Notes on Vision Language Pretraining

Nikos Giakoumoglou[1]

[1]Imperial College London

These notes provide a structured overview of vision-language pretraining, covering 49 methods published between 2021 and 2025. Starting from the foundational CLIP framework, we present each method with unified mathematical notation, architectural details, and training objectives. The surveyed papers span a wide range of research directions. Some focus on modifying the contrastive loss itself, replacing the standard InfoNCE with alternatives such as pairwise sigmoid classification (SigLIP), distribution matching (nCLIP), or cyclic consistency regularization (CyCLIP). Others augment the base contrastive objective with complementary self-supervised signals on images, including SimCLR-style invariance (SLIP), masked self-distillation (MaskCLIP), or local-to-global distillation à la DINO (SILC, TIPS). A separate line of work targets fine-grained alignment between visual patches and text tokens without resorting to expensive cross-attention (FILIP, HiCLIP, GLIP). Several methods address the quality of training data by generating synthetic captions with large language models (LaCLIP, DreamLIP, CLIPS, TripletCLIP), adaptively weighting noisy web pairs (ALIP), or selecting informative subsets (CLIPcov, ZeroVL). Long-text and spatial understanding are tackled through positional embedding redesign (LongCLIP), learnable corner tokens (LoTLIP), frozen LLM text towers (FLAME), and location-aware decoder objectives (LocCa, CLOC). Encoder-decoder hybrids that unify discriminative and generative training are also covered (BLIP, CoCa, Cap/CapPa), as are scaling studies that push model and data size to their limits (ALIGN, BASIC, EVA-CLIP, EVA-CLIP-18B, OpenCLIP). Finally, we include methods that improve compositional reasoning through hard negatives (NegCLIP, TripletCLIP, SuperCLIP), domain-specific adaptations (PLIP), and post-hoc inference-time techniques (ZLaP). Together, these notes document the progression from simple dual-encoder contrastive learning to sophisticated multi-task frameworks combining discriminative objectives, generative captioning, self-distillation, and spatial grounding.

Date: March 17, 2026
Correspondence: Nikos Giakoumoglou <nikos@imperial.ac.uk>

IMPERIAL

1 Introduction

Vision-language pretraining (*a.k.a.* image-text pretraining) aims to learn joint representations of images and text by leveraging large-scale paired data collected from the web. Given a dataset of image-text pairs $\{(\mathbf{x}_i, \mathbf{c}_i)\}$, where $\mathbf{x}_i$ denotes an image and $\mathbf{c}_i$ its corresponding textual description, the goal is to train an image encoder and a text encoder such that their output representations capture the semantic alignment between the two modalities. The core training signal comes from the pairing itself: matching images and their captions are considered positive pairs, while non-matching combinations are treated as negatives. In practice, the paired data is often noisy or limited in scale. A common setting is to also have access to a larger set of unpaired images $\{\mathbf{x}_i\}$ with no corresponding text supervision. Leveraging this additional unlabeled visual data alongside the paired set is an active area of research (Mu et al., 2021; Dong et al., 2023; Naeem et al., 2024), as it can improve the quality and generalization of the learned representations.

Each model in this space is composed of two (or more) subnetworks, each processing a single modality; these are referred to as **towers**. The fundamental distinction lies in whether a tower operates as an **encoder** or a **decoder**:

- **Image Encoder:** Maps the entire image to a fixed-size representation in a single forward pass. The architecture typically uses bidirectional attention where every patch attends to every other patch. Common choices include Vision Transformers such as ViT-B/16, ViT-L/14, ViT-G/14 (Radford et al., 2021; Cherti et al., 2023), convolutional networks like ResNet-50, EfficientNet-B7 (Jia et al., 2021), or specialized architectures such as Swin Transformer (Li et al., 2022b), CoSwin (Yuan et al., 2021), DaViT (Xiao et al., 2024), and EVA-ViT (Sun et al., 2024).

- **Text Encoder:** Processes the entire caption bidirectionally and produces a single global embedding in one pass. Every token attends to every other token simultaneously. Standard choices include transformer encoders such as BERT (Jia et al., 2021; Singh et al., 2022), RoBERTa (Cherti et al., 2023), or the text encoder component of CLIP's transformer (Radford et al., 2021), which reads the full sequence and extracts a representation from the final token or a special [CLS] token.
- **Text Decoder:** Generates output sequences autoregressively, one token at a time, using causal masking where each token can only attend to previously generated tokens. This architecture is suited for generative tasks such as image captioning. Examples include GPT-style causal transformers and the multimodal decoder stages in CoCa (Yu et al., 2022) and BLIP (Li et al., 2022a), which may additionally receive visual features via cross-attention to condition text generation on image content.

The choice of encoder versus decoder for each tower is a key architectural decision that distinguishes the methods surveyed here. In CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021), both towers are encoders (image encoder + text encoder), enabling efficient similarity computation for retrieval and classification. In contrast, models like CoCa (Yu et al., 2022) employ a image encoder paired with a text decoder split into unimodal and multimodal stages, combining contrastive learning with autoregressive caption generation. BLIP (Li et al., 2022a) introduces a unified architecture where the text side can function as either an encoder or decoder depending on the task.

2 Notation

We consider a dataset of N image-text pairs $\{(\mathbf{x}_i, \mathbf{c}_i)\}_{i=1}^N$, where $\mathbf{x}_i$ denotes an image and $\mathbf{c}_i$ denotes its corresponding textual description (*e.g.*, a caption, alt-text, or label). The framework comprises two parameterized models: an image encoder $f = f_\theta$ with parameters θ (*e.g.*, a vision transformer such as ViT-B/16, ViT-L/14, or a convnet such as ResNet-50) and a text encoder $g = g_\phi$ with parameters ϕ (*e.g.*, a GPT-style causal transformer or a BERT-style bidirectional encoder). Given an input pair $(\mathbf{x}_i, \mathbf{c}_i)$, the encoders produce embeddings

$$\mathbf{v}_i = f(\mathbf{x}_i), \qquad \mathbf{t}_i = g(\mathbf{c}_i),$$

which are mapped to a common d-dimensional embedding space via linear projection layers (denoted by π), *i.e.*, $\mathbf{v}_i, \mathbf{t}_i \in \mathbb{R}^d$. The embeddings are ℓ_2-normalized:

$$\mathbf{v}_i \leftarrow \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2}, \qquad \mathbf{t}_i \leftarrow \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|_2}, \qquad \|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1,$$

such that their cosine similarity reduces to the inner product $\mathbf{v}_i^\top \mathbf{t}_i$.

When data augmentation or masking is applied, we denote augmented views of an image as $\tilde{\mathbf{x}}_i$ (*e.g.*, via cropping, color jitter, or other photometric transformations) and their corresponding embeddings as $\tilde{\mathbf{v}}_i = f(\tilde{\mathbf{x}}_i)$. For methods employing multiple views (*e.g.*, SimCLR, SLIP, DeCLIP), we distinguish views with superscripts. Two augmented crops of the same image are created as

$$\mathbf{x}_i^{(1)} = \mathrm{aug}(\mathbf{x}_i), \qquad \mathbf{x}_i^{(2)} = \mathrm{aug}(\mathbf{x}_i),$$

producing embeddings

$$\mathbf{v}_i^{(1)} = f(\mathbf{x}_i^{(1)}), \qquad \mathbf{v}_i^{(2)} = f(\mathbf{x}_i^{(2)}).$$

For local crops (*e.g.*, in SILC, TIPS), local and global crops are generated as

$$\mathbf{x}_i^{\mathrm{loc}} = \mathrm{aug}_{\mathrm{loc}}(\mathbf{x}_i), \qquad \mathbf{x}_i^{\mathrm{glob}} = \mathrm{aug}_{\mathrm{glob}}(\mathbf{x}_i),$$

with corresponding embeddings

$$\mathbf{v}_i^{\mathrm{loc}} = f(\mathbf{x}_i^{\mathrm{loc}}), \qquad \mathbf{v}_i^{\mathrm{glob}} = f(\mathbf{x}_i^{\mathrm{glob}}).$$

Similarly, text augmentations (*e.g.*, synonym replacement, random deletion) are denoted $\tilde{\mathbf{c}}_i$, producing text embeddings $\tilde{\mathbf{t}}_i = g(\tilde{\mathbf{c}}_i)$.

For methods that operate on patch- or token-level features (*e.g.*, FILIP), we denote the sequence of visual token embeddings (excluding the [CLS] token) as $\mathbf{V}_i = \{\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,N_v}\} \in \mathbb{R}^{N_v \times d}$, where N_v is the number of visual patches (typically $N_v = H \times W$ for a grid of height H and width W). Analogously, the sequence of text token embeddings is denoted $\mathbf{T}_i = \{\mathbf{t}_{i,1}, \ldots, \mathbf{t}_{i,N_t}\} \in \mathbb{R}^{N_t \times d}$, where N_t is the number of text tokens. For masked inputs (*e.g.*, in MaskCLIP, FLIP, MAE-style methods), a masked image is created as

$$\mathbf{x}_i^{\text{mask}} = \text{mask}(\mathbf{x}_i, \rho),$$

where a fraction $\rho \in [0, 1]$ of patches are removed or replaced by a [MASK] token, producing the embedding

$$\mathbf{v}_i^{\text{mask}} = f(\mathbf{x}_i^{\text{mask}}).$$

The set of masked positions is denoted $\mathcal{M}_i \subseteq \{1, \ldots, N_v\}$.

In self-distillation frameworks (*e.g.*, MaskCLIP, SILC, TIPS), we introduce a teacher encoder $f' = f_{\theta'}$, whose parameters θ' are updated via an exponential moving average (EMA) of the student parameters:

$$\theta' \leftarrow \lambda\theta' + (1 - \lambda)\theta,$$

where $\lambda \in [0, 1]$ is the momentum coefficient. The teacher produces target embeddings

$$\mathbf{v}_i' = f'(\mathbf{x}_i),$$

which are used to supervise the student encoder.

For a batch of N input samples, the bi-directional contrastive loss is computed as

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{N}\left(\underbrace{\sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau)}}_{\text{image-to-text}} + \underbrace{\sum_{i=1}^{N} \log \frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{t}_i^\top \mathbf{v}_j/\tau)}}_{\text{text-to-image}}\right), \tag{1}$$

where τ is a learnable temperature parameter. The two directional terms correspond to image-to-text and text-to-image alignment objectives, respectively. This symmetric formulation treats each modality as both a query and a key, maximizing agreement between matched pairs $(\mathbf{v}_i, \mathbf{t}_i)$ while minimizing agreement with all other pairs in the batch, which serve as negatives.

3 Methodologies

This section presents a comprehensive survey of vision-language pretraining methods, organized chronologically and thematically to highlight the evolution of architectural innovations and training paradigms. We begin with the foundational dual-encoder approaches—CLIP and ALIGN—which established the contrastive learning framework that underlies most subsequent work. We then explore methods that enhance this foundation through various strategies: self-supervised augmentation techniques that leverage unlabeled visual data (SLIP, DeCLIP, MaskCLIP, SILC), fine-grained alignment mechanisms that capture token-level correspondences (FILIP, HiCLIP, GLIP), and loss function innovations that improve training efficiency and representation quality (SigLIP, nCLIP, CyCLIP). The survey covers encoder-decoder hybrids that unify discriminative and generative objectives (BLIP, CoCa), multi-tower architectures that incorporate additional supervision signals (FLAVA, 3T), and unified frameworks designed for broad task coverage (Florence, Florence-2). We examine methods addressing practical challenges such as noisy web data (ALIP, ZeroVL, NegCLIP), long-text understanding (DreamLIP, LongCLIP, LotLIP, FLAME), and spatial grounding (TIPS, LocCa, CLOC). Finally, we discuss recent scaling efforts (BASIC, EVA-CLIP, EVA-CLIP-18B, OpenCLIP) and specialized approaches for compositional reasoning (TripletCLIP, SuperCLIP) and domain-specific applications (PLIP, ZLaP).

3.1 CLIP (OpenAI, 2021, ICML 2021)

CLIP (Radford et al., 2021) is a dual-encoder model. Both towers are encoders: the image encoder $f = f_\theta$ (e.g., a vision transformer like ViT-B/16 or a ConvNet like ResNet-50) and the text encoder $g = g_\phi$ (e.g., a GPT-style causal transformer that processes the full caption and outputs a single embedding extracted from the final token) independently map their respective inputs into a shared embedding space, using linear projection layers (`nn.Linear`), with no interaction between the two during encoding. Given an image $\mathbf{x}_i$ and its caption $\mathbf{c}_i$, they produce

$$\mathbf{v}_i = f(\mathbf{x}_i), \qquad \mathbf{t}_i = g(\mathbf{c}_i),$$

and both embeddings are mapped to a common d-dimensional space via π and ℓ_2-normalized so that $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$, reducing cosine similarity to $\mathbf{v}_i^\top \mathbf{t}_i$.

Given a batch of N image-text pairs, CLIP is trained to predict which of the N possible pairings actually occurred, i.e., maximize the cosine similarity $\mathbf{v}_i^\top \mathbf{t}_i$ of the N correct pairs while minimizing it for the $N^2 - N$ incorrect ones. The model is optimized with the symmetric InfoNCE loss:

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{N}\left(\sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau)} + \sum_{i=1}^{N} \log \frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{t}_i^\top \mathbf{v}_j/\tau)}\right),$$

where τ is a learnable temperature parameter, initialized at 0.07; this matches the bi-directional contrastive loss defined in Equation (1) with the same image-to-text and text-to-image terms.

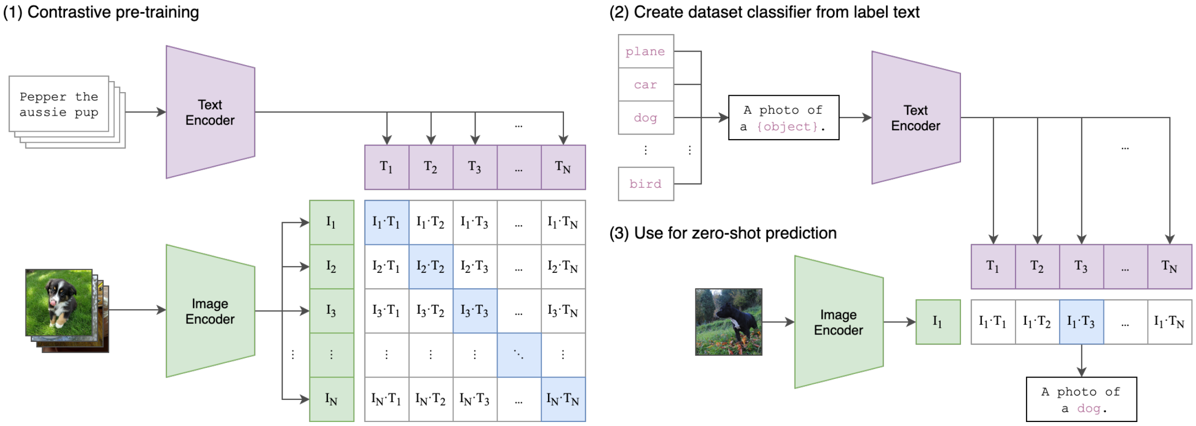

(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ⋮ I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
⋮ ⋮ ⋮ ⋱ ⋮
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
T_1 T_2 T_3 ... T_N
I_1
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
A photo of a dog.

Figure 1 CLIP uses dual encoders to map images and text into a shared embedding space, trained with symmetric InfoNCE loss on image-text pairs.

3.2 ALIGN (Google, 2021, ICML 2021)

ALIGN (Jia et al., 2021) follows the same dual-encoder architecture as CLIP: an image encoder $f = f_\theta$ (e.g. EfficientNet) and a text encoder $g = g_\phi$ (e.g. BERT with [CLS] token) map images $\mathbf{x}_i$ and captions $\mathbf{c}_i$ to embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$ in a shared space, optimized with the same symmetric InfoNCE loss $\mathcal{L}_{\text{CLIP}}$ defined in Equation (1). The key difference lies in the data regime: while CLIP trains on a curated dataset of 400M image-text pairs, ALIGN scales to over 1 billion pairs obtained directly from image alt-text on the web, with no expensive filtering or post-processing. ALIGN demonstrates that the sheer scale of the training data can compensate for its inherent noise, achieving state-of-the-art performance and surpassing models trained on cleaner but smaller datasets.

3.3 DeCLIP (SenseTime, 2021, ICLR 2022)

DeCLIP (Li et al., 2022c) improves vision-language pretraining efficiency by exploiting widespread supervision within image-text pairs. Following CLIP's dual-encoder architecture, it employs an image encoder $f = f_\theta$ (e.g., ResNet-50 or ViT-B/32) and a text encoder $g = g_\phi$ (e.g., CLIP's Transformer), both outputting embeddings via linear projection layers (`nn.Linear`) followed by ℓ_2-normalization so that $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$.

For each training sample, DeCLIP generates **two augmented views** of both modalities:

$$\mathbf{x}_i = \text{aug}_{\text{img}}(\mathbf{x}_i), \quad \tilde{\mathbf{x}}_i = \text{aug}_{\text{img}}(\mathbf{x}_i), \quad \mathbf{c}_i = \text{aug}_{\text{text}}(\mathbf{c}_i), \quad \tilde{\mathbf{c}}_i = \text{aug}_{\text{text}}(\mathbf{c}_i),$$

where image augmentations include RandomResizedCrop and ColorJitter, and text augmentations use EDA (Easy Data Augmentation: synonym replacement, random swap, random deletion). These are encoded to produce:

$$\mathbf{v}_i = f(\mathbf{x}_i), \quad \tilde{\mathbf{v}}_i = f(\tilde{\mathbf{x}}_i), \quad \mathbf{t}_i = g(\mathbf{c}_i), \quad \tilde{\mathbf{t}}_i = g(\tilde{\mathbf{c}}_i).$$

DeCLIP combines **five** supervision signals:

1. **CLIP loss**: The symmetric InfoNCE loss between the first augmented image-text pair:

$$\mathcal{L}_{\text{CLIP}} = \mathcal{L}_{\text{InfoNCE}}(\mathbf{v}_i, \mathbf{t}_i) = -\frac{1}{N}\left(\sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau)} + \sum_{i=1}^{N} \log \frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{t}_i^\top \mathbf{v}_j/\tau)}\right),$$

which matches the standard CLIP loss $\mathcal{L}_{\text{CLIP}}$ in Equation (1) evaluated on the original views $(\mathbf{v}_i, \mathbf{t}_i)$.

2. **Image self-supervision ($\mathcal{L}_{\text{I-SS}}$)**: Following SimSiam, a predictor MLP h is applied to one view, and the loss maximizes similarity with the stop-gradient of the other view:

$$\mathcal{L}_{\text{I-SS}} = -\frac{1}{2N}\sum_{i=1}^{N}\left(\frac{h(\mathbf{v}_i)^\top \text{sg}(\tilde{\mathbf{v}}_i)}{\|h(\mathbf{v}_i)\|_2 \|\text{sg}(\tilde{\mathbf{v}}_i)\|_2} + h(\tilde{\mathbf{v}}_i)^\top \text{sg}(\mathbf{v}_i) / \big(\|h(\tilde{\mathbf{v}}_i)\|_2 \|\text{sg}(\mathbf{v}_i)\|_2\big)\right).$$

3. **Text self-supervision ($\mathcal{L}_{\text{T-SS}}$)**: Following BERT's Masked Language Modeling (MLM), 15% of tokens in each caption are randomly masked and the model predicts the original token using cross-entropy loss:

$$\mathcal{L}_{\text{T-SS}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t \in \mathcal{M}_i} \log P(c_{i,t} \mid \mathbf{c}_{i,\setminus t}; g_\phi),$$

where $\mathcal{M}_i$ denotes the set of masked positions in caption $\mathbf{c}_i$.

