

“Longer” CLIP

V&L Lab Meeting Paper Reading
21 May 2025

Two papers about long-text CLIP

- Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. "[Long-clip: Unlocking the long-text capability of clip.](#)" In ECCV 2024.
- Ivona Najdenkoska, Mohammad Mahdi Derakhshani, Yuki M. Asano, Nanne van Noord, Marcel Worring, and Cees GM Snoek. "[Tulip: Token-length upgraded clip.](#)" In ICLR 2025.

Long-CLIP: Unlocking the Long-Text Capability of CLIP

Beichen Zhang^{§1,2}, Pan Zhang¹, Xiaoyi Dong^{1,3*},
Yuhang Zang¹, and Jiaqi Wang^{1*}

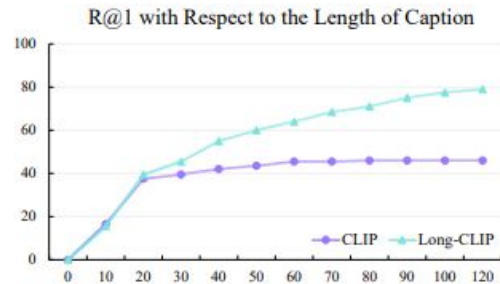
¹Shanghai AI Laboratory ²Shanghai Jiao Tong University

³The Chinese University of Hong Kong

CLIP's text encoder supports at most 77 tokens

Empirical study shows CLIP's actual effective length is even **less than 20 tokens!**

- Cannot faithfully encode long, detailed textual descriptions.
- Image encoder trained on short captions focuses on coarse cues; fails on fine-grained attributes and relational reasoning.



a) Low effective token length

Image-Text Retrieval

The image shows a city street scene . On the left side, there is a red brick building with a series of **white columns and lamp posts leading down a sidewalk**. A few pedestrians are walking on the sidewalk. There is a **blue mailbox and a black trash can** near the pedestrians. In the center of the image, there is a **multi-lane road**. To the right, there are **more buildings including a tall beige building with many windows**, and a shorter red brick building. **Street traffic lights** are visible. The road curves slightly to the right in the distance.



CLIP ✗



Long-CLIP ✓

Text-to-Image Generation

This picture shows a quiet and beautiful park. In distance, there is a **mountain that reaches to the sky**. In the park, a deep path leads to a distant forest. The trees on both sides of the road are **maples**, and **yellow maple leaves are falling on the path**.



The afternoon sun casts diagonal rays onto the alley, creating intertwining shadows on the cobblestone path. An **antique street lamp** stands in a corner, patiently awaiting the arrival of dusk. An **old-fashioned bicycle** is parked against the wall, seemingly waiting for its next rider. The Alley is filled with tranquility, a testament to the stillness of time.



Text

CLIP

Long-CLIP

Motivation for 'longer' CLIP

Unlock long-text understanding while preserving CLIP's zero-shot robustness.

Desired features:

- Accept hundreds of tokens (>77).
- Maintain alignment in the original CLIP latent space (plug-and-play)
- Cheap to train/ finetune based on original CLIP

Long-CLIP: Unlocking the Long-Text Capability of CLIP

Key idea: Retain the original CLIP architecture, apply lightweight modifications, and fine-tuning to extend context length and enhance fine-grained alignment.

Two novel strategies:

1. **Knowledge-Preserved Stretching (KPS)** of positional embeddings.
2. **Primary Component Matching (PCM)** for coarse-/fine-grained alignment.

Why Linear Positional Interpolation Fails in CLIP?

Simple linear interpolation is a common way to extend context length. **But in CLIP, this naive strategy fails due to:**

1. **High-position embeddings are poorly trained**
 - CLIP is trained mostly on short texts.
 - Embeddings beyond position 20 are unreliable.

Why Linear Positional Interpolation Fails in CLIP?

Simple linear interpolation is a common way to extend context length. **But in CLIP, this naive strategy fails due to:**

1. **High-position embeddings are poorly trained**
 - CLIP is trained mostly on short texts.
 - Embeddings beyond position 20 are unreliable.
2. **Interpolating all positions breaks well-trained ones**
 - Replacing early positions disrupts short-text performance.
 - Leads to degraded zero-shot classification and retrieval.

Why Linear Positional Interpolation Fails in CLIP?

Simple linear interpolation is a common way to extend context length. **But in CLIP, this naive strategy fails due to:**

1. **High-position embeddings are poorly trained**

- CLIP is trained mostly on short texts.
- Embeddings beyond position 20 are unreliable.

2. **Interpolating all positions breaks well-trained ones**

- Replacing early positions disrupts short-text performance.
- Leads to degraded zero-shot classification and retrieval.

3. **Semantic misalignment**

- Linear interpolation assumes smooth transitions.
- Actual position embeddings may encode non-linear semantics.

Knowledge Preserving Stretching (KPS)

Implementation Strategy:

- Keep: Positions 0–20 untouched (high-quality, well-trained)
- Stretch: Positions 21–77 using interpolation with a larger ratio λ_2 ($\lambda_2 = 4$)

→ Prevents overwriting meaningful representations

$$PE^*(pos) = \begin{cases} PE(pos), & pos \leq 20 \\ (1 - \alpha) \times PE(\lfloor \frac{pos}{\lambda_2} \rfloor) + \alpha \times PE(\lceil \frac{pos}{\lambda_2} \rceil), & \alpha = \frac{pos \% \lambda_2}{\lambda_2}, \text{ otherwise} \end{cases}$$

Knowledge Preserving Stretching (KPS)

Implementation Strategy:

- Keep: Positions 0–20 untouched (high-quality, well-trained)
- Stretch: Positions 21–77 using interpolation with a larger ratio λ_2 ($\lambda_2 = 4$)

→ Prevents overwriting meaningful representations

$$PE^*(pos) = \begin{cases} PE(pos), & pos \leq 20 \\ (1 - \alpha) \times PE(\lfloor \frac{pos}{\lambda_2} \rfloor) + \alpha \times PE(\lceil \frac{pos}{\lambda_2} \rceil), & \alpha = \frac{pos \% \lambda_2}{\lambda_2}, \text{ otherwise} \end{cases}$$

Benefits:

- Enables longer text input (248 tokens = 4 x 57 + 20)
- Maintains performance on short text tasks
- Smooth extension with minimal long-text fine-tuning

