# OPENCLIP FINE-TUNING FOR MULTI-MODAL RETRIEVAL TASK ON FASHIONGEN DATASET

**Di Wu Fangyuan Shi Xiaowei Deng Navaneeth Biju Paulette Rever**
Georgia Institute of Technology
{dwu400, fshi46, xdeng304, nbiju6, Prever3}@gatech.edu

## ABSTRACT

Reliable and efficient image–to–language and language–to–image retrieval is essential in many real-world settings, with e-commerce being one of the most prominent domains where service providers must match product images with user-generated natural language queries. In this work, we explore CLIP-style multimodal retrieval models for a domain that is highly relevant to e-commerce and other product-description scenarios: Fashion. Specifically,

(1) we fine–tune three OpenCLIP variants—ViT-B/32, ViT-B/16 and the recent SigLIP2—on the Fashion-Gen dataset (∼270k image–caption pairs), using 260,490 pairs for training and 7,000 pairs for evaluation;
(2) we evaluate two loss functions, Information Noise–Contrastive Estimation (InfoNCE) and Binary Cross–Entropy (BCE), characterizing their effects on optimization and training dynamics; and
(3) we augment the Fashion-Gen captions using `gpt-5.1-mini`, and retrain the strongest model/loss configuration, yielding additional improvements.

Models are evaluated using Recall@5 and Recall@10 for both image–to–text and text–to–image retrieval. Across all models, fine–tuning provides ∼30% relative improvement over zero–shot baselines. ViT-B/16 outperforms ViT-B/32, suggesting that smaller patches and longer visual context better capture fine–grained apparel attributes. SigLIP2 attains the best performance, indicating that its architectural modifications and BCE–based objective are effective in this domain. We further analyze the impact of loss function choice and caption augmentation on overall retrieval quality.

Overall, our results show that for multimodal retrieval:

(i) CLIP–style dual–encoder architectures with linear projections remain strong, practical baselines for production retrieval,
(ii) BCE objectives are competitive and stable relative to InfoNCE, and are often preferable for distributed implementations, and
(iii) caption augmentation using generative models can provide a valuable source of additional training signal.

## 1 INTRODUCTION AND RELATED WORK

Modern consumer platforms—including e–commerce, social media, and search engines—must reliably link visual content to language, such as matching product images to catalog descriptions, user queries, or personalized tags. Dual–encoder CLIP–style models are well suited for such retrieval problems: they produce compact image/text embeddings that enable large–scale nearest–neighbor search while being one to two orders of magnitude smaller and cheaper to serve than multimodal LLMs, which are typically designed for generative tasks.

In industry, CLIP–style architectures already support production experiences via multimodal embeddings and vector similarity search, including:

- **Google Lens (“Shop Similar”)**: retrieves visually and semantically related fashion products using image–text or image–image similarity,

- **Amazon StyleSnap**: matches user–uploaded photos with catalog products via joint visual–text embeddings, and
- **Pinterest "Shop the Look"**: identifies clothing items within images and retrieves related products using multimodal vector search.

Research–wise, Radford et al. (2021) introduced the CLIP framework to learn a joint image–text embedding space using a contrastive objective. OpenCLIP (Ilharco et al., 2021) extends this formulation with additional model variants and open–source training code. More recent variants (SigLIP/SigLIP2) replace the softmax InfoNCE loss with pairwise sigmoid (BCE) objectives (Zhai et al., 2023), which decouple negatives and can stabilize optimization in small–batch or distributed settings. SigLIP2 (Tschannen et al., 2025) builds on this formulation by increasing model capacity and introducing alignment–focused objectives with systematic tuning across languages and domains, scaling BCE–based contrastive learning for vision–language pre–training.

Our contribution is in this direction. We provide an apples–to–apples comparison of ViT-B/32, ViT-B/16, and ViT-B/16-SigLIP2–256 using two loss functions (InfoNCE vs. BCE) fine–tuned on the Fashion-Gen dataset under modest computational budgets (a single T4 GPU). We develop an efficient fine–tuning pipeline and study the impact of caption augmentation on retrieval performance.

## 2 Dataset and Data Pipeline

The domain dataset, FashinGen, was first introduced in 2018 (Moda et al., 2018). It was one of the largest text–image datasets specifically curated for fashion AI. Although originally created to facilitate multimodal generative tasks, because of the way it is organized, it fits perfectly with our project's purpose. We used 260,490 pairs for training and 7,000 pairs for evaluation.