4. **Multi-view supervision ($\mathcal{L}_{\text{MVS}}$)**: Inspired by SwAV's multi-crop strategy, DeCLIP contrasts all $2 \times 2 = 4$ image-text view combinations. Since $\mathcal{L}_{\text{CLIP}}$ already covers $(\mathbf{v}_i, \mathbf{t}_i)$, the remaining 3 pairs (each symmetrized) provide additional supervision:

$$\mathcal{L}_{\text{MVS}} = \mathcal{L}_{\text{InfoNCE}}(\mathbf{v}_i, \tilde{\mathbf{t}}_i) + \mathcal{L}_{\text{InfoNCE}}(\tilde{\mathbf{v}}_i, \mathbf{t}_i) + \mathcal{L}_{\text{InfoNCE}}(\tilde{\mathbf{v}}_i, \tilde{\mathbf{t}}_i).$$

5. **Nearest-neighbor supervision ($\mathcal{L}_{\text{NNS}}$)**: Following NNCLR, a FIFO queue $\mathcal{Q}$ of size 65K stores text embeddings. For each image embedding $\mathbf{v}_i$, the nearest-neighbor text embedding in the queue is retrieved:

$$\mathbf{t}_i^{\text{NN}} = \arg\max_{\mathbf{q} \in \mathcal{Q}} (\mathbf{v}_i^\top \mathbf{q}),$$

and contrastive loss is computed with both image views:

$$\mathcal{L}_{\text{NNS}} = \mathcal{L}_{\text{InfoNCE}}(\mathbf{v}_i, \mathbf{t}_i^{\text{NN}}) + \mathcal{L}_{\text{InfoNCE}}(\tilde{\mathbf{v}}_i, \mathbf{t}_i^{\text{NN}}).$$

The **total DeCLIP loss** combines these five objectives with weights $\alpha = \beta = \gamma = 0.2$:

$$\mathcal{L}_{\text{DeCLIP}} = (1 - \alpha - \beta - \gamma)\mathcal{L}_{\text{CLIP}} + \alpha(\mathcal{L}_{\text{I-SS}} + \mathcal{L}_{\text{T-SS}}) + \beta\mathcal{L}_{\text{MVS}} + \gamma\mathcal{L}_{\text{NNS}}.$$

3.4 FILIP (Huawei, 2021, ICLR 2022)

FILIP (Yao et al., 2022) modifies the dual-encoder architecture (e.g., ViT-L/14 and Transformer) to capture finer interactions without the computational cost of a full cross-attention multimodal encoder. Instead of computing cosine similarity between global embeddings $\mathbf{v}_i$ and $\mathbf{t}_i$ in $\mathbb{R}^d$, FILIP computes token-wise similarities between all image patches and all text tokens.

Let $\mathbf{V}_i = \{\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,N_v}\} \in \mathbb{R}^{N_v \times d}$ be the sequence of visual token embeddings (excluding the [CLS] token) and $\mathbf{T}_j = \{\mathbf{t}_{j,1}, \ldots, \mathbf{t}_{j,N_t}\} \in \mathbb{R}^{N_t \times d}$ be the sequence of text token embeddings, with ℓ_2-normalized entries $\|\mathbf{v}_{i,m}\|_2 = \|\mathbf{t}_{j,k}\|_2 = 1$. The similarity between the k-th text token $\mathbf{t}_{j,k}$ and the image $\mathbf{x}_i$ is defined as its maximum similarity to any visual token:

$$s(\mathbf{t}_{j,k}, \mathbf{V}_i) = \max_{1 \le m \le N_v} \mathbf{t}_{j,k}^\top \mathbf{v}_{i,m}.$$

The total image-text similarity is the average of these maximum similarities:

$$S(\mathbf{x}_i, \mathbf{c}_j) = \frac{1}{N_t} \sum_{k=1}^{N_t} s(\mathbf{t}_{j,k}, \mathbf{V}_i).$$

This fine-grained similarity $S(\mathbf{x}_i, \mathbf{c}_j)$ replaces the standard dot product $\mathbf{v}_i^\top \mathbf{t}_j$ in the symmetric InfoNCE loss $\mathcal{L}_{\text{FILIP}}$, which otherwise follows the same bi-directional form as Equation (1). This allows the model to align specific objects in the image with specific words in the caption (e.g., matching the word "*dog*" to the dog patch) efficiently.

3.5 BASIC (Google, 2021, NeurIPS 2021)

BASIC (Pham et al., 2023) demonstrates that the performance of vision-language models scales predictably with the expansion of three key dimensions: data size, model size, and batch size. While CLIP and ALIGN established the dual-encoder foundation—using image encoder $f = f_\theta$ and text encoder $g = g_\phi$ to map images $\mathbf{x}_i$ and captions $\mathbf{c}_i$ to ℓ_2-normalized embeddings $\mathbf{v}_i, \mathbf{t}_i \in \mathbb{R}^d$ and training with the symmetric InfoNCE loss $\mathcal{L}_{\text{CLIP}}$ in Equation (1)—BASIC pushes these limits by training a massive image tower f_θ (a CoAtNet-7 with 2.4B parameters) and a text tower g_ϕ (a large BERT-style transformer) on a dataset of 6.6B image-text pairs.

3.6 LiT (Google, 2021, CVPR 2022)

LiT (Zhai et al., 2022) investigates the most effective way to align pre-trained models across modalities. Architecturally, LiT is a dual-encoder model similar to CLIP, featuring an image encoder $f = f_\theta$ (e.g., a large pre-trained ViT-g/14) and a text encoder $g = g_\phi$ (e.g., a BERT-style transformer). The model is optimized using the standard symmetric InfoNCE loss $\mathcal{L}_{\text{CLIP}}$ to align the visual and textual embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$ in a shared space, with $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$.

The central premise of LiT is the "*locking*" of one or more towers during the alignment process. The "*unlocked*" tower can be either pretrained or random init. The authors evaluate three distinct training configurations:

1. 🔥 **Unlocked Image,** ❄ **Locked Text:** Tuning the image encoder to match a frozen text representation.
2. 🔥 **Unlocked Image,** 🔥 **Unlocked Text:** The standard CLIP-style training where both encoders are optimized simultaneously (can be either random init, i.e., CLIP or pretrained).
3. ❄ **Locked Image,** 🔥 **Unlocked Text:** The image tower is initialized with a high-capacity model pre-trained on a massive dataset (e.g., ImageNet-21k) and is frozen, while only the text tower g_ϕ is trained.

The authors investigate the following combinations: *"LU"*, *"Lu"*, *"UU"*, *"Uu"*, *"uU"*, *"uu"*, *"UL"*, *"uL"*, *"LL"* (first letter image, second letter text tower; capital letter pretrained, lowercase letter random init).

The empirical findings show that **Locked Image (pretrained)** ❄**, Unlocked Text (random init)** 🔥 is the most effective setting. By keeping the vision tower frozen ❄, the model retains the robust visual features learned from clean, supervised data, which might otherwise be degraded by the noise inherent in web-scale image-text pairs. This method allows the text tower to "*learn the language*" of a pre-existing, high-quality visual space.

3.7 Florence (Microsoft, 2021, arXiv 2021)

Florence (Yuan et al., 2021) adopts the same dual-encoder architecture and UniCL objective as in the CLIP-style setup, with image encoder $f = f_\theta$ and text encoder $g = g_\phi$ mapping images $\mathbf{x}_i$ and captions $\mathbf{c}_i$ to ℓ_2-normalized embeddings $\mathbf{v}_i, \mathbf{t}_i \in \mathbb{R}^d$, optimized with a symmetric InfoNCE-style loss, but scales it massively: a hierarchical Vision Transformer named CoSwin for vision and CLIP's transformer for text, trained on the FLD-900M dataset (900M+ image-text pairs with 5M+ unique concepts). Florence uses the identical UniCL loss $\mathcal{L}_{\text{Florence}} = \mathcal{L}_{\text{UniCL}}$, which corresponds to bi-directional contrastive alignment as in Equation (1). The key difference from UniCL is **scale and scope**: Florence demonstrates that this unified objective scales to foundation-model size, and introduces task-specific adapter modules for diverse downstream tasks (object detection, segmentation, VQA, etc.), establishing it as a universal vision foundation model rather than just a learning framework.

3.8 VIOLET (Microsoft, 2021, CVPR 2022)

VIOLET (Fu et al., 2022) is an end-to-end video-language transformer. It uses a Video Swin Transformer to extract sparse visual tokens $\mathbf{V}_i \in \mathbb{R}^{N_v \times d}$ (excluding [CLS]-style tokens), which are fed along with text tokens $\mathbf{T}_i \in \mathbb{R}^{N_t \times d}$ extracted from a language embedder BERT into a multimodal transformer block, also BERT-style. The model is trained with three supervision signals:

1. **Video-text contrastive loss ($\mathcal{L}_{\text{VTC}}$)**: A symmetric InfoNCE-style contrastive loss between global video and text embeddings $\mathbf{v}_i = h(\mathbf{V}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$, analogous to $\mathcal{L}_{\text{CLIP}}$ in Equation (1), aligning matched video-text pairs while treating others as negatives.
2. **Masked visual-token modeling ($\mathcal{L}_{\text{MVM}}$)**: Randomly masked video tokens are reconstructed by predicting discrete VAE (d-VAE) codes (i.e., DALL-E-style targets), with cross-entropy over the discrete visual token space.
3. **Masked language modeling ($\mathcal{L}_{\text{MLM}}$)**: Standard BERT-style MLM on the text tokens, maximizing the likelihood of the original tokens given the context.

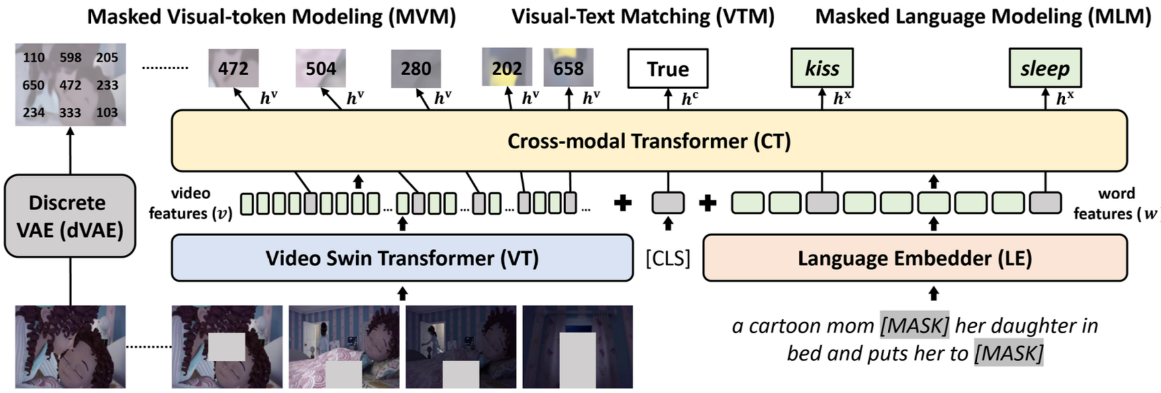

Masked Visual-token Modeling (MVM)
Visual-Text Matching (VTM)
Masked Language Modeling (MLM)
110 598 205
650 472 233
234 333 103
472
504
280
202
658
True
kiss
sleep
h^v
h^c
h^x
Cross-modal Transformer (CT)
Discrete VAE (dVAE)
video features (v)
word features (w)
Video Swin Transformer (VT)
[CLS]
Language Embedder (LE)
a cartoon mom [MASK] her daughter in bed and puts her to [MASK]

Figure 2 VIOLET extends vision-language pretraining to video by encoding temporal sequences with masked video token modeling and video-text matching objectives.

3.9 GLIP (Microsoft, 2021, CVPR 2022)

GLIP (Li et al., 2022b) reformulates object detection as a phrase grounding task to unify vision-language pre-training and object recognition. Unlike CLIP, which aligns global embeddings $\mathbf{v}_i, \mathbf{t}_i \in \mathbb{R}^d$, GLIP performs word-level alignment between N image region features and M text tokens. The architecture typically uses an Swin-Transformer image encoder $f = f_\theta$ and a BERT text encoder $g = g_\phi$.

The model computes an alignment score between N visual region features and M word tokens, $\mathbf{V}_i \in \mathbb{R}^{N \times d}$ and $\mathbf{T}_i \in \mathbb{R}^{M \times d}$. The core innovation is the **Deep Cross-Modality Fusion**, where layers of the two towers interact via cross-attention before the final embedding. The grounding loss is formulated to align specific regions with their corresponding phrases:

$$\mathcal{L}_{\text{GLIP}} = \mathcal{L}_{\text{grounding}} + \mathcal{L}_{\text{box\_regression}}.$$

3.10 FLAVA (Meta, 2021, CVPR 2022)

FLAVA (Singh et al., 2022) introduces a three-tower architecture: an image encoder $f = f_\theta$ (e.g., ViT-B/16), a text encoder $g = g_\phi$ (e.g., BERT-base), and a multimodal encoder $h = h_\psi$ (e.g., a 12-layer Transformer). The multimodal encoder accepts the concatenated sequence of visual tokens $\mathbf{V}_i$ and textual tokens $\mathbf{C}_i$ from the respective unimodal encoders. The global multimodal representation is $\mathbf{m}_i = h([\mathbf{V}_i; \mathbf{C}_i])$.

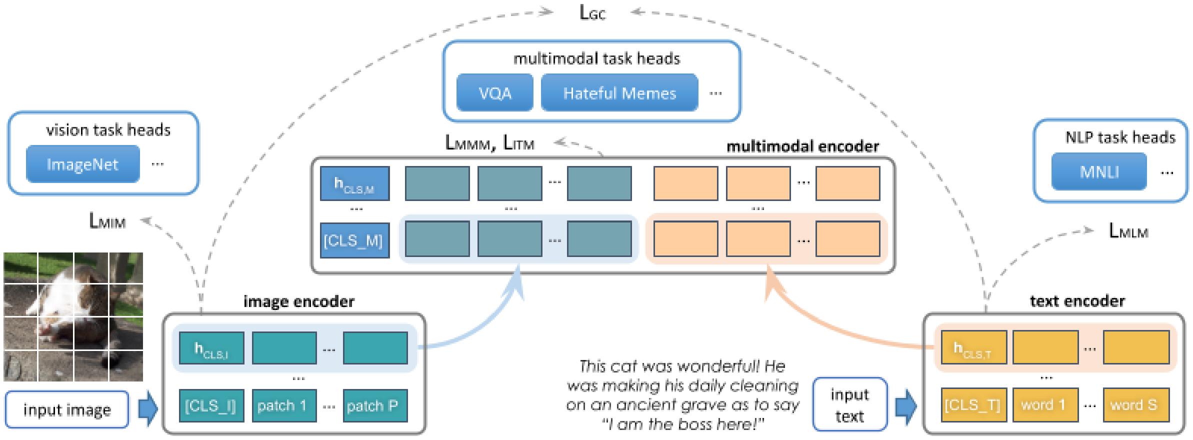


Figure 3 FLAVA uses separate image, text, and multimodal encoders with masked modeling objectives (MIM, MLM, and MMM) alongside contrastive learning.

FLAVA combines **four** supervision signals:

1. **Global contrastive loss ($\mathcal{L}_{\text{CLIP}}$):** This aligns the unimodal ℓ_2-normalized embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$ via the symmetric InfoNCE loss with a learnable τ, matching the bi-directional form in Equation (1).

2. **Masked multimodal modeling ($\mathcal{L}_{\text{MMM}}$):** A hybrid BEiT/BERT objective. Tokens in both $\mathbf{V}_i$ and $\mathbf{C}_i$ are masked, and h_ψ must reconstruct them using cross-modal context:

$$\mathcal{L}_{\text{MMM}} = -\mathbb{E}\left[\log P(\text{masked patches} \mid \mathbf{V}_{\setminus m}, \mathbf{C}_{\setminus m}) + \log P(\text{masked tokens} \mid \mathbf{V}_{\setminus m}, \mathbf{C}_{\setminus m})\right].$$

3. **Unimodal masked modeling:** The encoders f_θ and g_ϕ are trained with Masked Image Modeling ($\mathcal{L}_{\text{MIM}}$) using discrete VAE tokens (BEiT style) and Masked Language Modeling ($\mathcal{L}_{\text{MLM}}$) (BERT style):

$$\mathcal{L}_{\text{MIM}} = -\sum_{k \in \text{masked}} \log P(q_k \mid \mathbf{V}_{\setminus k}), \qquad \mathcal{L}_{\text{MLM}} = -\sum_{t \in \text{masked}} \log P(c_t \mid \mathbf{C}_{\setminus t}).$$

The final loss is $\mathcal{L}_{\text{FLAVA}} = \mathcal{L}_{\text{CLIP}} + \mathcal{L}_{\text{MMM}} + \mathcal{L}_{\text{MIM}} + \mathcal{L}_{\text{MLM}}$.

3.11 ZeroVL (Google, 2021, ECCV 2022)

ZeroVL (Cui et al., 2022) is a dual-encoder framework that retains the core CLIP architecture (e.g., ViT-B/16 or ViT-L/14 and a Transformer text encoder), with image encoder $f = f_\theta$ and text encoder $g = g_\phi$ producing ℓ_2-normalized embeddings $\mathbf{v}_i, \mathbf{t}_i \in \mathbb{R}^d$ optimized via symmetric InfoNCE as in Equation (1), but introduces a suite of specific modifications to the training recipe:

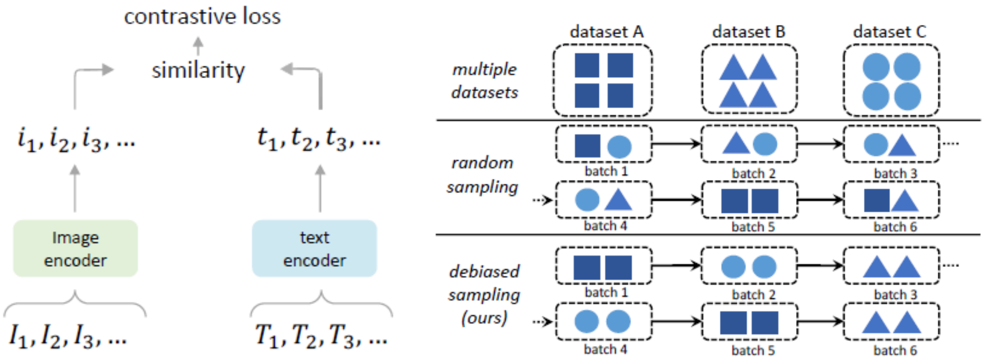


Figure 4 **Left**: CLIP dual-encoder architecture. **Right**: Debiased sampling.

1. **Debiased sampling:** When a batch is composed of samples from different datasets, models could be driven to distinguish negative samples by hacking the source information, i.e., learning the dataset bias. Debiased sampling ensures instances within each batch come from the same dataset, so that the InfoNCE negatives are not confounded by distribution shifts.
2. **Mixup augmentation:** To expand the training distribution, it linearly interpolates between image-text pairs: $\tilde{\mathbf{x}} = \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j$ and $\tilde{\mathbf{c}} = \lambda \mathbf{c}_i + (1-\lambda)\mathbf{c}_j$, forcing the model to predict the mixed target under the same contrastive loss $\mathcal{L}_{\text{CLIP}}$.
3. **Decoupled gradient accumulation:** Large batch sizes are crucial for InfoNCE. ZeroVL decouples the gradient computation to allow accumulating gradients over micro-batches while maintaining the large effective batch size required for negative sampling, keeping $\mathcal{L}_{\text{CLIP}}$ scalable.
4. **Token dropping & distillation:** It randomly masks input tokens (similar to FLIP/MAE) to speed up training and uses a distillation loss $\mathcal{L}_{\text{Distill}}$ where the student network mimics the soft targets of a moving-average teacher network, stabilizing the learning process.

The total loss combines these strategies: $\mathcal{L}_{\text{ZeroVL}} = \mathcal{L}_{\text{Debiased\_InfoNCE}} + \mathcal{L}_{\text{Distill}}$.

3.12 SLIP (Meta, 2021, ECCV 2022)

SLIP (Mu et al., 2021) improves the dual-encoder CLIP framework by introducing a self-supervised objective on the visual modality. Following CLIP, SLIP uses an image encoder $f = f_\theta$ (e.g., ViT-B/16) and the text encoder $g = g_\phi$ (e.g., a CLIP's transformer). For each image $\mathbf{x}_i$ in a batch of size N, the model generates one crop and two heavily augmented views

$$\mathbf{x}_i = \text{crop}(\mathbf{x}_i), \qquad \mathbf{x}_i^1 = \text{aug}(\mathbf{x}_i), \qquad \mathbf{x}_i^2 = \text{aug}(\mathbf{x}_i),$$

which are passed through the image encoder f_θ followed by different projection heads: a linear head h_{CLIP} (`nn.Linear`) for the crop and an MLP projection head h_{SSL} for the augmented views, both followed by ℓ_2-normalization to obtain

$$\mathbf{v}_i = h_{\text{CLIP}}(f_\theta(\mathbf{x}_i)), \qquad \mathbf{v}_i^1 = h_{\text{SSL}}(f_\theta(\mathbf{x}_i^1)), \qquad \mathbf{v}_i^2 = h_{\text{SSL}}(f_\theta(\mathbf{x}_i^2)).$$

The text encoder g_ϕ processes the caption to produce $\mathbf{t}_i = g_\phi(\mathbf{c}_i)$, using a linear projection layer (`nn.Linear`) at the output, followed by ℓ_2-normalization so that $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$.

