MLIP: Efficient Multi-Perspective Language-Image Pretraining with Exhaustive Data Utilization

Yu Zhang [1] **Qi Zhang** [*1] **Zixuan Gong** [*1] **Yiwei Shi** [2] **Yepeng Liu** [3] **Duoqian Miao** [1]
Yang Liu [4] **Ke Liu** [5] **Kun Yi** [6] **Wei Fan** [7] **Liang Hu** [1] **Changwei Wang** [8]

Abstract

Contrastive Language-Image Pretraining (CLIP) has achieved remarkable success, leading to rapid advancements in multimodal studies. However, CLIP faces a notable challenge in terms of inefficient data utilization. It relies on a single contrastive supervision for each image-text pair during representation learning, disregarding a substantial amount of valuable information that could offer richer supervision. Additionally, the retention of non-informative tokens leads to increased computational demands and time costs, particularly in CLIP's ViT image encoder. To address these issues, we propose **M**ulti-Perspective **L**anguage-**I**mage **P**retraining (**MLIP**). In MLIP, we leverage the frequency transform's sensitivity to both high and low-frequency variations, which complements the spatial domain's sensitivity limited to low-frequency variations only. By incorporating frequency transforms and token-level alignment, we expand CILP's single supervision into multi-domain and multi-level supervision, enabling a more thorough exploration of informative image features. Additionally, we introduce a token merging method guided by comprehensive semantics from the frequency and spatial domains. This allows us to merge tokens to multi-granularity tokens with a controllable compression rate to accelerate CLIP. Extensive experiments validate the effectiveness of our design.

[*]Equal contribution [1]Tongji University [2]University of Bristol [3]University of Florida [4]National University of Defense Technology [5]National Clinical Research Center for Mental Disorders & National Center for Mental Disorders, Beijing Anding Hospital, Capital Medical University [6]Beijing Institute of Technology [7]University of Oxford [8]Institute of Automation, Chinese Academy of Sciences. Correspondence to: Duoqian Miao <dqmiao@tongji.edu.cn>, Ke Liu <liuke0222@126.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

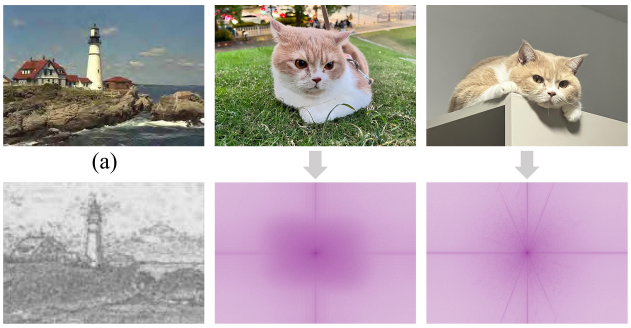

Figure 1. (a) A distorted image. (b) An objective error map. The house and the sky regions are easily observable, and those on textural regions (e.g. rocks) are less noticeable, i.e., HVS is much more sensitive to the low-frequency variations than the high-frequency variations. (c) The original images and spectrums of the same lying cat in different scenes. It shows spectrum is pretty effective in extracting and differentiating features such as the complexity and noise of a scene (the high-frequency variations).

1. Introduction

In recent times, multimodal study (Xu et al., 2024a; Zhang et al., 2024; Xu et al., 2024b; Lin et al., 2023) has gained significant popularity, leading to rapid advancements in Vision-Language Pre-training (VLP). One noteworthy development is Contrastive Language-Image Pretraining (CLIP) (Radford et al., 2021), which has demonstrated remarkable performance across a range of downstream tasks. CLIP achieves this by training on 400 million image-text pairs with a contrastive mechanism to effectively bring the representations of intra-pairs closer together while pushing apart those of inter-pairs. However, upon closer examination, it is evident that CLIP encounters a significant obstacle in terms of inefficient data utilization. For example, only one contrastive supervision is utilized for each pair during the forward process, thereby leaving substantial uni-modal and cross-modal information untapped, which can potentially enhance representation. Additionally, the presence of non-informative tokens leads to increased computational requirements and time costs, especially in CLIP's image encoder, i.e., a Vision Transformer (ViT) (Dosovitskiy et al., 2020). This additional workload hampers cross-modality alignment and significantly slows down the overall training speed of CLIP.

There has been a surge in studies focusing on the development of data-efficient CLIP-like models. Prevailing approaches include self-supervision or image enhancement techniques to increase the diversity of supervision (Mu et al., 2022; Li et al., 2021; Lee et al., 2022) or probe token-level alignments to refine feature learning (Yao et al., 2021; Zou et al., 2022). Although these approaches have shown promising results, they primarily focus on enhancing feature learning in the single spatial domain. However, it is crucial to recognize that a 2D image signal contains a wealth of additional important features that can be extracted in the frequency domain. CNNs and ViTs, which primarily operate in the spatial domain, are devised to mimic the human visual system (HVS) (Kim & Lee, 2017). However, HVS exhibits varying sensitivity to different frequency components, as illustrated in Figure 1(a) and (b). Fortunately, frequency transformation techniques can naturally differentiate and isolate less sensitive frequency components, cf. Figure 1(c). This persuades us that image signals in the frequency domain offer valuable information, potentially promoting multi-domain supervision to enhance data efficiency.

Moreover, the additional merits of frequency analysis, including computation efficiency, energy compacting, and a global view (Yi et al., 2023a;b), further encourage us to contemplate data utilization from multiple perspectives to improve CLIP accuracy and efficiency. i) *Multi-domain*: how to benefit from complementary supervision in the frequency domain to enhance the spatial domain? ii) *Multi-level*: How to introduce token-level alignments to promote instance-level alignments with fine-grained representation learning? Note that there is a fundamental distinction between frequency tokens and spatial tokens: Frequency tokens are numerous in quantity but in relatively low-frequency semantics, while spatial tokens are reduced in number, akin to text tokens, but carry high-frequency semantics. The distinction necessitates the multi-level alignments to alleviate semantics mismatch and suggests a strict one-to-one alignment at the spatial token level and a loose one-to-many alignment at the token-level alignment with text, respectively. iii) *Multi-granularity*: Does global frequency effectively guide to merge tokens at multi-granularity tokens for computational efficiency? Existing studies to accelerating CLIP, like FLIP (Li et al., 2023) and A-CLIP (Yang et al., 2023b), utilize masking image patches to achieve reduced accelerating CLIP in ViT, however, suffering from information loss from unreliable masking (Liang et al., 2022). Figure 2(a) and (b) show that tokens exhibit higher semantic similarity in deeper layers where reducing the token number should be more reliable. Frequency information, especially high frequency, effectively complements HVS with comprehensive semantics, potentially enhancing reliable token merging.

In light of the above discussion, we present a novel **M**ulti-Perspective **L**anguage-**I**mage **P**retraining (**MLIP**). The approach utilizes frequency analysis to ground the model and offers joint spatial-frequency token alignment, enabling exhaustive data utilization. Specifically, we propose splitting the image encoder into two stages: the Frequency Stage and the Spatial Stage, to provide frequency and spatial features of images, respectively. The Frequency Stage leverages the Discrete Fourier Transform (DFT) to efficiently mix tokens, allowing the image encoder to capture high-frequency variation features such as textures. On the other hand, the Spatial Stage utilizes the attention mechanism to learn local or global spatial features, including shape and position. These two stages generate a frequency embedding and a spatial embedding for each image, which are then used in contrastive learning alongside text embeddings at the instance level. Additionally, MLIP aligns tokens from the Frequency Stage and Spatial Stage with the text tokens at the token level, employing a loose one-to-one and a strict one-to-many matching respectively to fine-grain representation learning. To accelerate MLIP, we employ a token merging method guided by frequency-spatial supervision to reduce the token number at a controlled compression rate. Briefly, we devise a light *Guide* module to process the low-resolution counterpart of the image and send its class token and global features to cross-attention layers of the Spatial Stage, to enhance image tokens. This injection of high-level semantic information provides additional reliable guidance for selecting similar image tokens to merge.

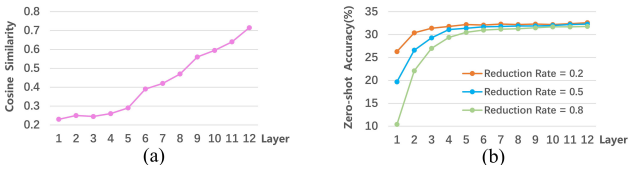

Figure 2. The observation of performing similarity calculation and reduction operations on tokens on ImageNet. (a) Average token similarity in each layer of CLIP-ViT-B/32. (b) Zero-shot accuracy of random token reductions on different layers.

Our contributions can be summarized as follows:

- Multi-domain supervision: We introduce the frequency transform into the CLIP paradigm for the first time, breaking the previous practice of only mining information from image-text pairs in the spatial domain. This enables a more thorough exploration of image features, leading to a more powerful CLIP paradigm.
- Merging leads to acceleration: We utilize token merging to reduce the token number while maintaining information integrity. As far as we know, this is the first application of token merging in the CLIP paradigm.
- Multi-perspective optimization: MLIP optimizes CLIP from multi-domain, multi-level and multi-granularity perspectives. Extensive experiments validate the effectiveness of our methods, demonstrating efficiency in both data utilization and model training.

