FineLIP: Extending CLIP's Reach via Fine-Grained Alignment with Longer Text Inputs

CVPR2025

Contrastive Language-Image Pre-training (CLIP):

Contrastive Language-Image Pre-training (CLIP) has set a high standard for vision-language models (VLMs). However, it identifies two primary limitations that hinder its performance in complex scenarios requiring detailed understanding:

1.Text Length Limitation:

Popular CLIP models have a text encoder limited to processing only 77 text tokens. This prevents them from effectively encoding longer, detail-rich captions.

2.Lack of Fine-Grained Understanding:

CLIP is primarily trained on image-short caption pairs. Its contrastive loss focuses on aligning global image and text features,thereby overlooking the correspondence between local, fine-grained visual and textual information. This limits the model's capability for tasks requiring understanding of intricate attributes, spatial relationships, etc.

Existing works often address only one issue: for instance, Long-CLIP and TULIP extend the context window for longer text but still rely on global feature alignment; while methods like FILIP and SPARC introduce fine-grained token-level alignment, they are designed for short captions and typically refine only visual representations. Therefore, the core objective of FineLIP is to simultaneously enable the handling of longer text inputs and achieve finer-grained cross-modal alignment within the CLIP framework.

Method

FineLIP extends a pre-trained CLIP model with three core components, as illustrated in its workflow: first stretching positional embeddings, then aggregating local image and text tokens separately, and finally enforcing fine-grained cross-modal alignment.

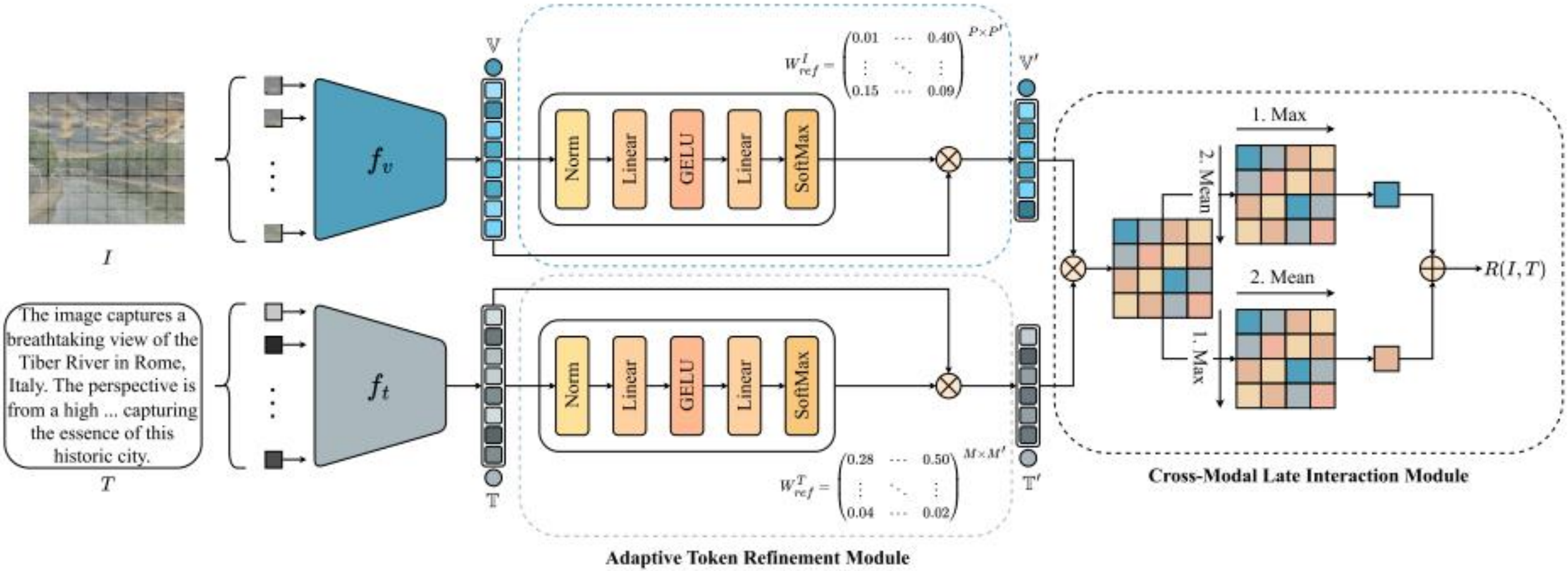
Input:

Image I
Text T

Process:

CLIP Encoder
↓
Token Refinement（ATRM）
↓
Fine-grained Alignment（CLIM）
↓
Loss（Triplet Loss）

Method


I
The image captures a breathtaking view of the Tiber River in Rome, Italy. The perspective is from a high ... capturing the essence of this historic city.
T
f_v
f_t
V
T
Norm
Linear
GELU
Linear
SoftMax
$W^I_{ref} = \begin{pmatrix} 0.01 & \cdots & 0.40 \\ \vdots & \ddots & \vdots \\ 0.15 & \cdots & 0.09 \end{pmatrix}^{P \times P'}$
$W^T_{ref} = \begin{pmatrix} 0.28 & \cdots & 0.50 \\ \vdots & \ddots & \vdots \\ 0.04 & \cdots & 0.02 \end{pmatrix}^{M \times M'}$
V'
T'
Adaptive Token Refinement Module
1. Max
2. Mean
2. Mean
1. Max
$R(I,T)$
Cross-Modal Late Interaction Module

Positional Embedding Stretching

Purpose: To break CLIP's native 77-token limit, enabling the model to process longer descriptions (extended to 248 tokens in the paper).

Method: Employs "knowledge-preserved stretching." It preserves the well-trained positional embeddings of the first 20 tokens. For positions beyond 20, an adaptive interpolation method stretches the embeddings to 4x their original length. This approach leverages CLIP's pretrained knowledge while minimizing disruption to its established cross-modal alignment.

ATRM（Adaptive Token Refinement Module）

Purpose: In vision-language tasks, individual image or text tokens can often be ambiguous due to a lack of context. Before applying fine-grained token-to-token cross-modal alignment, refining the tokens to reduce ambiguity and increase information density is crucial. Unlike some works that employ token selection, ATRM adopts a token aggregation strategy, aiming to minimize information loss while producing a more discrimina -tive set of tokens.

Method: The ATRM module takes the original token sets from the image and text encoders as input:
Visual Features: $\mathbb{V} = \{v_{cls}, v_1, \ldots, v_P\} \in \mathbb{R}^{(P+1)\times d}$
Textual Features: $\mathbb{T} = \{t_0, t_1, \ldots, t_{eos}, \ldots, t_{247}\} \in \mathbb{R}^{248\times d}$
ATRM performs dynamic aggregation separately for the visual and textual branches (the ATRM modules share the same architecture but use separate parameters for each branch)

Input: Let $X = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^{N \times d}$ represent N input tokens for a modality.

Aggregation Operation: ATRM aggregates N tokens into N' refined tokens ($N' < N$) via a learnable weight matrix $W_{ref} \in \mathbb{R}^{N' \times N}$

Aggregation Ratio: N'/N controls the degree of compression, set to 0.2 by default in the paper.

Constraint: $\sum_{i=1}^{N'} (W_{ref})_{i,j} = 1$ ensuring the refinement process is differentiable for end-to-end learning.

Weight Matrix Calculation: W_{ref} is computed in a manner inspired by self-attention but optimized for efficiency:

$$W_{ref} = SoftMax(\frac{W_q \sigma(X W_k)^T}{\tau})$$

$W_k \in \mathbb{R}^{d \times d_k}$ and $W_q \in \mathbb{R}^{N' \times d_k}$ are trainable projection matrices ($d_k < d$)

σ is a nonlinear function (GELU).

τ is a learnable temperature parameter that encourages sparse attention, making aggregation more focused.

Output: After ATRM processing, visual features are updated to $\mathbb{V}' = \{v_{cls}, v'_1, \ldots, v'_{P'}\}$ and textual features to $\mathbb{T}' = \{t'_1, \ldots, t'_{M'}, t_{eos}\}$. Here, P' and M' are the numbers of aggregated visual and textual tokens, respectively. Global tokens are not aggregated but retained in the sets. Padding tokens in the text are also ignored.

Design Features and Advantages:

Dual-Modality Aggregation: Unlike existing methods that only refine the visual branch, ATRM aggregates tokens from both image and text, addressing ambiguity in both modalities.

Avoiding Information Loss: By using aggregation instead of selection, it preserves information from all original tokens and compresses it into more condensed representations through learning.

Improving Alignment Efficiency: Reducing the number of tokens ($N'<N$) not only lowers the computational cost of subsequent fine-grained alignment but also results in each aggregated token containing richer semantic information, facilitating more accurate cross-modal correspondence.

