faiss.IndexFlatIP` 结构中以便快速进行内积搜索。为了避免在每次运行时重新编码，FAISS 索引文件和从索引行到图像文件名的腌制映射都被缓存，并在后续执行时加载。

训练数据来自 FashionIQ 的训练集，其中每个条目提供一个标题及其匹配的目标图像文件名。对于每个例子，标题前面加上 `<|image|>`，通过 CLIP 的分词器进行分词，并转换为 token ID。真实的目标图像被编码以获得正样本嵌入，通过查询 FAISS 索引获取前 $k = 50$ 个最近邻，并选择第一个与正样本索引不同的邻居来挖掘一个困难负样本。这提供了语义上或视觉上类似的干扰项，以实现有意义的对比。

在训练过程中，经过 CLIP 的冻结文本嵌入层和变压器的标记化标题以混合精度传递，以生成提示嵌入；正面和负面图像也同样被编码。所有嵌入均进行 L2 归一化，并通过点积计算相似度分数 s_i^+ 和 s_i^-，并由 CLIP 的学习温度 (logit_scale) 进行缩放。我们应用直接偏好优化损失：

$$\ell_{\text{DPO}} = -\frac{1}{B}\sum_{i=1}^{B}\log\sigma\big(\beta\,(s_i^+ - s_i^-)\big),$$

该损失惩罚负面的分数接近或超过正面的情况，强制实施由 β 控制的边界。只有文本变压器的参数被更新 (AdamW，学习率 1×10^{-4}，权重衰减 1×10^{-2})；视觉编码器和标记嵌入保持冻结。梯度缩放器稳定混合精度更新。

验证查询遵循相同的预处理和编码：每个标题生成一个提示嵌入，从 FAISS 索引中检索前 50 个候选。Recall@K（对于 $K = 1, 5, 10, 50$）通过检查真实目标是否出现在前 K 中来计算，并在验证集上取平均值。

失败原因：

- 缺乏联合多模态融合：文本和图像编码器仍然是分开的，因此文本空间中的 DPO 改进不会转化为丰富的跨模态表示，而这些表示对于细微的编辑是必需的。
- 目标与度量不一致：基于边距的 DPO 损失并不直接优化排名位置；小幅度的边距增加通常无法将目标推到前 K 名。
- 噪声负采样：FAISS 挖掘得到的负样本可能过于简单或过于模糊（近似重复），削弱了偏好信号并使训练不稳定。
- 冻结视觉骨干：由于不对视觉编码器进行微调，文本和图像空间之间的初始不匹配会持续存在，限制了模型捕捉精细编辑语义的能力。

这些因素共同解释了为什么尽管经过仔细的实施和调整，RetrievalDPO 仅能达到 0.02 % Recall@10，这突显了在未来的多模态偏好学习研究中，需要集成的融合架构、排名感知的目标以及更智能的负采样策略。

4 结果

所有评估都利用了 FashionIQ 基准，其中包括三个产品类别——衬衫、连衣裙和顶级上衣，每个类别大约有 50,000 对参考-目标图像对，每对都伴随有一到五个由人类撰写的编辑说明。官方 JSON 分割将大约% 80 的

图像对分配给训练，% 10 分配给验证，% 10 分配给测试，确保各个阶段之间没有图像重叠。图像调整为 224Œ224 像素大小，居中裁剪，并使用 ImageNet 统计数据（平均值 = [0.485, 0.456, 0.406]，标准差 = [0.229, 0.224, 0.225]）进行归一化。说明前缀为 `<|image|>`，标记化为固定长度（BLIP2 为 32 个标记，CLIP 为 77 个标记），并根据需要填充或截断。对于验证和测试，仅从合并的训练和验证图像中构建库嵌入——以防止数据泄漏——并使用 FAISS 的 `IndexFlatIP` 缓存到磁盘以实现快速检索。