### 2.1 Data preparation with `webdataset`

We preprocessed the original dataset into `webdataset` shards for training and validation datasets, leveraging the `webdataset`'s capability of shuffling and streaming behavior. Each shard consists of 1,000 sample pairs, with the last training shard having the remaining 490 pairs.

### 2.2 Caption Augmentation

In addition to the original product descriptions, we augment each Fashion-Gen caption with two synthetic variants generated by the `gpt-5.1-mini` model. Our goal is to enrich the textual corpus while preserving the underlying semantic content, enabling the model to learn more robust image–text alignment from multiple natural language realizations of the same set of visual attributes.

To generate synthetic captions, we provide an instruction prompt that constrains the model to rephrase the original stylist–authored description without introducing new information. The model may reorder, regroup, or reword the content to produce paraphrases, but must preserve all attributes stated in the original text, including measurements, colors, materials, and specific style names. This ensures that the augmentation increases linguistic diversity without altering the attribute distribution of the dataset or adding hallucinated details.

The prompt used for caption generation is as follows:

```
Given a product description, your job is to create {num_aug}
alternative descriptions that:
- keep all and only the attributes present in the original
text
- do not add or guess any new details
- can reorder or regroup information
- can change wording and sentence structure
- preserve measurements, colors, materials, and style names
exactly
```

This controlled paraphrasing strategy increases linguistic variability while maintaining the fidelity of stylist annotations, providing additional supervised signal for contrastive image–text learning.

### 2.3 Dataset Availability

Please refer to Appendix A.2 for the WebDataset we prepared for both the original and augmented data.

## 3 Architectural Review

In this section we briefly review the CLIP-style dual-encoder architecture used in our experiments and then compare the three concrete model variants we fine-tune: ViT-B/32, ViT-B/16, and ViT-B/16-SigLIP2-256.

### 3.1 CLIP-Style Dual-Encoder Architecture

CLIP and OpenCLIP implement a symmetric dual-encoder design: an image encoder $f_{\text{img}}$ and a text encoder $f_{\text{text}}$ map inputs from their respective modalities into a shared embedding space of dimension $D$. At a high level:

$$x \in \mathbb{R}^{3\times H\times W} \xrightarrow{f_{\text{img}}} z^{\text{img}} \in \mathbb{R}^{D}, \qquad t \in \{\text{token ids}\}^{L} \xrightarrow{f_{\text{text}}} z^{\text{text}} \in \mathbb{R}^{D},$$

followed by $\ell_2$-normalization to get $\hat{z}^{\text{img}}$, $\hat{z}^{\text{text}}$. Their dot product $\hat{z}^{\text{img}} \cdot \hat{z}^{\text{text}}$ is exactly the cosine similarity used by the contrastive loss.

OpenCLIP adopts different implmentation details for the two towers. The image tower is a *ViT encoder* with bidirectional self-attention and a learned `[CLS]` token for global aggregation. The text tower is a *GPT-style decoder* with causal attention and uses an end-of-text (`EOT`) token to summarize the caption. Below we detail the tensor flow for a concrete base model, OpenCLIP ViT-B/32.

#### 3.1.1 Image Encoder (ViT-B/32)

The ViT image encoder processes a normalized batch of shape $X \in \mathbb{R}^{B\times 3\times 224\times 224}$ through the following steps:

1. **Patch embedding.** The image is divided into non-overlapping $32 \times 32$ patches, producing $7 \times 7 = 49$ tokens. Each flattened patch in $\mathbb{R}^{3072}$ is projected to $D_v = 768$:

   $$H^{(0)} \in \mathbb{R}^{B\times 49\times 768}.$$

2. **Adding `[CLS]` and positional embeddings.** A learned `[CLS]` token is prepended and absolute position embeddings are added:

   $$H^{(0)}_{\text{cls}} \in \mathbb{R}^{B\times 50\times 768}.$$

3. **Bidirectional Transformer encoder.** A 12-layer encoder applies unrestricted (bidirectional) self-attention:

   $$H^{(\ell)} \in \mathbb{R}^{B\times 50\times 768}, \quad \ell = 1, \ldots, 12.$$

4. **Global image representation.** The final hidden state of the `[CLS]` token provides the global embedding:

   $$e^{\text{img}} = H^{(12)}[:, 0, :] \in \mathbb{R}^{B\times 768}.$$

5. **Projection to joint space.** A linear projection maps the image representation into a $D = 512$-dimensional shared space:

   $$z^{\text{img}} = W_{\text{img}} e^{\text{img}}, \qquad W_{\text{img}} : \mathbb{R}^{768} \to \mathbb{R}^{512},$$

   followed by $\ell_2$-normalization.