SLIP combines **two** supervision signals:

1. **CLIP loss**: The first component of the objective is the standard symmetric InfoNCE loss used in CLIP between the paired image and text embeddings:

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{N}\left(\sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau)} + \sum_{i=1}^{N} \log \frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{t}_i^\top \mathbf{v}_j/\tau)}\right),$$

where τ is a trainable temperature parameter, initialized at 0.07, identical to CLIP.

2. **Image self-supervised loss ($\mathcal{L}_{\text{SSL}}$):** The second component is the image self-supervised InfoNCE loss, which enforces representation consistency between the two augmented views of the same image:

$$\mathcal{L}_{\text{SSL}} = -\frac{1}{N}\left(\sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_i^{1\top}\mathbf{v}_i^{2}/\tau_s)}{\sum_{j=1}^{N}\exp(\mathbf{v}_i^{1\top}\mathbf{v}_j^{2}/\tau_s)} + \sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_i^{2\top}\mathbf{v}_i^{1}/\tau_s)}{\sum_{j=1}^{N}\exp(\mathbf{v}_i^{2\top}\mathbf{v}_j^{1}/\tau_s)}\right),$$

   where τ_s is a constant temperature parameter (typically set to 0.1).

The total SLIP loss is defined as the sum of these two objectives, scaled by a hyperparameter $\lambda = 1$:

$$\mathcal{L}_{\text{SLIP}} = \mathcal{L}_{\text{CLIP}} + \lambda\mathcal{L}_{\text{SSL}}.$$

3.13 BLIP (Salesforce, 2022, ICML 2022)

BLIP (Li et al., 2022a) introduces a multimodal mixture of encoder-decoder (MED) architecture. The image encoder $f = f_\theta$ is typically a ViT-B/16. The text subnetwork $g = g_\phi$ (based on BERT weights) is a unified transformer that operates in three modes: unimodal encoder, multimodal encoder, and causal decoder. Given an image $\mathbf{x}_i$, the image encoder f_θ produces a set of patch-level tokens $\mathbf{V}_i = f_\theta(\mathbf{x}_i)$, from which a global visual embedding $\mathbf{v}_i$ is derived.

BLIP combines **three** supervision signals:

1. **Image-text contrastive loss ($\mathcal{L}_{\text{ITC}}$):** The Image-Text Contrastive loss ($\mathcal{L}_{\text{ITC}}$) is the standard symmetric InfoNCE loss between the global image embedding $\mathbf{v}_i$ and the text embedding $\mathbf{t}_i$ produced by the unimodal encoder, matching the form of $\mathcal{L}_{\text{CLIP}}$ in Equation (1).

2. **Image-text matching loss ($\mathcal{L}_{\text{ITM}}$):** The Image-Text Matching loss ($\mathcal{L}_{\text{ITM}}$) uses the multimodal encoder mode, where the text tokens attend to the visual tokens $\mathbf{V}_i$ through cross-attention to produce a joint representation $\mathbf{h}_{i,j}$. A binary classification head then predicts a matching probability $p_{i,j}$ for the pair $(\mathbf{x}_i, \mathbf{c}_j)$:

$$\mathcal{L}_{\text{ITM}} = -\mathbb{E}_{(\mathbf{x},\mathbf{c})\sim D}\left[y \log p_{i,j} + (1-y)\log(1-p_{i,j})\right].$$

3. **Language modeling loss ($\mathcal{L}_{\text{LM}}$):** The Language Modeling loss ($\mathcal{L}_{\text{LM}}$) trains the subnetwork as a causal decoder to autoregressively generate the caption $\mathbf{c}_i$ conditioned on the visual features $\mathbf{V}_i$:

$$\mathcal{L}_{\text{LM}} = -\sum_{t=1}^{T} \log P(c_{i,t} \mid c_{i,<t}, \mathbf{V}_i).$$

The total BLIP loss is

$$\mathcal{L}_{\text{BLIP}} = \mathcal{L}_{\text{ITC}} + \mathcal{L}_{\text{ITM}} + \mathcal{L}_{\text{LM}}.$$

This multi-task approach allows BLIP to excel at both discriminative tasks like retrieval and generative tasks like image captioning.

3.14 UniCL (Microsoft, 2022, CVPR 2022)

UniCL (Yang et al., 2022) uses the same dual-encoder architecture as CLIP: an image encoder $f = f_\theta$ (e.g. ResNet or Swin Transformer) and a text encoder $g = g_\phi$ (e.g. CLIP's transformer) that independently map images $\mathbf{x}_i$ and captions $\mathbf{c}_i$ into a shared embedding space, with ℓ_2-normalized embeddings $\mathbf{v}_i = f_\theta(\mathbf{x}_i)$ and $\mathbf{t}_i = g_\phi(\mathbf{c}_i)$ such that $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$.

The key innovation is unifying three types of supervision: image-text pairs, image-label pairs, and text-label relationships. UniCL treats classification labels as a special case of text descriptions and extends CLIP's contrastive objective to handle multiple positives per image.

Given a batch of N triplets $(\mathbf{x}_i, \mathbf{c}_i, y_i)$, where $\mathbf{x}_i$ is an image, $\mathbf{c}_i$ is its associated text (caption or label), and y_i is the semantic class or label identifier (e.g., "dog", "car"), the encoders produce:

$$\mathbf{v}_i = f_\theta(\mathbf{x}_i), \qquad \mathbf{t}_i = g_\phi(\mathbf{c}_i).$$

UniCL optimizes:

$$\mathcal{L}_{\text{UniCL}} = -\left(\sum_{i=1}^{N} \frac{1}{|P(i)|} \sum_{k \in P(i)} \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_k/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau)} + \sum_{j=1}^{N} \frac{1}{|P(j)|} \sum_{k \in P(j)} \log \frac{\exp(\mathbf{t}_j^\top \mathbf{v}_k/\tau)}{\sum_{i=1}^{N} \exp(\mathbf{t}_j^\top \mathbf{v}_i/\tau)} \right),$$

where $P(i) = \{k \mid y_k = y_i, k \in \{1, \ldots, N\}\}$ is the set of all positive text indices for image i (all samples sharing the same label y_i). This reduces to CLIP when $|P(i)| = 1$ (one caption per image), but naturally handles multiple labels or descriptions per image.

3.15 CoCa (Google, 2022, JMLR 2022)

CoCa (Yu et al., 2022) unifies contrastive and autoregressive captioning into a single model. The caption $\mathbf{c}_i$ is a sequence of T tokens $\mathbf{c}_i = \{c_1, c_2, \ldots, c_T\}$. The image encoder $f = f_\theta$ is a standard ViT (e.g., ViT-L/14 or ViT-G/14), producing image features $\mathbf{v}_i = f(\mathbf{x}_i)$. The text side, in contrast to CLIP's single bidirectional encoder, is a causal Transformer decoder split at an intermediate layer into two functional parts: a **unimodal decoder** g_ϕ^{uni} (bottom layers) and a **multimodal decoder** g_ϕ^{mm} (top layers).

The unimodal decoder produces a text embedding $\mathbf{t}_i = g_\phi^{\text{uni}}(\mathbf{c}_i)$ by extracting the output at a learnable [CLS] token appended to the end of the caption. The embeddings $\mathbf{v}_i$ and $\mathbf{t}_i$ are mapped to a common d-dimensional space via linear projection layers (denoted by π) and ℓ_2-normalized so that $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$, reducing cosine similarity to $\mathbf{v}_i^\top \mathbf{t}_i$.

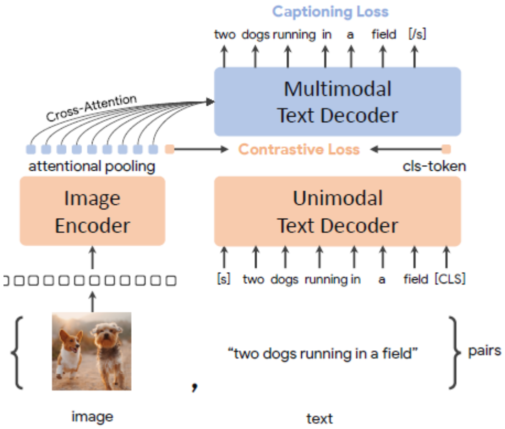


Figure 5 CoCa combines contrastive learning (dual-encoder) with generative captioning (decoder).

CoCa combines **two** supervision signals:

1. **Contrastive loss ($\mathcal{L}_{\text{Con}}$):** The unimodal decoder produces $\mathbf{t}_i$ and the image encoder produces $\mathbf{v}_i$, used to compute the contrastive loss, which is the same symmetric InfoNCE as in CLIP:

$$\mathcal{L}_{\text{Con}} = -\frac{1}{N} \left(\sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau)} + \sum_{i=1}^{N} \log \frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{t}_i^\top \mathbf{v}_j/\tau)} \right),$$

   matching the bi-directional contrastive loss $\mathcal{L}_{\text{CLIP}}$ in Equation (1).

2. **Captioning loss ($\mathcal{L}_{\text{Cap}}$):** The multimodal decoder g_ϕ^{mm} takes over from the unimodal decoder's hidden states and performs autoregressive next-token prediction over the caption, now conditioned on the image via cross-attention:

$$\mathcal{L}_{\text{Cap}} = -\sum_{t=1}^{T} \log P_\theta(c_t \mid c_{<t}, \mathbf{x}_i).$$

The two objectives are combined via a weighted sum:

$$\mathcal{L}_{\text{CoCa}} = \lambda_{\text{Con}} \cdot \mathcal{L}_{\text{Con}} + \lambda_{\text{Cap}} \cdot \mathcal{L}_{\text{Cap}}.$$

3.16 CyCLIP (Meta, 2022, NeurIPS 2022)

CyCLIP (Goel et al., 2022) addresses the geometric inconsistencies in the shared embedding space of standard CLIP. It identifies that matching pairs are often pulled together without considering the relationship between mismatched pairs or the internal structure of each modality. To enforce a more meaningful geometry, CyCLIP introduces two cyclic consistency regularizers.

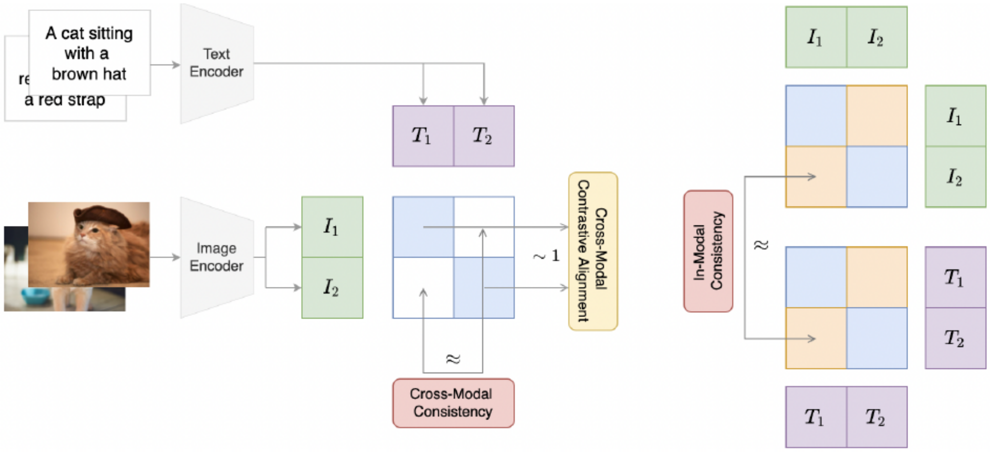


Figure 6 CyCLIP enhances CLIP by adding consistency constraints between image-text-image and text-image-text cycles to improve cross-modal alignment.

CyCLIP combines **three** supervision signals:

1. **CLIP loss**: The standard CLIP loss $\mathcal{L}_{\text{CLIP}}$ aligns image–text pairs $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$ in the shared space using the symmetric InfoNCE in Equation (1), with $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$.
2. **Cross-modal consistency loss ($\mathcal{L}_{\text{C-Cyclic}}$)**: The **cross-modal consistency** regularizer ensures that the similarity between an image j and text k is consistent with the similarity between image k and text j:

$$\mathcal{L}_{\text{C-Cyclic}} = \frac{1}{N^2} \sum_{j=1}^{N} \sum_{k=1}^{N} \left| \mathbf{v}_j^\top \mathbf{t}_k - \mathbf{v}_k^\top \mathbf{t}_j \right|^2.$$

3. **In-modal consistency loss ($\mathcal{L}_{\text{I-Cyclic}}$)**: The **in-modal consistency** regularizer encourages the relative distances between images to be identical to the relative distances between their corresponding captions:

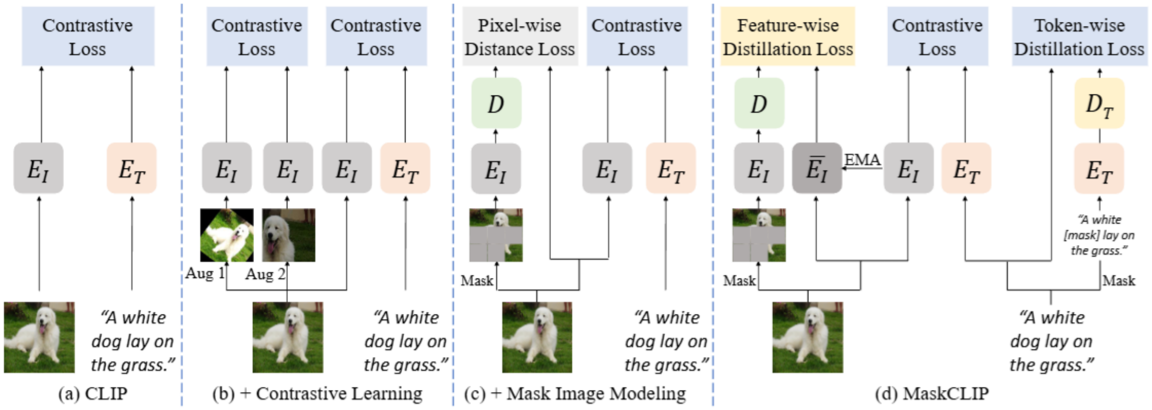
$$\mathcal{L}_{\text{I-Cyclic}} = \frac{1}{N^2} \sum_{j=1}^{N} \sum_{k=1}^{N} \left| \mathbf{v}_j^\top \mathbf{v}_k - \mathbf{t}_j^\top \mathbf{t}_k \right|^2.$$

The total loss is the sum of the standard CLIP loss and these two structural constraints:

$$\mathcal{L}_{\text{CyCLIP}} = \mathcal{L}_{\text{CLIP}} + \lambda_1 \mathcal{L}_{\text{I-Cyclic}} + \lambda_2 \mathcal{L}_{\text{C-Cyclic}}.$$

3.17 MaskCLIP (Microsoft, 2022, CVPR 2023)

MaskCLIP (Dong et al., 2023) improves the CLIP framework by introducing a masked self-distillation branch. MaskCLIP utilizes a student image encoder $f = f_\theta$ and a teacher image encoder $f' = f_{\bar{\theta}}$, where the teacher's


Contrastive Loss
E_I
E_T
"A white dog lay on the grass."
Contrastive Loss
Contrastive Loss
E_I
E_I
E_I
E_T
Aug 1
Aug 2
"A white dog lay on the grass."
Pixel-wise Distance Loss
Contrastive Loss
D
E_I
E_I
E_T
Mask
"A white dog lay on the grass."
Feature-wise Distillation Loss
Contrastive Loss
Token-wise Distillation Loss
D
D_T
E_I
$\bar{E}_I$
EMA
E_I
E_T
E_T
"A white [mask] lay on the grass."
Mask
Mask
"A white dog lay on the grass."

(a) CLIP (b) + Contrastive Learning (c) + Mask Image Modeling (d) MaskCLIP

Figure 7 **(a)** CLIP. **(b)** SLIP: CLIP + SimCLR. **(c)** CLIP + MAE. **(d)** MaskCLIP = CLIP + Mean Teacher + BERT.

weights $\bar{\theta}$ are an exponential moving average (EMA) of the student's weights $\bar{\theta}_t = \lambda\bar{\theta}_{t-1} + (1-\lambda)\theta_t$, with $\lambda \in [0, 1]$ as the momentum coefficient. The teacher produces target embeddings $\mathbf{v}'_i = f'(\mathbf{x}_i)$, which are used to supervise the student.

During training, an image $\mathbf{x}_i$ is processed in two ways. The teacher encoder f' ❄ receives the full image to produce a target representation $\mathbf{v}'_i = f'(\mathbf{x}_i)$. Simultaneously, the student encoder f receives a **masked image**

$$\mathbf{x}_i^{\text{mask}} = \text{mask}(\mathbf{x}_i, \rho),$$

where a fraction $\rho \in [0, 1]$ of patches (e.g., 75%) are removed or replaced by a [MASK] token, producing the student embedding $\mathbf{v}_i^{\text{mask}} = f(\mathbf{x}_i^{\text{mask}})$. The set of masked positions is denoted $\mathcal{M}_i \subseteq \{1, \ldots, N_v\}$.

MaskCLIP combines **three** supervision signals:

1. **CLIP loss**: The standard CLIP loss $\mathcal{L}_{\text{CLIP}}$ is computed between the student's global image embedding $\mathbf{v}_i$ (from the unmasked branch) and the text embedding $\mathbf{t}_i = g(\mathbf{c}_i)$, with $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$, using the symmetric InfoNCE in Equation (1).

2. **Masked Self-Distillation (MSD) loss ($\mathcal{L}_{\text{MSD}}$)**: The **MSD** loss is computed between the masked-image student embeddings $\mathbf{v}_i^{\text{mask}}$ and the full-image teacher embeddings $\mathbf{v}'_i$:

$$\mathcal{L}_{\text{MSD}} = \text{distill}(\mathbf{v}_i^{\text{mask}}, \mathbf{v}'_i) = \frac{1}{|\mathcal{M}_i|} \sum_{k \in \mathcal{M}_i} -h(\mathbf{v}'_i)^\top \log h(\mathbf{v}_i^{\text{mask}}),$$

   where $h(\cdot)$ is an online quantizer that transforms features into soft codeword distributions.

3. **Masked Language Modeling (MLM) loss ($\mathcal{L}_{\text{MLM}}$)**: For the text modality, MaskCLIP follows a **BERT-style masking** approach, where a percentage of input text tokens are masked and the model is trained to recover them. The **MLM** loss is:

$$\mathcal{L}_{\text{MLM}} = -\sum_{t \in \mathcal{M}_{\text{text}}} \log p(c_{i,t} \mid \mathbf{c}_{i,\setminus t}; g_\phi),$$

   where $\mathcal{M}_{\text{text}}$ denotes the set of masked token indices in caption $\mathbf{c}_i$.

The total objective combines the standard CLIP contrastive loss with these local distillation signals:

$$\mathcal{L}_{\text{MaskCLIP}} = \mathcal{L}_{\text{CLIP}} + \lambda_1 \mathcal{L}_{\text{MSD}} + \lambda_2 \mathcal{L}_{\text{MLM}}.$$

By distilling knowledge from the full-image teacher to the masked-image student, MaskCLIP bridges the gap between discriminative and generative pre-training.

3.18 NegCLIP (2022, ICLR 2023)

NegCLIP (Yuksekgonul et al., 2023) is a simple but effective extension of CLIP. The architecture remains a dual-encoder model with image encoder $f = f_\theta$ and text encoder $g = g_\phi$, producing embeddings

$$\mathbf{v}_i = f(\mathbf{x}_i), \qquad \mathbf{t}_i = g(\mathbf{c}_i),$$

which are mapped to a common d-dimensional space via linear projection layers (denoted by π) and ℓ_2-normalized so that $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$, reducing cosine similarity to $\mathbf{v}_i^\top \mathbf{t}_i$.