Knowledge Preserving Stretching (KPS)

Implementation Strategy:


- Keep: Positions 0–20 untouched (high-quality, well-trained)
- Stretch: Positions 21–77 using interpolation with a larger ratio λ_2 ($\lambda_2 = 4$)

→ Prevents overwriting meaningful representations

$$PE^*(pos) = \begin{cases} PE(pos), & pos \leq 20 \\ (1 - \alpha) \times PE(\lfloor \frac{pos}{\lambda_2} \rfloor) + \alpha \times PE(\lceil \frac{pos}{\lambda_2} \rceil), & \alpha = \frac{pos \% \lambda_2}{\lambda_2}, \text{ otherwise} \end{cases}$$

Benefits:

- Enables longer text input (248 tokens = 4 x 57 + 20)
- Maintains performance on short text tasks
- Smooth extension with minimal long-text fine-tuning

 KPS uses a "semi-trust strategy": it acknowledges that high positions are weak, but leverages them carefully through **interpolation as a 'gentle' initialization**, and corrects them via **fine-tuning to re-learn the semantics of those positions**.

Limitations of KPS only

Merely relaxing the input length constraint through KPS strategy is not sufficient to fully unleash the model's ability to process long text. Therefore, an effective fine-tuning strategy is needed:

- Truly **unlock long-text capability**: Enable the model to understand and utilize the detailed information in long text descriptions.
- Simultaneously **maintain short-text capability**: Prevent the model from forgetting how to handle short text, or experiencing performance degradation on short-text tasks, while learning from long text.
- Learn to **distinguish information importance**: Allow the model not only to capture various attributes in an image but also to understand their relative importance.

Primary Component Matching (PCM) Overview

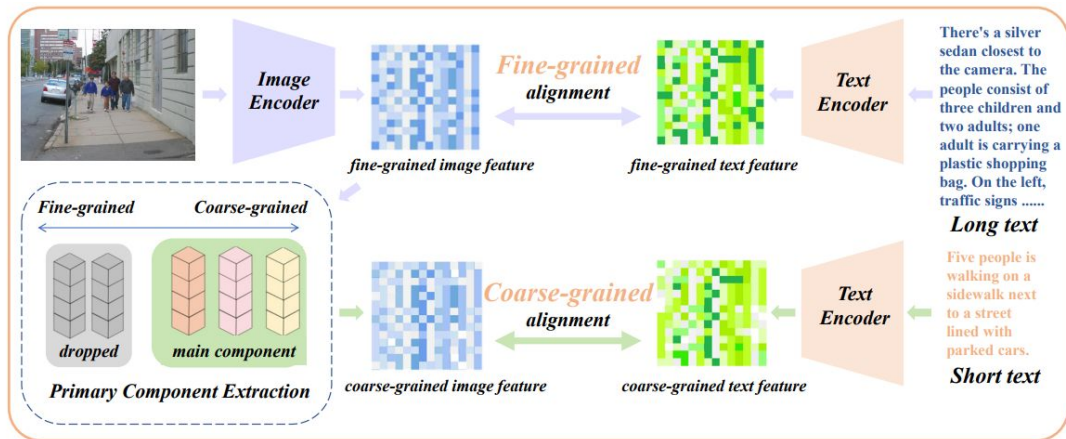


Fig. 3: The pipeline of our training process. We align the fine-grained image feature with the a long detailed caption. Moreover, we also apply **primary component extraction** to keep the main component and extract the coarse-grained image feature. Then, it is aligned with a short summary caption.

```

1 #image_encoder
2 #text_encoder
3 #I         - minibatch of input image
4 #T_long    - minibatch of input long caption
5 #T_short   - minibatch of input short caption
6 #t         - learned temperature parameter
7 #alpha     - hyperparameter for balancing coarse-grained
8 #          - and fine-grained alignment
9 #PCE       - Our primary component extraction algorithm
10
11 # extract and align the fine-grained feature of each modality
12 T_fine = text_encoder(T_long)
13 I_fine = image_encoder(I)
14
15 #compute the logits and the loss for fine-grained alignment
16 logits_fine = np.dot(I_fine, T_fine.T) * np.exp(t)
17 labels      = np.arange(n)
18 loss_fine   = cross_entropy_loss(logits_fine, labels)
19
20 #reconstruct and align the coarse-grained feature of each modality
21 T_coarse = text_encoder(T_short)
22 I_coarse = PCE(I_fine)
23
24 #compute the logits and the loss for fine-grained alignment
25 logits_coarse = np.dot(I_coarse, T_coarse.T) * np.exp(t)
26 labels        = np.arange(n)
27 loss_coarse   = cross_entropy_loss(logits_coarse, labels)
28
29 #compute the total loss
30 loss = loss_fine + alpha * loss_coarse

```

Primary Component Extraction (PCE)

A method for **extracting coarse-grained image features from fine-grained image features** to extract different attributes from the fine-grained image feature and analyze their importance.

Implementation:

- **Component-decomposition function \mathbf{F}** : decomposes the feature into several vectors that represent different attributes and also analyzes the importance of each attribute.
- **Component-filtration function $\mathbf{\mathcal{E}}$** : filters out less important attributes based on their analyzed importance.
- **Component-reconstruction function \mathbf{F}^{-1}** : reconstructs the image feature with different attribute vectors and their corresponding importance.

Primary Component Extraction (PCE)

Given a fine-grained image feature I_{fine} , we first extract its different component vectors v_t and the corresponding importance i_t as Eq. 3

$$(v_1, i_1), (v_2, i_2), \dots, (v_n, i_n) = \mathcal{F}(I_{fine}) \quad (3)$$

Then, we apply our component-filtration function \mathcal{E} to select the key components and wipe out the others as Eq. 4.

$$(v_{k_1}, i_{k_1}), (v_{k_2}, i_{k_2}), \dots, (v_{k_m}, i_{k_m}) = \mathcal{E}[(v_1, i_1), (v_2, i_2), \dots, (v_n, i_n)], \quad m \ll n \quad (4)$$

Finally, we apply our component-reconstruction function \mathcal{F}^{-1} to reconstruct the image feature with only the key components and their importance as Eq. 5.

