CLIP Is Shortsighted: Paying Attention Beyond the First Sentence

Supplementary Material

Appendices

In Sec. A, we present the training details of our main results. In Sec. B, we present the pseudocode of our DeBias-CLIP method. In Sec. C, we present additional ablations, including on the embedding stretching for SigLIP, training length, sampling, padding, and token padding methods. In Sec. D and Sec. E, we present additional discussion and results on CLIP and Long-CLIP biases, respectively. In Sec. F, we present visualizations that highlight the superior long-text retrieval capabilities of DeBias-CLIP and demonstrate its advantages in text-to-image diffusion.

A. Training Details

Complete training parameters are provided in Tab. S1. Our setup largely follows Long-CLIP [8], with two modifications: we train for 3 epochs instead of only 1 and use a smaller batch size of 256 instead of 1024 to ensure consistency across all experimental runs. Experiments were conducted on a variety of hardware configurations, subject to compute cluster availability, but the main results in Sec. 5.3 of the paper were all run on 4 A100 GPUs.

Parameter	Value
Batch size	256
Training epochs	3
Warm-up iterations	200
Weight decay	1e-2
Learning rate	1e-6
AdamW β_1	0.9
AdamW β_2	0.999
AdamW ϵ	1e-8

Table S1. **Training parameters**. We use AdamW as the optimizer and run on 4 A100 GPUs for the main paper results.

B. Pseudocode

We present pseudocode of our sampling method in Algorithm 1, where $\varphi(\cdot)$ is the text tokenizer, $\psi_{\text{text}}(\cdot)$ and $\psi_{\text{img}}(\cdot)$ are respectively the text and image encoders, and $f(\cdot)$ is an operator that reconstructs feature vectors from their principal components (PCA).

Algorithm 1 DeBias-CLIP Caption Sampling

Require: Long-text caption $C = [s_1, s_2, \ldots, s_{n_{\text{sents}}}]$, image I, loss weight λ^s
Ensure: Training loss $\mathcal{L}$

$C^\ell \leftarrow C$ ▷ Long caption
$T^\ell \leftarrow \varphi(C^\ell)$ ▷ Tokenize long caption
$C^{\text{no\_sum}} \leftarrow [s_2, \ldots, s_{n_{\text{sents}}}]$ ▷ Remove first (summary) sentence
Sample $n_{\text{samp}} \sim \mathcal{U}\{1, \ldots, n_{\text{sents}} - 1\}$
Sample a subset $\mathcal{S} \subset \{2, \ldots, n_{\text{sents}}\}$ with $|\mathcal{S}| = n_{\text{samp}}$ (without replacement)
$C^{\text{samp}} \leftarrow$ concatenate $\{s_i : i \in \mathcal{S}\}$
$T^{\text{samp}} \leftarrow \varphi(C^{\text{samp}})$ ▷ Tokenize short caption
$T^{\text{samp}} \leftarrow [\texttt{SOT}, t_1, \ldots, t_{n_{samp}}, \texttt{EOT}, \texttt{PAD}_{\text{post}}]$
Let n_{post} be the number of post-padding tokens in the tokenized short caption T^{samp}
Sample $n_{\text{pre}} \sim \mathcal{U}\{0, \ldots, n_{\text{post}}\}$
$n_{\text{post}} \leftarrow n_{\text{post}} - n_{\text{pre}}$
$T^s \leftarrow [\texttt{SOT}, \underbrace{\texttt{PAD}, \ldots, \texttt{PAD}}_{n_{\text{pre}}}, t_1, \ldots, t_{n_{samp}}, \texttt{EOT}, \underbrace{\texttt{PAD}, \ldots, \texttt{PAD}}_{n_{\text{post}}}]$
Encode captions and image:
$u^s \leftarrow \psi_{\text{text}}(T^s)$ ▷ Short (sampled) caption
$u^\ell \leftarrow \psi_{\text{text}}(T^\ell)$ ▷ Long caption
$v \leftarrow \psi_{\text{img}}(I)$ ▷ Image
$\mathcal{L}^s \leftarrow \mathcal{L}^s(u^s, f(v))$ ▷ Short caption loss
$\mathcal{L}^\ell \leftarrow \mathcal{L}^\ell(u^\ell, v)$ ▷ Long caption loss
$\mathcal{L} \leftarrow \lambda^s \mathcal{L}^s + (1 - \lambda^s)\mathcal{L}^\ell$
return $\mathcal{L}$