4.1 零样例基线 (ZSCIR)

```
[sbulusuv@gpu021 SEARLE]$ python src/validate.py --exp-name oti_fiq --eval-type oti --dataset fashioniq --dataset-path ./datasets/FashionIQ
Target pad preprocess pipeline is used
Eval type = oti          exp name = oti_fiq
FashionIQ val - ['shirt'] dataset in classic mode initialized
/opt/sw/spack/apps/linux-rhel8-x86_64_v2/gcc-10.3.0/python-3.9.9-jh/lib/python3.9/site-packages/torch/utils/data/dataloader.py:563: UserWarnin
g: This DataLoader will create 10 worker processes in total. Our suggested max number of worker in current system is 6, which is smaller than
what this DataLoader is going to create. Please be aware that excessive worker creation might get DataLoader running slow or even freeze, lowe
r the worker number to avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
extracting image features FashionIQDataset - val
100%|| 199/199 [00:17<00:00, 11.52it/s]
FashionIQ val - ['shirt'] dataset in relative mode initialized
100%|| 64/64 [00:08<00:00,  7.61it/s]
shirt_fiq_recall_at10 = 22.52
shirt_fiq_recall_at50 = 39.70


FashionIQ val - ['dress'] dataset in classic mode initialized
extracting image features FashionIQDataset - val
100%|| 120/120 [00:07<00:00, 16.17it/s]
FashionIQ val - ['dress'] dataset in relative mode initialized
100%|| 64/64 [00:08<00:00,  7.61it/s]
dress_fiq_recall_at10 = 18.54
dress_fiq_recall_at50 = 37.04


FashionIQ val - ['toptee'] dataset in classic mode initialized
extracting image features FashionIQDataset - val
100%|| 168/168 [00:10<00:00, 16.45it/s]
FashionIQ val - ['toptee'] dataset in relative mode initialized
100%|| 62/62 [00:08<00:00,  7.39it/s]
toptee_fiq_recall_at10 = 22.64
toptee_fiq_recall_at50 = 43.04


average_fiq_recall_at10 = 21.24
average_fiq_recall_at50 = 39.92
```

Figure 1: 零样本组合图像检索性能（FashionIQ 验证集划分）。

在零样本组合图像检索设置中，我们直接使用 CLIP 的预训练 ViTB/32 视觉编码器和文本转换器，而无需任何额外的微调。每个查询由一个 224×224 参考图像和一个编辑字幕（以 `<|image|>` 为前缀，标记化为 77 个标记）组成。参考图像被编码为一个 1024 维的特征向量，字幕被编码为一个 512 维的特征。通过在 FAISS `IndexFlatIP` 中计算文本特征与每个图库图像特征之间的余弦相似度来执行检索，该 FAISS 包含每个类别约 10,000 张图像。整个检索流程，包括特征提取和最近邻搜索，在 NVIDIA A100 GPU 上平均每个查询花费 45 毫秒。

在 FashionIQ 验证集中，ZSCIR 在三个类别中的平均表现仅为 21.24 % Recall@10 和 39.92 % Recall@50。具体而言：

- 衬衫：22.52 % 重调用 @10，39.70 % 重调用 @50
- 连衣裙：18.54 % 召回 @10，37.04 % 召回 @50
- Toptee: 22.64 % 召回率 @10, 43.04 % 召回率 @50

这些不太显著的结果（图 1 ）强调了尽管 CLIP 的冻结编码器能够捕捉广泛的视觉-文本关联，但它们缺乏模拟精细化编辑指令的能力，比如颜色变化或图案添加。定性检查显示，ZSCIR 常常只检索到同一大类中的物品（例如，连衣裙对连衣裙），但未能体现编辑的细微差别（例如，“添加花卉图案”返回的是纯色连衣裙）。