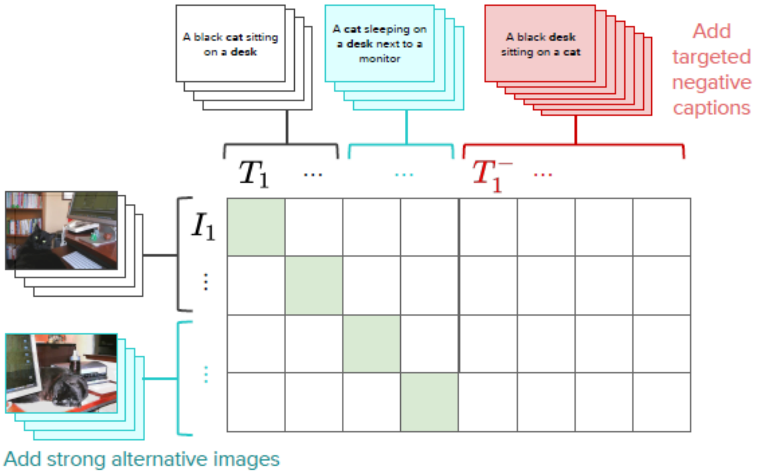

A black cat sitting on a desk
A cat sleeping on a desk next to a monitor
A black desk sitting on a cat
Add targeted negative captions
T_1 ...
...
T_1^- ...
I_1
⋮
⋮
Add strong alternative images

Figure 8 NegCLIP augments each batch with composition-aware hard negatives: nearest-neighbor images and captions with swapped linguistic elements.

For each batch of N image-caption pairs, NegCLIP augments the training data with two types of hard negatives:

1. **Negative captions $\mathbf{c}_i^-$:** Generated by swapping linguistic elements (nouns, adjectives, adverbs, verb phrases, noun phrases) in the original caption $\mathbf{c}_i$. For example:
   - Original: “The horse is eating the grass”
   - Negative: “The grass is eating the horse” (noun swap)
   - Or: “The horse is drinking the grass” (verb phrase swap)
2. **Positive images $\mathbf{x}_i^-$:** Sampled from the $K = 3$ nearest-neighbor images to $\mathbf{x}_i$ based on pre-computed CLIP similarity.

Given a batch with N positive pairs and their augmented negatives, the extended batch contains N images and $2N$ captions (original + negative). The similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times 2N}$ is computed, and the modified loss is:

$$\mathcal{L}_{\text{NegCLIP}} = -\frac{1}{N}\left(\sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{2N} \exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau)} + \sum_{i=1}^{N} \log \frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{t}_i^\top \mathbf{v}_j/\tau)}\right),$$

where the image-to-text term now includes negative captions in the denominator, but the text-to-image term only computes loss for the N correct captions (not the negatives, as they have no matching image).

3.19 FLIP (Meta, 2022, CVPR 2023)

FLIP (Li et al., 2023b) is a scaling-oriented improvement of CLIP that introduces random masking to significantly reduce the computational footprint during training. While CLIP processes the entire image and text, FLIP randomly masks out and removes a large portion of the input tokens. For an image $\mathbf{x}_i$ divided into N_p non-overlapping patches, a masking ratio $\rho \in [0, 1]$ (typically 50% or 75%) is applied, resulting in a masked image

$$\mathbf{x}_i^{\text{mask}} = \text{mask}(\mathbf{x}_i, \rho),$$

with only a subset of visible patches $\tilde{\mathbf{x}}_i \subset \mathbf{x}_i$ processed by the image encoder $f = f_\theta$. The number of patches processed is $N_{\text{vis}} = (1 - \rho)N_p$.

The image tower f_θ (typically a ViT) operates only on these N_{vis} patches. This sparse computation reduces the complexity of the image encoding to roughly $(1 - \rho)$ times that of the original CLIP. Optionally, a similar masking strategy is applied to the text encoder $g = g_\phi$ (e.g., a GPT-style transformer): for a caption $\mathbf{c}_i$ with T tokens, the model only encodes a visible subset $\tilde{\mathbf{c}}_i$ containing $(1 - \rho_t)T$ tokens. Unlike BERT, which replaces masked tokens with a [MASK] symbol, FLIP entirely removes them from the sequence, following the MAE approach.

The resulting embeddings $\mathbf{v}_i = f(\tilde{\mathbf{x}}_i)$ and $\mathbf{t}_i = g(\tilde{\mathbf{c}}_i)$ are projected into the shared space and ℓ_2-normalized so that $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$, used to compute the standard symmetric InfoNCE loss:

$$\mathcal{L}_{\text{FLIP}} = -\frac{1}{N}\left(\sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau)} + \sum_{i=1}^{N} \log \frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{t}_i^\top \mathbf{v}_j/\tau)}\right).$$

By reducing the per-sample computation, FLIP allows for much larger batch sizes and more training epochs within the same wall-clock time budget. This trade-off between "looking carefully" at each image and "seeing more" image-text pairs proves highly effective: FLIP trains up to 3× faster than CLIP while achieving superior performance on downstream tasks. After pre-training, the model can be directly applied to unmasked, intact images during inference.

3.20 OpenCLIP (LAION, 2022, CVPR 2023)

OpenCLIP (Ilharco et al., 2021; Cherti et al., 2023) is an open-source reproduction and scaling of CLIP's dual-encoder. Using the LAION-5B dataset, it scales architectures up to ViT-G/14 and RoBERTa-style text encoders, with image encoder $f = f_\theta$ and text encoder $g = g_\phi$ mapping images $\mathbf{x}_i$ and captions $\mathbf{c}_i$ to ℓ_2-normalized embeddings $\mathbf{v}_i, \mathbf{t}_i \in \mathbb{R}^d$ optimized via the symmetric InfoNCE loss $\mathcal{L}_{\text{CLIP}}$ in Equation (1).

3.21 BLIP-2 (Salesforce, 2023, ICML 2023)

BLIP-2 (Li et al., 2023a) introduces the Querying Transformer (Q-Former) to bridge the gap between frozen image encoders and frozen LLMs. The architecture leverages a frozen ViT-g/14 as the image tower $f = f_\theta$ and a frozen Flan-T5 as the text tower, with $\mathcal{L}_{\text{CLIP}}$-style alignment objectives used in the first stage and causal-generation losses later, but the paper is not framed as a multi-signal addition in the same explicit way as the earlier entries, so it is kept as a single method description rather than a bullet-style loss decomposition.

The Q-Former uses learnable query tokens to extract visual features from the frozen vision tower. Pre-training occurs in two stages:

1. **Representation Learning:** Aligning queries with text via contrastive and matching losses.
2. **Generative Learning:** Connecting the Q-Former output to a frozen LLM, enabling the LLM to process visual information.

This strategy allows BLIP-2 to achieve state-of-the-art multimodal performance while remaining significantly more computationally efficient than end-to-end trained models.

3.22 HiCLIP (Google, 2023, ICLR 2023)

HiCLIP (Geng et al., 2023) addresses the limitation of standard CLIP in capturing fine-grained, local semantic alignments. While CLIP focuses on global image–text matching, HiCLIP utilizes a hierarchical vision transformer as the image tower $f = f_\theta$ and a hierarchical transformer as the text tower $g = g_\phi$ to learn representations at multiple granularities.

The model decomposes an image $\mathbf{x}_i$ and its caption $\mathbf{c}_i$ into hierarchies, such as local patches to global images and words to sentences. The encoders produce embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$, which are mapped to a common d-dimensional embedding space via linear projection layers (denoted by π) and ℓ_2-normalized so that $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$, reducing cosine similarity to $\mathbf{v}_i^\top \mathbf{t}_i$.

In addition to the standard global contrastive loss $\mathcal{L}_{\text{CLIP}}$ in Equation (1), HiCLIP incorporates a **hierarchical constraint** that enforces consistency between local and global embeddings through hierarchy-aware attention.

Standard transformer attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_h}}\right) V$$

HiCLIP modifies this with an attention mask C that encodes the tendency to merge semantically and spatially similar patches/tokens:

$$\text{Hierarchy-Attention} = \left(C \odot \text{softmax}\left(\frac{QK^\top}{\sqrt{d_h}}\right)\right) V.$$

3.23 CLIPcov (2023, AISTATS 2024)

CLIPcov (Joshi et al., 2024) addresses CLIP's data inefficiency by identifying small, generalizable subsets from massive image-caption datasets. Given CLIP's dual-encoder framework with image encoder $f = f_\theta$ and text encoder $g = g_\phi$ producing embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$, CLIPcov theoretically proves that subsets preserving the **cross-covariance matrix**

$$C_D^V = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \mu_{\mathbf{x}})(\mathbf{c}_i - \mu_{\mathbf{c}})^\top$$

of image-caption pairs achieve similar zero-shot performance to training on the full data. Using a pre-trained CLIP model to compute cross-modal similarities $\text{sim}(i, j) = \mathbf{v}_i^\top \mathbf{t}_j + \mathbf{v}_j^\top \mathbf{t}_i$, CLIPcov selects examples that (1) preserve class centers while encouraging diversity within each latent class k, (2) have high self-similarity $\mathbf{v}_i^\top \mathbf{t}_i$ (CLIP score), capturing central examples in subgroups, and (3) maximize separation between different classes. On CC3M, a 10% subset found by CLIPcov achieves 1.75× the ImageNet accuracy of the next best baseline (CLIP Score filtering), demonstrating that **data quality trumps quantity** when subsets preserve the cross-covariance structure that determines CLIP's learned representations.

3.24 EVA-CLIP (BAI, 2023, arXiv 2023)

EVA-CLIP scales the dual-encoder framework to 1.1B parameters by leveraging a high-capacity EVA-ViT image encoder and a giant Transformer text encoder. The core philosophy of EVA-CLIP is that initializing a massive dual-encoder with weights from a self-supervised visual model significantly stabilizes training and improves convergence.

Beyond the standard CLIP training protocol, EVA-CLIP introduces the following key improvements:

1. **EVA initialization:** The image encoder $f = f_\theta$ is initialized with weights from the EVA model.
2. **LAMB optimizer:** To handle the potential instabilities associated with extremely large batch sizes and high-dimensional parameter spaces, the model utilizes the LAMB optimizer instead of standard AdamW.
3. **FLIP masking:** To accelerate training and reduce computational overhead, EVA-CLIP adopts the random patch masking strategy from FLIP, processing only a subset of the image tokens during the contrastive stage. A masked image is created as

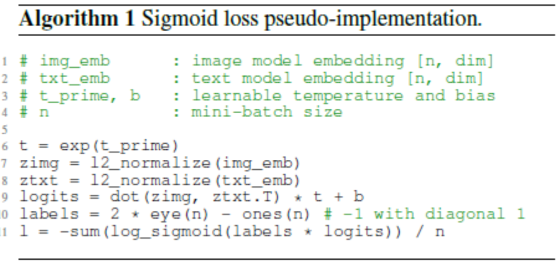
$$\mathbf{x}_i^{\text{mask}} = \text{mask}(\mathbf{x}_i, \rho),$$

   with a fraction $\rho \in [0, 1]$ of patches removed, producing the embedding $\mathbf{v}_i^{\text{mask}} = f(\mathbf{x}_i^{\text{mask}})$.

4. **Flash Attention:** The model incorporates Flash Attention kernels in both towers to optimize memory throughput and increase training speed.
5. **Gradient checkpointing:** To fit the massive 1.1B parameters and large batches into GPU memory, gradient checkpointing is used throughout the networks.
6. **More data:** EVA-CLIP is trained on massive datasets such as LAION-2B and COYO-700M, providing a significantly larger and more diverse signal than the original CLIP-400M.
7. **OpenAI initialization:** The text encoder $g = g_\phi$ is initialized with pre-trained weights from OpenAI's original CLIP text tower to facilitate faster cross-modal alignment.

3.25 SigLIP (Google, 2023, ICCV 2023)

SigLIP (Zhai et al., 2023) proposes a fundamental change to the contrastive learning objective. While CLIP relies on a symmetric InfoNCE loss that requires a global softmax over the entire batch, SigLIP replaces this with a simple pairwise sigmoid loss.



Algorithm 1 Sigmoid loss pseudo-implementation.

```
# img_emb       : image model embedding [n, dim]
# txt_emb       : text model embedding [n, dim]
# t_prime, b    : learnable temperature and bias
# n             : mini-batch size

t = exp(t_prime)
zimg = l2_normalize(img_emb)
ztxt = l2_normalize(txt_emb)
logits = dot(zimg, ztxt.T) * t + b
labels = 2 * eye(n) - ones(n) # -1 with diagonal 1
l = -sum(log_sigmoid(labels * logits)) / n
```

In the standard CLIP setting, the encoders $f = f_\theta$ and $g = g_\phi$ produce ℓ_2-normalized embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$, with $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$, and the loss for a batch of N pairs is computed using

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{N}\left(\sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau)} + \sum_{i=1}^{N} \log \frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{t}_i^\top \mathbf{v}_j/\tau)}\right),$$

matching the bi-directional form in Equation (1).

SigLIP simplifies this by treating the problem as a set of independent binary classification tasks. For every pair of image $\mathbf{v}_i$ and text $\mathbf{t}_j$ in a batch, the model predicts whether they match ($z_{ij} = 1$ if $i = j$) or not ($z_{ij} = -1$ if $i \neq j$). The sigmoid loss is defined as:

$$\mathcal{L}_{\text{SigLIP}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} \log\left(\sigma(z_{ij}(\beta \cdot \mathbf{v}_i^\top \mathbf{t}_j + b))\right) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} \log \frac{1}{1 + e^{z_{ij}(-\beta \cdot \mathbf{v}_i^\top \mathbf{t}_j + b)}},$$

where $\sigma(x) = 1/(1 + e^{-x})$, β is a learnable temperature (log-scale), and b is a learnable bias. By decoupling the samples, SigLIP allows for significantly larger batch sizes and more efficient distributed training. This simple change leads to better performance than CLIP, especially at smaller batch sizes, and enables the training of massive models like ViT-p/14.

3.26 nCLIP & xCLIP (Microsoft, 2023, CVPR 2023)

Zhou et al. (2023) introduces **nCLIP** and **xCLIP**, which depart from the traditional InfoNCE framework by treating language–image alignment as a distribution-matching problem rather than a discriminative task. Given CLIP's dual-encoder framework with image encoder $f = f_\theta$ and text encoder $g = g_\phi$ producing embeddings

$$\mathbf{v}_i = f(\mathbf{x}_i), \qquad \mathbf{t}_i = g(\mathbf{c}_i),$$

nCLIP transforms the projected features into probability distributions $\mathbf{p} = \text{softmax}(\mathbf{V})$ and $\mathbf{q} = \text{softmax}(\mathbf{T})$, where $\mathbf{V}$ and $\mathbf{T}$ are the image and text batch features respectively (i.e., $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_N]$ and $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]$). The model then minimizes the cross-entropy between these distributions:

$$\mathcal{L}_{\text{CE}} = -\mathbf{p}^\top \log(\mathbf{q}).$$

nCLIP combines **three** supervision signals:

1. **Distribution matching loss ($\mathcal{L}_{\mathbf{CE}}$):** The core nCLIP objective is $\mathcal{L}_{\text{CE}}$, pushing the empirical image and text distributions to match.
2. **Entropy minimization ($\mathcal{L}_{\mathbf{EH}}$):** To avoid collapsing solutions where the model maps all inputs to a single point, nCLIP adds an entropy minimization regularizer:

$$\mathcal{L}_{\text{EH}} = -\mathbf{p}^\top \log(\mathbf{p}).$$

3. **Mean entropy maximization ($\mathcal{L}_{\mathbf{HE}}$):** To preserve diversity, nCLIP also includes a mean-entropy maximization term:

$$\mathcal{L}_{\text{HE}} = -\bar{\mathbf{p}}^\top \log(\bar{\mathbf{p}}),$$

where $\bar{\mathbf{p}}$ is the mean probability over the batch.

The complete nCLIP objective is

$$\mathcal{L}_{\text{nCLIP}} = \mathcal{L}_{\text{CE}} + \lambda_1 \cdot \mathcal{L}_{\text{EH}} - \lambda_2 \cdot \mathcal{L}_{\text{HE}}.$$

Given that the two objectives each have their own limitations, the authors further seek the complementarity between $\mathcal{L}_{\text{CLIP}}$ and $\mathcal{L}_{\text{nCLIP}}$, and pre-train the models with both objectives, written as:

$$\mathcal{L}_{\text{xCLIP}} = \lambda_{\text{CLIP}} \cdot \mathcal{L}_{\text{CLIP}} + \lambda_{\text{nCLIP}} \cdot \mathcal{L}_{\text{nCLIP}}.$$

3.27 3T (Google, 2023, NeurIPS 2023)

The 3T (Three Towers) (Kossen et al., 2023) approach introduces a third tower to the standard dual-encoder setup to leverage specialized visual features from a pre-trained teacher. In this framework, the model consists of a learnable image encoder $f = f_\theta$, a learnable text encoder $g = g_\phi$, and a third auxiliary image tower f_ξ. Typically, f_ξ is a high-capacity model (e.g., pre-trained on ImageNet-21k) that remains frozen ❄ during alignment.

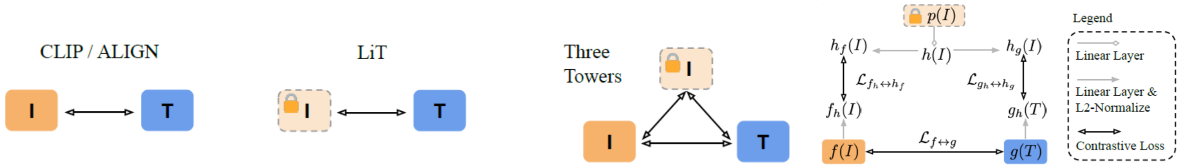


Figure 9 3T introduces a third tower which contains fixed pretrained image embeddings; extra loss terms align the main towers to the third tower.

To align these representations, 3T employs learned linear projection heads (adaptors) denoted as h_i. These heads allow for flexibility, ensuring the main towers do not strictly collapse into the teacher's fixed feature space. For a given image $\mathbf{x}_i$ and text $\mathbf{c}_i$, the encoders produce:

$$\mathbf{v}_i = f_\theta(\mathbf{x}_i), \qquad \mathbf{t}_i = g_\phi(\mathbf{c}_i), \qquad \mathbf{p}_i = \texttt{nn.Linear}\left(f_\xi(\mathbf{x}_i)\right).$$

All embeddings are mapped to a common d-dimensional space via projection layers and ℓ_2-normalized so that $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = \|\mathbf{p}_i\|_2 = 1$, reducing cosine similarities to inner products.

3T combines **three** supervision signals:

1. **CLIP loss between main towers ($\mathcal{L}_{\mathbf{CLIP}}(\mathbf{v}, \mathbf{t})$):** The standard CLIP contrastive loss aligns the main image and text embeddings $\mathbf{v}_i$ and $\mathbf{t}_i$ via the symmetric InfoNCE in Equation (1).
2. **Image-to-teacher alignment loss ($\mathcal{L}_{\mathbf{CLIP}}(h_1(\mathbf{v}), h_2(\mathbf{p}))$):** A second CLIP-style term aligns the main image tower $\mathbf{v}_i$ to the teacher image embeddings $\mathbf{p}_i$ via learned projection heads h_1, h_2, both implemented as linear layers followed by ℓ_2-normalization.

3. **Text-to-teacher alignment loss ($\mathcal{L}_{\mathbf{CLIP}}(h_3(\mathbf{t}), h_4(\mathbf{p}))$):** A third term aligns the text embeddings $\mathbf{t}_i$ to the teacher image embeddings $\mathbf{p}_i$ using projection heads h_3, h_4, again following the CLIP form.

The training objective is

$$\mathcal{L}_{3\text{T}} = \mathcal{L}_{\text{CLIP}}(\mathbf{v}, \mathbf{t}) + \lambda_1 \mathcal{L}_{\text{CLIP}}(h_1(\mathbf{v}), h_2(\mathbf{p})) + \lambda_2 \mathcal{L}_{\text{CLIP}}(h_3(\mathbf{t}), h_4(\mathbf{p})).$$

By incorporating f_ξ as a stable target, 3T facilitates the learning of more robust and fine-grained visual representations. For downstream tasks, the third tower and its associated projection heads are discarded, and the model is evaluated as a standard dual-encoder using only the main towers f_θ and g_ϕ. This ensures that the final model retains the inference efficiency of CLIP while benefiting from the teacher's knowledge during pre-training.