$$I_{coarse} = \mathcal{F}^{-1}[(v_{k_1}, i_{k_1}), (v_{k_2}, i_{k_2}), \dots, (v_{k_m}, i_{k_m})] \quad (5)$$

$$I_{coarse} = \mathcal{F}^{-1}(\mathcal{E}(\mathcal{F}(I_{fine}))) \quad (6)$$

Primary Component Extraction (PCE)

Given a fine-grained image feature I_{fine} , we first extract its different component vectors v_t and the corresponding importance i_t as Eq. 3

$$(v_1, i_1), (v_2, i_2), \dots, (v_n, i_n) = \mathcal{F}(I_{fine}) \quad (3)$$

Then, we apply our component-filtration function \mathcal{E} to select the key components and wipe out the others as Eq. 4.

$$(v_{k_1}, i_{k_1}), (v_{k_2}, i_{k_2}), \dots, (v_{k_m}, i_{k_m}) = \mathcal{E}[(v_1, i_1), (v_2, i_2), \dots, (v_n, i_n)], \quad m \ll n \quad (4)$$

Finally, we apply our component-reconstruction function \mathcal{F}^{-1} to reconstruct the image feature with only the key components and their importance as Eq. 5.

$$I_{coarse} = \mathcal{F}^{-1}[(v_{k_1}, i_{k_1}), (v_{k_2}, i_{k_2}), \dots, (v_{k_m}, i_{k_m})] \quad (5)$$

$$I_{coarse} = \mathcal{F}^{-1}(\mathcal{E}(\mathcal{F}(I_{fine}))) \quad (6)$$

- Utilize a widely-used dimensionality reduction algorithm, **Principal Component Analysis (PCA)**. Specifically, the **Eigen Value Decomposition (EVD)** of the covariance matrix serves as component-decomposition function \mathcal{F} .
- Select the eigenvectors corresponding to the **top 32 largest eigenvalues**, which serves as the component-filtration function \mathcal{E} .
- Reconstruct the image by the **linear combination of the selected eigenvectors** as component-reconstruction function \mathcal{F}^{-1}

Experiments

Evaluation Benchmarks:

1) **zero-shot image classification:**

ImageNet-1K, ImageNet-V2, ImageNet-O, CIFAR-10, CIFAR-100

2) **short-caption image-text retrieval:**

COCO2017-val, Flickr30k

3) **long-caption image-text retrieval:**

ShareGPT4V (random 1k image-long text pairs from it), Urban-200 (custom: 200 urban images + long GPT-4V generated captions)

Fine-tuning Dataset: ShareGPT4V (approx. 1M long caption-image pairs, excluding the 1k validation).

Base: CLIP ViT-B/16, CLIP ViT-L/14

Fine-tuning: 1 epoch, batch 1024, AdamW ($\eta = 1 \times 10^{-4}$), weight decay 10^{-2} .

Hardware: 8 A100 GPUs, <0.25 h per model.

Results

Table 1: The R@1 of long-caption text-image retrieval on 1k ShareGPT4V [2] validation set and Urban-200 dataset. Best result is in **bold**.

		ShareGPT4V		Urban-200	
		Image-to-Text	Text-to-Image	Image-to-Text	Text-to-Image
B/16	CLIP	78.2	79.6	46.5	46.0
	Direct Fine-tuning	94.1	93.6	78.5	78.0
	Long-CLIP(Ours)	94.6	93.3	79.5	79.0
L/14	CLIP	81.8	84.0	47.0	47.0
	Direct Fine-tuning	95.3	95.4	78.0	76.5
	Long-CLIP(Ours)	95.8	95.6	81.5	81.5

Table 2: Results of short-caption text-image retrieval on the 5k COCO2017 validation set and the whole **30k** Flickr30K dataset. Best result is in **bold**.

		COCO						Flickr30k					
		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	CLIP	51.8	76.8	84.3	32.7	57.7	68.2	44.1	68.2	77.0	24.7	45.1	54.6
	Direct Fine-tuning	37.4	62.3	72.1	21.8	43.4	54.5	25.7	45.8	55.4	17.9	34.5	43.1
	Long-CLIP(Ours)	57.6	81.1	87.8	40.4	65.8	75.2	46.8	71.4	79.8	34.1	56.3	65.7
L/14	CLIP	56.1	79.5	86.8	35.4	60.1	70.2	48.5	72.6	80.8	28.0	49.3	58.7
	Direct Fine-tuning	37.9	63.1	72.2	23.1	45.1	55.9	26.0	46.3	55.6	17.9	34.9	43.5
	Long-CLIP(Ours)	62.8	85.1	91.2	46.3	70.8	79.8	53.4	77.5	85.3	41.2	64.1	72.6

Results

Table 3: Results of zero-shot image classification in the above five validation sets. Best result is in **bold**.

		ImageNet	ImageNet-O	ImageNet-V2	Cifar10	Cifar100	Average
B/16	CLIP	68.4	42.2	61.9	90.8	67.3	66.12
	Direct Fine-tuning	55.1	31.7	44.8	83.9	59.2	54.94
	Long-CLIP(Ours)	66.8	42.7	61.2	90.7	69.3	66.14
L/14	CLIP	75.5	31.9	69.9	95.5	76.8	69.92
	Direct Fine-tuning	58.4	29.2	52.7	92.7	68.7	60.3
	Long-CLIP(Ours)	73.5	33.7	67.9	95.3	78.5	69.78

Long-CLIP preserves short-text performance, unlike naive interpolation

Demonstrates compatibility with original CLIP in zero-shot settings

Ablation Study: KPS and PCM Contributions

Table 4: A comparison of whether to use our two strategies. Best result is in **bold**.

KPS	PCM	ImageNet	Cifar100	COCO T2I R@5	Flickr I2T R@5	urban T2I R@1
✗	✗	55.1	59.2	43.4	45.8	78.0
✗	✓	58.8	63.5	46.1	46.0	76.5
✓	✗	65.6	65.9	64.3	70.4	78.0
✓	✓	66.8	69.3	65.8	71.4	79.0

Removing either KPS or PCM results in a significant loss of short-text capabilities (e.g., on COCO/Flickr).

Both KPS and PCM are crucial for optimal performance, especially for maintaining and improving short-text performance while enabling long-text understanding.