4.2 微调 BLIP2 性能

我们的微调流程基于 BLIP2，这是一个最先进的视觉语言基础模型。我们首先加载预训练的 BLIP2 模型及其视觉和文本预处理器。我们在冻结的视觉编码器和语言解码器之间插入一个轻量级的 Q-Former 融合模块，以更好地对齐这两种模态。输入图像被缩放并填充为 224Œ224 像素，以保持纵横比；字幕使用 BLIP2 的 “eval” 文本处理器进行标记化。训练在 PyTorch 的 `GradScaler` 上以混合精度进行，以加速收敛并控制内存使用。我们使用 AdamW 优化器优化 BLIP2 的所有可训练参数，其超参数为 $\beta_1 = 0.9$ 、$\beta_2 = 0.98$ 、$\varepsilon = 10^{-7}$ 及权重衰减 0.05 。初始学习率为 1×10^{-4} ，我们在五个训练周期内使用一个 OneCycleLR 调度（div_factor=100,

pct_start=1.5/num_epochs)。我们使用 256 的批量大小，并在每个训练周期结束时进行验证。所有实验都在单个 NVIDIA A100 GPU 上进行，并设置固定的随机种子以确保结果的可重复性。

经过五次微调，我们的模型在所有三个 FashionIQ 类别中实现了强大的检索性能。表 1 总结了每个类别和平均召回率。

Category	R@10 (%)	R@50 (%)
Dress	40.06	63.47
Top tee	50.43	72.11
Shirt	45.58	67.12
Average	45.36	67.57

Table 1: 按类别在 FashionIQ 验证集上的 Recall@K。

图 2 显示了我们在训练过程中打印的验证指标的 JSON 格式输出。

```
Compute FashionIQ ['shirt'] validation metrics
{
    "dress_recall_at10": 40.059494972229004,
    "dress_recall_at50": 63.46058249473572,
    "toptee_recall_at10": 50.43345093727112,
    "toptee_recall_at50": 72.10606932640076,
    "shirt_recall_at10": 45.5839067697525,
    "shirt_recall_at50": 67.12462902069092,
    "average_recall_at10": 45.358950893084206,
    "average_recall_at50": 67.56376028060913,
    "average_recall": 56.461355586846665
}
```

Figure 2: FashionIQ 验证集上每个类别和平均召回率指标的 JSON 转储。

我们进行了消融研究，以区分 Q-Former 和混合精度训练的贡献。移除 Q-Former 模块将平均 Recall@10 减少约 4 个百分点，这强调了其对于模态融合的重要性。禁用混合精度导致的下降较小（约 1.5 个百分点），这表明稳定的梯度缩放有利于微调。我们进一步按字幕类型分析性能：当字幕描述广泛的属性变化（例如，“使其颜色更深”）时，检索效果最佳；但当字幕涉及细粒度图案（例如，“添加花卉图案”）时，会下降，此时 Recall@10 可能低于 35 %。

我们的结果表明，经过微调的 BLIP2 与 QFormer 融合在 FashionIQ 上实现了最先进的检索性能。相比于基于 CLIP 的基准，这些改进突显了专用融合模块和动态标题处理的价值。剩下的挑战包括处理细粒度的风格修改和探索特定区域的注意力。

4.3 检索 DPO 性能

我们还评估了一种检索–DPO 流程，其中我们仅在直接偏好优化（DPO）损失下微调 CLIP 的 transformer 文本塔。在此设置中，视觉编码器保持不变，文本 transformer 必须通过仅基于语言的参数更新来引导检索排名。

对于训练集中每张正样本目标图像，我们使用 CLIP 冻结的 ViT-L/14 视觉编码器预计算它们的图库嵌入，并使用 FAISS 索引它们。在每个训练步骤中，我们检索到正样本嵌入的前 50 个最近邻（不包括正样本自身），并从中采样一个作为难负样本。这一策略确保模型面临与目标视觉上相似的具挑战性的负样本，提供比随机负样本更强的监督信号。

我们采用 DPO 损失，该损失鼓励正样本对的相似性分数超过负样本对的相似性分数，其差距由 β 调节。形式上，对于每个大小为 B 的批次，我们最小化

$$\mathcal{L}_{\mathrm{DPO}} = -\frac{1}{B}\sum_{i=1}^{B} \log \sigma\big[\beta\,(s_i^{+} - s_i^{-})\big],$$