3.28 LaCLIP (Google, 2023, NeurIPS 2024)

LaCLIP (Fan et al., 2023) enhances CLIP training by introducing text augmentation through language rewrites generated by large language models. The architecture remains identical to CLIP with dual encoders $f = f_\theta$ and $g = g_\phi$, producing ℓ_2-normalized embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$, and optimizing the standard symmetric InfoNCE loss $\mathcal{L}_{\text{CLIP}}$ in Equation (1), but incorporates diverse caption variations during training.

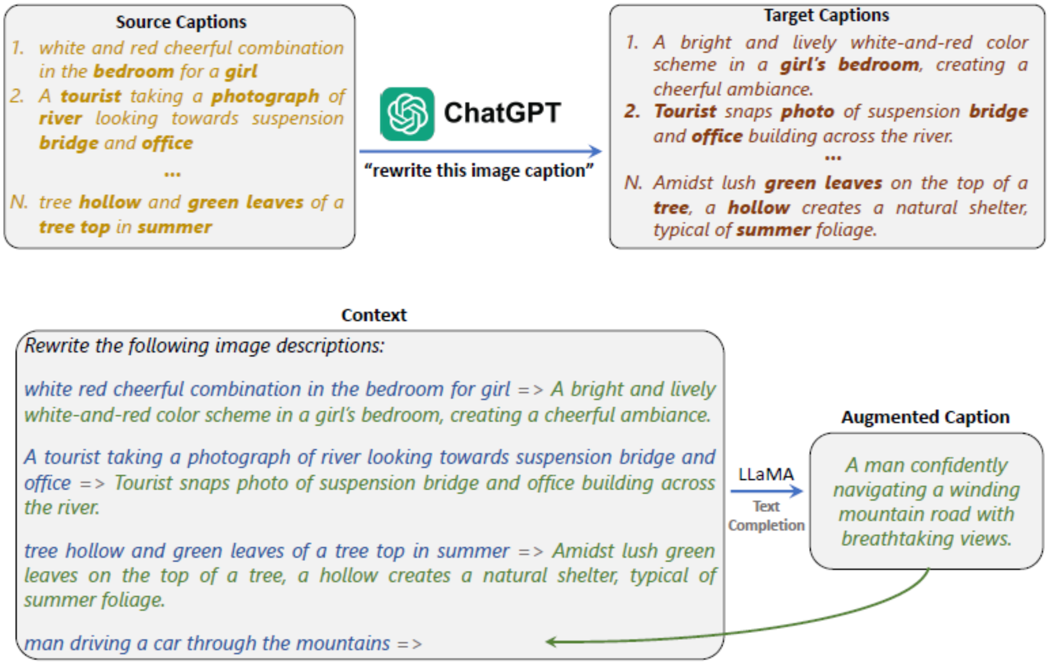

Source Captions
1. white and red cheerful combination in the bedroom for a girl
2. A tourist taking a photograph of river looking towards suspension bridge and office
...
N. tree hollow and green leaves of a tree top in summer
ChatGPT
"rewrite this image caption"
Target Captions
1. A bright and lively white-and-red color scheme in a girl's bedroom, creating a cheerful ambiance.
2. Tourist snaps photo of suspension bridge and office building across the river.
...
N. Amidst lush green leaves on the top of a tree, a hollow creates a natural shelter, typical of summer foliage.
Context
Rewrite the following image descriptions:
white red cheerful combination in the bedroom for girl => A bright and lively white-and-red color scheme in a girl's bedroom, creating a cheerful ambiance.
A tourist taking a photograph of river looking towards suspension bridge and office => Tourist snaps photo of suspension bridge and office building across the river.
tree hollow and green leaves of a tree top in summer => Amidst lush green leaves on the top of a tree, a hollow creates a natural shelter, typical of summer foliage.
man driving a car through the mountains =>
LLaMA
Text Completion
Augmented Caption
A man confidently navigating a winding mountain road with breathtaking views.

Figure 10 **Top**: Illustration of using ChatGPT to generate meta-input-output pairs: we first sample source captions randomly from a few datasets. We then use prompts such as "*Rewrite this image caption*" to guide the ChatGPT model to generate rewritten target captions. **Bottom**: Illustration of our proposed in-context learning based strategy for language rewriting.

1. **Meta-input-output pair creation:** Generate 16 example caption rewrite pairs using four strategies (4 examples each):
   - ChatGPT: Prompt ChatGPT to rewrite captions vividly.
   - Bard: Use Bard to generate alternative descriptions.
   - Human: Human annotators rewrite captions based on images.

- MSCOCO: Sample two different captions from MSCOCO's 5 captions per image.

2. **In-context learning with LLaMA:** For each caption $\mathbf{c}_i$ in the dataset, use LLaMA-7B with in-context learning to generate 4 rewritten versions $\{\mathbf{c}_i^1, \mathbf{c}_i^2, \mathbf{c}_i^3, \mathbf{c}_i^4\}$. Each rewrite is prompted using 3 randomly sampled meta-pairs from one of the four strategies above.

During CLIP training, one caption is randomly sampled from the original or rewritten versions:

$$\text{aug}_T(\mathbf{c}_i) \sim \text{Uniform}([\mathbf{c}_i^0, \mathbf{c}_i^1, \mathbf{c}_i^2, \mathbf{c}_i^3, \mathbf{c}_i^4]),$$

and the standard $\mathcal{L}_{\text{CLIP}}$ is computed between $\mathbf{v}_i$ and the sampled text embedding $\mathbf{t}_i = g(\text{aug}_T(\mathbf{c}_i))$.

3.29 Cap & CapPa (Google, 2023, NeurIPS 2023)

Cap and CapPa (Tschannen et al., 2023) replace CLIP's dual-encoder setup by a single vision encoder–text decoder architecture trained with an image captioning objective instead of a contrastive loss. The image encoder $f = f_\theta$ is a Vision Transformer (e.g., ViT-B/16 or ViT-L/14) that maps an input image $\mathbf{x}_i$ to a sequence of visual tokens $\mathbf{V}_i \in \mathbb{R}^{N_v \times d}$. A Transformer decoder $g = g_\xi$ (same width as the encoder, half the depth) receives visual tokens via cross-attention and predicts the caption tokens $\mathbf{T}_i$ autoregressively. The model is trained with standard next-token cross-entropy under a causal mask:

$$\mathcal{L}_{\text{Cap}} = -\sum_{t=1}^{N_t} \log P(t_{i,t} \mid \mathbf{t}_{i,<t}, \mathbf{V}_i).$$

CapPa modifies this training by mixing autoregressive and *parallel* prediction. For each training example, with probability $p = 0.75$, the decoder is trained in parallel mode: all input text tokens are replaced by a special mask token and the causal self-attention mask is removed. The parallel loss is

$$\mathcal{L}_{\text{Pa}} = -\sum_{t=1}^{N_t} \log P(t_{i,t} \mid \mathbf{t}_{i,\text{mask}}, \mathbf{V}_i),$$

where $\mathbf{t}_{i,\text{mask}}$ denotes the fully masked text sequence. The overall CapPa objective is the mixture

$$\mathcal{L}_{\text{CapPa}} = (1-p)\,\mathcal{L}_{\text{Cap}} + p\,\mathcal{L}_{\text{Pa}},$$

with the same encoder–decoder architecture and optimizer settings as CLIP-style models, but trained purely via captioning on web-scale image–alt-text pairs.

3.30 ACLIP (2023, ICCV 2023)

ACLIP (Yang et al., 2023b) improves CLIP-style pre-training by using **text-guided attentional masking** on image patches. Given a batch of image–text pairs $\{(\mathbf{x}_i, \mathbf{c}_i)\}_{i=1}^N$, an image encoder f_θ and a text encoder g_ϕ first produce global features

$$\mathbf{v}_i = f_\theta(\mathbf{x}_i), \qquad \mathbf{t}_i = g_\phi(\mathbf{c}_i),$$

with $\mathbf{v}_i, \mathbf{t}_i \in \mathbb{R}^d$ projected and ℓ_2-normalized. ACLIP then uses cross-attention to estimate the importance of each visual patch for the given caption and selectively masks less informative patches before computing the CLIP loss.

Concretely, ACLIP is built on a vision transformer and uses patch-level features $\mathbf{V}_i = \{\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,N_v}\} \in \mathbb{R}^{N_v \times d}$ (excluding the [CLS] token). For a given text embedding $\mathbf{t}_i$, an attention module computes scalar importance scores $a_{i,m}$ for each patch $\mathbf{v}_{i,m}$ through a text-conditioned attention head:

$$a_{i,m} = \text{Attn}(\mathbf{t}_i, \mathbf{v}_{i,m}) \in \mathbb{R}, \qquad m = 1, \ldots, N_v.$$

Scores are normalized (e.g., by softmax over patches) and a fraction $\rho \in [0,1]$ of the least important patches is masked:

$$\mathcal{M}_i = \text{TopK}_{\text{low}}(\{a_{i,m}\}_{m=1}^{N_v},\ \rho N_v),$$

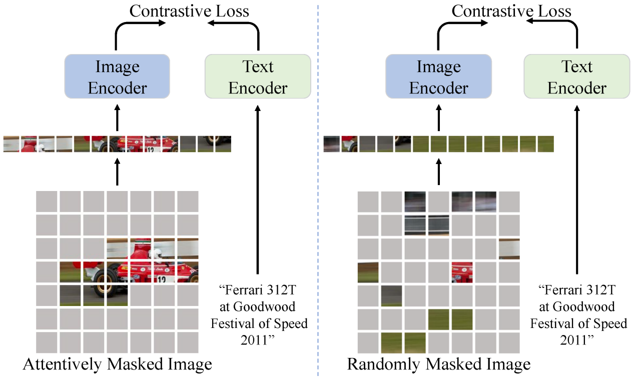


Figure 11 Attentive mask vs. random mask. **Left** part is the attentive mask applied CLIP training process, and **right** part is random mask applied

$$\mathbf{x}_i^{\text{mask}} = \text{mask}(\mathbf{x}_i, \mathcal{M}_i), \qquad \mathbf{v}_i^{\text{mask}} = f_\theta(\mathbf{x}_i^{\text{mask}}).$$

Here $\mathcal{M}_i$ denotes the set of masked patch indices. The *masked* image embedding $\mathbf{v}_i^{\text{mask}}$ replaces $\mathbf{v}_i$ in the contrastive loss, while the text side remains unchanged.

ACLIP combines **two supervision signals**:

1. **Standard CLIP loss** $\mathcal{L}_{\text{CLIP}}$ (on full images): the symmetric InfoNCE loss between $\mathbf{v}_i$ and $\mathbf{t}_i$ for unmasked images, as in Eq. (1):

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{2N}\left(\sum_{i=1}^{N}\log\frac{\exp(\mathbf{v}_i^\top\mathbf{t}_i/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{v}_i^\top\mathbf{t}_j/\tau)} + \sum_{i=1}^{N}\log\frac{\exp(\mathbf{t}_i^\top\mathbf{v}_i/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{t}_i^\top\mathbf{v}_j/\tau)}\right).$$

2. **Attentional-mask CLIP loss** $\mathcal{L}_{\text{ACLIP}}$: a symmetric InfoNCE loss where image embeddings are computed from attentively masked images $\mathbf{x}_i^{\text{mask}}$:

$$\mathcal{L}_{\text{ACLIP}} = -\frac{1}{2N}\left(\sum_{i=1}^{N}\log\frac{\exp((\mathbf{v}_i^{\text{mask}})^\top\mathbf{t}_i/\tau)}{\sum_{j=1}^{N}\exp((\mathbf{v}_i^{\text{mask}})^\top\mathbf{t}_j/\tau)} + \sum_{i=1}^{N}\log\frac{\exp(\mathbf{t}_i^\top\mathbf{v}_i^{\text{mask}}/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{t}_i^\top\mathbf{v}_j^{\text{mask}}/\tau)}\right).$$

The total ACLIP training objective interpolates between full-image and attentively masked supervision:

$$\mathcal{L}_{\text{total}} = (1-\lambda)\,\mathcal{L}_{\text{CLIP}} + \lambda\,\mathcal{L}_{\text{ACLIP}},$$

where $\lambda \in [0,1]$ controls the strength of the attentional masking branch. This encourages the encoder to rely on text-relevant regions and improves image–text alignment and data efficiency compared to purely random masking.

3.31 ALIP (Huawei, 2023, ICCV 2023)

ALIP (Yang et al., 2023a) augments CLIP-style contrastive pre-training with synthetic captions and adaptive weighting. For each image–text pair in the dataset, we assume an image $\mathbf{x}_i$, its raw web text caption $\mathbf{c}_i$, and a synthetic caption $\mathbf{c}_i^{\text{syn}}$ generated offline by a captioner (e.g., OFA) using a prompt like "*What does the image describe?*". These triplets $(\mathbf{x}_i, \mathbf{c}_i, \mathbf{c}_i^{\text{syn}})$ are passed through a shared dual-encoder backbone: an image encoder $f = f_\theta$ and a text encoder $g = g_\phi$, yielding ℓ_2-normalized embeddings

$$\mathbf{v}_i = f(\mathbf{x}_i), \qquad \mathbf{t}_i = g(\mathbf{c}_i), \qquad \mathbf{t}_i^{\text{syn}} = g(\mathbf{c}_i^{\text{syn}}),$$

with $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = \|\mathbf{t}_i^{\text{syn}}\|_2 = 1$.

Using these embeddings, ALIP computes two kinds of adaptive weights:

1. A *sample weight* W_i^s from a **Language Consistency Gate**, based on the agreement between raw caption and synthetic caption.
2. An *image–description weight* W_i^t and W_i^c from a **Description Consistency Gate**, measuring how well the image aligns with the raw caption and synthetic caption, respectively.

Given a batch $\{(\mathbf{v}_i, \mathbf{t}_i, \mathbf{t}_i^{\text{syn}})\}_{i=1}^N$, ALIP replaces the standard CLIP InfoNCE with two weighted InfoNCE terms—one for image–raw-text and one for image–synthetic-caption—combined into

$$\begin{aligned}
\mathcal{L}_{xt} &= -\sum_{i=1}^{N} W_i^s W_i^t \left[\log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^N \exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau)} + \log \frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i/\tau)}{\sum_{j=1}^N \exp(\mathbf{t}_i^\top \mathbf{v}_j/\tau)} \right], \\
\mathcal{L}_{xc} &= -\sum_{i=1}^{N} W_i^s W_i^c \left[\log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i^{\text{syn}}/\tau)}{\sum_{j=1}^N \exp(\mathbf{v}_i^\top \mathbf{t}_j^{\text{syn}}/\tau)} + \log \frac{\exp(\mathbf{t}_i^{\text{syn}\top} \mathbf{v}_i/\tau)}{\sum_{j=1}^N \exp(\mathbf{t}_j^{\text{syn}\top} \mathbf{v}_i/\tau)} \right], \\
\mathcal{L}_{\text{ALIP}} &= \mathcal{L}_{xt} + \mathcal{L}_{xc},
\end{aligned}$$

where τ is a learnable temperature as in CLIP. Intuitively, the Language Consistency Gate produces W_i^s to down-weight globally unreliable samples, while the Description Consistency Gate produces W_i^t and W_i^c to emphasize well-aligned image–caption and image–synthetic-caption pairs. The resulting adaptive contrastive loss reduces the impact of noisy web text while exploiting both human and synthetic descriptions for image–language representation learning.

3.32 SILC (Google, 2023, ECCV 2024)

SILC (Naeem et al., 2024) improves vision–language pretraining by pairing CLIP-style image–text contrastive learning with local-to-global self-distillation on images (as in DINO). Following CLIP, it uses an image encoder $f = f_\theta$ (student, e.g., ViT-B/16) and a text encoder $g = g_\phi$ (e.g., a CLIP Transformer) that both output ℓ_2-normalized embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$ in a common d-dimensional space via projection layers, so that $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$. In addition, SILC maintains an EMA teacher image encoder $f' = f_{\bar{\theta}}$ for self-distillation. The teacher parameters $f_{\bar{\theta}}$ are updated from the student by EMA:

$$\bar{\theta} \leftarrow \lambda \bar{\theta} + (1 - \lambda)\theta,$$

with $\lambda \in [0, 1]$. The teacher produces target embeddings $\mathbf{v}'_i = f'(\mathbf{x}_i)$, which are used to supervise the student encoder.

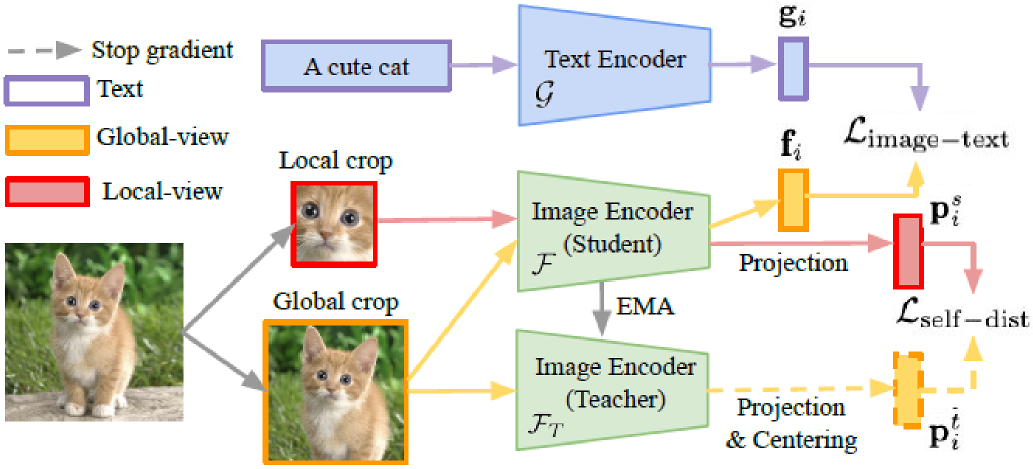


Figure 12 SILC combines standard CLIP-style contrastive training with DINO self-distillation.

For each image–caption pair $(\mathbf{x}_i, \mathbf{c}_i)$, SILC samples a **local** crop and a **global** crop of the image:

$$\mathbf{x}_i^{\text{loc}} = \text{aug}_{\text{loc}}(\mathbf{x}_i), \qquad \mathbf{x}_i^{\text{glob}} = \text{aug}_{\text{glob}}(\mathbf{x}_i),$$

and encodes them with the student and teacher:

$$\mathbf{v}_i^{\text{loc}} = f(\mathbf{x}_i^{\text{loc}}), \qquad \mathbf{v}_i^{\text{glob}} = f(\mathbf{x}_i^{\text{glob}}), \qquad \bar{\mathbf{v}}_i^{\text{glob}} = f'(\mathbf{x}_i^{\text{glob}}),$$

where all embeddings are ℓ_2-normalized. The caption is processed by the text encoder to obtain $\mathbf{t}_i = g(\mathbf{c}_i)$.

SILC combines **two** supervision signals:

1. **Image–text contrastive loss ($\mathcal{L}_{\mathbf{CLIP}}$)**: SILC retains the standard CLIP-style InfoNCE loss between image and text. Given a batch $\{(\mathbf{v}_i^{\text{glob}}, \mathbf{t}_i)\}_{i=1}^N$, the symmetric contrastive loss is

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{N}\left(\sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_i^{\text{glob}\top}\mathbf{t}_i/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{v}_i^{\text{glob}\top}\mathbf{t}_j/\tau)} + \sum_{i=1}^{N} \log \frac{\exp(\mathbf{t}_i^{\top}\mathbf{v}_i^{\text{glob}}/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{t}_i^{\top}\mathbf{v}_j^{\text{glob}}/\tau)}\right),$$

   where τ is a learnable temperature parameter matching the form in Equation (1).