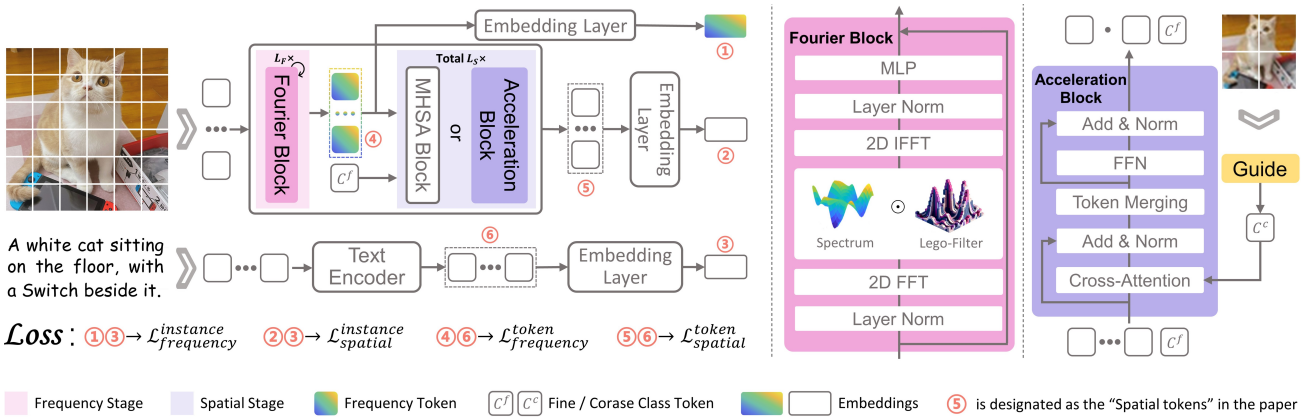

Figure 3. The overall framework of MLIP. We modify the image encoder, and related design lies in the colorful areas and indexes.

2. Related Work

CLIP is a simple yet powerful paradigm of representation learning. It is widely applied to various downstream tasks (Wan et al., 2024; Liu et al., 2024; Gong et al., 2024). However, CLIP is in inefficient data utilization. Several studies are attempting to address the issue. For instance, SLIP (Mu et al., 2022) and DeCLIP (Li et al., 2021) expand contrastive supervision; FILIP (Yao et al., 2021) explores token-level alignment; CLIP-PSD (Andonian et al., 2022) and SoftCLIP (Gao et al., 2023) soften one-hot labels; FLIP (Li et al., 2023) achieves acceleration by randomly masking patches, and A-CLIP (Yang et al., 2023b) further masks patches with weak semantic correlation to speed up. Recent studies combine various aforementioned techniques to explore new approaches (Gao et al., 2022; Yang et al., 2023a; Dong et al., 2023; Geng et al., 2023). Diverging from the above works, we solve the issue from new perspectives: frequency transforming and token merging.

Frequency transforming plays a crucial role in signal processing and has shown surprising performance when applied to various fields of deep learning (Zheng et al., 2021; Cao et al., 2020; Qin et al., 2021). These studies utilize Fourier Transform (FT) in converting signals from the spatial or time domain to the frequency domain. FNet (Lee-Thorp et al., 2021), as the first work to explore the application of frequency transforming to Transformer (Vaswani et al., 2017), finds that FT can replace the Self-Attention layer to achieve fast token mixing. GFNet (Rao et al., 2021), applying Fast Fourier Transform (FFT) to ViT, improves the image classification performance of ViT. Subsequent studies, such as AFNO (Guibas et al., 2021) and AFFNet (Huang et al., 2023), delve deeper into the application of FFT to ViT. Currently, frequency transforming is rarely discussed in VLP. We consider introducing frequency transforming in CLIP to achieve efficiency. However, unlike above works, we use both frequency and spatial domain features of images and have made modifications to the process of transforming.

3. Methodology

In this section, we first introduce some CLIP preliminaries and the overall MLIP loss. Sequentially present our three methods: supervision expansion via frequency transforming, joint spatial-frequency token alignment, and acceleration via token merging. Figure 3 shows the overall framework.

3.1. CLIP Preliminaries and MLIP Overall Loss

For a batch of N image-text pairs $\{(I_j, T_j)\}_{j=1}^{N}$, I_j and T_j are the image and text of the j-th pair. y_j and z_j represent the normalized embeddings of I_j and T_j, respectively, obtained from the image encoder and text encoder. The InfoNCE loss (Oord et al., 2018) is used for contrastive learning, and the loss for image-to-text can be computed as:

$$\mathcal{L}_{IT} = \frac{1}{N}\sum_{j=1}^{N}\log\frac{\exp(sim(y_j, z_k)/\tau)}{\sum_{k=1}^{N}\exp(sim(y_j, z_k)/\tau)}, \quad (1)$$

where τ is a learnable temperature hyper-parameter, it is typically set to 0.07. The function $sim(,)$ is used to compute the similarity by dot product, and the text-to-image loss L_{TI} can be obtained as Equation 1. Therefore, the overall loss of CLIP is calculated through $\mathcal{L}_{CLIP} = \frac{1}{2}\mathcal{L}_{IT} + \frac{1}{2}\mathcal{L}_{TI}$.

Similarly, the overall loss of MLIP is donated as:

$$\mathcal{L}_{MLIP} = \alpha\mathcal{L}^{ins}_{fre} + \beta\mathcal{L}^{ins}_{spa} + \gamma\mathcal{L}^{tok}_{fre} + \delta\mathcal{L}^{tok}_{spa}, \quad (2)$$

where ins and tok respectively represent instance-level alignment and token-level alignment, while fre and spa refer to aligning text tokens with frequency tokens and spatial tokens of the image, respectively. We set mixing coefficients α, β, γ and δ to balance multiple losses.

3.2. Supervision Expansion via Frequency Transforming

Frequency Stage. Frequency Stage contains $L_F\times$ Fourier Blocks for transforming tokens into the frequency domain for mixing. For an image Y with the resolution of $H\times$

W, we first split it into $h \times w$ non-overlapping patches, h and w represent the number of patches split in the H and W directions, respectively. After patch embedding, the collection of these C-dimensional tokens, which serve as the input to the image encoder, is denoted as $\mathbf{y}(p,q), 1 \leq p \leq h, 1 \leq q \leq w$.

The spectrum is the representation of a signal in the frequency domain. Therefore, to process a discrete signal in the frequency domain, it's essential first to obtain its spectrum through Discrete Fourier Transform (DFT). The separability of 2D DFT indicates that, for a given 2D image signal $f(m,n), 1 \leq m \leq \mathcal{M}, 1 \leq n \leq \mathcal{N}$, its 2D DFT can be separated into two 1D DFTs: first perform a 1D DFT of length $\mathcal{N}$ along one dimension of the variable n, then take the computed result and perform a 1D DFT of length $\mathcal{M}$ along the other dimension of the variable m to obtain the spectrum $F^{2D}(u,v)$ of the 2D signal:

$$F^{1D}(m,v) = \sum_{n=1}^{\mathcal{N}} f(m,n) e^{-i2\pi vn/\mathcal{N}}, \tag{3}$$

$$F^{2D}(u,v) = \sum_{m=1}^{\mathcal{M}} F^{1D}(m,v) e^{-i2\pi um/\mathcal{M}}. \tag{4}$$

Further, as for $\mathbf{y}(p,q)$, $1 \leq p \leq h$,$1 \leq q \leq w$, we obtain:

$$Y(u,v) = \sum_{p=1}^{h} \sum_{q=1}^{w} \mathbf{y}(p,q) e^{-i2\pi(up/h + vq/w)}, \tag{5}$$

where, i is the imaginary unit, and $Y(u,v)$ is the spectrum of the 2D signal at (ω_u, ω_v). $\omega_u = 2\pi u/h$ and $\omega_v = 2\pi v/w$ correspond to the discrete frequency components in the orthogonal dimensions. Here, we adopt the standard FFT algorithm (Cooley & Tukey, 1965) to calculate the DFT.

DFT and its inverse process are lossless. Therefore, based on the fundamental properties of DFT, given a 1D spectrum $F^{1D}(n)$, we can reconstruct the original signal $f(n)$ by Inverse DFT (IDFT):

$$f(n) = \frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} F^{1D}(v) e^{i2\pi vn/\mathcal{N}}. \tag{6}$$

Consequently, We can reconstruct the original 2D signal $\mathbf{y}(p,q)$ from the 2D spectrum $Y(u,v)$:

$$\mathbf{y}(p,q) = \frac{1}{hw} \sum_{u=1}^{h} \sum_{v=1}^{w} Y(u,v) e^{i2\pi(up/h + vq/w)}. \tag{7}$$

It is noteworthy that $\mathbf{y}(p,q) \in \mathbb{R}$. According to the fundamental properties of DFT, the spectrum $Y(u,v)$ obtained by 2D DFT is conjugate symmetric about the origin, which means $Y(u,v) = Y^*(-u,-v)$. Moreover, considering the periodicity of DFT, which states $Y(u,v) = Y(u+P, v+q)$, one can derive that $Y(p-u, q-v) = Y^*(p,q)$. This implies that half of the spectrum $Y(u,v)$ can be used to reconstruct the complete 2D signal $\mathbf{y}(p,q)$. Therefore, we adopt a smaller equivalent spectrum $Y'(u,v)$ to replace $Y(u,v)$ for signal reconstruction:

$$Y' = Y(:, 1 : w/2). \tag{8}$$

Overall, we define $\mathcal{F}(\cdot)$ as the 2D DFT, for token collection $\mathbf{y} \in \mathbb{R}^{h \times w \times C}$, the spectrum of $\mathbf{y}$ can be represented as:

$$Y = \mathcal{F}(\mathbf{y}) \in \mathbb{C}^{h \times w \times C}, \tag{9}$$

where Y is a complex tensor. In order to reduce computation, we take half of the spectrum Y, denoted as $Y' \in \mathbb{C}^{h \times \frac{w}{2} \times C}$, to effectively reconstruct the original signal. Then we introduce the Lego-Filter to modulate the spectrum to the Y'. We utilize $X = [x_1, x_2, ..., x_{\aleph}]$ to represent the Lego-Filter, where $\aleph$ is the number of piece filters in the Lego-Filter:

$$\hat{Y} = 2 \sum_{j=1}^{\aleph} \frac{1}{hw} |Y'|^2 \odot x_j cos((2j-1)\pi/2\aleph), \tag{10}$$

where $\odot$ is the element-wise multiplication (Hadamard product), $|Y'|^2$ is the power spectrum of Y', which smooths the spectrum, highlighting the main components of the spectrum and facilitating the subsequent learning. $cos((2j-1)\pi/2\aleph)$ compacts better energy and can aggregate the more important information in a 2D signal.