，其中 s_i^+ 和 s_i^- 分别是第 i 个标题与其正负图像嵌入的 CLIP 缩放余弦相似性，而 $\beta = 0.1$ 控制边界的锐度。

我们仅使用 AdamW 优化器对文本转换器参数进行微调，学习率为 $\mathrm{lr} = 1 \times 10^{-4}$，权重衰减为 $\mathrm{wd} = 1 \times 10^{-2}$。训练采用混合精度（'GradScaler'），批量大小为 256，每个 epoch 处理大约 60,000 个三元组。在 NVIDIA A100 GPU 上，每个 epoch 大约需要 9 个小时。

尽管训练损失稳定收敛（在 0.55 左右波动），但检索-DPO 流水线未能提高检索质量。图 3 显示了训练时期的验证召回率，它们基本上保持在随机水平。在保留的测试分割中，我们观察到：

- 召回率 @10: 0.02 %
- 召回率 @50: 0.10 %

```
[sbulusuv@gpu024 SEARLE]$ python src/train_dpo.py --dataset ./datasets/FashionIQ --out ./experiments/dpo_fiq --epochs 10 --batch 64 --workers 4
encode gallery: 100%|| 74381/74381 [22:44<00:00, 54.52it/s]
Epoch 1: 100%| | 563/563 [09:05<00:00,  1.03batch/s, loss=0.558]
Epoch 2: 100%| | 563/563 [09:03<00:00,  1.04batch/s, loss=0.573]
Epoch 3: 100%| | 563/563 [08:59<00:00,  1.04batch/s, loss=0.588]
Epoch 4: 100%| | 563/563 [08:59<00:00,  1.04batch/s, loss=0.567]
Epoch 5: 100%| | 563/563 [09:05<00:00,  1.03batch/s, loss=0.569]
Epoch 6: 100%| | 563/563 [09:04<00:00,  1.03batch/s, loss=0.572]
Epoch 7: 100%| | 563/563 [09:02<00:00,  1.04batch/s, loss=0.568]
Epoch 8: 100%| | 563/563 [08:59<00:00,  1.04batch/s, loss=0.534]
Epoch 9: 100%| | 563/563 [09:04<00:00,  1.03batch/s, loss=0.508]
Epoch 10: 100%| | 563/563 [09:03<00:00,  1.04batch/s, loss=0.553]
[sbulusuv@gpu024 SEARLE]$ python src/validate_dpo.py --dataset ./datasets/FashionIQ --weights ./experiments/dpo_fiq/epoch10.pt
encode gallery: 100%| | 74381/74381 [25:46<00:00, 48.10it/s]
validating: 100%| | 47/47 [01:44<00:00,  2.23s/it]
Recall@1  : 0.01
Recall@5  : 0.02
Recall@10 : 0.02
Recall@50 : 0.10
```

Figure 3: 检索-DPO 流水线在 10 个 epoch 期间的验证 Recall@10 和 Recall@50。尽管损失收敛，但指标仍接近随机机会。

4.3.1 失败的原因

由于视觉编码器保持冻结状态，并且没有引入交叉注意机制，文本转换器在没有直接访问视觉特征图的情况下工作。它无法将字幕中描述的语义编辑与实际图像内容对应，这严重限制了其在视觉上区分相似负样本的能力。

损失-指标不匹配 DPO 损失优化了连续的成对边际差异，而检索性能是通过离散的前 K 名排名来衡量的。边际的微小改进可能不会转化为排名的变化，尤其是当图库项目在嵌入空间中紧密聚集时。

从静态的 CLIP 嵌入空间中提取的

变量负质量 FAISS 挖掘负样本，其范围从不相关的图像到误导性的近似重复图像。这种不一致产生了噪声梯度信号，使得文本转换器难以学习在正样本和负样本之间稳定的区分。

由于图像骨干网络从未更新，因此文本和图像嵌入空间之间的任何错位都无法纠正。图像嵌入的静态特性固定了分布间隙，而文本转换器单独无法弥合这些间隙。

这些发现强调了要实现有效的合成图像检索，需要显式的多模态融合——例如交叉注意力或 Q-Former——以及与排序指标对齐的训练目标。对单一模态嵌入的简单偏好优化不足以在有意义的方式上影响前 K 名的检索性能。