Ablation Study: Strategies for Keeping Short-Text Capability

- **Undistinguished Image Feature**: Align same image feature with both long and short texts.
- **Mixed-length Text**: Randomly replace 10% long texts with short texts during training.
- **Bounded Text Encoder**: Compute SmoothL1 loss to keep current short text features close to frozen original CLIP's.

Table 5: A comparison of different strategies aiming to keep the short-text capability. Best result is in **bold**.

Strategy	ImageNet	Cifar100	COCO T2I R@5	Flickr I2T R@5	Urban-200 T2I R@1
Undistinguished	65.5	67.5	64.6	66.8	71.0
Mixed-length text	66.4	67.8	64.2	68.2	67.5
Bounded encoding	66.8	67.9	65.7	69.3	80.0
Ours	66.8	69.3	65.8	71.4	79.0

Plug-and-Play in Image Generation using CLIP

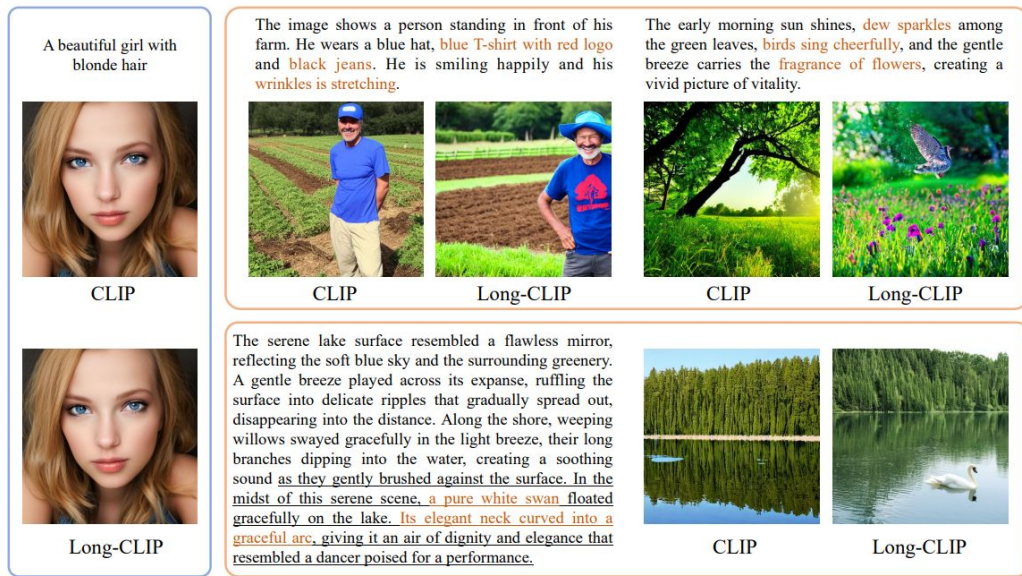


Fig. 6: Our Long-CLIP model can benefit the **text-to-image generation** in a plug-and-play manner in these three aspects. For short simple captions (left), the image generated is nearly the same. For a detailed caption (right top), our model can take in more detailed attributes in it. The caption marked in brown are the detailed attributes missed by the original CLIP, but successfully captured by us. For a long caption (right bottom) exceeding 77 tokens, our model can take in the complete sentence. The underlined caption will be truncated in original CLIP, but can be kept by us.

Conclusion

Summary:

- **Knowledge–Preserved Stretching (KPS)** of positional embeddings extension.
- **Primary Component Matching (PCM)** for coarse–/fine–grained alignment.

Potential Limitations

- **Upper Bound on Length:** Still has a maximum input token length (248 tokens), though significantly improved over original CLIP.
- **Comparison to Relative PEs:** Relative positional embeddings (like RoPE) might offer theoretically unbounded length, but their performance can degrade substantially on very long inputs.

Scaling-up Potential

- **Current Data:** Leveraged 1M long text-image pairs from ShareGPT4V due to data scarcity.
- **Future Improvement:** Significant potential for improvement if trained on larger datasets of long text-image pairs.
- **Richness of Long Texts:** Sufficient long texts can provide complex information (world knowledge, object properties, spatial relationships, aesthetics), which could greatly enhance the model's abilities.



TULIP: TOKEN-LENGTH UPGRADED CLIP

Ivona Najdenkoska* Mohammad Mahdi Derakhshani* Yuki M. Asano

Nanne van Noord Marcel Worring[†] Cees G. M. Snoek[†]

University of Amsterdam

Amsterdam, the Netherlands

Shortcomings of Long-CLIP

1. Relies on Absolute Positional Encodings (via Interpolation)
 - Long-CLIP merely stretches existing position embeddings rather than fundamentally improving how the model handles token relationships.
 - It still uses absolute encodings, which are inherently limited in long-range dependencies.

Shortcomings of Long-CLIP

1. Relies on Absolute Positional Encodings (via Interpolation)
 - Long-CLIP merely stretches existing position embeddings rather than fundamentally improving how the model handles token relationships.
 - It still uses absolute encodings, which are inherently limited in long-range dependencies.
2. Unable to Capture Fine-Grained Relative Positions
 - Absolute encodings don't generalize well to sequences longer than those seen during training.
 - Long-CLIP lacks the capacity to model how tokens relate relatively to each other across long distances, which is important for long-caption understanding.

Key Contributions

TULIP (Token-Length Upgraded CLIP): A generalizable method with relative positional encodings to upgrade token length for CLIP-like models to any length.

- Propose a two-step training: **Relative Position Distillation + Position Expansion.**
- Evaluate on multiple datasets and tasks, showing strong gains on long-caption understanding.
- Introduce Long-DCI benchmark for dense, diverse caption evaluation.

Position Encodings in Transformer Models

- Absolute Positional Encodings (Vaswani et al., 2017): Fixed embeddings added to token embeddings.
- Relative Positional Encodings (Shaw et al., 2018; Press et al., 2021): Capture pairwise distances.
- Rotary Position Embedding (RoPE) (Su et al., 2024): Applies relative PEs without modifying self-attention, computationally efficient.
- Contextual Position Encodings (COPE) (Golovneva et al., 2024): General PE for attending to specific words/sentences.
- **Focus of this paper: Effective integration of PEs across modalities in VL models.**

Architecture: RoPE Encoding

- Replace absolute position encodings with Rotary Position Embeddings (RoPE).
- RoPE enables token pairwise distance modeling with relative position information.
- RoPE applied via rotation matrices in self-attention (Su et al., 2024).