### 3.1.2 Text Encoder (GPT-style Decoder)

Unlike the ViT encoder, the text tower uses causal attention and does not rely on a `[CLS]` token. Instead, an explicit `EOT` token is appended to each caption, and its final hidden state serves as the sequence representation.

1. **Tokenization.** Captions are tokenized with a maximum length $L_{\max} = 77$:

$$T \in \{0, \ldots, V-1\}^{B \times L}, \qquad L \leq 77.$$

2. **Embedding and positional encoding.** Tokens are embedded and added to learned positional encodings:

$$E^{(0)} = E_{\text{token}}(T) + E_{\text{pos}} \in \mathbb{R}^{B \times L \times 512}.$$

3. **Causal Transformer.** A 12-layer decoder applies left-to-right masked attention:

$$E^{(\ell)} \in \mathbb{R}^{B \times L \times 512}, \quad \ell = 1, \ldots, 12.$$

4. **Global text representation via `EOT`.** The hidden state of the final `EOT` token—which accumulates information from all preceding tokens—is taken as the caption embedding:

$$e^{\text{text}} \in \mathbb{R}^{B \times 512}.$$

5. **Projection to joint space.** The text embedding is mapped into the same shared space:

$$z^{\text{text}} = W_{\text{text}} e^{\text{text}}, \qquad W_{\text{text}} : \mathbb{R}^{512} \to \mathbb{R}^{512},$$

followed by $\ell_2$-normalization.

### 3.1.3 Shared Embedding Space and Similarity

Architecturally, CLIP and OpenCLIP can thus be viewed as two modality-specific transformers (vision and text) followed by small projection heads $W_{\text{img}}$ and $W_{\text{text}}$ that map both outputs into a common embedding space $\mathbb{R}^D$. For a batch of normalized embeddings $\hat{z}^{\text{img}} \in \mathbb{R}^{B \times D}$ and $\hat{z}^{\text{text}} \in \mathbb{R}^{B \times D}$, the similarity matrix

$$S_{ij} = \hat{z}_i^{\text{img}} \cdot \hat{z}_j^{\text{text}}$$

contains pairwise cosine similarities between all images and texts in the batch.

Because inference reduces to computing embeddings and cosine similarities in $\mathbb{R}^D$, this architecture is naturally compatible with embedding-based search methods such as approximate nearest neighbors (ANN), which is critical for large-scale retrieval systems.

## 3.2 Model Variants and Architectural Differences

We fine-tune three CLIP-style models on Fashion-Gen:

- **ViT-B/32 (OpenCLIP)**: baseline with $32 \times 32$ patches at $224 \times 224$ resolution.
- **ViT-B/16 (OpenCLIP)**: higher spatial resolution with $16 \times 16$ patches at $224 \times 224$.
- **ViT-B/16-SigLIP2-256**: a SigLIP2-based variant with $16 \times 16$ patches at $256 \times 256$, SigLIP-style BCE loss, and a smaller joint embedding dimension $D = 256$.

Table 1 summarizes the key architectural differences.

ViT-B/16 differs from ViT-B/32 only in patch size ($16 \times 16$ vs. $32 \times 32$), increasing image tokens from 49 to 196. They are both the older variants. The SigLIP2 variant introduces several design changes relative to OpenCLIP:

- **Pooling strategy.** Instead of a dedicated [CLS] token, SigLIP2 uses mean or MAP-style pooling over patch tokens to form the image representation. This mitigates the well-known "CLS bottleneck" in ViTs and can improve generalization.