2. **Local-to-global self-distillation loss ($\mathcal{L}_{\mathbf{dist}}$)**: To learn locally aware image features, SILC adds a DINO-style self-distillation loss between the *local* student embedding and the *global* teacher embedding for the same image. A projection head maps image features to a higher-dimensional space, after which temperature-scaled softmax and centering are applied to define probability distributions

$$P_i^{\text{teacher}} = \operatorname{softmax}\left(\frac{\bar{\mathbf{z}}_i^{\text{glob}} - \mathbf{c}}{\tau_{\text{t}}}\right), \qquad P_i^{\text{student}} = \operatorname{softmax}\left(\frac{\mathbf{z}_i^{\text{loc}}}{\tau_{\text{s}}}\right),$$

   where $\mathbf{z}_i^{\text{loc}}$ and $\bar{\mathbf{z}}_i^{\text{glob}}$ are the projected features of $\mathbf{v}_i^{\text{loc}}$ and $\bar{\mathbf{v}}_i^{\text{glob}}$, $\tau_{\text{t}}, \tau_{\text{s}}$ are temperatures, and $\mathbf{c}$ is a running center vector. The self-distillation loss is a cross-entropy between teacher and student distributions:

$$\mathcal{L}_{\text{dist}} = -\frac{1}{N}\sum_{i=1}^{N}\left(P_i^{\text{teacher}}\right)^{\top}\log P_i^{\text{student}}.$$

The final SILC training objective is a weighted sum

$$\mathcal{L}_{\text{SILC}} = \mathcal{L}_{\text{CLIP}} + \lambda_{\text{dist}}\,\mathcal{L}_{\text{dist}},$$

which jointly enforces image–text alignment and local-to-global consistency. This yields language-grounded visual features that are both strong for global image-level tasks (classification, retrieval) and rich in local semantics for dense prediction tasks (segmentation, detection).

3.33 Florence-2 (Microsoft, 2023, CVPR 2024)

Florence-2 (Xiao et al., 2024) is a unified vision-language model that handles a variety of tasks (captioning, detection, grounding, segmentation) via a prompt-based representation. The model uses a DaViT vision encoder $f = f_\theta$ and a Transformer seq-to-seq architecture with a text encoder–decoder $g = g_\phi$ that jointly processes images and text prompts. Given an image $\mathbf{x}_i$ and a task prompt (encoded into $\mathbf{c}_i$) by the model, the encoder–decoder treats every task as a translation problem: it generates text or coordinate tokens as output. The model is trained on image–prompt–output triplets, where the outputs can be class-agnostic boxes, masks, or descriptive captions, enabling Florence-2 to cover a wide range of tasks within a single unified framework.

3.34 EVA-CLIP-18B (BAI, 2024, arXiv 2024)

EVA-CLIP-18B (Sun et al., 2024) represents the extreme scaling of the CLIP framework, creating the largest open-source vision-language model to date. It utilizes a massive EVA-18B vision encoder (18 billion parameters) pre-trained via Masked Image Modeling (MIM) and a Giant text encoder $g = g_\phi$. The image encoder $f = f_\theta$ produces ℓ_2-normalized embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ in a shared space with text embeddings $\mathbf{t}_i = g(\mathbf{c}_i)$, and the model is trained with a sigmoid-style contrastive loss $\mathcal{L}_{\text{SigLIP}}$ (as in the SigLIP notation) to handle the massive batch sizes efficiently.

To train such a colossal model, the authors employ a progressive training strategy:

- **MIM initialization:** The vision tower is initialized from EVA, which learns robust visual structures without labels via masked image modeling.
- **Quadruple-stage training:** The image resolution is progressively increased (from 224^2 to 448^2) and the batch size is scaled up to stabilize the massive gradient updates.

EVA-CLIP-18B demonstrates that scaling the vision tower is more critical than scaling the text tower for discriminative tasks, achieving exceptional zero-shot top-1 accuracy on numerous benchmarks despite using only publicly available datasets such as LAION-2B and COYO-700M.

3.35 LongCLIP (2024, ECCV 2024)

LongCLIP (Zhang et al., 2024) extends CLIP's text encoder to handle long, detailed captions by redesigning positional embeddings and a careful fine-tuning scheme on long-text image–caption data. Instead of naively interpolating all absolute position embeddings to a longer sequence (which would distort well-trained low positions and hurt zero-shot performance), it introduces "*knowledge-preserving stretching*": the first ∼20 positions (where CLIP actually sees most tokens during pretraining) are kept fixed, while only higher, poorly trained positions are interpolated with a larger stride to support longer inputs. On top of this, LongCLIP is fine-tuned with long captions using a primary-component matching strategy: fine-grained image features are aligned with long, detailed descriptions, while a coarse-grained "primary component" of the same image feature (obtained by decomposing, filtering, and reconstructing attributes) is aligned with short summary captions, preserving CLIP's short-text behavior while unlocking robust long-text capability. The resulting model retains the standard CLIP objective $\mathcal{L}_{\text{CLIP}}$ in Equation (1) over the reparameterized text encoder, enhanced with this long-text-aware positional and fine-tuning scheme.

3.36 DreamLIP (2024, ECCV 2024)

DreamLIP (Zheng et al., 2024) extends CLIP by leveraging long captions generated from MLLMs. The architecture remains a dual-encoder model with image encoder $f = f_\theta$ (e.g., ViT-B/16) and text encoder $g = g_\phi$ (e.g., a CLIP Transformer), both outputting ℓ_2-normalized embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$ in a common d-dimensional space via linear projection layers (`nn.Linear`), so that $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$.

DreamLIP combines **a generation process and two supervision signals**:

1. **Synthetic caption generation:** Given an image–text dataset $(\mathbf{x}_i, \mathbf{c}_i)$, DreamLIP uses a pre-trained MLLM F (e.g., ShareGPT4V, LLaVA-1.5, or InstructBLIP) to generate both long and short captions:

$$\mathbf{c}_i^l = F(\mathbf{x}_i, q_l), \qquad \mathbf{c}_i^s = F(\mathbf{x}_i, q_s),$$

   where $\mathbf{c}_i^l$ is the long caption (generated with prompt "*Describe the image in details*") and $\mathbf{c}_i^s$ is the short caption (generated with prompt "*Describe the image in short*"). Long captions typically contain 8–12 sentences (200–400 tokens), with each sentence describing local regions or specific objects in the image.

2. **Multi-positive contrastive loss ($\mathcal{L}_{\text{MPCL}}$):** For each image $\mathbf{x}_i$, DreamLIP constructs a sub-caption set by decomposing the long caption $\mathbf{c}_i^l = [\mathbf{c}_1, \ldots, \mathbf{c}_M]$ into individual sentences and combining with the raw caption $\mathbf{c}_i$ and short caption $\mathbf{c}_i^s$. During training, K sub-captions are uniformly sampled:

$$S_{i,j} \sim \text{Uniform}([\mathbf{c}_i, \mathbf{c}_i^s, \mathbf{c}_1, \ldots, \mathbf{c}_M]),$$

   where $j \in \{1, \ldots, K\}$ indexes the sampled sub-captions. Given image $\mathbf{x}_i$ and its sampled sub-captions $\{S_{i,j}\}_{j=1}^K$, the embeddings are:

$$\mathbf{v}_i = f(\mathbf{x}_i), \qquad \mathbf{t}_{i,j} = g(S_{i,j}).$$

   Following LaCLIP, DreamLIP applies a multi-positive contrastive loss that treats all sub-captions as positive pairs for the same image:

$$\mathcal{L}_{\text{MPCL}}^{t2v} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\log\frac{\exp(\mathbf{v}_i^\top \mathbf{t}_{i,j}/\tau)}{\sum_{n=1}^{N}\exp(\mathbf{v}_n^\top \mathbf{t}_{i,j}/\tau)},$$

where τ is a learnable temperature parameter. The symmetric vision-to-text loss is computed analogously, and the total multi-positive loss is $\mathcal{L}_{\text{MPCL}} = (\mathcal{L}^{t2v}_{\text{MPCL}} + \mathcal{L}^{v2t}_{\text{MPCL}})/2$, matching the bi-directional CLIP form in Equation (1).

3. **Sub-caption-specific grouping loss ($\mathcal{L}_{\text{Sub}}$):** To enhance fine-grained alignment, DreamLIP introduces a grouping loss that matches each sub-caption with its corresponding local image patches. The image encoder extracts patch embeddings $\mathbf{V}_i = [\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,HW}]$ where HW is the number of patches. For each sub-caption embedding $\mathbf{t}_{i,j}$, attention weights are computed as cosine similarities with all patch embeddings and then sparsified to select only the most relevant patches:

$$\hat{w}_{i,j,n} = \mathbf{t}_{i,j}^{\top}\mathbf{v}_{i,n}, \qquad \tilde{w}_{i,j,n} = \begin{cases} \hat{w}_{i,j,n} & \text{if } \hat{w}_{i,j,n} \geq \sigma \\ 0 & \text{otherwise} \end{cases},$$

where σ is a sparsity threshold. The sparsified weights are normalized and used to pool the patch embeddings into a sub-caption-specific representation:

$$\hat{\mathbf{v}}_{i,j} = \sum_{n=1}^{HW} \frac{\tilde{w}_{i,j,n}}{\sum_{n'} \tilde{w}_{i,j,n'}} \mathbf{v}_{i,n}.$$

The grouping loss aligns the pooled patch embedding $\hat{\mathbf{v}}_{i,j}$ with the sub-caption embedding $\mathbf{t}_{i,j}$:

$$\mathcal{L}_{\text{Sub}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M+2} \log \frac{\exp(\hat{\mathbf{v}}_{i,j}^{\top}\mathbf{t}_{i,j}/\tau)}{\sum_{n=1}^{K}\exp(\hat{\mathbf{v}}_{i,n}^{\top}\mathbf{t}_{i,j}/\tau)}.$$

The total DreamLIP loss combines both objectives:

$$\mathcal{L}_{\text{DreamLIP}} = \lambda_{\text{MPCL}}\mathcal{L}_{\text{MPCL}} + \lambda_S \mathcal{L}_{\text{Sub}},$$

where $\lambda_{\text{MPCL}} = 1.0$ and $\lambda_S \in [0.1, 0.9]$ are hyperparameters.

3.37 LocCa (Google, 2024, NeurIPS 2024)

LocCa (Wan et al., 2024) enhances vision–language pretraining by incorporating location-aware tasks into an encoder–decoder captioning framework. Unlike dual-encoder models such as CLIP, LocCa uses an image encoder $f = f_\theta$ (e.g., ViT-L/14) and a Transformer decoder $g = g_\phi$ with cross-attention to integrate visual and textual information. The decoder is trained autoregressively to generate text sequences conditioned on image features $\mathbf{V}_i = f(\mathbf{x}_i)$.

Bounding box annotations. LocCa requires bounding box annotations for location-aware pretraining. Given an image–text dataset $\{(\mathbf{x}_i, \mathbf{c}_i)\}_{i=1}^{N}$, a pretrained object detector (e.g., OWL-ViT) generates pseudo bounding boxes $\mathbf{b}_i \in \mathbb{N}^4$ with corresponding region descriptions $\mathbf{r}_i$. Box coordinates are represented as integral numbers (0–500) and directly converted to strings for tokenization (e.g., $[20, 480, 150, 200]$).

Multi-task pretraining. LocCa pretrains on three tasks simultaneously using task-specific prefixes:

1. **Standard captioning (Cap):** Given image $\mathbf{x}_i$, the model generates the full caption $\mathbf{c}_i$. The input sequence is prefixed with "`Cap`:" and the model predicts:

$$\mathbf{y}_i = [\text{“Cap:”}, \mathbf{c}_i].$$

2. **Automatic referring expression (ARef):** Given image $\mathbf{x}_i$ and region description $\mathbf{r}_i$, the model predicts the bounding box $\mathbf{b}_i$. The input sequence is:

$$\mathbf{y}_i = [\text{“ARef:”}, \mathbf{r}_i, \mathbf{b}_i].$$

The loss is applied to both $\mathbf{r}_i$ (teaching the model to identify and describe regions) and $\mathbf{b}_i$ (teaching box coordinate regression).

3. **Grounded captioning (GCap):** Given image $\mathbf{x}_i$, the model jointly predicts both the bounding box $\mathbf{b}_i$ and its description $\mathbf{r}_i$. The input sequence is:

$$\mathbf{y}_i = [\text{“GCap:”}, \mathbf{b}_i, \mathbf{r}_i].$$

For each image, the vision encoder f_θ extracts patch embeddings $\mathbf{V}_i$, and the decoder g_ϕ uses cross-attention to process these features while predicting the output sequence $\mathbf{y}_i$ autoregressively. The optimization objective is to maximize the log-likelihood:

$$\mathcal{L}_{\text{LocCa}} = -\sum_{i=1}^{|\mathbf{y}|} \log P_\theta(\mathbf{y}_i \mid \mathbf{y}_{<i}, \mathbf{x}_i),$$

where the loss is applied to the entire sequence excluding the task prefix.

Parallel prediction. Following recent captioning work, LocCa applies parallel prediction to 50% of training examples in the Cap task. This removes the causal attention mask and replaces the decoder input with mask tokens, forcing the model to predict caption tokens independently based solely on visual information without relying on preceding text tokens.

3.38 ZLaP (Naver Labs, 2024, CVPR 2024)

ZLaP (Kalantidis et al., 2024) is a non-parametric post-hoc method that improves zero-shot classification with vision–language models by running label propagation on a k-nearest-neighbor graph built over both class text embeddings and unlabeled image embeddings. In the notation above, class prompts yield ℓ_2-normalized text embeddings $\mathbf{t}_1, \ldots, \mathbf{t}_C \in \mathbb{R}^d$ and test images yield $\mathbf{v}_1, \ldots, \mathbf{v}_U \in \mathbb{R}^d$. ZLaP treats the class prompts as labeled nodes and images as unlabeled nodes, then uses the resulting geodesic (graph-based) similarities instead of raw cosine similarities for prediction. The graph is carefully adapted to the bimodal setting by using separate image–image and image–text neighborhoods and reweighting cross-modal edges. The framework derives both a transductive solution (propagate labels over the test graph) and an efficient inductive variant that precomputes a sparse propagated label matrix so that new images can be classified via a single graph solve or a fast weighted combination step, while the underlying VLM (e.g., CLIP with $\mathcal{L}_{\text{CLIP}}$) remains frozen.

3.39 ClusterMasking (2024, CVPR 2024)

ClusterMasking (Wei et al., 2024) improves upon the random masking used in FLIP by introducing a more structure-aware masking strategy based on visual similarity between patches. Instead of dropping patches independently, it first partitions an image $\mathbf{x}_i$ into patches and computes pairwise cosine similarities in a low-level feature space (pixel-normalized RGB, optionally combined with patch embeddings from $f = f_\theta$). A small subset of patches is randomly selected as anchor patches; for each anchor, a cluster is formed by collecting all patches within a similarity radius, and the entire cluster is then masked. This produces masks that tend to remove coherent visual regions (e.g., parts of objects) rather than scattered pixels, while still being simple and cheap to compute. The image encoder f_θ is applied only to the remaining visible patches $\tilde{\mathbf{x}}_i$, and the model is trained with the standard symmetric InfoNCE loss $\mathcal{L}_{\text{CLIP}}$ in Equation (1) between $\mathbf{v}_i = f(\tilde{\mathbf{x}}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$, with $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$. By focusing computation on a subset of patches and masking entire similarity-based clusters, ClusterMasking matches FLIP's efficiency and yields consistently better representations, especially at high masking ratios (e.g. 50–75%).

3.40 LoTLIP (2024, NeurIPS 2024)

LoTLIP (Wu et al., 2024) improves long-text understanding in contrastive language–image pre-training by (i) creating a 100M-scale long-caption dataset and (ii) augmenting the text encoder with learnable **corner tokens**. Images are collected from CC3M, CC12M, YFCC15M, LAION, and COYO; for each image $\mathbf{x}_i$ they keep the original short caption $\mathbf{c}_i^{\text{short}}$ (avg ∼18 tokens) and three detailed captions $\mathbf{c}_i^{(1)}, \mathbf{c}_i^{(2)}, \mathbf{c}_i^{(3)}$ generated by InstructBLIP, LLaVA, and ShareGPT4V (avg ∼136 tokens, ∼6 sentences).

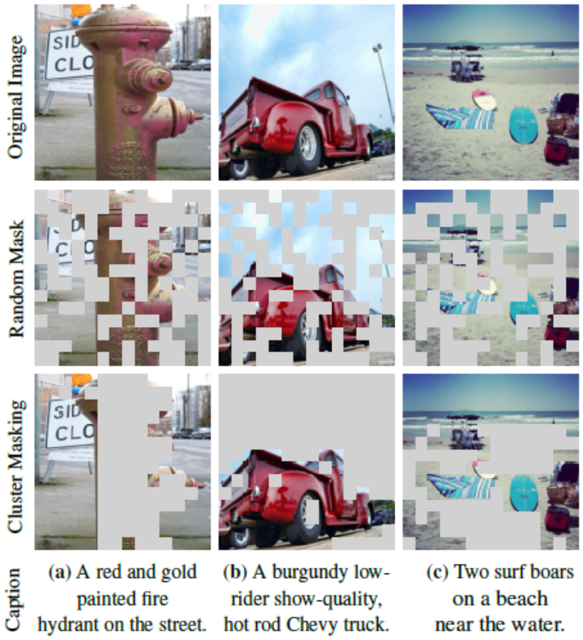

Original Image
Random Mask
Cluster Masking
Caption
(a) A red and gold painted fire hydrant on the street.
(b) A burgundy low-rider show-quality, hot rod Chevy truck.
(c) Two surf boars on a beach near the water.

Figure 13 Cluster masking randomly masks clusters of visually similar image patches when training like CLIP/FLIP.

Following LiT, the image encoder $f = f_\theta$ (e.g., ViT-B/16, frozen) produces $\mathbf{v}_i = f(\mathbf{x}_i)$ and a BERT-style text encoder $g = g_\phi$ produces text features $\mathbf{t}_i = g(\mathbf{c}_i^{\text{short}})$, trained with the standard symmetric InfoNCE loss on short captions:

$$\mathcal{L}_{\text{short}} = \mathcal{L}_{\text{i2t}} + \mathcal{L}_{\text{t2i}},$$

where each term follows the CLIP form in Equation (1), using ℓ_2-normalized embeddings $\mathbf{v}_i, \mathbf{t}_i$ and cosine similarity $\mathbf{v}_i^\top \mathbf{t}_i$.

For **long captions**, LoTLIP introduces a BERT-based text encoder with m learnable corner tokens $[\texttt{Cor}_1], \ldots, [\texttt{Cor}_m]$ inserted immediately after [CLS] in the tokenized sequence:

$$[\texttt{CLS}], [\texttt{Cor}_1], \ldots, [\texttt{Cor}_m], \text{tokens of sub-cap}_1, [\texttt{SEP}], \ldots$$

After the Transformer, the [CLS] embedding $\mathbf{t}_{g,i}$ is the global text feature; the corner embeddings $\mathbf{t}_{c_k,i}$ are additional global "views" of the same long caption. An attention mask is used so that [CLS] and corner tokens *cannot* attend to each other (they only attend to regular text tokens), and corner tokens cannot attend to one another. This forces each corner token to aggregate diverse textual information independently of [CLS] and of other corners.

LoTLIP combines **two supervision signals**:

1. **Short-text contrastive loss ($\mathcal{L}_{\text{short}}$)**: The standard CLIP-style symmetric InfoNCE aligns short captions $\mathbf{c}_i^{\text{short}}$ with images $\mathbf{x}_i$, as above.
2. **Long-text contrastive loss ($\mathcal{L}_{\text{long}}$)**: Given image feature $\mathbf{v}_i$ and long-text features $\mathbf{t}_{g,i}$, $\mathbf{t}_{c_k,i}$, LoTLIP adds a long-text contrastive loss that aligns the image with *all* global views:

$$\mathcal{L}_{\text{long}} = \mathcal{L}_{\text{long}}^{\text{i2t}} + \mathcal{L}_{\text{long}}^{\text{t2i}},$$

where $\mathcal{L}_{\text{long}}^{\text{i2t}}$ and $\mathcal{L}_{\text{long}}^{\text{t2i}}$ are InfoNCE terms computed once for $\mathbf{t}_{g,i}$ and once for each corner $\mathbf{t}_{c_k,i}$ (each with its own softmax over the batch).

The total training objective is:

$$\mathcal{L}_{\text{LoTLIP}} = \mathcal{L}_{\text{short}} + \mathcal{L}_{\text{long}},$$

which jointly improves both short- and long-text understanding in the same model.

3.41 PLIP (2024, NeurIPS 2024)

PLIP (Zuo et al., 2024) is a dual-encoder vision-language pre-training framework tailored for **person representation learning**. The visual encoder $f = f_\theta$ (e.g., ResNet-50) and text encoder $g = g_\phi$ (e.g., BERT with the last 5 layers unfrozen) independently encode their inputs, producing ℓ_2-normalized embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$ that are aligned via a symmetric InfoNCE loss as in CLIP. Unlike general CLIP, PLIP introduces three specialized pretext tasks to capture person-specific characteristics: fine-grained attributes and identity-level associations. PLIP is trained on SYNTH-PEDES, a large-scale person Re-ID dataset.