Problem Statement

Let model f be a contrastive VL model (e.g., CLIP).

Its text encoder f_T is constrained to $T_f = 77$ tokens due to fixed absolute positional encodings $P_f \in \mathbb{R}^{77 \times d}$

For an input sequence $x = [x_1, \dots, x_n]$ with $n > T_f$, f_T truncates x to $x' = [x_1, \dots, x_{T_f}]$.

Objective: Transform f into model g (with text encoder g_T) capable of processing sequences of arbitrary length $T_g > 77$, without retraining from scratch.

Positional Encoding Swapping

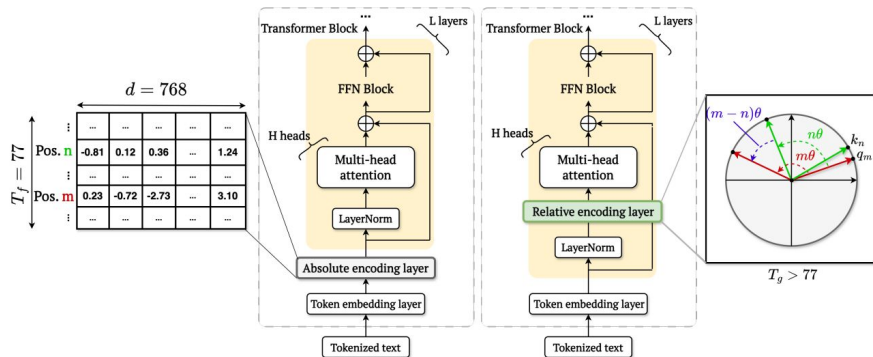


Figure 1: **Swapping the Positional Encoding.** We update CLIP models by replacing the absolute positional encoding with relative positional encoding in each transformer block. This modification

- Replace absolute $P_f(i)$ with a function $P_g(i)$ that scales with input length T_g , implemented using RoPE.
- RoPE rotates embeddings based on relative distance between tokens.

Positional Encoding Swapping

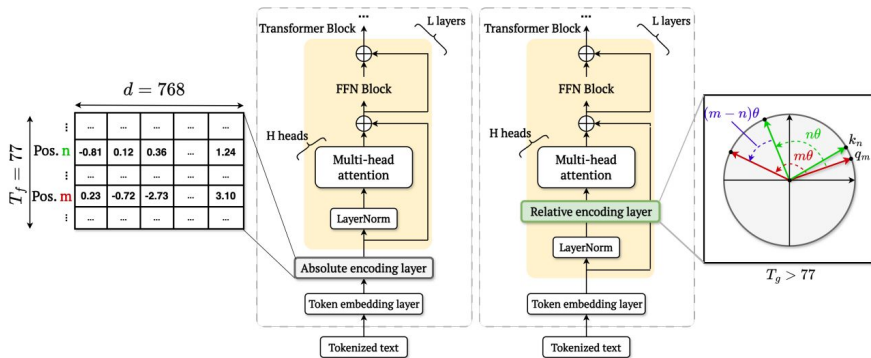


Figure 1: **Swapping the Positional Encoding.** We update CLIP models by replacing the absolute positional encoding with relative positional encoding in each transformer block. This modification

- Replace absolute $P_f(i)$ with a function $P_g(i)$ that scales with input length T_g , implemented using RoPE.
- RoPE rotates embeddings based on relative distance between tokens.

- Original self-attention: $\text{softmax}(\frac{q_m^T k_n}{\sqrt{d}})$
 $q_m = W_q x_m, k_n = W_k x_n,$
- With RoPE, position info of each token is injected into query and key vectors $q_m = R_{\Theta, m} W_q x_m, k_n = R_{\Theta, n} W_k x_n$ where $R_{\Theta, m}, R_{\Theta, n}$ are rotation matrices, Θ is rotational frequency.
- Allows for long caption understanding and better modeling of pairwise token dependencies.

Training Phase 1: Relative Position Distillation

- Adapt text encoder g_T to short/long text while retaining image-text alignment of f .
- Teacher: f_T (original text encoder). Student: s_T (text encoder with new RPEs).
- Input caption x with length $n < T_f$.
- Teacher output: $z_{fT} = f_T(x)$. Student output: $z_{sT} = s_T(x)$.
- Distillation Loss (cosine similarity):

$$\mathcal{L}_{\text{distill}} = \frac{z_{fT} \cdot z_{sT}}{\|z_{fT}\| \|z_{sT}\|}.$$

- Efficient, generalizable, transfers capabilities without full retraining.

Training Phase 2: Relative Position Expansion

- Expand context length of model g beyond $T_f = 77$ tokens
- Initialize g_T with weights from distilled student s_T
- Employ NTK-aware scaled RoPE (bloc97, 2023):
 - Adapts RoPE to changing input lengths by scaling rotational frequency
 - Resolves loss of high-frequency info when interpolating RoPE to longer positions
 - Scales the rotational frequency Θ by a factor $(\alpha * \frac{T_g}{T_f}) - (\alpha - 1)$, where α is a hyperparameter

Training Phase 2: Relative Position Expansion

- Expand context length of model g beyond $T_f = 77$ tokens
- Initialize g_T with weights from distilled student s_T
- Employ NTK-aware scaled RoPE (bloc97, 2023):
 - Adapts RoPE to changing input lengths by scaling rotational frequency
 - Resolves loss of high-frequency info when interpolating RoPE to longer positions
 - Scales the rotational frequency Θ by a factor $(\alpha * \frac{T_g}{T_f}) - (\alpha - 1)$, where α is a hyperparameter
- Fine-tune model g (both text encoder and vision encoder) with longer captions ($T_g > T_f$)
- Joint contrastive loss for short (x_{T_f}) and long (x_{T_g}) captions:

$$\mathcal{L}_{\text{total}}(x_{T_g}, x_{T_f}, y) = \lambda \times \mathcal{L}_{\text{short}}(x_{T_f}, y) + (1 - \lambda) \times \mathcal{L}_{\text{long}}(x_{T_g}, y)$$

$$\mathcal{L}(x, y) = -\log \frac{\exp(\cos(z_x, z_y)/\tau)}{\sum_{y'} \exp(\cos(z_x, z_{y'})/\tau)}$$