Table 1: Summary of model variants used in our experiments. "Image tokens" counts patch tokens plus a [CLS] token where applicable.

| **Architecture** | **ViT-B/32** | **ViT-B/16** | **ViT-B/16-SigLIP2-256** |
|---|---|---|---|
| Patch size | $32 \times 32$ | $16 \times 16$ | $16 \times 16$ |
| Image resolution | $224 \times 224$ | $224 \times 224$ | $256 \times 256$ |
| Image tokens | $49 + 1$ (CLS) | $196 + 1$ (CLS) | 256 (MAP pooling) |
| Text context length | up to 77 tokens | up to 77 tokens | up to 64 tokens |
| Hidden dim (V / T) | $768/512$ | $768/512$ | $768/768$ |
| Transformer layers | 12 | 12 | 12 |
| Joint embedding dim $D$ | 512 | 512 | 256 |

- **Multilingual text encoder.** SigLIP2 employs a larger multilingual vocabulary (on the order of hundreds of thousands of tokens), leading to a larger text tower hidden size (768). However, in any given batch only a small subset of tokens is active, so the additional parameters do not linearly translate into training cost.
- **Smaller joint embedding dimension.** Both image and text towers output 768-dimensional features, but SigLIP2 projects them into a smaller joint space with $D = 256$. As reported in the SigLIP2 paper, this stabilizes training and is attractive for deployment: lower-dimensional embeddings reduce storage requirements and speed up nearest-neighbor search.
- **Loss function.** SigLIP2 uses a BCE-based contrastive objective instead of InfoNCE, which we analyze in detail in Section 4 and 5.5. This change decouples per-pair contributions from batch composition and, in our experiments, improves retrieval performance under small-batch training.

Although SigLIP2 uses a much larger text vocabulary, this does not affect training or inference speed, since only the token embeddings corresponding to the tokens present in a caption are accessed and updated. The runtime difference between our ViT-B/16 and ViT-B/16-SigLIP2-256 models comes almost entirely from the image side: the $256 \times 256$ resolution produces more patch tokens and therefore higher self-attention cost. In practice, ViT-B/16 fine-tunes in about 7 hours on a single T4 GPU, whereas SigLIP2 requires roughly 9 hours, with the gap attributable to the longer image sequences rather than the vocabulary size.

## 4 Loss Functions

We compare two contrastive objectives used in CLIP-style image–text retrieval: the standard softmax-based InfoNCE loss, and a pairwise binary cross-entropy (BCE) loss used in recent variants such as SigLIP and SigLIP2. Our goal is to understand how each image–text pair contributes to the loss, and how differences in formulation affect the learning signal and resulting embeddings.

We first present both losses using a unified notation. Let a mini-batch of $B$ aligned image–text pairs be given by $\{(x_i, t_i)\}_{i=1}^{B}$, and let the encoders produce $\ell_2$-normalized embeddings:

$$\tilde{e}_i^{\text{img}} = \frac{f_{\text{img}}(x_i)}{\|f_{\text{img}}(x_i)\|_2}, \qquad \tilde{e}_j^{\text{text}} = \frac{f_{\text{text}}(t_j)}{\|f_{\text{text}}(t_j)\|_2},$$

with cosine similarities

$$s_{ij} = \tilde{e}_i^{\text{img}} \cdot \tilde{e}_j^{\text{text}}.$$

A learnable temperature parameter $\tau = 1/\alpha$ rescales the logits $L_{ij} = s_{ij}/\tau$.

### 4.1 InfoNCE (softmax) loss

The standard CLIP loss treats each image as a query over the texts in the batch, using the paired text as the positive and all others as implicit negatives. The image-to-text direction is:

$$\mathcal{L}_{\text{i2t}}^{\text{InfoNCE}} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp(L_{ii})}{\sum_{j=1}^{B} \exp(L_{ij})}.$$

The text-to-image direction mirrors this:

$$\mathcal{L}_{\text{t2i}}^{\text{InfoNCE}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(L_{ii})}{\sum_{j=1}^{B} \exp(L_{ji})}.$$

The final loss is the symmetric average:

$$\mathcal{L}_{\text{InfoNCE}} = \tfrac{1}{2}\left(\mathcal{L}_{\text{i2t}}^{\text{InfoNCE}} + \mathcal{L}_{\text{t2i}}^{\text{InfoNCE}}\right).$$

Let $y_{ij} = 1$ if $i = j$ and $y_{ij} = 0$ otherwise. Define the softmax probability:

$$p_{ij} = \frac{\exp(L_{ij})}{\sum_{k=1}^{B} \exp(L_{ik})}.$$

For a fixed query $i$, the gradient of the loss with respect to $L_{ij}$ takes the familiar cross-entropy form:

$$\frac{\partial \mathcal{L}_{\text{i2t}}^{\text{InfoNCE}}}{\partial L_{ij}} = p_{ij} - y_{ij}.$$

Thus each pair $(i, j)$ contributes a term $(p_{ij} - y_{ij})$, bounded in $[-1, 1]$. For a positive pair $(i = i)$, we have $y_{ii} = 1$, and $p_{ii} \in (0, 1)$, so the gradient is negative, moving the image embedding toward the paired text embedding in the shared space. For negatives $j \neq i$, the gradient is positive, pushing embeddings apart.