CLIM（Cross-Modal Late Interaction）

Purpose: In vision-language tasks, nuanced correspondences (e.g., spatial, semantic relationships) often exist between images and text. Traditional CLIP-style coarse-grained alignment methods, which only optimize the similarity of global image and text features, can easily lose these details. The fine-grained cross-modal alignment in FineLIP aims to enable the model to focus on the intricate interactions between local visual tokens and their corresponding textual tokens, thereby enhancing the model's ability to parse complex, detail-rich descriptions. The CLIM module is the core component designed to achieve this goal.

Method:

Token-Level Similarity Calculation:

First, compute the cosine similarity S($v_i^{'}$, $t_j^{'}$)between each aggregated visual token $v_i^{'}$ and each aggregated textual token $t_j^{'}$. This step constructs a fine-grained similarity matrix that directly captures the detailed correspondence between specific parts of the image and individual text tokens.

Bidirectional Pooling for Overall Alignment Score:

To derive a comprehensive matching score R(I , T)for an image-text pair (I ,T) from the fine-grained similarities, the following pooling strategy is used:

$$R(I,T) = \frac{1}{P'}\sum_{i=1}^{P'} \max_j S(v'_i, t'_j) + \frac{1}{M'}\sum_{j=1}^{M'} \max_i S(t'_i, v'_j)$$

First Term (Image-to-Text): For each visual token, find the most similar text token, then average these maximum similarities across all visual tokens. This measures " how well the image can find corresponding descriptions in the text. "

Second Term (Text-to-Image): For each text token, find the most similar visual token, then average these maximum similarities across all text tokens. This measures " how well the text can find corresponding regions in the image. "

This bidirectional design ensures both the image and text are accurately represented in the relation, leading to more robust alignment.

Loss Function: Triplet Marginal Loss:

The model employs a Triplet Marginal Loss for optimization instead of CLIP's traditional contrastive loss. This loss is applied in both image-to-text and text-to-image directions.

For a query image I_q, its loss with a positive text T^+ and a negative text T^- is:

$$\mathcal{L}_{i2t} = \max(0, R(I_q, T^-) - R(I_q, T^+) + \alpha)$$

Symmetrically, for a query text T_q, its loss is:

$$\mathcal{L}_{t2i} = \max(0, R(T_q, I^-) - R(T_q, I^+) + \alpha)$$

The total loss is:

$$\mathcal{L}_{triplet} = \mathcal{L}_{i2t} + \mathcal{L}_{t2i}$$

Here, α is a predefined margin (set to 0.2 in the paper). This loss forces the similarity score of positive pairs to exceed that of negative pairs by at least α, more effectively separating positive and negative samples.

Experiment

The paper evaluates FineLIP on two core tasks and conducts comprehensive ablation studies.

Experimental Setup:

Training Data: The ShareGPT4V dataset (~1.2M image-caption pairs) is used, featuring long captions with an average length of ~143.6 words.

Evaluation Tasks & Datasets:

Zero-Shot Long Caption Cross-Modal Retrieval: Evaluated on Urban1k and DOCCI datasets (containing long, detailed captions) for Image-to-Text (I2T) and Text-to-Image (T2I) retrieval. Metrics are Recall@1/5/10.

Long-Text-to-Image Generation: FineLIP's text encoder replaces the original CLIP encoder in Stable Diffusion XL (SDXL). The quality of images generated from long text is evaluated using the Fréchet Inception Distance (FID) metric (lower is better).

Compared Methods: Include the Baseline (only positional embedding stretching), long-text methods Long-CLIP and TULIP, and fine-grained alignment methods SPARC and LAPS.