TULIP Training Procedure

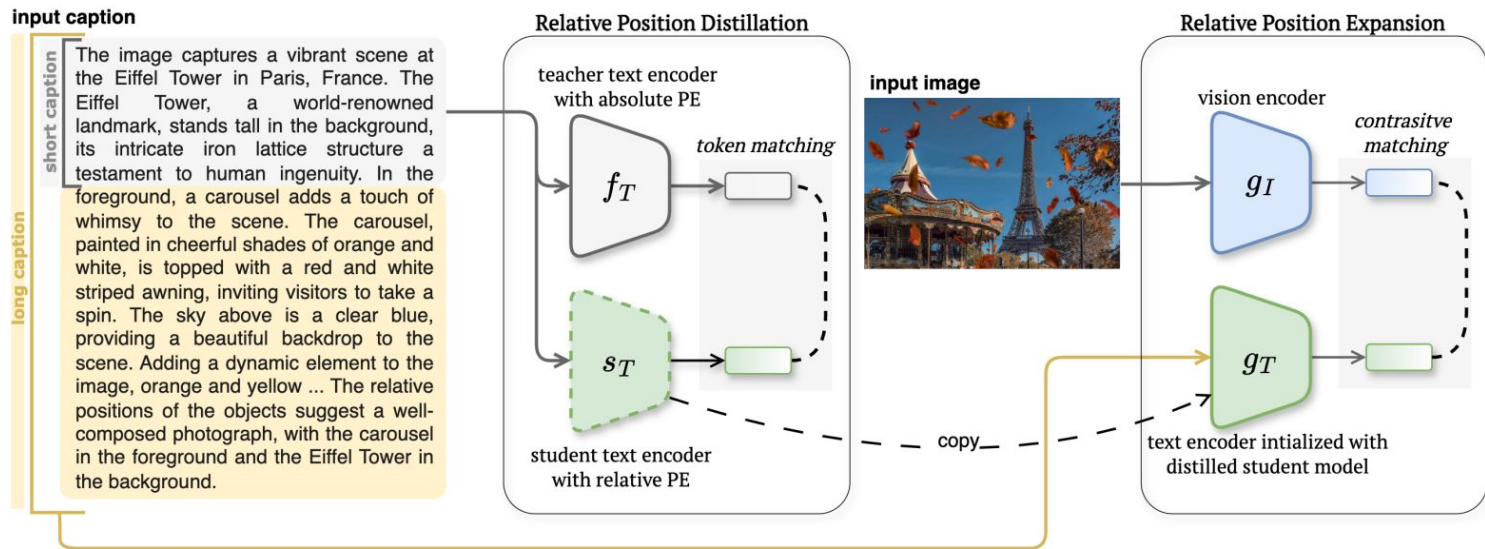


Figure 2: **TULIP training procedure.** First, we perform relative position adaptation by distilling the knowledge of the CLIP text encoder into a student text encoder initialized with relative position encodings. This stage uses the first 77 tokens of a long caption (the gray block). The second stage is the relative position expansion, where we fine-tune the distilled text encoder with captions longer than 77 tokens (the combined gray and yellow blocks), together with the vision encoder.

Experiments

Downstream Tasks:

- **Short Caption Retrieval:** COCO2017 5k validation, Flickr30k full
- **Long Caption Retrieval:** ShareGPT4V test split (1k pairs), Urban-1K (1k pairs, recaptioned by Long-CLIP), Long-DCI (7,000 human-annotated images/long captions, avg. 200 tokens/image) from DCI dataset.
- **Text-to-Image Generation:** Qualitative, using SDXL.

Base Models: OpenAI's pre-trained CLIP-ViT-B-16 and CLIP-ViT-L-14.

Data for TULIP training: ShareGPT4V (1.2M image-long caption pairs).

Distillation: Captions truncated to 77 tokens. Cosine loss. 20 epochs, batch 640, AdamW, LR 5e-4, 1000 warmup.

Expansion: Full-length captions (248 tokens for fair comparison with Long-CLIP). NTK-aware RoPE ($\alpha=8.0$). 1 epoch, batch 1280, AdamW, LR 1e-5, 1000 warmup.

Results: long caption-image retrieval

		Long-DCI		ShareGPT4V		Urban-1K	
		Img2Txt	Txt2Img	Img2Txt	Txt2Img	Img2Txt	Txt2Img
ViT-B-16	CLIP	35.9	33.7	78.2	79.6	68.1	53.6
	Fine-tuned CLIP	46.3	45.4	94.1	93.6	80.4	79.8
	Long-CLIP	42.1	48.4	94.6	93.3	78.9	79.5
	TULIP (Ours)	50.2	50.6	98.6	98.6	88.1	86.6
ViT-L-14	CLIP	35.0	37.0	81.8	84.0	68.7	52.8
	Fine-tuned CLIP	51.6	50.7	95.3	95.4	78.0	76.5
	Long-CLIP	54.0	46.1	95.8	95.6	82.7	86.1
	TULIP (Ours)	55.7	56.4	99.0	99.0	90.1	91.1

Table 1: **Long caption cross-modal retrieval comparison** on Long-DCI, ShareGPT4V and Urban-1K. TULIP consistently outperforms other CLIP variants across all evaluated datasets and tasks. Note that we adopt the results for CLIP and Long-CLIP from Zhang et al. (2024), while we fine-tune CLIP (Fine-tuned CLIP) on ShareGPT4V ourselves.

TULIP shows strong performance, especially on the more challenging Long-DCI dataset.

Results: short caption-image retrieval

		COCO				Flickr30k			
		Img2Txt		Txt2Img		Img2Txt		Txt2Img	
		R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
ViT-B-16	CLIP	51.8	76.8	32.7	57.7	44.1	68.2	24.7	45.1
	Fine-tuned CLIP	37.4	62.3	21.8	43.4	25.7	45.8	17.9	34.5
	Long-CLIP	57.6	81.1	40.4	65.8	46.8	71.4	34.1	56.3
	TULIP (Ours)	56.8	80.3	40.7	66.1	46.1	70.8	35.2	57.4
ViT-L-14	CLIP	56.1	79.5	35.4	60.1	48.5	72.6	28.0	49.3
	Fine-tuned CLIP	37.9	63.1	23.1	45.1	26.0	46.3	17.9	34.9
	Long-CLIP	62.8	85.1	46.3	70.8	53.4	77.5	41.2	64.1
	TULIP (Ours)	62.6	84.7	46.1	71.1	56.7	79.5	41.6	64.3

Table 2: **Short caption cross-modal retrieval comparison on COCO and Flickr30k.** TULIP shows competitive performance, often matching or exceeding Long-CLIP across different metrics and model backbones.