Next, we utilize IFFT $\mathcal{F}^{-1}(\cdot)$ to construct and update the token collection $\mathbf{y}$:

$$\mathbf{y} \leftarrow \mathcal{F}^{-1}(\hat{Y}). \tag{11}$$

Finally, the tokens within the token collection, after transforming through the Frequency Stage, are termed *frequency tokens* $\mathbf{y}^{fre}$. After being embedded, frequency tokens generate an instance-level frequency embedding y^{fre}, which is used for alignment with the instance-level embedding z coming from the text encoder.

Spatial Stage. Based on $L_S\times$ MHSA (Multi-Head Self-Attention)Blocks, some MHSA Blocks are replaced with Acceleration Blocks to form the Spatial Stage. Spatial Stage takes frequency tokens as input and outputs *spatial tokens* $\mathbf{y}^{spa}$, also using the attention mechanism for token interaction. Similarly, spatial tokens $\mathbf{y}^{spa}$ produce an instance-level spatial embedding y^{spa}, which aligns with z.

Instance-level alignment loss. In MLIP, instance-level alignment losses include the alignment loss of (image) frequency-text $\mathcal{L}^{ins}_{fre}$ and (image) spatial-text $\mathcal{L}^{ins}_{spa}$. Taking

$\mathcal{L}_{fre}^{ins}$ as an example, it can be represented as:

$$\mathcal{L}_{fre}^{ins} = \frac{1}{2N}\sum_{j=1}^{N}\log\frac{\exp(sim(y_j^{fre}, z_k)/\tau)}{\sum_{k=1}^{N}\exp(sim(y_j^{fre}, z_k)/\tau)} + \frac{1}{2N}\sum_{j=1}^{N}\log\frac{\exp(sim(z_j, y_k^{fre})/\tau)}{\sum_{k=1}^{N}\exp(sim(z_j, y_k^{fre})/\tau)}. \quad (12)$$

3.3. Joint Spatial-Frequency Token Alignment

CLIP only summarizes global visual and textual presentations (instance-level alignment), consequently overlooking a substantial amount of fine-grained information. However, token-level alignment can utilize fine-grained information to assist the model in learning more detailed features. In MLIP, token-level alignment is also categorized into (image) frequency-text and (image) spatial-text.

Frequency-text. We still utilize frequency tokens for token-level alignment of image-text, adopting the original approach (Yao et al., 2021). Specifically, We denote l_1 and l_2 as the number of frequency tokens $\mathbf{y}^{fre}$ and the number of non-padded tokens z involved in late interaction, respectively. The corresponding embeddings are a and b. We require that through calculating cosine similarities, each token involved in token-level alignment finds its most similar cross-modal token. For instance, for the r-th frequency token $\mathbf{y}_r^{fre}$, we compute the similarity of its embedding a^r with all text token embeddings $\{b^s\}_{s=1}^{l_2}$, and select the highest one to represent the matching completion of $\mathbf{y}_r^{fre}$:

$$\max_{1\le s\le l_2}\frac{a^r\cdot b^s}{\|a^r\|_2\,\|b^s\|_2}. \quad (13)$$

Subsequently, we use the average of matchings to represent the token-level alignment ϖ^{IT} from image to text:

$$\varpi^{IT} = \frac{1}{l_1}\sum_{r=1}^{l1}\frac{a^r\cdot b^{s_r^{IT}}}{\|a^r\|_2\,\|b^{s_r^{IT}}\|_2}, \quad (14)$$

where $s_r^{IT} = \max_{1\le s\le l_2}\frac{a^r\cdot b^s}{\|a^r\|_2\,\|b^s\|_2}$. Therefore, for N image-text pairs, we can formulate the frequency token-level alignment loss from image to text $\mathcal{L}_{fre-IT}^{tok}$. Similarly, we can derive the corresponding loss from text to image $\mathcal{L}_{fre-TI}^{tok}$. Assigning a mixing coefficient of $1/2$ to each loss, we get the full loss $\mathcal{L}_{fre}^{tok}$, denoted as:

$$\mathcal{L}_{fre}^{tok} = -\frac{1}{2N}\sum_{j=1}^{N}\varpi_j^{IT} - \frac{1}{2N}\sum_{j=1}^{N}\varpi_j^{TI}. \quad (15)$$

Spatial-text. Spatial tokens perform a spatial-text token-level alignment. We define the number of spatial tokens as l_3, and their corresponding embeddings as $\{c^t\}_{t=1}^{l_3}$. Due to token merging, many spatial tokens are the products of merging previously similar tokens, which have a higher level of semantic concepts compared to frequency tokens, and their number is close to that of test tokens. Therefore, we adopt a one-to-one alignment scheme, which means that every token, during cross-modal alignment, should not only look for the most similar token but also consider the cross-modal alignment of other intra-modal tokens to avoid conflicts. Hence, we can view this as a bipartite matching problem. Further considering that the similarity between cross-modal tokens could naturally serve as a weight, we model it as a maximum weight bipartite matching problem and solve it using the Kuhn-Munkres (KM) algorithm.

For embedding collections $\{b^s\}_{s=1}^{l_2}$ and $\{c^t\}_{t=1}^{l_3}$, set $l^* = \max(l^2, l^3)$. To explain more succinctly, we use the embedding's index number within the collection to represent itself. Therefore, we construct two sets $S = \{1, 2, \cdots, l^*\}$ and $T = \{1, 2, \cdots, l^*\}$ to represent the embedding collections of text tokens and spatial tokens, respectively. If $l^2 < l^*$ or $l^3 < l^*$, add incremental elements until reaching l^* to complete the construction. Set up a weight matrix $W \in \mathbb{R}^{l^*\times l^*}$, whose element $\mathbf{w}(s,t)$ is defined as follows:

$$\mathbf{w}(s,t) = \begin{cases} 0 & \text{if } s \ge l^2 \text{ or } t \ge l^3 \\ \frac{b^s\cdot c^t}{\|b^s\|_2\,\|c^t\|_2} & \text{otherwise} \end{cases} \quad (16)$$

We initialize the index (denoting the embedding) : $L_b(s) = max_{t\in T}\mathbf{w}(s,t), \forall s \in S$ and $L_c(t) = 0, \forall t \in T$. We introduce M to record the matching scheme, setting $M[t] = -1, \forall t \in T$ and adopt $match$ to record whether a match has already been made, setting $match[s] = 0, \forall s \in S$. Set $\hat{S}$, $\hat{T}$, and array $Slack[\,]$ are used for adjustments and updates during matching. Algorithm 1 shows the core process.

Algorithm 1 KM for matching text and spatial embedding

Input: S, T, W, initialized settings (M, $match$, L_b, L_c)
Output: matching scheme M

```
for s ← 1 to l* do
    Ŝ ← {s}, T̂ ← ∅
    for t ← 1 to l* do
        Slack[t] ← L_b(s) + L_c(t) − w(s,t)
    while match[s] = 0 do
        if ∃t ∈ (T \ T̂) : L_b(s) + L_c(t) = w(s,t) then
            if M[t] = −1 then
                M[t] ← s, match[s] ← 1, break while
            S ← S ∪ {M[t]}, T ← T ∪ {t}
        else
            t* ← arg min_{t∉T̂} Slack[t], Δ ← Slack[t*]
            update: ∀j ∈ Ŝ, l_b(j) ← l_b(j) − Δ
                    ∀k ∈ T̂, l_c(k) ← l_c(k) + Δ
                    ∀t ∉ T̂, Slack[t] ← Slack[t] − Δ
return M
```

After obtaining the matching scheme M, we can use M to calculate the final spatial-text token-level alignment loss

$\mathcal{L}_{spa}^{tok}$. For N image-text pairs, $\mathcal{L}_{spa}^{tok}$ can be calculated by the following equation:

$$\mathcal{L}_{spa}^{tok} = -\frac{1}{Nmin(l^2, l^3)} \sum_{j=1}^{N} \sum_{t=1}^{l^*} \mathbf{w}_j(M[t], t). \quad (17)$$

3.4. Acceleration via Token Merging

Constitution of Spatial Stage. Accelerating training is realized in Acceleration Blocks, precisely owing to the Token Merging module within them. The key to acceleration lies in selecting tokens for merging to reduce their quantity. As shown in Figure 2(b), performing the same token reduction operation in different blocks leads to huge performance differences. Therefore, deciding when to reduce the token number, or in other words, how to place Acceleration Blocks in the Spatial Stage, is vital. Due to the input of the Spatial Stage, frequency tokens, do not interact in the spatial domain. To enable these tokens to capture spatial features (such as shape, position, etc.), and establish local and global relationships, we set a few MHSA Blocks in the early Spatial Stage for token interaction. The subsequent setting combines MHSA Blocks and Acceleration Blocks. Moreover, a (fine) class token C^f is appended to record the importance of each token for later guiding the merging.