Experiment

		Urban1k						DOCCI					
		Image-to-Text			Text-to-Image			Image-to-Text			Text-to-Image		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
B/16	Baseline	0.859	0.969	0.989	0.866	0.963	0.976	0.731	0.937	0.972	0.767	0.946	0.975
	SPARC [2]	0.854	0.963	0.989	0.853	0.957	0.977	0.700	0.922	0.960	0.720	0.925	0.965
	LAPS [6]	0.890	**0.987**	0.994	0.884	0.971	0.988	0.768	0.953	0.979	0.783	0.954	0.981
	TULIP [19]	0.881	-	-	0.866	-	-	-	-	-	-	-	-
	Long-CLIP [36]	0.789	-	-	0.795	-	-	-	-	-	-	-	-
	FineLIP (Ours)	0.907	0.983	**0.995**	0.893	0.975	0.987	0.771	0.954	0.980	0.795	0.958	**0.984**
	FineLIP* (Ours)	**0.912**	0.985	**0.995**	**0.900**	**0.977**	**0.990**	**0.781**	**0.961**	**0.982**	**0.800**	**0.959**	**0.984**
L/14	Baseline	0.892	0.985	0.995	0.907	0.982	0.988	0.791	0.954	0.981	0.817	0.967	0.985
	SPARC [2]	0.772	0.940	0.969	0.814	0.945	0.975	0.670	0.909	0.952	0.700	0.917	0.965
	LAPS [6]	0.921	0.988	0.994	0.916	0.981	0.988	0.818	0.965	0.985	0.818	0.965	0.985
	TULIP [19]	0.901	-	-	0.911	-	-	-	-	-	-	-	-
	Long-CLIP [36]	0.827	-	-	0.861	-	-	-	-	-	-	-	-
	FineLIP (Ours)	0.932	0.988	**0.996**	0.930	0.984	0.994	0.822	0.967	0.985	0.831	0.971	0.990
	FineLIP* (Ours)	**0.945**	**0.993**	**0.996**	**0.939**	**0.987**	**0.995**	**0.837**	**0.972**	**0.989**	**0.844**	**0.974**	**0.991**
bigG/14	Baseline	0.908	0.986	**0.996**	0.911	0.977	0.985	0.813	0.965	0.984	0.842	0.972	0.987
	LAPS [6]	**0.932**	0.988	**0.996**	0.928	0.984	0.991	0.832	0.968	0.988	0.848	0.975	0.990
	FineLIP (Ours)	0.924	0.991	0.995	0.933	0.985	0.991	0.836	0.972	0.988	0.853	0.975	0.989
	FineLIP* (Ours)	**0.932**	**0.992**	**0.996**	**0.941**	**0.986**	**0.994**	**0.845**	**0.974**	**0.990**	**0.860**	**0.978**	**0.992**

L/14	FID ↓	
	Urban1k	DOCCI
Baseline	29.535	16.884
SPARC [2]	31.604	17.241
LAPS [6]	**26.743**	15.426
FineLIP (Ours)	27.261	**15.410**

Experiment

A: Impact of Initialization and Positional Embedding Stretching.

Different Settings	Urban1k				DOCCI			
	Image-to-Text		Text-to-Image		Image-to-Text		Text-to-Image	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
Baseline_rand_init	0.298	0.582	0.306	0.559	0.155	0.344	0.115	0.276
Pretrained-CLIP[†]	0.684	0.888	0.559	0.796	0.658	0.892	0.631	0.871
Baseline_nPE	0.697	0.907	0.733	0.911	0.611	0.873	0.666	0.891
Baseline_Lv [35]	0.775	0.950	0.755	0.931	0.650	0.902	0.679	0.905
Baseline	**0.859**	**0.969**	**0.866**	**0.963**	**0.731**	**0.937**	**0.767**	**0.946**

B: Impact of ATRM in our model.

Different Settings	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
CLIM (only)	0.854	0.967	0.782	0.940	0.721	0.928	0.701	0.915
ATRM (image) + CLIM	0.893	**0.985**	0.890	0.972	0.767	0.953	0.786	0.953
ATRM (text) + CLIM	0.887	0.975	0.827	0.964	0.767	**0.955**	0.742	0.936
ATRM (both) + CLIM	**0.907**	0.983	**0.893**	**0.975**	**0.771**	0.954	**0.795**	**0.958**

C: Effect of different aggregation ratio in ATRM.

Aggregation Ratio	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
0.4	0.899	0.980	0.891	0.974	0.768	0.954	0.785	0.957
0.2	**0.907**	0.983	**0.893**	0.975	0.771	0.954	**0.795**	0.958
0.1	0.905	**0.984**	0.883	**0.978**	**0.776**	**0.958**	0.790	**0.959**

D: Importance of retaining global features in alignment.

B/16	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
No Global Tokens	0.894	0.982	**0.893**	0.973	0.760	0.948	0.773	0.949
FineLIP (Ours)	**0.907**	**0.983**	**0.893**	**0.975**	**0.771**	**0.954**	**0.795**	**0.958**

Thanks