Long-CLIP’s specific tailoring for first 20 tokens benefits short captions.

TULIP achieves competitive performance without such specific tailoring, showing RPE flexibility.

Text-to-Image Generation (Qualitative)
















Original CLIP ViT-L-14 text encoder in Stable Diffusion XL (SDXL) replaced with TULIP. No additional training of the diffusion model.

TULIP demonstrates improvements in:

- Long and short caption understanding.
- Modeling of nuanced details.

Compared to T5-based models (PIXART-Alpha, ELLA) and CLIP/Long-CLIP based SDXL.

TULIP text encoder enhances long caption comprehension for more accurate image generation.

T5-based models		CLIP-based models		
PIXART - Alpha	ELLA	SDXL + CLIP	SDXL + Long-Clip	SDXL + TULIP (Ours)
				
<p>In a quiet garden, with tall green bushes on the left and a clear pond on the right, a wooden box sits on the grass in the middle. The box is slightly open, and inside is a bright red tulip, standing tall. The sunlight shines on the tulip, making it stand out against the smooth wood of the box. A light breeze moves through the garden, // but the tulip stays still, standing tall inside the box.</p>				
				
<p>The painting captures a serene moment in nature. At the center, a calm lake reflects the sky, its surface rippled only by the gentlest of breezes. The sky above is a brilliant mix of blues and whites, with fluffy clouds drifting leisurely across. On the banks of the lake, tall trees stand gracefully, their leaves rustling in the wind. // The soft light of the setting sun bathes the entire scene in a warm glow, creating a sense of peace and tranquility. The colors are muted yet vibrant, and the details are captured with precision, giving the painting a sense of realism while still retaining a dreamlike quality. In the foreground, an old man sits on a rock, seemingly lost in deep thought or meditation.</p>				
				
<p>An illustration from a graphic novel. A bustling city street under the shine of a full moon. The sidewalks bustling with pedestrians enjoying the nightlife. At the corner stall, a young woman with fiery red hair, dressed in a signature velvet cloak, is haggling with the grumpy old vendor. the grumpy vendor, a tall, sophisticated man is wearing a sharp suit, sports a // noteworthy moustache is animatedly conversing on his steampunk telephone.</p>				

Ablation: Different Positional Encodings

Positional encodings	Long-DCI		ShareGPT4V		Urban-1K	
	Img2Txt	Txt2Img	Img2Txt	Txt2Img	Img2Txt	Txt2Img
Absolute	41.9	40.0	96	93.8	72.9	69.4
CoPE	50.8	49.9	98.5	97.8	86.7	82.8
RoPE	55.7	56.4	99.0	99.0	90.1	91.1

Table 3: **Ablation comparing different positional encodings such as Absolute, RoPE, and CoPE in TULIP.** RoPE generalizes better across varying or extended sentence lengths, especially on out-of-distribution datasets, namely Long-DCI and Urban-1k.

RoPE outperforms CoPE, especially on out-of-distribution datasets (Long-DCI, Urban-1K).

RoPE generalizes better across varying/extended sentence lengths.

CoPE embeddings more dependent on specific training context, struggle with generalization to longer/different sequences.

Ablation: Impact of Caption Length

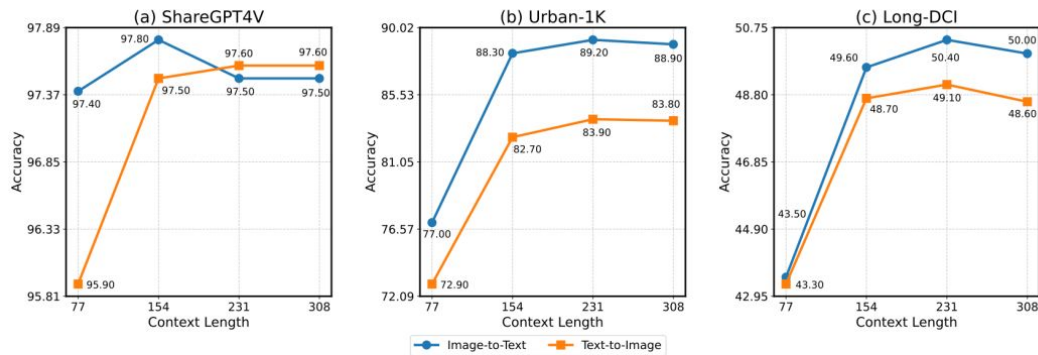


Figure 4: **Impact of the sequence length on cross-modal retrieval tasks.** We observe general improvement in performance with increased sequence length, particularly from 77 to 154 tokens, across all datasets and tasks.

General improvement up to 154 tokens.

Plateau/minor decline for 308-length:

- Additional tokens might introduce noise/redundancy.
- Average caption length in ShareGPT4V training data is 174.02 tokens.

Ablation: Benefit of Cosine Distillation Loss

Distillation loss	Long-DCI		ShareGPT4V		Urban-1K	
	Img2Txt	Txt2Img	Img2Txt	Txt2Img	Img2Txt	Txt2Img
CLIP	35.0	37.0	81.8	84.0	68.7	52.8
L2	39.4	35.8	84.8	83.9	70.2	56.3
MSE	36.4	31.8	83.0	81.3	74.0	53.9
Cosine	38.5	35.8	84.8	84.2	73.6	56.6

Table 4: **Ablation comparing different distillation loss terms in TULIP.** Cosine loss yields the best performance across different datasets and tasks. Note that here we report the performance of the distilled models before the relative position expansion phase.

Cosine loss performs best overall.

Aligns with normalized embedding space of CLIP.

Scale invariance property is beneficial:

- Student model (with RoPE/CoPE) can produce embeddings of different magnitudes than teacher.
- Cosine loss focuses on directional information, robust to scale discrepancies.

Additional Analysis: Attention Spread Visualization

Compared TULIP vs. Long-CLIP on CLS token attention to preceding tokens (248-token caption).