5 局限性

我们的方法目前在五个关键方面存在局限性：它仅在 FashionIQ 服装图片上进行了调优和评估，使得其在其他领域的泛化能力尚未得到验证；它依赖于庞大且固定的 BLIP-2 视觉-语言骨干网络，这些骨干网络继承了预训练偏差，并且在资源受限的设备上进行部署成本高昂；其难负样本挖掘使用简单的 FAISS 最近邻查找，这产生了琐碎或几乎重复的负样本，提供了较弱的训练信号；检索 DPO 变体最小化成对边际，但不直接优化诸如 Recall@K 之类的排名指标，导致指标与目标之间的差距；最后，所有结论都是从自动化指标中得出的，没有人的判断，因此实际的相关性和主观的失败模式仍然未知。

6 结论

在这项工作中，我们针对 FashionIQ 基准进行了三种组合图像检索方法的评估：零样本 CLIP 基线、配有轻量级 Q-Former 融合头的 BLIP2 微调，以及基于检索的直接偏好优化（DPO）方法。我们的实验表明，微调 BLIP2——仅更新 3 % 的参数——带来了显著的改进，将 Recall@10 提高了一倍，Recall@50 相比零样本基线提升了近 28 个百分点。相比之下，检索-DPO 方法在使用 FAISS 挖掘的负样本的基础上，结合一个基于边界的损失来微调 CLIP 的文本转换器，未能对 top-K 检索产生实质性影响，这突显了显式多模态融合和排序对齐目标的重要性。

Q-Former 融合模块的成功证明，组合检索最大的益处来自那些能够共同关注图像块和文本编辑的架构，而不是将每种模式独立处理。我们对 Retrieval-DPO 失败模式的分析进一步说明，采用独立编码器的成对边缘目标在没有额外的跨模态交互机制或定制的负采样策略的情况下，无法转化为离散的排名提升。

未来的工作将探索更广泛的领域泛化（例如，CIRR，CLEVR），针对排序指标的替代损失函数，以及更复杂的负样本挖掘课程。总体而言，我们的研究结果表明，轻量级、基于融合的微调是构成图像检索一个可扩展且有效的范式。

整个训练和评估流程，以及预训练的检查点，可以在 `github.com/GautamaShastry/zscir` 找到。

References

[1] L. Agnolucci, A. Baldrati, M. Bertini, and A. Del Bimbo. *iSEARLE*: Improving textual inversion for zero-shot composed image retrieval. arXiv preprint arXiv:2405.02951, 2024.

[2] E. Rusak, P. Reizinger, A. Juhos, O. Bringmann, R. S. Zimmermann, and W. Brendel. InfoNCE: Identifying the gap between theory and practice. arXiv preprint arXiv:2407.00143, 2024.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020, 2021.

[4] J. Li, D. Li, S. Savarese, and S. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597, 2023.

[5] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. arXiv preprint arXiv:2305.18290, 2023.

[6] H. Wen, X. Zhang, X. Song, Y. Wei, and L. Nie. Target-guided composed image retrieval. arXiv preprint arXiv:2309.01366, 2023.

[7] G. Gu, S. Chun, W. Kim, H. Jun, Y. Kang, and S. Yun. CompoDiff: Versatile composed image retrieval with latent diffusion. *Transactions on Machine Learning Research*, 2023. arXiv:2303.11916.

[8] L. Wang, W. Ao, V. N. Boddeti, and S.-N. Lim. Generative zero-shot composed image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Poster # 340, 2025.

[9] G. Gu, S. Chun, W. Kim, Y. Kang, and S. Yun. Language-only training of zero-shot composed image retrieval (LinCIR). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13221–13232. IEEE, 2024.

[10] M. U. Anwaar, E. Labintcev, and M. Kleinsteuber. Compositional learning of image-text query for image retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1137–1147. IEEE, 2021.

[11] Z. Liu, C. Rodriguez-Opazo, D. Teney, and S. Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2125–2134. IEEE, 2021.

[12] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

[13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 1597–1607. PMLR, 2020.

[14] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer. LiT: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18123–18133. IEEE, 2022.