Guide and cross-attention. Due to the computational complexity of self-attention being quadratically related to the token number, MHSA Blocks should be minimized. However, this might lead to insufficient interaction between tokens before merging. A lack of global understanding may result in suboptimal merging. To resolve this paradox, we equipped Acceleration Blocks with a *Guide*. Simply put, Guide is a lightweight pre-trained ViT. We split the low-resolution counterpart of the image into fewer patches as the Guide's input, so the Guide could learn more global features of the image with less computational cost, which has higher semantic concepts. Then we input its (coarse) class token C^c into the cross-attention layer, where it interacts with original image tokens and (fine) class token. This process injects higher semantics into them, allowing them to acquire more global features and directional guides during their merging.

Token Merging. In the Token Merging module, we propose a controllable compression ratio token merging strategy: 1) set a compression rate $\mathcal{C}$, 2) sort all tokens from largest to smallest, based on the ranking of each token's attention score in (fine) class token C^c, 3) take the last $2\mathcal{C}$ tokens for merging. The merging process is shown in Figure 4.

Our method is inspired by ToMe (Bolya et al., 2022) but differs in three main aspects: (1) Our method not only uses the fine features of the original image to determine to merge but also adds the features with higher semantics brought by Guide in the cross-attention layer to jointly determine the tokens involved in merging. (2) ToMe first performs matching, and then selects tokens for merging; while our method first selects tokens for merging, and then performs matching, this can help us achieve a controllable compression rate. (3) In our method, tokens have undergone frequency transforming, possess complementary frequency features that can better facilitate token merging.

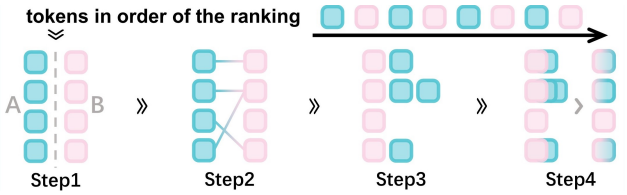

Figure 4. The process of Token Merging: Step1. Divide tokens at odd positions into set A and those at even positions into set B. Step2. Find the most similar token in B for each token in A by calculating cosine similarity. Step3. Put similar tokens together to complete the match. Step4. Merge the similar tokens by weights.

4. Experiment

4.1. Experiment Setup

Pre-training datasets. To enable a fair comparison with as many methods as possible, we use YFCC15M (Cui et al., 2022), which is commonly adopted by many methods, for the pre-training of MLIP. We also adopt CC3M (Sharma et al., 2018) and CC12M (Changpinyo et al., 2021b) for pre-training to verify that MLIP is data-efficient across multiple datasets of different scales.

Table 1. Comparison against CLIP baselines with zero-shot (ZS) and linear probing (LP) classification Top-1 accuracy on ImageNet, $^{\diamondsuit}$Reported in (Cui et al., 2022), $^{\heartsuit}$Our implementation.

Method	Baesd Encoder	ZS TOP-1	LP TOP-1
CLIP	ViT-B / 32	32.8$^{\diamondsuit}$	62.4$^{\heartsuit}$
SLIP	ViT-B / 32	34.3$^{\diamondsuit}$	67.5$^{\heartsuit}$
FILIP	ViT-B / 32	39.5$^{\diamondsuit}$	—
DeCLIP	ViT-B / 32	43.2$^{\diamondsuit}$	70.4$^{\heartsuit}$
MLIP	ViT-B / 32	41.1	70.2
CLIP	ViT-B / 16	39.0$^{\heartsuit}$	64.7$^{\heartsuit}$
SLIP	ViT-B / 16	43.2$^{\heartsuit}$	72.3$^{\heartsuit}$
DeCLIP	ViT-B / 16	47.9$^{\heartsuit}$	77.8$^{\heartsuit}$
MLIP	ViT-B / 16	46.3	77.1

Implementation details. We utilize two ViT variants, ViT-B/32 and ViT-B/16, as the basis for constructing the image encoders, corresponding to MLIP-ViT-B/32 and MLIP-ViT-B/16, respectively. More details are in Appendix C. The image resolution is 224×224, and the Guide employs DeiT-Tiny (Wang et al., 2023) based on MAE pre-training to process the counterpart with a resolution of 64×64. The text encoder follows the original design of CLIP. Drawing

Table 2. Zero-shot and linear-probe classification Top-1 accuracy (%) on 10 smaller datasets, based on variant ViT-B/32, against CLIP baselines, C10/100/F101/FLOW/SUN/DTD/CAL/AIR is CIFAR10/CIFAR-100/Food101/Flowers/SUN397/Describable Textures/Caltech-101/Aircraft. AVG is average accuracy across 10 datasets, AVG (+ImageNet) is average accuracy across 11 datasets, including ImageNet. LS denotes label smoothing. **Black text** indicates the best performance, while underlined text indicates the second-best performance.

Method	C10	C100	F101	PETS	FLOW	SUN	CARS	DTD	CAL	AIR	AVG	AVG (+ImageNet)
zero-shot classification:												
CLIP	63.7	33.2	34.6	20.1	50.1	35.7	2.6	15.5	59.9	1.2	31.7	31.8
SLIP	50.7	25.5	33.3	23.5	49.0	34.7	2.8	14.4	59.9	1.7	29.5	30.0
FILIP	65.5	33.5	43.1	24.1	52.7	50.7	3.3	24.3	68.8	**3.9**	37.0	37.2
DeCLIP	66.7	38.7	**52.5**	**33.8**	60.8	50.3	**3.8**	27.7	74.1	2.1	41.1	41.3
MLIP	65.8	37.0	48.5	31.7	64.7	52.9	3.0	36.8	75.9	3.1	41.9	41.9
MLIP+LS	**67.1**	**38.9**	49.6	32.5	**65.3**	**53.5**	3.3	**37.8**	**76.1**	3.2	**42.7**	**42.8**
linear-probe classification:												
CLIP	86.5	64.7	69.2	64.6	90.6	66.0	24.9	61.3	79.1	23.1	63.0	63.2
SLIP	86.4	65.1	73.9	69.5	89.2	70.6	27.0	64.1	82.8	25.7	65.4	65.6
DeCLIP	89.2	69.0	**75.4**	72.2	94.4	71.6	**31.0**	68.8	87.9	27.6	68.7	68.8
MLIP	88.6	67.0	72.3	69.9	96.7	75.1	26.8	83.3	92.2	26.1	69.8	69.8
MLIP+LS	**90.3**	**70.7**	73.4	**72.5**	**97.0**	**75.6**	27.9	**84.6**	**92.5**	**28.2**	**71.3**	**71.3**

from experience, we set the mixing coefficients α, β, γ, and δ to 0.15, 0.65, 0.1, and 0.1, respectively. We train all models for 32 epochs with the same hyperparameter setting. More details are in Appendix D.

Downstream tasks for evaluation. We evaluate MLIP on three downstream tasks: zero-shot and linear-probe image classification, and zero-shot image-text retrieval. For image classification, we perform experiments on ImageNet (Deng et al., 2009) and 10 other smaller datasets, a total of 11 datasets. For image-text retrieval, we set experiments on Flickr30K (Hodosh et al., 2013) and MS-COCO (Chen et al., 2015). More information is in Appendix B.

4.2. Main Results

Zero-shot and linear-probe image classification. Zero-shot and linear-probe classification results of MLIP on ImageNet are shown in Table 1. It can be seen that MLIP's accuracy surpasses that of CLIP and CLIP-like baselines. However, it's worth noting that DeCLIP outperforms our MLIP, primarily because Nearest-Neighbor Supervision (essentially a kind of label smoothing) significantly enhances performance, and (Gao et al., 2023) holds the same view. Therefore, we further conduct experiments with label smoothing and find that MLIP could exceed the performance of DeCLIP. More details are in Section 4.3. In Table 2, we also present the zero-shot and linear-probe classification results on other datasets, where our MLIP is still competitive overall. Especially when combined with label smoothing, MLIP demonstrates a significant performance advantage. Notably, on datasets like FLOW and DTD, which contain more scenes, edges, and textures, MLIP's superiority is particularly evident. This aligns well with our expectations when introducing frequency domain transformation.

Zero-shot image-text retrieval. We evaluate MLIP's zero-shot image-text retrieval performance in Table 4, indicating that MLIP outperforms CLIP or CLIP-like methods. We can find notable improvements in most recall@1 metrics, which we attribute to the increased supervision and finer alignment. Additionally, unlike classification, image-text retrieval involves processing more complex and noisy image information, such as scene details, an area where frequency transforming excels, hence yielding better results.