The key structural property of the softmax is that $p_{ij}$ is defined *relative to all items in the batch*.

### 4.2 Pairwise BCE loss

The BCE formulation treats each similarity score independently, applying a sigmoid transformation to obtain a bounded scalar:

$$\sigma_{ij} = \sigma(L_{ij}) = \frac{1}{1 + \exp(-L_{ij})}.$$

With the same definition of $y_{ij}$, the pairwise BCE loss is

$$\ell_{ij}^{\text{BCE}} = -\left[y_{ij} \log \sigma_{ij} + (1 - y_{ij}) \log(1 - \sigma_{ij})\right],$$

and the full loss averages all pairs in both directions:

$$\mathcal{L}_{\text{BCE}} = \frac{1}{2B^2} \sum_{i=1}^{B} \sum_{j=1}^{B} \ell_{ij}^{\text{BCE}}.$$

The gradient for each pair is

$$\frac{\partial \ell_{ij}^{\text{BCE}}}{\partial L_{ij}} = \sigma_{ij} - y_{ij}.$$

As in InfoNCE, positives yield negative gradients and negatives yield positive gradients. However, unlike the softmax case, $\sigma_{ij}$ depends only on the pair $(i, j)$; it is entirely *independent* of the rest of the batch. In Section 5, we investigate how the loss function choice affects retrieval performance.

## 5 Evaluation & Results

### 5.1 Training Setup

All experiments are conducted on a single NVIDIA T4 GPU (16 GB). The memory constraints of this setup determine several training defaults: the global batch size, the use of mixed precision, and the decision to follow the standard OpenCLIP fine–tuning procedure where the transformer backbone is frozen and only the projection layers and temperature parameter are updated.

Contrastive learning typically benefits from large batch sizes (e.g., thousands of paired samples per iteration), since the loss relies on in–batch negatives. Due to hardware limitations, we adopt a batch size of 64 with `num_workers = 2`, which already saturates available VRAM on the T4. While not

Table 2: Final Recall@10 results on Fashion-Gen validation set. We report zero-shot performance, best fine-tuned checkpoint, and the improvement from fine-tuning.

| Model (loss) | Zero-shot | Best | Abs. Gain | Rel. Gain |
|---|---|---|---|---|
| ViT-B/32 (InfoNCE) | 0.5281 | 0.6936 | +0.1656 | +31.36% |
| ViT-B/16 (InfoNCE) | 0.5862 | 0.7635 | +0.1774 | +30.26% |
| ViT-B/16-SigLIP2 (InfoNCE) | 0.6185 | 0.7772 | +0.1588 | +25.67% |
| ViT-B/16-SigLIP2 (BCE) | 0.6185 | 0.7964 | +0.1779 | +28.77% |
| ViT-B/16-SigLIP2 (BCE, Aug. Dataset) | 0.6185 | 0.8150 | +0.1965 | +31.77% |

ideal for maximizing contrastive performance, this configuration enables a controlled comparison between InfoNCE and BCE objectives under identical training conditions.

We use AdamW as the optimizer for all models and loss functions. Following common practice in CLIP–style fine–tuning, we set the initial learning rate to $10^{-5}$ and apply decoupled weight decay. Convergence is smooth and stable across architectures and loss objectives, and we do not require per–model tuning.

We trained all models (except for the one on augmented dataset) with 3 epochs. Most of them plateau between 1 to 2 epochs.

### 5.2 Metric: Recall@$k$

We evaluate cross-modal retrieval performance using the standard metric Recall@$k$ (top-$k$ recall), specifically $R$@5 and $R$@10. Recall@$k$ directly measures how often the correct item is retrieved among the model's top-$k$ ranked results.