PLIP combines **three pretext tasks**:

1. **Text-guided Image Colorization (TIC):** Given a grayscale image $\mathbf{x}_i^{\text{gray}}$ and complete text description $\mathbf{c}_i^{\text{complete}}$, PLIP reconstructs the color image $\mathbf{x}_i^{\text{color}}$ by conditioning the visual encoder on the text features, thereby establishing fine-grained correspondence between person-related regions and color-part textual phrases.
2. **Image-guided Attributes Prediction (IAP):** Given a color image $\mathbf{x}_i^{\text{color}}$ and masked text $\mathbf{c}_i^{\text{masked}}$ with attribute tokens replaced by [MASK], the model predicts the missing attribute tokens based on the image features, mining fine-grained attribute information from the person's body in the image.
3. **Identity-based Vision-Language Contrast (IVLC):** Unlike instance-level CLIP, IVLC operates at the **identity level**. For each visual embedding $\mathbf{v}_i$, positive pairs include all text embeddings $\mathbf{t}_j$ corresponding to the same person identity; negatives are all text embeddings from different identities. This encourages cross-modal representations that are robust to viewpoint and occlusion while preserving fine-grained person-specific semantics.

The full PLIP objective is a weighted sum over the IVLC contrastive loss plus the TIC and IAP reconstruction losses, all optimized jointly on SYNTH-PEDES.

3.42 CLIPS (2024, arXiv 2024)

CLIPS (Liu et al., 2024) enhances CLIP training with synthetic captions via two key designs: (1) **sub-caption sampling** for contrastive learning, and (2) an **asymmetric autoregressive decoder** to predict full synthetic captions. The architecture uses image encoder $f = f_\theta$ (e.g., ViT-S/16, ViT-B/16, ViT-L/16, ViT-H/14) and text encoder $g = g_\phi$ (standard CLIP Transformer, without causal mask) that output ℓ_2-normalized embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$. The model is trained with a symmetric InfoNCE loss $\mathcal{L}_{\text{CLIP}}$ in Equation (1) over image–text pairs.

CLIPS combines **two supervision signals**:

1. **Multi-positive contrastive learning with sub-captions:** For each training sample, CLIPS samples one random sentence $\mathbf{c}_{syn}^{short}$ from the full synthetic caption and combines it with the original web-crawled caption $\mathbf{c}_{orig}$. Both captions are fed to the text encoder to produce embeddings $\mathbf{t}_{orig}$ and $\mathbf{t}_{syn}^{short}$, which are treated as positive pairs for the same image $\mathbf{x}_i$. The model then uses a multi-positive contrastive loss over $\{\mathbf{t}_{orig}, \mathbf{t}_{syn}^{short}\}$ in the same batch, generalizing the standard CLIP loss to multiple gold captions per image.
2. **Asymmetric autoregressive captioning with synthetic targets:** To fully utilize synthetic captions, CLIPS adds an autoregressive text decoder that predicts the **full synthetic caption** $\mathbf{c}_{syn}^{full}$ conditioned on the image $\mathbf{x}_i$ and the web-crawled caption $\mathbf{c}_{web}$. Unlike CoCa (symmetric: input = output = $\mathbf{c}_{web}$), CLIPS uses an **asymmetric** design, where the input to the text encoder is the web-crawled caption $\mathbf{c}_{web}$ and the prediction target is the full-length synthetic caption $\mathbf{c}_{syn}^{full}$. The decoder is optimized with a standard cross-entropy captioning loss

$$\mathcal{L}_{\text{cap}} = -\sum_t \log P(\mathbf{c}_{syn,t}^{full} \mid \mathbf{c}_{web,<t}, \mathbf{V}_i),$$

where $\mathbf{V}_i = f(\mathbf{x}_i)$ are the image patch embeddings, and $\mathbf{c}_{syn,t}^{full}$ is the synthetic token at position t.

The full CLIPS training objective is:

$$\mathcal{L}_{\text{CLIPS}} = \mathcal{L}_{\text{CLIP}} + \lambda_{\text{cap}}\mathcal{L}_{\text{cap}},$$

where λ_{cap} trades off the contrastive and captioning objectives, better leveraging synthetic knowledge while preserving the discriminative power of CLIP-style alignment.

3.43 TIPS (Google, 2025, ICLR 2025)

TIPS (Maninis et al., 2025) improves vision–language pretraining by combining CLIP-style image–text contrastive learning with both DINO-style self-distillation and iBOT-style masked image modeling. Following CLIP, it uses an image encoder $f = f_\theta$ (student, e.g., ViT-B/14, ViT-g/14) and a text encoder $g = g_\phi$ (e.g., a 12-layer Transformer) that both output ℓ_2-normalized embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$ in a common d-dimensional embedding space via projection layers, so that $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$. In addition, TIPS maintains an EMA teacher image encoder $f' = f_{\bar{\theta}}$ for self-supervision. The teacher parameters are updated from the student by EMA: $\bar{\theta} \leftarrow \lambda\bar{\theta} + (1-\lambda)\theta$, with $\lambda \in [0,1]$. The teacher produces target embeddings $\mathbf{v}'_i = f'(\mathbf{x}_i)$, which are used to supervise the student encoder.

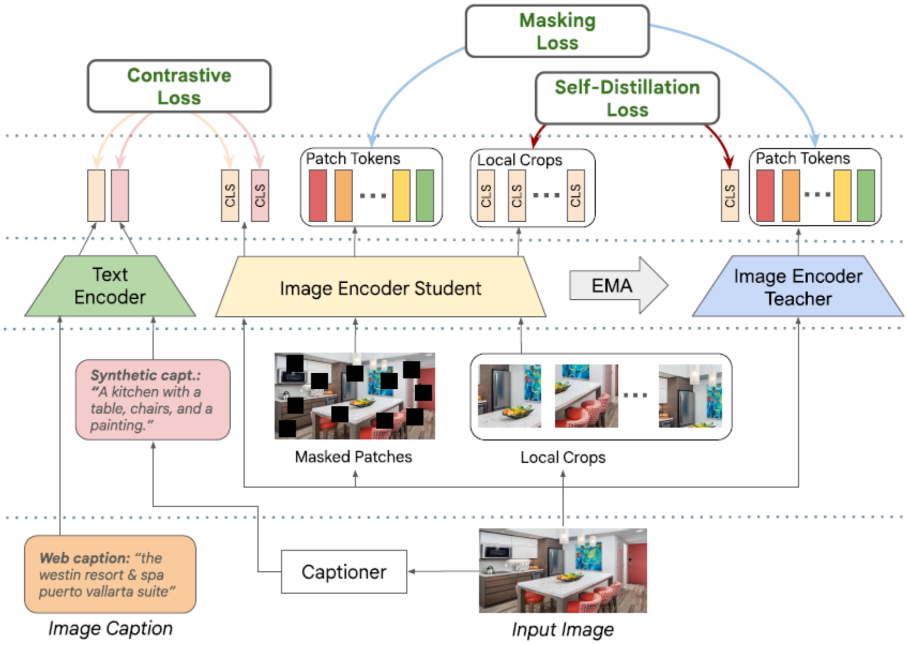


Figure 14 TIPS combines CLIP-style contrastive training with dual captions (noisy web and synthetic), DINO self-distillation, and iBOT masked image modeling.

For each image with noisy web caption $(\mathbf{x}_i, \mathbf{c}_i)$ and synthetic caption $\mathbf{c}'_i$, TIPS samples $M = 6$ **local** crops and 1 **global** crop:

$$\mathbf{x}_i^{\text{loc},m} = \text{aug}_{\text{loc}}(\mathbf{x}_i) \text{ for } m = 1,\ldots,M, \qquad \mathbf{x}_i^{\text{glob}} = \text{aug}_{\text{glob}}(\mathbf{x}_i),$$

and creates a **masked** version of the global crop with 75% random masking:

$$\mathbf{x}_i^{\text{mask}} = \text{mask}(\mathbf{x}_i^{\text{glob}}).$$

The student encoder processes all crops with a **dual** CLS **token** design:

$$[\mathbf{v}_i^g, \mathbf{v}_i^{g'}, \{\mathbf{v}_i^n\}] = f(\mathbf{x}_i^{\text{glob}}), \qquad \mathbf{v}_i^{\text{loc},m} = f(\mathbf{x}_i^{\text{loc},m})[\text{CLS}],$$

$$[\mathbf{v}_i^{g,\text{mask}}, \mathbf{v}_i^{g',\text{mask}}, \{\mathbf{v}_i^{n,\text{mask}}\}] = f(\mathbf{x}_i^{\text{mask}}),$$

where $\mathbf{v}_i^g$ is the object-centric CLS token, $\mathbf{v}_i^{g'}$ is the spatially-aware CLS token, and $\{\mathbf{v}_i^n\}_{n=1}^N$ are patch tokens. All embeddings are projected to the same space and ℓ_2-normalized. The teacher processes only the unmasked global crop:

$$[\bar{\mathbf{v}}_i^g, \bar{\mathbf{v}}_i^{g'}, \{\bar{\mathbf{v}}_i^n\}] = f'(\mathbf{x}_i^{\text{glob}}).$$

The captions are processed by the text encoder to obtain

$$\mathbf{t}_i = g(\mathbf{c}_i), \qquad \mathbf{t}'_i = g(\mathbf{c}'_i).$$

TIPS combines **four** supervision signals:

1. **Dual image–text contrastive loss ($\mathcal{L}_{\text{CLIP}}$)**: TIPS uses two separate CLIP-style InfoNCE losses with dual embeddings. Given a batch $\{(\mathbf{v}_i^g, \mathbf{t}_i)\}_{i=1}^N$ and $\{(\mathbf{v}_i^{g'}, \mathbf{t}'_i)\}_{i=1}^N$, the dual contrastive loss is

$$\mathcal{L}_{\text{CLIP}} = \frac{1}{2}\left[\mathcal{L}_{\text{CLIP}}(\mathbf{v}^g, \mathbf{t}) + \mathcal{L}_{\text{CLIP}}(\mathbf{v}^{g'}, \mathbf{t}')\right],$$

where each term is the symmetric InfoNCE loss matching the form in Equation (1):

$$\mathcal{L}_{\text{CLIP}}(\mathbf{v}, \mathbf{t}) = -\frac{1}{N}\left(\sum_{i=1}^N \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^N \exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau)} + \sum_{i=1}^N \log \frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i/\tau)}{\sum_{j=1}^N \exp(\mathbf{t}_i^\top \mathbf{v}_j/\tau)}\right).$$

2. **Local-to-global self-distillation loss ($\mathcal{L}_{\text{distill}}$)**: To learn locally aware image features, TIPS adds a DINO-style self-distillation loss between the *local* student CLS embeddings $\mathbf{v}_i^{\text{loc},m}$ and the *global* teacher CLS embedding $\bar{\mathbf{v}}_i^{g'}$. A projection head P_s (3-layer MLP + weight-normalized projection to 32k prototypes) maps student features, and a teacher projection head P_t (EMA-updated from P_s) maps teacher features. Temperature-scaled softmax with centering defines probability distributions

$$P_i^{\text{teacher}} = \text{softmax}\left(\frac{P_t(\bar{\mathbf{v}}_i^{g'}) - \mathbf{c}}{\tau_{\text{t}}}\right), \qquad P_{i,m}^{\text{student}} = \text{softmax}\left(\frac{P_s(\mathbf{v}_i^{\text{loc},m})}{\tau_{\text{s}}}\right),$$

where $\mathbf{c}$ is a running center vector (EMA with momentum 0.9), $\tau_{\text{t}} = 0.07$, and $\tau_{\text{s}} = 0.1$. The self-distillation loss is

$$\mathcal{L}_{\text{distill}} = -\frac{1}{N}\sum_{i=1}^N \sum_{m=1}^M \left(P_i^{\text{teacher}}\right)^\top \log P_{i,m}^{\text{student}}.$$

3. **Masked image modeling loss ($\mathcal{L}_{\text{mask}}$)**: To encourage spatial coherence in patch embeddings, TIPS adds an iBOT-style masked reconstruction loss. Separate projection heads $\tilde{P}_s$ and $\tilde{P}_t$ (same architecture as above but unshared weights) map masked student patches $\mathbf{v}_i^{n,\text{mask}}$ and unmasked teacher patches $\bar{\mathbf{v}}_i^n$. For each masked patch n, probability distributions are defined as

$$\tilde{P}_{i,n}^{\text{teacher}} = \text{softmax}\left(\frac{\tilde{P}_t(\bar{\mathbf{v}}_i^n) - \tilde{\mathbf{c}}}{\tilde{\tau}_{\text{t}}}\right), \qquad \tilde{P}_{i,n}^{\text{student}} = \text{softmax}\left(\frac{\tilde{P}_s(\mathbf{v}_i^{n,\text{mask}})}{\tilde{\tau}_{\text{s}}}\right),$$

where $\tilde{\mathbf{c}}$ is a separate running center. The masked image modeling loss is

$$\mathcal{L}_{\text{mask}} = -\frac{1}{N}\sum_{i=1}^N \sum_{n\in\text{masked}} \left(\tilde{P}_{i,n}^{\text{teacher}}\right)^\top \log \tilde{P}_{i,n}^{\text{student}}.$$

The final TIPS training objective is a weighted sum

$$\mathcal{L}_{\text{TIPS}} = \mathcal{L}_{\text{CLIP}} + \lambda_{\text{dist}}\,\mathcal{L}_{\text{distill}} + \lambda_{\text{mask}}\,\mathcal{L}_{\text{mask}},$$

with $\lambda_{\text{dist}} = 1$ and $\lambda_{\text{mask}} = 2$, which jointly enforces image–text alignment, local-to-global consistency, and spatially coherent patch representations, yielding strong frozen features for both dense and global downstream tasks.

3.44 TripletCLIP (NeurIPS 2024)

TripletCLIP (Patel et al., 2024) improves compositional reasoning in CLIP via synthetic vision-language negatives. Following CLIP, it uses an image encoder $f = f_\theta$ (e.g., ViT-B/32) and a text encoder $g = g_\phi$ (standard Transformer) that independently map inputs into a shared embedding space, producing ℓ_2-normalized embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$ and $\mathbf{t}_i = g(\mathbf{c}_i)$ optimized with the symmetric InfoNCE loss $\mathcal{L}_{\text{CLIP}}$ in Equation (1). Unlike standard CLIP, TripletCLIP generates hard negative image–text pairs $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{c}}_i)$ using LLMs and text-to-image diffusion models:

1. **Negative caption generation:** Using Mistral-7B-Instruct-v0.2 with in-context learning, the model generates linguistically accurate hard negative captions $\tilde{\mathbf{c}}_i$ from positive captions $\mathbf{c}_i$. The prompt instructs the LLM to create similar descriptions with significant differences such that they cannot describe the same image. Example: "A dog to the left of the cat" → "A dog to the right of the cat".

2. **Negative image generation:** Using SDXL-turbo, the model generates negative images $\tilde{\mathbf{x}}_i$ conditioned on negative captions $\tilde{\mathbf{c}}_i$.

The negative image and caption are encoded to obtain

$$\tilde{\mathbf{v}}_i = f(\tilde{\mathbf{x}}_i), \qquad \tilde{\mathbf{t}}_i = g(\tilde{\mathbf{c}}_i),$$

with $\|\tilde{\mathbf{v}}_i\|_2 = \|\tilde{\mathbf{t}}_i\|_2 = 1$.

TripletCLIP combines **two NegCLIP-style triplet terms**:

1. **Triplet contrastive loss for $(\mathbf{x}, \mathbf{c}, \tilde{\mathbf{c}})$:** For the first triplet, the standard NegCLIP-style image-to-text loss augments the CLIP denominator with negative captions:

$$\mathcal{L}_{\text{img}\to\text{txt}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau) + \sum_{m=1}^{N}\exp(\mathbf{v}_i^\top \tilde{\mathbf{t}}_m/\tau)},$$

where τ is the temperature parameter. The asymmetric NegCLIP loss for this triplet is

$$\mathcal{L}_{\text{NegCLIP}}(\mathbf{v}, \mathbf{t}, \tilde{\mathbf{t}}) = \mathcal{L}_{\text{txt}\to\text{img}} + \mathcal{L}_{\text{img}\to\text{txt}},$$

where $\mathcal{L}_{\text{txt}\to\text{img}}$ is the standard CLIP text-to-image loss without negative augmentation.

2. **Triplet contrastive loss for $(\tilde{\mathbf{x}}, \tilde{\mathbf{c}}, \mathbf{c})$:** The second term ensures that negative images $\tilde{\mathbf{x}}$ are closer to negative captions $\tilde{\mathbf{c}}$ than to positive captions $\mathbf{c}$, providing additional supervision for understanding compositional differences. The full triplet contrastive loss alternates between positive and negative image–text pairs:

$$\mathcal{L}_{\text{TripletCLIP}} = \mathcal{L}_{\text{NegCLIP}}(\mathbf{v}, \mathbf{t}, \tilde{\mathbf{t}}) + \mathcal{L}_{\text{NegCLIP}}(\tilde{\mathbf{v}}, \tilde{\mathbf{t}}, \mathbf{t}).$$

3.45 FLAME (2024, CVPR 2025)

FLAME (Cao et al., 2025) addresses the limitations of traditional CLIP text encoders, such as limited capacity and difficulty processing long-form text. The framework leverages a frozen Large Language Model (LLM) as the text tower $g = g_\phi$ to achieve superior multilingual and long-text generalization, while a dual-encoder image tower $f = f_\theta$ maps images $\mathbf{x}_i$ into ℓ_2-normalized embeddings $\mathbf{v}_i = f(\mathbf{x}_i)$. The model is trained with a symmetric InfoNCE loss $\mathcal{L}_{\text{CLIP}}$ in Equation (1) over $(\mathbf{v}_i, \mathbf{t}_i)$ pairs, but with richer text signals extracted from the frozen LLM.

The architecture comprises two primary components:

- **Multifaceted prompt distillation:** This technique uses frozen prompts to query the LLM over long captions and extract diverse semantic representations (e.g., objects, attributes, relationships), which are distilled into the vision tower f_θ, ensuring better alignment between the image features and the rich information in the text.

- **Data-efficient alignment:** By utilizing the pre-trained knowledge of frozen LLMs, FLAME achieves high performance with significantly less training data than standard VLM pre-training methods, via facet-decomposed text features and an offline embedding strategy that reduces training-time compute overhead while preserving the long-context and multilingual capabilities of the original LLM.

3.46 SuperCLIP (2024, NeurIPS 2025)

SuperCLIP (Zhao et al., 2025) augments CLIP with classification-based supervision to improve fine-grained visual-text alignment. Following CLIP, it uses an image encoder $f = f_\theta$ (e.g., ViT-B/16, ViT-L/14) and a text encoder $g = g_\phi$ (standard Transformer) that independently map inputs into a shared embedding space. Given an image $\mathbf{x}_i$ and its caption $\mathbf{c}_i$, the encoders produce ℓ_2-normalized embeddings:

$$\mathbf{v}_i = \frac{f(\mathbf{x}_i)}{\|f(\mathbf{x}_i)\|_2}, \qquad \mathbf{t}_i = \frac{g(\mathbf{c}_i)}{\|g(\mathbf{c}_i)\|_2},$$

matching the CLIP form in Equation (1).

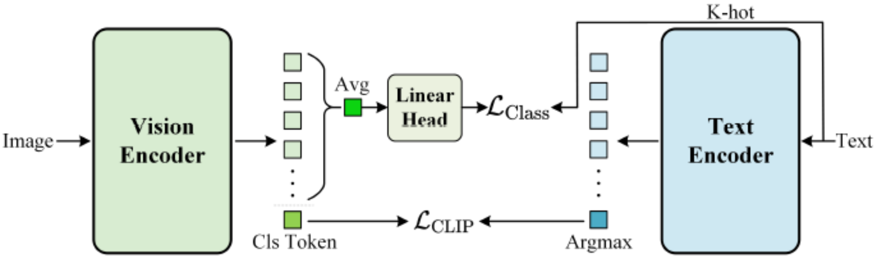


Figure 15 SuperCLIP introduces a simple classification-based supervision into the CLIP framework by adding a linear layer to the image encoder that maps the averaged image features to text classification targets.