TULIP shows:

- More uniform attention distribution across input tokens.
- Increased attention to punctuation (e.g., commas), enhancing parsing of longer texts.

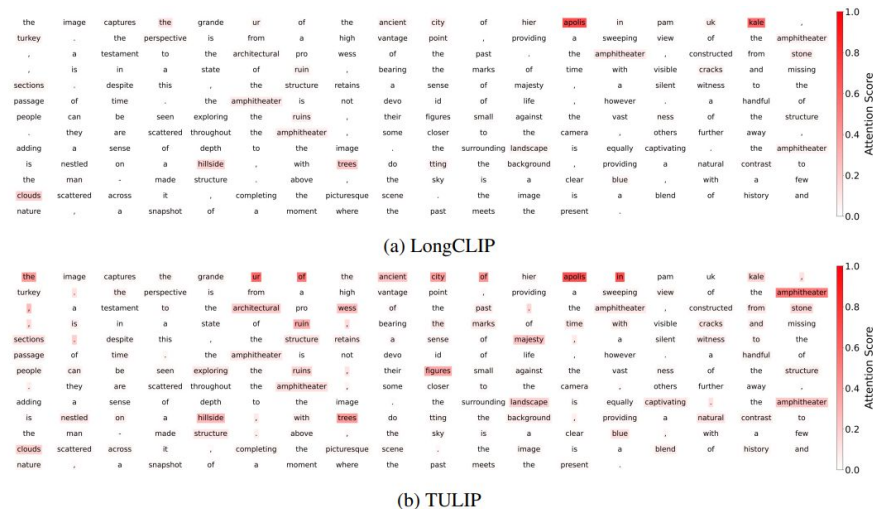


Figure 5: **Attention Spread Visualization** comparing (a) LongClip and (b) TULIP. Our model achieves uniform attention across tokens, demonstrating superior capabilities in parsing and segmenting longer texts with precision.

Additional Analysis: Caption-Image Relevance Distribution

Analyzed where relevant info is in long captions given an image(ShareGPT4V).

Cosine similarity between image embedding and text embeddings of sliding subwindows (20, 33, 55 tokens).

Findings:

- Image-relevant information is spread throughout captions.
- Larger context windows capture more cohesive/pronounced relevance.
- Reinforces need to leverage entire textual sequence.

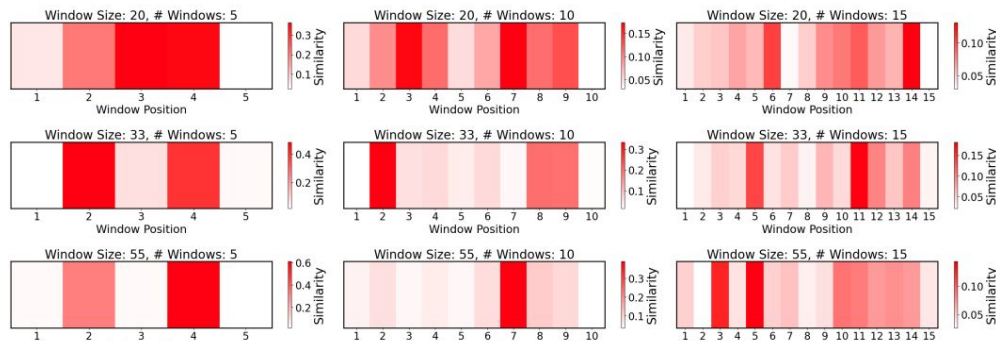


Figure 6: **Caption-image relevance distribution analysis** across varying window sizes and positions. It shows that image-relevant information is spread throughout captions, emphasizing the need for models to process longer text inputs to capture all pertinent details.

Further Appendix Results

Compositional Understanding

Method	ARO				VL-Checklist		
	VGR	VGA	Flickr	COCO	Obj	Att	Rel
CLIP	59.9	63.1	60.2	47.9	81.1	67.6	61.9
Long-CLIP	64.6	66.6	24.8	23.3	84.3	71.6	63.5
TULIP (Ours)	63.4	66.2	52.3	43.9	85.2	74.3	62.7

Table 6: **Comparison across compositional understanding benchmarks, namely ARO and VL-Checklist.** We can observe that TULIP consistently outperforms CLIP and surpasses Long-CLIP in 4 out of 7 settings, demonstrating its ability to handle fine-grained captions. Note that we use CLIP-ViT-L-14 as vision encoder.

Further Appendix Results

Training Data Amount for Student Model

Amount of data	Long-DCI		ShareGPT4V		Urban-1K	
	Img2Txt	Txt2Img	Img2Txt	Txt2Img	Img2Txt	Txt2Img
33%	15.2	12.2	88.6	90.0	33.6	36.9
66%	28.7	26.2	94.6	95.5	37.6	45.0
100%	55.7	56.4	99.0	99.0	90.1	91.1

Table 8: Performance comparison across different dataset sizes.

It is important to utilize the full dataset, particularly for improving the performance on out-of-distribution data, i.e. Long-DCI and Urban1k

Conclusion

TULIP leverages relative positional encodings (RoPE) for effective modeling of pairwise token relationships.

Efficient two-step training process (distillation + expansion) adapts models to longer captions without compromising short-caption performance significantly.

TULIP considerably improves long-caption performance on cross-modal retrieval and text-to-image generation.

Limitations

Reliance on ShareGPT4V dataset quality: TULIP's performance is linked to the quality of these long captions.

Practical token length constraint:

- Theoretically, TULIP can handle much longer contexts.
- Practically, its effective token length is constrained by the average token length of the ShareGPT4V training captions (around 174 tokens).
Performance plateaus observed beyond this.

Long-CLIP vs TULIP

	Long-CLIP	TULIP
Positional Encoding	Stretch absolute with knowledge-preserving interpolation	Switch from absolute to relative (RoPE)
Training Strategy	Joint fine-tuning with long and short text using PCM	Distillation + expansion (long-short caption fine-tuning)
Loss Functions	Contrastive loss; dual alignment (fine + coarse features)	Cosine loss for distillation; contrastive loss for fine-tuning
Handling Long Context	Stretches embeddings and selectively interpolates positions	Uses RoPE + NTK-aware scaling to extend attention field
Handling Short Text	Preserves first 20 PEs; Primary component matching aligns with short summaries.	Distillation step focuses on short text; Joint loss in expansion phase includes short text.