In our setting, each query (either an image or a text caption) has exactly one correct match in the dataset. Therefore, for a given query $q$, there are only two possible outcomes: (i) the correct match appears within the top-$k$ ranked candidates returned by the model (a true positive), or (ii) it does not (a false negative). Recall@$k$ asks: *"How often does the model put the correct answer in its top-$k$ predictions?"*

Formally, let TP denote the number of queries for which the correct match appears in the top-$k$, and FN the number of queries for which it does not. Recall@$k$ is defined as:

$$\text{Recall@}k = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

We compute Recall@$k$ separately in both directions:

- **Text-to-Image** (T2I)**:** given a caption, rank all images by cosine similarity and check whether the correct image appears in the top-$k$.
- **Image-to-Text** (I2T)**:** given an image, rank all captions and check whether the correct caption appears in the top-$k$.

The reported values represent the average recall over all validation queries.

### 5.3 Results Summary

We track the training dynamics using $R$@10 over all optimization steps and report the best checkpoint for each model. Table 2 summarizes the final Recall@$k$ values for the three fine-tuned models, compared against their corresponding zero-shot baselines. For a more comprehensive view, we present the complete step-wise results in the Appendix Section A.1. As shown in Figure 1, Recall@10 improves steadily during fine-tuning for all architectures. We can conclude that the domain finetuning works very well for the Fashion-Gen dataset, achieving the primary goal of this project.

### 5.4 Impact of Visual Patches

Moving from ViT-B/32 to ViT-B/16 increases the number of visual patch tokens from 49 to 196 for the same $224 \times 224$ input resolution. This finer patch granularity preserves more local structure (e.g.,

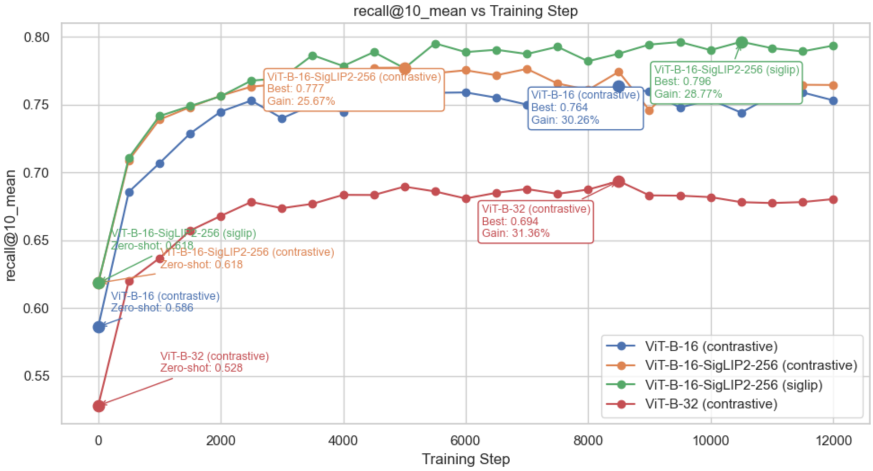

Figure 1: Training dynamics measured by average Recall@10 on the Fashion-Gen validation set.

fabric texture, decorative details, pattern alignment) while keeping the transformer depth and hidden sizes unchanged. In our results, this higher token count translates into a consistent performance gain: ViT-B/16 improves Recall@10 by $+0.014$ absolute over ViT-B/32 under identical training conditions.

While improved performance with smaller patches is expected theoretically, the effect is especially pronounced on Fashion-Gen. Many product descriptions are distinguished not by global silhouette but by subtle visual attributes: the presence of trim, lace patterns, clustered buttons, stitching structure, or fabric variations that cannot be captured reliably with larger patches. In this setting, increasing patch density is more than a generic improvement in spatial resolution—it enables the model to link fine-grained textual attributes to localized visual evidence. Thus, we attribute a significant portion of the gain in ViT-B/16 to its ability to retain fine detail relevant to deep attribute-level retrieval, rather than only improving global representation quality.

### 5.5 Impact of loss functions

The BCE-based SigLIP2 consistently outperforms InfoNCE-based SigLIP2, with the best checkpoints having a $1.9\%$ absolute advantage. We believe the performance difference comes from how the logit score is computed:

**Batch-coupled vs. pairwise-decoupled weighting.** In InfoNCE, softmax score $p_{ij}$ reflects the relative strength of $(i, j)$ compared to all $(i, k)$ in the batch. Two nearly identical examples placed in the same batch will compete with each other in the denominator, lowering their $p_{ij}$ values. If the same examples appear in different batches without similar neighbors, each receives a higher $p_{ij}$. This means the learning signal for a given pair can vary substantially depending on batch composition, especially at small batch sizes.