4.3. Ablation Study

In this section, we investigate the effectiveness of every design in MLIP. Unless specifically stated, all experiments use the MLIP-ViT-B/16 model pre-trained for 25 epochs on CC3M to evaluate its zero-shot classification on ImageNet and zero-shot image-text retrieval on MS-COCO.

Data efficiency across multiple datasets of different scales. To confirm MLIP's data efficiency across varying dataset scales, we test its performance on various pretraining datasets and datasets with fewer image-text pairs. Table 3 shows MLIP's data efficiency on different scales.

Table 3. Data efficiency experiments on datasets of different scales.

Method	Pretraing Dataset	Baesd Encoder	ZS TOP-1
CLIP$^\heartsuit$	CC3M	ViT-B / 16	16.3
MLIP	CC3M	ViT-B / 16	**18.4(+2.1)**
MLIP	CC3M(95%)	ViT-B / 16	17.9(+1.6)
CLIP$^\heartsuit$	CC12M	ViT-B / 16	30.4
MLIP	CC12M	ViT-B / 16	**33.2(+2.8)**
MLIP	CC12M(90%)	ViT-B / 16	31.1(+0.7)
CLIP$^\heartsuit$	YFCC15M	ViT-B / 16	37.6
MLIP	YFCC15M	ViT-B / 16	**42.9(+5.3)**
MLIP	YFCC15M(90%)	ViT-B / 16	39.4(+1.8)

Table 4. Zero-shot image-text retrieval results on Flickr30k and MS-COCO.

	Image-to-text retrieval						Text-to-image retrieval					
	Flickr30k			MS-COCO			Flickr30k			MS-COCO		
Method	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
CLIP-ViT-B / 32	34.9	63.9	75.9	20.8	43.9	55.7	23.4	47.2	58.9	13.0	31.7	42.7
SLIP-ViT-B / 32	47.8	76.5	85.9	27.7	52.6	63.9	32.3	58.7	58.8	18.2	39.2	51.0
DeCLIP-ViT-B / 32	51.4	80.2	88.9	28.3	53.2	64.5	34.3	60.3	70.7	18.4	39.6	51.4
UniCLIP-ViT-B / 32	52.3	81.6	89.0	32.0	57.7	69.2	34.8	62.0	72.0	20.2	43.2	54.4
MLIP-ViT-B / 32	**53.1**	**84.0**	**93.8**	**32.6**	**59.1**	**71.3**	**35.2**	**62.9**	**74.7**	**20.4**	**43.7**	**56.2**
MLIP-ViT-B / 16	56.5	88.7	98.5	34.7	64.4	75.8	37.3	64.8	78.1	24.5	47.7	62.1

Influence of label softening. We employ Nearest-Neighbor Supervision (NNS) (Li et al., 2021) and the label smoothing method (LS) from PyramidCLIP (Gao et al., 2022) to explore how label softening enhances MLIP's performance. As shown in Table 5, we can see that label softening significantly boosts MLIP's performance. Therefore, without the label softening trick, MLIP outperforms DeCLIP.

Table 5. The influence of label softening on MLIP's performance.

Method	ZS TOP-1	LP TOP-1
DeCLIP-ViT-B/32	43.2	70.4
MLIP-ViT-B/32	41.1	70.2
MLIP-ViT-B/32 + NNS	43.1	71.6
MLIP-ViT-B/32 + LS	**43.6**	**71.7**
DeCLIP-ViT-B/16	47.9	77.8
MLIP-ViT-B/16	46.3	77.1
MLIP-ViT-B/16 + NNS	48.2	78.5
MLIP-ViT-B/16 + LS	**48.6**	**78.9**

Effectiveness of frequency transforming and token-level alignment. To verify the effectiveness of these methods, we conduct an ablation experiment as shown in Table 6. We can observe that both frequency transforming and token-level alignment markedly enhance the performance. This indicates that expanding the supervision of learning representation through these two methods is quite effective.

Table 6. Ablation study on the effects of frequency transforming (Fre-T) and token-level alignment (Tok-A). 'I2T' and 'T2I' mean image-to-text and text-to-image, respectively. [1]Only use the one-to-many matching strategy. [2]Use both one-to-many and one-to-one matching strategies, i.e., our token-level alignment method.

Method	I2T R@1	T2I R@1	ZS TOP-1
CLIP (ViT-B / 16)	10.4	6.6	16.3
+ Fre-T	12.7	8.0	17.6
+ Tok-A[1]	13.1	8.2	17.9
+ Fre-T + Tok-A[2]	**14.8**	**9.5**	**18.7**

Influence of matching strategies. We design a set of experiments to analyze the influence of using different matching strategies on performance. As shown in Table 7, the MLIP's matching strategy realizes the best result, underscoring its importance in token-level alignment.

Table 7. Ablation study on the influence of matching strategies. 'o-to-o' and 'o-to-m' are one-to-one and one-to-many, respectively.

Fre-Text	Spa-Text	I2T R@1	T2I R@1	ZS TOP-1
o-to-o	o-to-m	13.0	8.2	17.8
o-to-o	o-to-o	13.4	8.5	18.0
o-to-m	o-to-m	14.1	8.9	18.3
o-to-m	o-to-o	**14.6**	**9.3**	**18.4**

Effectiveness of Guide. Table 8 shows the results with and without Guide, demonstrating that Guide is essential for better performance. Furthermore, combining data from Table 6, it's evident that the performance loss due to token merging operations can be largely compensated for by Guide. Additionally, even when compared to other baselines without the pre-trained Guide, MLIP remains competitive. This indicates that MLIP's performance gains are also attributed to other well-designed components beyond Guide.

Table 8. Ablation study on the effectiveness of Guide.

Method	I2T R@1	T2I R@1	ZS TOP-1
CLIP	14.1	8.9	18.0
SLIP	14.1	8.9	18.0
DeCLIP w/o NNS	14.1	8.9	18.0
MLIP w/o Guide	14.1	8.9	18.0
MLIP	**14.6**	**9.3**	**18.4**

Comparison of computational efficiency. Since MLIP also reduces the computational cost by token merging, we compare its balance of performance and computation with other CLIP-like models, as shown in Table 9. We measure the amount of computation required for each model by metric GFLOPs. In the experiment, despite MLIP obtaining sub-optimal performance, it achieves the best computation-performance balance, suggesting MLIP's efficiency is more comprehensive.

Table 9. Comparison on computational efficiency. Metric ZS TOP-1/GFLOPs is used to represent computation-performance balance.

Method	ZS TOP-1	GFLOPs	ZS TOP-1/GFLOPs
CLIP$^{\heartsuit}$	16.3	19.78	0.82
SLIP$^{\heartsuit}$	16.9	22.61	0.74
DeCLIP$^{\heartsuit}$	**18.7**	26.29	0.71
MLIP	18.4	**19.54**	**0.94**

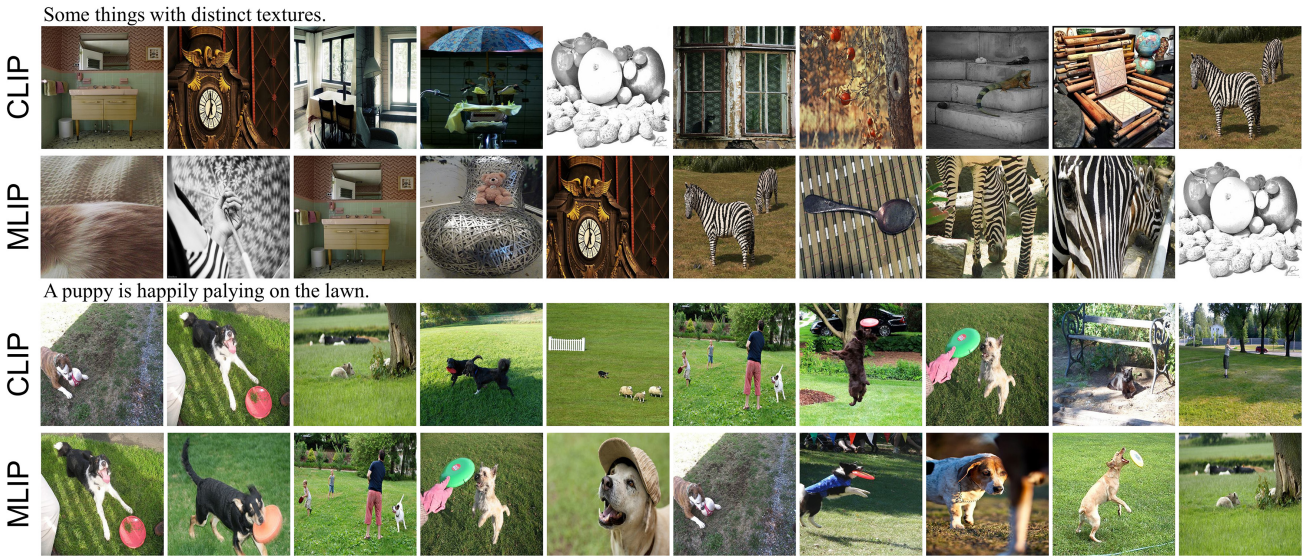

Figure 5. Text-to-image top 10 retrieval results on MS-COCO.

Comparison of training wall-clock time. Reporting the wall-clock time for training more effectively demonstrates computational efficiency. Table 10 reports the wall-clock times for training MLIP, CLIP, DeCLIP, SLIP. Although the overall training process of MLIP is about 17 minutes slower than that of the fastest CLIP in terms of Wall-clock time, the GPU Hours are still fewer. Given the better performance of MLIP, we compare its balance of performance and training time with other CLIP-like models, it is evident that MLIP achieves the best balance between training time and performance.