SuperCLIP combines **two** supervision signals:

1. **Image–text contrastive loss ($\mathcal{L}_{\text{CLIP}}$):** SuperCLIP retains the standard CLIP-style symmetric InfoNCE loss

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{2N}\left(\sum_{i=1}^{N}\log\frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{v}_i^\top \mathbf{t}_j/\tau)} + \sum_{i=1}^{N}\log\frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{t}_i^\top \mathbf{v}_j/\tau)}\right),$$

where τ is the temperature parameter.

2. **Classification head and CE loss ($\mathcal{L}_{\text{CE}}$):** SuperCLIP adds a lightweight linear head on top of the vision encoder to predict which text tokens appear in the caption. Each $\mathbf{c}_i$ is tokenized with CLIP's vocabulary of size V, yielding a token set $\mathcal{C}_i \subset \{1, \dots, V\}$ and a K-hot label vector $\mathbf{y}_i \in \{0,1\}^V$, where $y_{i,c} = 1$ iff $c \in \mathcal{C}_i$. IDF weights are computed as

$$w_c = \log\left(\frac{|\mathcal{D}|}{1 + \text{df}(c)}\right),$$

and converted to a normalized target distribution

$$\hat{y}_{i,c} = \frac{w_c y_{i,c}}{\sum_{c'=1}^{V} w_{c'} y_{i,c'}}.$$

Let $\mathbf{h}_i$ be the pooled image feature (e.g., average over patch tokens or a CLS token), and $W \in \mathbb{R}^{V\times d}$ the classification weight. The logits are $\mathbf{z}_i = W\mathbf{h}_i$, and the classification loss is the cross-entropy between $\hat{\mathbf{y}}_i$ and softmax($\mathbf{z}_i$):

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{V}\hat{y}_{i,c}\log\frac{\exp(z_{i,c})}{\sum_{c'=1}^{V}\exp(z_{i,c'})}.$$

The total SuperCLIP loss is a weighted sum of contrastive and classification supervision

$$\mathcal{L}_{\text{SuperCLIP}} = \mathcal{L}_{\text{CLIP}} + \lambda_{\text{CE}}\,\mathcal{L}_{\text{CE}},$$

where λ_{CE} controls the strength of the auxiliary CE loss (typically $\lambda_{\text{CE}} \geq 1$).

3.47 SigLIP-2 (Google, 2025, arXiv 2025)

SigLIP-2 (Tschannen et al., 2025) is trained in two main stages, always using the standard CLIP-style dual encoder $f = f_\theta$, $g = g_\phi$ with embeddings

$$\mathbf{v}_i = f(\mathbf{x}_i), \qquad \mathbf{t}_i = g(\mathbf{c}_i),$$

and a multilingual tokenizer for g_ϕ, producing ℓ_2-normalized representations $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$.

Stage 1: Sigmoid contrastive + LocCa-style decoder.

1. **Sigmoid (SigLIP) image–text loss ($\mathcal{L}_{\textbf{sigmoid}}$):** Instead of InfoNCE, SigLIP treats every pair $(\mathbf{v}_i, \mathbf{t}_j)$ in the batch as a binary label (match vs non-match) and applies a sigmoid loss over all N^2 pairs:

   $$\mathcal{L}_{\text{sigmoid}} = -\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \big[y_{ij} \log \sigma(\mathbf{v}_i^\top \mathbf{t}_j) + (1 - y_{ij}) \log(1 - \sigma(\mathbf{v}_i^\top \mathbf{t}_j))\big],$$

   where $y_{ij} = 1$ iff $i = j$ and σ is the logistic sigmoid, generalizing the SigLIP objective.

2. **LocCa-style captioning/grounded decoding loss ($\mathcal{L}_{\textbf{LocCa}}$):** A Transformer decoder with cross-attention is attached to the vision tokens (before pooling). It is trained with:

   - *Image captioning* loss (autoregressive and parallel prediction) on full-image captions,
   - *Dense captioning* loss on region-level captions,
   - *Referring expression* loss: predicting region boxes from text (and region text from boxes).

   This yields a decoder loss $\mathcal{L}_{\text{LocCa}}$ over captioning, dense captioning, and referring targets, all computed from the image–text pretraining signals.

Stage 2: SILC/TIPS-style self-distillation and masking (last 20% of training).

1. **Local-to-global self-distillation loss ($\mathcal{L}_{\textbf{distill}}$):** An EMA teacher image encoder $f' = f_{\bar{\theta}}$ is introduced; for each step, the student sees multiple local crops and the teacher sees a global image. A DINO-style soft-label loss $\mathcal{L}_{\text{distill}}$ enforces

   $$f(\text{local crops}) \approx f'(\text{global image})$$

   in a prototype space, promoting local-to-global consistency.

2. **Masked image prediction loss ($\mathcal{L}_{\textbf{mask}}$):** In parallel, a fraction of image patches (e.g., 50%) in the student stream are replaced by mask tokens, and the student is trained to match the teacher's per-patch features at those masked locations, yielding a masked loss $\mathcal{L}_{\text{mask}}$ in the TIPS style, using a separate projection head and center vector.

Final training objective. Over the full run, SigLIP-2 minimizes:

$$\mathcal{L}_{\text{SigLIP2}} = \mathcal{L}_{\text{sigmoid}} + \mathcal{L}_{\text{LocCa}} + \alpha\, \mathcal{L}_{\text{distill}} + \beta\, \mathcal{L}_{\text{mask}},$$

where $\mathcal{L}_{\text{distill}}$ and $\mathcal{L}_{\text{mask}}$ are activated only in the last 20% of training, with scale factors α, β set by model size (smaller for larger encoders).

3.48 CLOC (2025, ICML 2025)

CLOC (Chen et al., 2025) builds on the standard CLIP dual-encoder. Given an image–caption pair $(\mathbf{x}_i, \mathbf{c}_i)$, the image encoder $f = f_\theta$ and text encoder $g = g_\phi$ produce embeddings

$$\mathbf{v}_i = f(\mathbf{x}_i), \qquad \mathbf{t}_i = g(\mathbf{c}_i),$$

which are ℓ_2-normalized before computing similarities $\mathbf{v}_i^\top \mathbf{t}_i$, following Equation (1).

CLOC combines **two** supervision signals along CLIP's contrastive loss:

1. **Region–text contrastive loss ($\mathcal{L}_{\text{loc}}$)**: Let $f'(\mathbf{x}_i)$ denote the ViT patch tokens (before global pooling). For image i, we have M region prompts $\{\boldsymbol{\ell}_i^{(m)}\}_{m=1}^M$ (e.g., bounding boxes) and corresponding region captions $\{\mathbf{c}_i^{(m)}\}_{m=1}^M$. Region texts are encoded as

$$\mathbf{t}_i^{(m)} = g(\mathbf{c}_i^{(m)}),$$

and a lightweight Prompter produces region embeddings

$$\mathbf{z}_i^{(m)} = \text{Prompter}\big(\boldsymbol{\ell}_i^{(m)}, f'(\mathbf{x}_i)\big).$$

After ℓ_2-normalization, define region–text similarities

$$R_{(i,m),(j,m')} = \frac{\big(\mathbf{z}_i^{(m)}\big)^\top \mathbf{t}_j^{(m')}}{\tau}.$$

The region-to-text loss is

$$\mathcal{L}_{R\to T} = -\frac{1}{MN}\sum_{i=1}^{N}\sum_{m=1}^{M}\log\frac{\exp\big(R_{(i,m),(i,m)}\big)}{\sum_{j=1}^{N}\sum_{m'=1}^{M}\exp\big(R_{(i,m),(j,m')}\big)},$$

and the symmetric text-to-region loss $\mathcal{L}_{T\to R}$ is defined analogously by normalizing over (j, m') for each fixed (i, m). The localized CLIP loss is

$$\mathcal{L}_{\text{loc}} = \frac{1}{2}\left(\mathcal{L}_{R\to T} + \mathcal{L}_{T\to R}\right).$$

2. **Optional grounding loss ($\mathcal{L}_{\text{ground}}$)**: For text-conditioned prompting, CLOC can also regress the region box from text. Using

$$\tilde{\mathbf{z}}_i^{(m)} = \text{Prompter}\big(g(\mathbf{c}_i^{(m)}), f'(\mathbf{x}_i)\big),$$

a small MLP head Head$(\cdot)$ predicts the box $\boldsymbol{\ell}_i^{(m)}$, and a regression loss is applied:

$$\mathcal{L}_{\text{ground}} = \frac{1}{4MN}\sum_{i=1}^{N}\sum_{m=1}^{M}\big\|\boldsymbol{\ell}_i^{(m)} - \text{Head}\big(\tilde{\mathbf{z}}_i^{(m)}\big)\big\|_2^2.$$

The final loss is:

$$\mathcal{L}_{\text{CLOC-total}} = \mathcal{L}_{\text{CLIP}} + \lambda_{\text{loc}}\,\mathcal{L}_{\text{loc}} + \lambda_{\text{ground}}\,\mathcal{L}_{\text{ground}},$$

where λ_{loc} and λ_{ground} control the strength of region-level contrastive and grounding supervision (with λ_{ground} often set small or zero if grounding is not used).

3.49 CLIPpy (2023, ICCV 2023)

CLIPpy (Ranasinghe et al., 2023) keeps the standard CLIP dual-encoder framework. Given an image–caption pair $(\mathbf{x}_i, \mathbf{c}_i)$, the image encoder $f = f_\theta$ (ViT-B/16) and text encoder $g = g_\phi$ (Sentence-T5) independently map inputs into a shared space:

$$\mathbf{v}_i = f(\mathbf{x}_i) \in \mathbb{R}^{H\times W\times D}, \qquad \mathbf{t}_i = g(\mathbf{c}_i) \in \mathbb{R}^D,$$

where $\mathbf{v}_i$ is a spatial token map and $\mathbf{t}_i$ a sentence embedding. The final image–text similarities are computed after projecting $\bar{\mathbf{v}}_i, \mathbf{t}_i$ into the same d-dimensional space and normalizing so that $\|\bar{\mathbf{v}}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$, reducing cosine similarity to $\bar{\mathbf{v}}_i^\top \mathbf{t}_i$.

CLIPpy introduces two minimal but crucial changes:

1. **Pretraining of backbones:**

- f_θ is initialized from a self-supervised ViT (e.g., DINO) trained on ImageNet-1K, so spatial tokens already reflect perceptual grouping.
- g_ϕ is initialized from Sentence-T5 (a contrastive sentence encoder) for robust sentence-level embeddings.

2. **Max-pooling image aggregation:** Instead of a `CLS` token or global average pooling, CLIPpy aggregates spatial features with global max-pooling:

$$\bar{\mathbf{v}}_i[d] = \max_{h,w} \mathbf{v}_i[h, w, d], \qquad d = 1, \ldots, D,$$

   and uses $\bar{\mathbf{v}}_i \in \mathbb{R}^D$ as the image embedding for contrastive training. This concentrates gradients on a small set of spatial positions and encourages object-aligned feature peaks.

The contrastive loss remains the standard symmetric CLIP InfoNCE on $(\bar{\mathbf{v}}_i, \mathbf{t}_i)$, matching the form in Equation (1):

$$\mathcal{L}_{\text{CLIPpy}} = -\frac{1}{N}\left(\sum_{i=1}^{N} \log \frac{\exp(\bar{\mathbf{v}}_i^\top \mathbf{t}_i/\tau)}{\sum_{j=1}^{N} \exp(\bar{\mathbf{v}}_i^\top \mathbf{t}_j/\tau)} + \sum_{i=1}^{N} \log \frac{\exp(\mathbf{t}_i^\top \bar{\mathbf{v}}_i/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{t}_i^\top \bar{\mathbf{v}}_j/\tau)}\right).$$

With these changes, spatial tokens $\mathbf{v}_i$ support:

- *bottom-up* grouping (e.g., unsupervised segmentation, via clustering in token space), and
- *top-down* grouping (e.g., zero-shot semantic segmentation via token–text similarity),

without any segmentation labels or task-specific heads.

References

Anjia Cao, Xing Wei, and Zhiheng Ma. Flame: Frozen large language models enable data-efficient language-image pre-training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4080–4090, 2025.

Hong-You Chen, Zhengfeng Lai, Haotian Zhang, Xinze Wang, Marcin Eichner, Keen You, Meng Cao, Bowen Zhang, Yinfei Yang, and Zhe Gan. Contrastive localized language-image pre-training. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 8386–8402. PMLR, 13–19 Jul 2025. https://proceedings.mlr.press/v267/chen25ah.html.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2829, June 2023.

Quan Cui, Boyan Zhou, Yu Guo, Weidong Yin, Hao Wu, Osamu Yoshie, and Yubo Chen. Contrastive vision-language pre-training with limited resources. *European Conference on Computer Vision*, 2022.

Xiaoyi Dong, Jianmin Bao, Yinglin Zheng, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Maskclip: Masked self-distillation advances contrastive language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10995–11005, June 2023.

Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving CLIP training with language rewrites. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. https://openreview.net/forum?id=SVjDiiVySh.

Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet : End-to-end video-language transformers with masked visual-token modeling, 2022. https://arxiv.org/abs/2111.12681.

Shijie Geng, Jianbo Yuan, Yu Tian, Yuxiao Chen, and Yongfeng Zhang. HiCLIP: Contrastive language-image pretraining with hierarchy-aware attention. In *The Eleventh International Conference on Learning Representations*, 2023. https://openreview.net/forum?id=0eTTKOOOQkV.

Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan A. Rossi, Vishwa Vinay, and Aditya Grover. CyCLIP: Cyclic contrastive language-image pretraining. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. https://openreview.net/forum?id=I-6yh2-dkyD.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. https://doi.org/10.5281/zenodo.5143773. If you use this software, please cite it as below.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR, 18–24 Jul 2021. https://proceedings.mlr.press/v139/jia21b.html.

Siddharth Joshi, Arnav Jain, Ali Payani, and Baharan Mirzasoleiman. Data-efficient contrastive multi-modal learning: Prioritizing data quality over quantity. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

Yannis Kalantidis, Giorgos Tolias, et al. Label propagation for zero-shot classification with vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23209–23218, 2024.

Jannik Kossen, Mark Collier, Basil Mustafa, Xiao Wang, Xiaohua Zhai, Lucas Beyer, Andreas Peter Steiner, Jesse Berent, Rodolphe Jenatton, and Effrosyni Kokiopoulou. Three towers: Flexible contrastive learning with pretrained image models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. https://openreview.net/forum?id=LSYQB4CwD3.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 17–23 Jul 2022a. https://proceedings.mlr.press/v162/li22n.html.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR, 23–29 Jul 2023a. https://proceedings.mlr.press/v202/li23q.html.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10965–10975, June 2022b.

Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *International Conference on Learning Representations*, 2022c. https://openreview.net/forum?id=zq1iJkNk3uN.

Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23390–23400, June 2023b.

Yanqing Liu, Xianhang Li, Zeyu Wang, Bingchen Zhao, and Cihang Xie. Clips: An enhanced clip framework for learning with synthetic captions, 2024. https://arxiv.org/abs/2411.16828.

Kevis-Kokitsi Maninis, Kaifeng Chen, Soham Ghosh, Arjun Karpur, Koert Chen, Ye Xia, Bingyi Cao, Daniel Salz, Guangxing Han, Jan Dlabal, Dan Gnanapragasam, Mojtaba Seyedhosseini, Howard Zhou, and Andre Araujo. Tips: Text-image pretraining with spatial awareness. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 64256–64279, 2025. https://proceedings.iclr.cc/paper_files/paper/2025/file/a15a2ece7f0663d1ba7db91103ac61c9-Paper-Conference.pdf.

Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. *arXiv preprint arXiv:2112.12750*, 2021.

Muhammad Ferjad Naeem, Yongqin Xian, Xiaohua Zhai, Lukas Hoyer, Luc Van Gool, and Federico Tombari. Silc: Improving vision language pretraining with self-distillation. In *European conference on computer vision*, pages 38–55. Springer, 2024.

Maitreya Patel, Abhiram Kusumba, Sheng Cheng, Changhoon Kim, Tejas Gokhale, Chitta Baral, and Yezhou Yang. Tripletclip: Improving compositional reasoning of clip via synthetic vision-language negatives. *Advances in Neural Information Processing Systems*, 2024.

Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, Mingxing Tan, and Quoc V. Le. Combined scaling for zero-shot transfer learning. *Neurocomputing*, 555:126658, 2023. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.126658. https://www.sciencedirect.com/science/article/pii/S0925231223007816.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PmLR, 2021.

Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens. Perceptual grouping in contrastive vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5571–5584, 2023.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15638–15650, June 2022.

Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. Eva-clip-18b: Scaling clip to 18 billion parameters. *arXiv preprint arXiv:2402.04252*, 2024.

Michael Tschannen, Manoj Kumar, Andreas Peter Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. Image captioners are scalable vision learners too. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. https://openreview.net/forum?id=A7feCufBhL.

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, 2025. https://arxiv.org/abs/2502.14786.

Bo Wan, Michael Tschannen, Yongqin Xian, Filip Pavetic, Ibrahim Alabdulmohsin, Xiao Wang, André Susano Pinto, Andreas Steiner, Lucas Beyer, and Xiaohua Zhai. Locca: Visual pretraining with location-aware captioners. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 116355–116387. Curran Associates, Inc., 2024. doi: 10.52202/079017-3695. https://proceedings.neurips.cc/paper_files/paper/2024/file/d303b4f1ef8d8274ae6b152df70f5406-Paper-Conference.pdf.

Zihao Wei, Zixuan Pan, and Andrew Owens. Efficient vision-language pre-training by cluster masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26815–26825, 2024.

Wei Wu, Kecheng Zheng, Shuailei Ma, Fan Lu, Yuxin Guo, Yifei Zhang, Wei Chen, Qingpei Guo, Yujun Shen, and Zheng-Jun Zha. Lotlip: Improving language-image pre-training for long text understanding. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 64996–65019. Curran Associates, Inc., 2024. doi: 10.52202/079017-2075. https://proceedings.neurips.cc/paper_files/paper/2024/file/77828623211df05497ce3658300dafd9-Paper-Conference.pdf.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829, 2024.

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19163–19173, June 2022.

Kaicheng Yang, Jiankang Deng, Xiang An, Jiawei Li, Ziyong Feng, Jia Guo, Jing Yang, and Tongliang Liu. Alip: Adaptive language-image pre-training with synthetic caption. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2922–2931, 2023a.

Yifan Yang, Weiquan Huang, Yixuan Wei, Houwen Peng, Xinyang Jiang, Huiqiang Jiang, Fangyun Wei, Yin Wang, Han Hu, Lili Qiu, and Yuqing Yang. Attentive mask clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2771–2781, October 2023b.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. In *International Conference on Learning Representations*, 2022. https://openreview.net/forum?id=cpDhcsEDC2.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. https://openreview.net/forum?id=Ee277P3AYC.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision, 2021. https://arxiv.org/abs/2111.11432.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations*, 2023. https://openreview.net/forum?id=KRLUvxh8uaX.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas 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, June 2022.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986, October 2023.

Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip. In *European Conference on Computer Vision*, pages 310–325. Springer, 2024.

Weiheng Zhao, Zilong Huang, Jiashi Feng, and Xinggang Wang. SuperCLIP: CLIP with simple classification supervision. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. https://openreview.net/forum?id=EeIEvZlmVg.

Kecheng Zheng, Yifei Zhang, Wei Wu, Fan Lu, Shuailei Ma, Xin Jin, Wei Chen, and Yujun Shen. Dreamlip: Language-image pre-training with long captions. In *European Conference on Computer Vision*, pages 73–90. Springer, 2024.

Jinghao Zhou, Li Dong, Zhe Gan, Lijuan Wang, and Furu Wei. Non-contrastive learning meets language-image pre-training. *Computer Vision and Pattern Recognition (CVPR)*, 2023.

Jialong Zuo, Jiahao Hong, Feng Zhang, Changqian Yu, Hanyu Zhou, Changxin Gao, Nong Sang, and Jingdong Wang. Plip: Language-image pre-training for person representation learning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 45666–45702. Curran Associates, Inc., 2024. doi: 10.52202/079017-1452. https://proceedings.neurips.cc/paper_files/paper/2024/file/510ad3018bbdc5b6e3b10646e2e35771-Paper-Conference.pdf.