In BCE, sigmoid score $\sigma_{ij}$ is only a function of $s_{ij}$, so each pair contributes to the loss in a stable way regardless of what else is in the batch. This reduces variance in the learning signal when batch size is constrained.

**Computational implications.** The softmax requires computing and storing the full similarity matrix of size $B \times B$ (per direction) to normalize logits, leading to higher memory footprint and synchronization. BCE accesses each pair independently; the matrix can be processed in blocks, enabling lower-memory accumulation and finer-grained parallel strategies. This property makes BCE attractive for multi-device pre-training, especially when the batch size is decided to be very large.

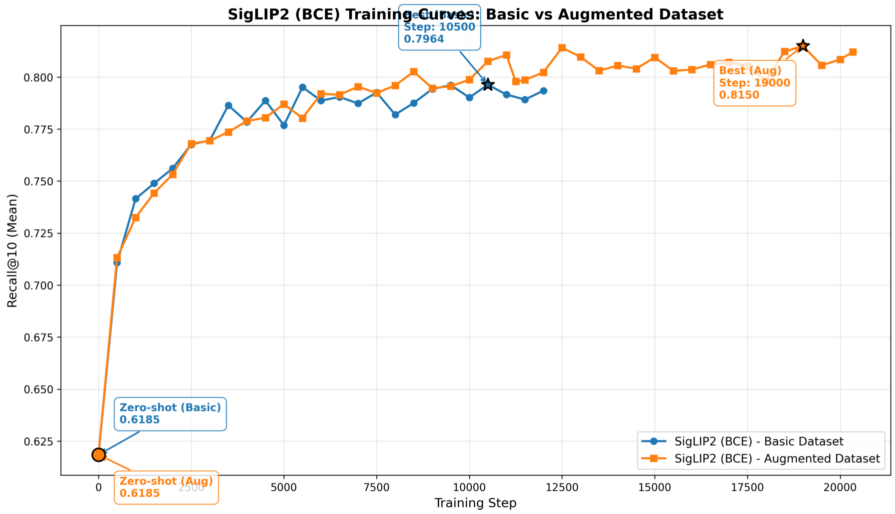

Figure 2: Training dynamics measured by average Recall@10 for basic dataset and augmented dataset.

### 5.6 Augmented Dataset Experiment

To assess whether caption augmentation improves multimodal alignment, we conduct an additional experiment using the strongest model and loss configuration identified above: ViT-B/16-SigLIP2 trained with the BCE objective. We execute the training based on the augmented dataset as described in Section 2.2.

During training, each WebDataset shard randomly emits *one* of the three captions (original + two paraphrases) for a given image at each sampling step. This implicitly enlarges the text distribution without modifying the image distribution or the total number of image–text pairs. At evaluation time, we only use the *original* captions from the Fashion-Gen validation split, ensuring a fair comparison with non-augmented baselines.

**Training Dynamics.** For the augmented dataset, we train the model for five epochs (versus three for the base dataset) to ensure sufficient exposure to distinct paraphrases for each image. As shown in Figure 2, both models improve at a similar rate early in training. After approximately 8,000 update steps, the augmented model begins to outperform the baseline consistently, while the baseline curve plateaus. We interpret this divergence as the point where repeated exposure to paraphrased captions begins to provide additional positive contrastive signal. By observing multiple linguistic realizations of identical visual content, the model learns a more robust mapping between semantic attributes and visual patterns, improving retrieval accuracy in the later stages of training.

**Results.** The augmented model achieves a Recall@10 of $0.8150$, an absolute improvement of $+0.019$ over the best non-augmented model ($0.7964$). Given the difficulty of fine-grained fashion retrieval and the consistently higher training curve beyond $80\%$, we consider this gain meaningful. These results indicate that controlled caption augmentation can strengthen image–text alignment in contrastive learning by increasing the effective linguistic variability per image. In contrastive terms, each image is associated with multiple positive examples that share the same attribute content but differ in lexical form, reducing overfitting to stylist-specific phrasing and encouraging attribute-level generalization within the shared embedding space.

**Data Release.** To support reproducibility and further research, we release all WebDataset shards used in this experiment, including both the original Fashion-Gen shards and the augmented dataset produced with our paraphrasing pipeline. Please refer to Appendix A.2.