Table 10. Comparison on training wall-clock time. Metric ZS TOP-1/WCT (wall-clock time) represents training-performance balance.

Method	ZS TOP-1	GPU Hours	ZS TOP-1/WCT
CLIP	16.3	**361**	1.44
SLIP	16.9	488	1.11
DeCLIP	**18.7**	545	1.09
MLIP	370	**1.59**	**0.94**

4.4. Visualization

Text-to-image retrieval. Figure 5 shows 2 sets of top 10 retrieval results on MS-COCO. In the first set, it can be seen that MLIP has a more global and distinct ability to recognize textures. And in another, MLIP also shows better retrieval.

Lego-Filter. Figure 6 visualizes Lego-Filter of MLIP-ViT-B/32, showing that Lego-Filter effectively captures both high and low-frequency variations.

Embedding space. We utilize t-SNE visualization to compare the embedding spaces of CLIP and MLIP on the CIFAR-10 dataset. From Figure 7, it's evident that MLIP, on the same pretrain dataset, exhibits better separation between samples of different classes, indicating that MLIP indeed improves data utilization and learns better representations.

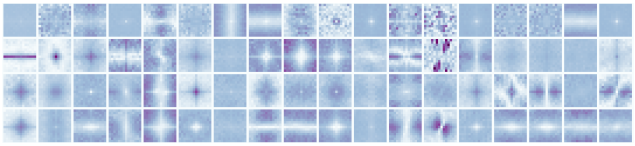

Figure 6. Visualization of Lego-Filter.

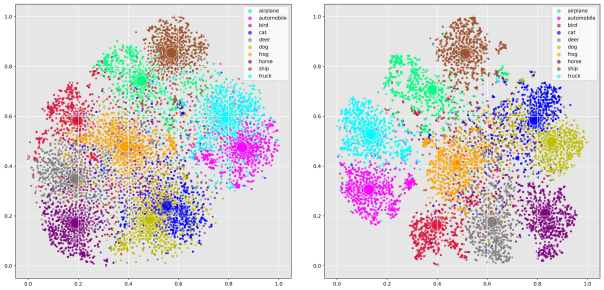

Figure 7. Visualization of embedding space.

5. Conclusion

In this article, we propose MLIP, a framework to develop an efficient CLIP via exhaustive data utilization in multi-perspective. MLIP introduces frequency transforming and alignments at both the token level and instance level to expand the supervision of learning representation in the image encoder. Additionally, MLIP also incorporates a modified token merging method, reducing the token number in the image encoder and accelerating the overall training. Extensive experiments validate the effectiveness of our design, and we hope our work can inspire its community. We also discuss the limitation of MLIP in Appendix E.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (No. 2022YFB3104700), the National Natural Science Foundation of China (No. 62376198, No. 62006172, No. 62106091, No. 62076182 and No. 62163016), the Shandong Provincial Natural Science Foundation (No. ZR2021MF054), the Jiangxi "Double Thousand Plan" and the Jiangxi Provincial Natural Science Foundation (No. 20212ACB202001). The authors would like to thank Jun Wang, Xuerong Zhao, Yayue Tan and Jiu for inspirational suggestions and helpful assistance.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Andonian, A., Chen, S., and Hamid, R. Robust cross-modal representation learning with progressive self-distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16430–16441, 2022.

Bolya, D., Fu, C.-Y., Dai, X., Zhang, P., Feichtenhofer, C., and Hoffman, J. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pp. 446–461. Springer, 2014.

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., Tong, Y., Xu, B., Bai, J., Tong, J., et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778, 2020.

Changpinyo, S., Sharma, P., Ding, N., and Soricut, R. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021a.

Changpinyo, S., Sharma, P., Ding, N., and Soricut, R. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3558–3568, 2021b.

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., and Qiao, Y. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Cooley, J. W. and Tukey, J. W. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

Cui, Y., Zhao, L., Liang, F., Li, Y., and Shao, J. Democratizing contrastive language-image pre-training: A clip benchmark of data, model, and supervision. *arXiv preprint arXiv:2203.05796*, 2022.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dong, X., Bao, J., Zheng, Y., Zhang, T., Chen, D., Yang, H., Zeng, M., Zhang, W., Yuan, L., Chen, D., et al. Maskclip: Masked self-distillation advances contrastive language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10995–11005, 2023.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pp. 178–178. IEEE, 2004.

Gao, Y., Liu, J., Xu, Z., Zhang, J., Li, K., Ji, R., and Shen, C. Pyramidclip: Hierarchical feature alignment for vision-language model pretraining. *Advances in neural information processing systems*, 35:35959–35970, 2022.

Gao, Y., Liu, J., Xu, Z., Wu, T., Liu, W., Yang, J., Li, K., and Sun, X. Softclip: Softer cross-modal alignment makes clip stronger. *arXiv preprint arXiv:2303.17561*, 2023.

Geng, S., Yuan, J., Tian, Y., Chen, Y., and Zhang, Y. Hiclip: Contrastive language-image pretraining with hierarchy-aware attention. *arXiv preprint arXiv:2303.02995*, 2023.

Gong, Z., Zhang, Q., Bao, G., Zhu, L., Zhang, Y., Liu, K., Hu, L., and Miao, D. Lite-mind: Towards efficient and robust brain representation network, 2024.

Guibas, J., Mardani, M., Li, Z., Tao, A., Anandkumar, A., and Catanzaro, B. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*, 2021.

Hodosh, M., Young, P., and Hockenmaier, J. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47: 853–899, 2013.

Huang, Z., Zhang, Z., Lan, C., Zha, Z.-J., Lu, Y., and Guo, B. Adaptive frequency filters as efficient global token mixers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6049–6059, 2023.

Kim, J. and Lee, S. Deep learning of human visual sensitivity in image quality assessment framework. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1676–1684, 2017.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lee, J., Kim, J., Shon, H., Kim, B., Kim, S. H., Lee, H., and Kim, J. Uniclip: Unified framework for contrastive language-image pre-training. *Advances in Neural Information Processing Systems*, 35:1008–1019, 2022.

Lee-Thorp, J., Ainslie, J., Eckstein, I., and Ontanon, S. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021.

Li, Y., Liang, F., Zhao, L., Cui, Y., Ouyang, W., Shao, J., Yu, F., and Yan, J. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021.

Li, Y., Mao, H., Girshick, R., and He, K. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pp. 280–296. Springer, 2022.

Li, Y., Fan, H., Hu, R., Feichtenhofer, C., and He, K. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23390–23400, 2023.

Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., and Xie, P. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

Lin, J., Gao, H., Xu, R., Wang, C., Guo, L., and Xu, S. The development of llms for embodied navigation. *arXiv preprint arXiv:2311.00530*, 2023.

Liu, C., Wan, Z., Ouyang, C., Shah, A., Bai, W., and Arcucci, R. Zero-shot ecg classification with multimodal learning and test-time clinical knowledge enhancement, 2024.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

Mu, N., Kirillov, A., Wagner, D., and Xie, S. Slip: Self-supervision meets language-image pre-training. In *European Conference on Computer Vision*, pp. 529–544. Springer, 2022.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

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

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Qin, Z., Zhang, P., Wu, F., and Li, X. Fcanet: Frequency channel attention networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 783–792, 2021.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rao, Y., Zhao, W., Zhu, Z., Lu, J., and Zhou, J. Global filter networks for image classification. *Advances in neural information processing systems*, 34:980–993, 2021.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, 2018.

Soliman, S. S. and Srinath, M. D. Continuous and discrete signals and systems. *Englewood Cliffs*, 1990.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

Tsai, Y.-H. H., Bai, S., Yamada, M., Morency, L.-P., and Salakhutdinov, R. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wan, Z., Liu, C., Zhang, M., Fu, J., Wang, B., Cheng, S., Ma, L., Quilodrán-Casas, C., and Arcucci, R. Medunic: Unifying cross-lingual medical vision-language pre-training by diminishing bias. *Advances in Neural Information Processing Systems*, 36, 2024.

Wang, D., Zhang, Q., Xu, Y., Zhang, J., Du, B., Tao, D., and Zhang, L. Advancing plain vision transformer toward remote sensing foundation model. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–15, 2022.

Wang, S., Gao, J., Li, Z., Zhang, X., and Hu, W. A closer look at self-supervised lightweight vision transformers. In *International Conference on Machine Learning*, pp. 35624–35641. PMLR, 2023.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Xu, R., Zhang, J., Sun, J., Wang, C., Wu, Y., Xu, S., Meng, W., and Zhang, X. Mrftrans: Multimodal representation fusion transformer for monocular 3d semantic scene completion. *Information Fusion*, pp. 102493, 2024a.

Xu, S., Chen, S., Xu, R., Wang, C., Lu, P., and Guo, L. Local feature matching using deep learning: A survey. *arXiv preprint arXiv:2401.17592*, 2024b.

Yang, K., Deng, J., An, X., Li, J., Feng, Z., Guo, J., Yang, J., and Liu, T. Alip: Adaptive language-image pre-training with synthetic caption. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2922–2931, 2023a.

Yang, Y., Huang, W., Wei, Y., Peng, H., Jiang, X., Jiang, H., Wei, F., Wang, Y., Hu, H., Qiu, L., et al. Attentive mask clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2771–2781, 2023b.

Yao, L., Huang, R., Hou, L., Lu, G., Niu, M., Xu, H., Liang, X., Li, Z., Jiang, X., and Xu, C. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021.

Yi, K., Zhang, Q., Fan, W., Wang, S., Wang, P., He, H., Lian, D., An, N., Cao, L., and Niu, Z. Frequency-domain mlps are more effective learners in time series forecasting. *CoRR*, abs/2311.06184, 2023a.

Yi, K., Zhang, Q., Wang, S., He, H., Long, G., and Niu, Z. Neural time series analysis with fourier transform: A survey. *CoRR*, abs/2302.02173, 2023b.

Zhang, B., Tian, Z., Tang, Q., Chu, X., Wei, X., Shen, C., et al. Segvit: Semantic segmentation with plain vision transformers. *Advances in Neural Information Processing Systems*, 35:4971–4982, 2022.

Zhang, B., Liu, L., Phan, M. H., Tian, Z., Shen, C., and Liu, Y. Segvitv2: Exploring efficient and continual semantic segmentation with plain vision transformers. *arXiv preprint arXiv:2306.06289*, 2023.

Zhang, J., Wang, K., Xu, R., Zhou, G., Hong, Y., Fang, X., Wu, Q., Zhang, Z., and He, W. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024.

Zheng, B., Yuan, S., Yan, C., Tian, X., Zhang, J., Sun, Y., Liu, L., Leonardis, A., and Slabaugh, G. Learning frequency domain priors for image demoireing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (11):7705–7717, 2021.

Zou, X., Wu, C., Cheng, L., and Wang, Z. Tokenflow: Rethinking fine-grained cross-modal alignment in vision-language retrieval. *arXiv preprint arXiv:2209.13822*, 2022.

A. Additional Fourier Theory Analysis

A.1. Discrete Fourier Transform

The Discrete Fourier Transform (DFT) can be understood through various approaches. In this context, we explore how DFT is developed from the conventional Fourier Transform (FT), which is primarily applicable to continuous signals. The FT transforms a continuous-time signal into its frequency domain representation, acting as a broader application of the Fourier series concept. In essence, the Fourier transform for a signal $f(t)$ is defined as follows:

$$F^{1D}(i\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt = F^{1D}[f(t)]. \tag{18}$$

The Inverse Fourier Transform (IFT) bears a resemblance in structure to the Fourier Transform:

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F^{1D}(i\omega)e^{i\omega t}d\omega. \tag{19}$$

The equations for the FT and the IFT provide insight into the duality characteristic of the FT, which bridges the time and frequency domains. This duality principle suggests that characteristics observed in the time domain find analogous expressions in the frequency domain. Among the many attributes of the Fourier transform, several fundamental ones include the transformation of a unit impulse function $\delta(t)$ (commonly referred to as the Dirac delta function), which is

$$F^{1D}(\delta(t)) = \int_{-\infty}^{\infty} \delta(t)e^{-i\omega t}dt = \int_{0-}^{0+} \delta(t)dt = 1, \tag{20}$$

and the time-shifting property:

$$F^{1D}(\delta(t-t_0)) = \int_{-\infty}^{\infty} f(t-t_0)e^{-i\omega t}dt = e^{-i\omega t_0}\int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt = e^{-i\omega t_0}F^{1D}(i\omega). \tag{21}$$

In practical scenarios, continuous signals are seldom directly dealt with. Instead, a common approach involves sampling the continuous signal to generate a sequence of discrete signals. This sampling process is typically carried out through a series of unit impulse functions.

$$f_s(t) = f(t)\sum_{n=-\infty}^{\infty}\delta(t-nT_s) = \sum_{n=-\infty}^{\infty} f(nT_s)\delta(t-nT_s), \tag{22}$$

When we take the Fourier Transform (FT) of the sampled signal $f_s(t)$ with a sampling interval T and then apply Equation (A.3) and Equation (A.4), we obtain

$$F_s^{1D}(i\omega) = \sum_{n=-\infty}^{\infty} f(nT_s)e^{-in\omega T_s}. \tag{23}$$

In the provided equation, it is evident that $F_s^{1D}(i\omega)$ exhibits periodic behavior with a fundamental period of $2\pi/T_s$. In fact, there is always a direct correspondence between discrete signals in one domain and periodic signals in the other domain. Typically, we prefer to work with a normalized frequency, denoted as $\omega \leftarrow \omega T_s$, which results in $F_s^{1D}(i\omega)$ having an exact period of 2π. We can also represent $f(n)$ as $f(nT_s)$, defining it as the sequence of discrete signals, and subsequently derive the Discrete-Time Fourier Transform (DTFT):

$$F^{1D}(e^{i\omega}) = \sum_{n=-\infty}^{\infty} f(n)e^{-in\omega}. \tag{24}$$

When the discrete signal $f(n)$ has a finite length $\mathcal{N}$, which is a common scenario in digital signal processing, the DTFT can be expressed as follows:

$$F^{1D}(e^{i\omega}) = \sum_{n=1}^{\mathcal{N}} f(n)e^{-in\omega}. \tag{25}$$

where the non-zero terms of the discrete signal $f(n)$ are assumed to lie in the range $[1, N]$ without loss of generality, the DTFT $F^{1D}(e^{i\omega})$ is indeed a continuous function of ω. You can obtain a sequence $F^{1D}[v]$ by sampling $F^{1D}(e^{i\omega})$ at discrete frequencies $\omega_v = \frac{2\pi v}{\mathcal{N}}$, resulting in:

$$F^{1D}(v) = F^{1D}(e^{i\omega})|_{\omega=2\pi n/N} = \sum_{n=1}^{\mathcal{N}} f[n]e^{-i(2\pi/\mathcal{N})kn}, \tag{26}$$

Indeed, the extension from the 1D Discrete Fourier Transform (DFT) to the 2D DFT is straightforward. The 2D DFT can be viewed as applying the 1D DFT independently to the two dimensions of the data. Specifically, the 2D DFT of a signal or image $f(m, n)$ is given by:

$$F^{2D}(u, v) = \sum_{m=1}^{\mathcal{M}} \sum_{n=1}^{\mathcal{N}} f(m, n)e^{-i2\pi\left(\frac{um}{\mathcal{M}} + \frac{vn}{\mathcal{N}}\right)}, \tag{27}$$

Therefore, we can obtain Equation 5.

A.2. The equivalence between self-attention and frequency-domain computation

A.2.1. (SELF-ATTENTION)

Consider an input tensor X (for clearer expression, we choose X different from F), where $x_n \in \mathbb{R}^d$ signifies the n-th element in the sequence, and N represents the sequence's length.

Definition 1 Self-Attention The mechanism of self-attention, denoted as Self-Att : $\mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d}$, is articulated through a kernel integration approach as found in (Kovachki et al., 2021; Guibas et al., 2021; Tsai et al., 2019):

$$\text{Self-Att} = \text{softmax}\left(\frac{(XW_q)(XW_k)^\top}{\sqrt{d}}\right)XW_v \tag{28}$$

Here, K is defined as the softmax-normalized score array of size $N \times N$: $K = \text{softmax}\left(\frac{(XW_q)(XW_k)^\top}{\sqrt{d}}\right)$. The operation of self-attention is then equivalent to an asymmetric kernel $\kappa : [N] \times [N] \to \mathbb{R}^{d\times d}$, with each entry $\kappa[s, t]$ constructed as $K[s, t] \otimes W_v^\top$. Thus, we interpret self-attention as a sum over this kernel.

$$\text{Self-Att}(X)[s] = \sum_{t=1}^{N} \kappa[s, t] \cdot X[t], \quad \forall s \in [N] \tag{29}$$

Expanding upon the notion of kernel summation, we incorporate the concept of continuous kernel integration. Within this framework, the tensor X encapsulates a spatial function over the space $X = (D, \mathbb{R}^d)$, with D being a subset of $\mathbb{R}^2$.

$$\text{Self-Att}(X)[s] = \mathcal{K}(X)(s) = \int_D \kappa(s, t)X(t)\, dt, \quad \forall s \in D \tag{30}$$

For any continuous element X within D, we define the kernel integral operator $\mathcal{K} : (\mathbb{R}^d, D) \to (D, \mathbb{R}^d)$ in the following manner:

Definition 2 Kernel Integral. The kernel integral operator, denoted as $\mathcal{K}$, maps pairs of the domain D and the Euclidean space $\mathbb{R}^d$ into themselves, symbolically represented as $\mathcal{K} : (D, \mathbb{R}^d) \to (D, \mathbb{R}^d)$. This operator is formally defined for all s within the domain D as follows:

$$\mathcal{K}(X)(s) = \int_D \kappa(s,t) \cdot X(t)\, dt, \ \ \forall s \in \mathcal{D} \tag{31}$$

where κ is a continuous function that takes two arguments from the domain D and returns a $d \times d$ real matrix. In the context where Green's kernel is applied, $\kappa(s,t)$ simplifies to $\kappa(s-t)$, which characterizes a specific instance of this kernel function.

Definition 3 Frequency-Domain Analysis

The convolution theorem, as stated by (Soliman & Srinath, 1990), posits that spatial domain convolution is functionally analogous to frequency domain multiplication. Hence, for any continuous input X belonging to domain $\mathcal{D}$, the kernel integration as outlined by (Guibas et al., 2021) can be expressed as:

$$\mathcal{K}(X)(s) = \mathcal{F}^{-1}(\mathcal{J}(k) \cdot \mathcal{F}(X))(s), \ \ \forall s \in \mathcal{D} \tag{32}$$

Here, the symbol $\cdot$ denotes element-wise multiplication.