# 6 Conclusion

In this work, we studied CLIP-style dual-encoder models for text–image retrieval in the fashion domain under realistic computational constraints. Using the Fashion-Gen dataset, we carried out an apples-to-apples comparison of three OpenCLIP variants (ViT-B/32, ViT-B/16, and ViT-B/16-SigLIP2-256) and two contrastive objectives (softmax-based InfoNCE vs. pairwise BCE), and evaluated their performance with Recall@5 and Recall@10 in both image-to-text and text-to-image directions.

Our experiments show that even on a single T4 GPU with modest batch sizes, fine-tuning yields substantial gains over zero-shot baselines, with relative improvements of roughly 25–32% in Recall@10. Increasing spatial resolution from ViT-B/32 to ViT-B/16 consistently improves retrieval quality, indicating that finer patch granularity better captures subtle fashion attributes such as pattern, trim, and fabric. Among loss functions, the BCE objective provides more stable and effective learning in this small-batch regime than InfoNCE, and the SigLIP2-based model with BCE achieves the strongest overall performance.

We further demonstrate that controlled caption augmentation can provide additional benefits. By generating paraphrases that preserve all stylist-annotated attributes, and randomly sampling one of three captions per image during training, we obtain an additional 1.9 percentage point absolute improvement in Recall@10 over our best non-augmented model. This suggests that increasing linguistic diversity without altering the underlying visual content can strengthen multimodal alignment, even without changing the visual dataset or training hardware.

There are several natural directions for future work. First, our study is limited to a single dataset and single-GPU setting; extending the comparison to larger-scale, multi-domain retrieval benchmarks and distributed training regimes would clarify how these findings scale. Second, we focused on standard CLIP-style pretraining and fine-tuning; incorporating hard-negative mining, curriculum strategies, or hybrid generative–contrastive objectives could further improve performance. Finally, from an applied perspective, integrating these models into a full retrieval stack with approximate nearest-neighbor search and latency/throughput measurements would close the gap between experimental results and production-ready fashion search systems.

# References

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cameron Gordon, Nicholas Carlini, Samir Gadre, Adam Roberts, Alexander D'Amour, Vivek Shankar, Ariel Kobren, Rohan Taori, Aparna Dave, Seungwon Yun, John Miller, Nitish Keskar, Armand Joulin, Justin Gilmer, Ben Recht, and Ludwig Schmidt. OpenCLIP. `https://github.com/mlfoundations/open_clip`, 2021. LAION/Research, 2021–2023.

Diego Moda, Rauan Gabbasov, et al. The fashion-gen dataset. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshop*, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision–language encoders with improved semantic understanding, localization, and dense features. `https://arxiv.org/abs/2502.14786v1`, 2025. arXiv preprint arXiv:2502.14786v1.

Xiaohua Zhai, Hugo Touvron, Basil Mustafa, Piotr Bojanowski, Zihang He, Hervé Jégou, Andrea Vedaldi, Matthijs Douze, Mathilde Caron, Armand Joulin, Diane Larlus, Oriol Vinyals Alexander Kolesnikov Araujo, Aaron van den Oord, and Laurent Cordonnier. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

# A Code and Resources

All code and experimental artifacts for this project are available in our public repository:

`https://github.com/Woodygoodenough/clip_finetuning`

The repository includes:

- **Training implementation** for all experiments, including model configuration, loss functions, and evaluation scripts.
- **Model variants** used in this work (ViT-B/32, ViT-B/16, and ViT-B/16-SigLIP2-256) with corresponding settings.
- **WebDataset support** and instructions for preparing the Fashion-Gen dataset in streaming shard format.
- **Caption augmentation pipeline** with prompt specification.
- **Evaluation suite** computing Recall@5 and Recall@10 for both retrieval directions.

## A.1 Step-wise Evaluation Logs

The repository contains all training logs used in the experiments, including:

- Full step-wise Recall@5 and Recall@10 metrics for both image-to-text and text-to-image retrieval.
- Logs for all three model architectures trained with different objectives.
- Separate logs for the augmented SigLIP2 model, which is trained for five epochs.

Results are stored in CSV format under `training_log/`. Usage instructions are provided in the repository README.

## A.2 Dataset Release

We release two versions of the Fashion-Gen dataset in WebDataset format (original and augmented). The dataset links and instructions for use are provided directly in the repository README.