To summarize, leveraging computations in the frequency domain to restructure self-attention mechanisms offers a method that is both effective and theoretically sound. This approach also provides a theoretical foundation for the practicality and validity of our proposed method.

B. Additional Information of Dataset

B.1. Pre-training Dataset

YFCC15M. RFCC15M (Radford et al., 2021) is a curated subset of YFCC-100M dataset (Thomee et al., 2016). It is specifically filtered to include only those images that have English titles or descriptions. The dataset comprises 14,829,396 images, each accompanied by natural language captions.

CC3M. The CC3M dataset (Sharma et al., 2018) is a large-scale collection of over 3 million images accompanied by natural-language captions, providing a diverse resource for automatic image captioning tasks. The images and captions in CC3M are sourced from the web, particularly from the Alt-text HTML attributes of images, offering a wide array of styles and contexts.

CC12M. The CC12M dataset (Changpinyo et al., 2021a) is a collection of approximately 12 million image-text pairs, significantly larger and more diverse than its predecessor, CC3M. Designed specifically for vision-and-language pre-training.

B.2. Downstream Task Dataset

B.2.1. Image Classification

In this section, we detail the ten datasets utilized for the classification task in our experiments, each of which is concisely summarized in Table 11.

ImageNet. The ImageNet dataset (Deng et al., 2009) consists of millions of labeled images across a wide variety of categories. It is structured according to the WordNet hierarchy, with each node of the hierarchy represented by hundreds of images.

Caltech-101. The Caltech 101 dataset (Fei-Fei et al., 2004) consists of approximately 9,000 images divided into 101 distinct object classes, along with an extra category for background/clutter. The dataset encompasses a diverse collection of objects in each class, with image counts per category ranging roughly from 40 to 800. For oriented items like airplanes and motorcycles, images have been flipped to align from left to right, and images of vertically structured objects like buildings have been rotated to be off-axis aligned.

CIFAR-10. The CIFAR-10 (Krizhevsky et al., 2009) dataset is a collection of 60,000 color images, each 32x32 pixels in size. These images are categorized into 10 different classes, with each class representing distinct objects.

CIFAR-100. The CIFAR-100 (Krizhevsky et al., 2009) dataset is similar to the CIFAR-10 dataset but with a higher granularity in classification. It contains 60,000 color images, each 32x32 pixels, divided into 100 classes. Each class has 600 images, providing a more challenging and diverse dataset for image recognition tasks.

Describable Texture. The Describable Textures dataset (Cimpoi et al., 2014) is a specialized collection of images focused on textures. It consists of texture images that are categorized based on describable attributes, rather than the object or material they represent. The dataset includes a wide range of textures, such as patterns found in nature, fabrics, or man-made materials. Each texture is annotated with a set of human-describable attributes, like "bubbly," "cracked," or "woven.". Its emphasis on describable attributes rather than just material types allows algorithms to better understand and interpret the various characteristics that make up a texture.

Food-101. The Food-101 dataset (Bossard et al., 2014) is specifically designed for food recognition tasks. It contains 101,000 images, divided into 101 food categories, with each category containing 1,000 images.

Oxford -IIIT Pets. The Oxford -IIIT Pets (Parkhi et al., 2012) dataset contains images of pets, specifically focused on cats and dogs. It includes 37 different breeds of cats and dogs, with roughly 200 images for each breed, totaling around 7,400 images. The images are varied in terms of scale, pose, and lighting.

Oxford Flowers. The Oxford Flowers dataset (Nilsback & Zisserman, 2008) consists of 8,189 images, each depicting one of 102 flower species commonly found in the United Kingdom. Each class (flower species) in the dataset is represented by between 40 and 258 images, ensuring a variety of examples for each type of flower. The images in the dataset vary in terms of scale, pose, and lighting conditions, which makes the dataset challenging for algorithms to process. In addition, there are categories that have large variations within the category and several very similar categories. The dataset is visualized using isomap with shape and color features.

SUN397. The SUN397 dataset (Xiao et al., 2010) is a comprehensive collection of images specifically designed for scene recognition and classification tasks in computer vision. This dataset is part of the Scene UNderstanding (SUN) database, which is focused on providing a rich variety of scene categories. The SUN397 dataset contains approximately 108,000 images that span 397 different scene categories, offering a remarkably broad spectrum of environments.

FGVC Aircraft. The FGVC Aircraft dataset (Maji et al., 2013) is a specialized image dataset used in fine-grained visual categorization (FGVC) tasks, particularly focusing on aircraft recognition and classification. The dataset contains 10,200 images of aircraft, with 100 images for each of 102 different aircraft model variants, most of which are airplanes.

Table 11. Overview of used datasets in our classification experiments.

Dataset	Abbreviation	Classes	Train Size	Test Size	Evaluation Metric
ImageNet	IN	1000	1,281,167	50,000	accuracy
Caltech-101	CAL	102	3,060	6,085	mean per class
CIFAR-10	C10	10	50,000	10,000	accuracy
CIFAR-100	C100	100	50,000	10,000	accuracy
Describable Textures	DTD	47	3,760	1,880	accuracy
Food-101	F101	101	75,750	25,250	accuracy
Oxford-IIIT Pets	PETS	37	3,680	3,669	mean per class
Oxford Flowers 102	FLOW	102	2,040	6,149	mean per class
SUN397	SUN	397	19,850	19,850	accuracy
FGVC Aircraft	AIR	100	6,667	3,333	mean per class

B.2.2. IMAGE-TEXT RETRIEVAL

Flickr30K. The Flickr30K dataset (Hodosh et al., 2013) is mainly used for image captioning and visual question-answering tasks. This dataset comprises approximately 30,000 images, and each image in the Flickr30k dataset is accompanied by five different textual descriptions (captions).

MS-COCO. The MS-COCO dataset (Chen et al., 2015) contains over 200,000 images with a diverse set of everyday scenes that include complex backgrounds and a variety of objects. It is richly annotated with details and multiple object labels.

C. Additional Information of MLIP Architecher

We follow the same architecture design as CLIP. Table 12 and Table 13 showcase the based and specific structures of the MLIP series.

Table 12. The architecture parameters for based models of MLIP.

Model	Embedding dimension	Input resolution	Image Encoder			Text Encoder		
			#layers	width	#heads	#layers	width	#heads
MLIP-ViT-B	512	224× 224	12	768	12	12	512	8
MLIP-ViT-L	512	224× 224	24	1024	16	12	768	12

Table 13. The specific structural parameters of the image encoder in MLIP.

Method	Frequency Stage	Spatial Stage	Acceleration Block	Compression rate
MLIP-ViT-B / 32	$[1, 2, 3, 4]$	$[5, \cdots, 12]$	$[9, 11]$	$[0.7, 0.7]$
MLIP-ViT-B / 16	$[1, 2, 3, 4]$	$[5, \cdots, 12]$	$[9, 11]$	$[0.5, 0.5]$
MLIP-ViT-L / 14	$[1, \cdots, 8]$	$[9, \cdots, 24]$	$[13, 16, 19, 22]$	$[0.7, 0.7, 0.65, 0.65]$

D. Additional Implementation Details

We used the AdamW optimizer (Loshchilov & Hutter, 2017), with a weight decay rate of 0.1 for pre-training. For the first 2000 warm-up iterations, the learning rate increases linearly to the peak value, then decays to 0 following a cosine strategy (Loshchilov & Hutter, 2016). We set the batch size to 4096 and conducted all experiments on 32 V100 GPUs. Additionally, to save GPU memory, we use automatic mixed-precision (Micikevicius et al., 2017) for training. Unless specifically stated, ablation studies were conducted with training 25 epochs on CC3M. Moreover, we also briefly tested the performance of MLIP-L/32 constructed based on the ViT variant of ViT-L/32, but differently, we only trained it on YFCC15M for 8 epochs.

E. Limitation

Structure limitation. Given that Transformers, as opposed to CNNs, can establish long-range dependencies and dominate multimodal applications, this paper only investigates MLIP based on the ViT structure.

Experiment limitation. Due to limited computational resources, we are unable to extend MLIP to larger-scale models such as ViT-Large for complete experimentation. However, similar works (Li et al., 2021; Lee et al., 2022; Gao et al., 2022; 2023; Geng et al., 2023; Dong et al., 2023; Yang et al., 2023a), have not expanded to ViT-Large either.

Comparison limitation. Since most similar works are not open-source, and downstream tasks, pre-training datasets, and training strategies vary widely, it is challenging for us to conduct a broader fair comparison.

Performance limitation. To be honest, MLIP approaches but not surpass state-of-the-art (SOTA) performance because we prioritize a more comprehensive efficiency. Some experiments show that if focusing solely on enhancing performance, with more refined optimization, it might be possible to achieve SOTA, which we will pursue in our future research.

Transfer limitation. Due to token merging, applying the image encoder of MLIP to other dense vision downstream tasks such as segmentation poses some challenges. However, studies (Li et al., 2022; Zhang et al., 2022; 2023; Chen et al., 2022; Wang et al., 2022) demonstrate that for such plain ViT models, modifications can still be made to perform dense downstream tasks like detection and segmentation.