C. Additional Ablations

C.1. Positional Embedding Stretching for SigLIP

For the different pretrained CLIP-style models presented in our main results, we follow Long-CLIP [8] and stretch the positional embeddings by freezing the first 20 text positions and using linear interpolation to extend the rest by a factor of 4. Results from Long-CLIP justify this for the OpenAI weights, and OpenCLIP weights are trained with the same approach. However, SigLIP models use a different loss, tokenizer, and attention mechanism (with no text causal masking), and therefore, it may not be necessarily appropriate to freeze the first few tokens. We present in Tab. S2 short and long retrieval results for our DeBias-CLIP method on SigLIP and SigLIP2 weights, comparing stretching all position embeddings against keeping the first 20 frozen. We find that stretching all embeddings leads to a significant reduction in short retrieval (-15.3% for Flickr T2I with SigLIP2), so we recommend using the standard Long-CLIP strategy. In

Method	COCO T2I	Flickr T2I	Urban1k T2I	DOCCI T2I
SigLIP				
Stretch all	31.8	27.4	82.9	80.9
Freeze first 20	48.7	40.8	85.6	83.3
SigLIP2				
Stretch all	39.5	28.3	87.0	82.9
Freeze first 20	51.9	43.6	87.2	83.0

Table S2. **Ablation on stretching all embeddings of SigLIP models for text-to-image (T2I) retrieval**. While performance on long retrieval (see Urban1k and DOCCI) does not change substantially, stretching all the embeddings leads to a significant reduction in short-text retrieval performance (see COCO and Flickr).

Epochs	COCO T2I	Flickr T2I	Urban1k T2I	DOCCI T2I
1 (Long-CLIP)	40.4	34.1	79.5	71.4
3 (Long-CLIP)	40.6	33.9	82.7	75.2
1	42.2	36.0	90.1	78.4
3	43.0	36.6	93.0	80.0
5	43.0	36.7	93.5	80.9
7	43.2	36.8	94.1	81.3
10	43.1	36.7	94.2	81.5

Table S3. **Ablation on the number of training epochs for text-to-image (T2I) retrieval**. We observe significant improvements when going from 1 to 3 epochs across all datasets. Extending training from 5 to 7 epochs yields smaller but still notable improvements on long retrieval (Urban1k and DOCCI). Results from 7 to 10 epochs are more marginal and mixed.

practice, we also observe a significant bias towards the first few tokens for SigLIP models (see Sec. 3.2), which further justifies freezing the first 20 embeddings.

C.2. Ablation on Number of Training Epochs

We investigate the effects of training duration by training for 1, 3, 5, 7, and 10 epochs. Results are reported in Tab. S3. We observe that performance improves substantially across datasets when increasing from 1 to 3 epochs ($+2.9\%$ on Urban1k T2I), with more marginal but consistent gains from 3 to 5 and 5 to 7 epochs. Improvements for long retrieval from 7 to 10 epochs are minor (0.2%), and short retrieval is starting to be negatively affected. We also include the original Long-CLIP results for comparison. Notably, even with an equivalent training budget (1 epoch), our DeBias-CLIP method significantly outperforms the Long-CLIP baseline for all datasets, with $+1.8\%$ and $+1.9\%$ for short-text retrieval on COCO and Flickr T2I, and $+10.6\%$ and $+7.0\%$ for long-text retrieval on Urban1k and DOCCI T2I.

Method	COCO T2I	Flickr T2I	Urban1k T2I	DOCCI T2I
Random	43.0	36.6	93.0	79.7
Ordered	43.2	37.6	91.7	80.2
Independent	42.7	36.5	92.7	80.3
Keep 4	42.7	36.4	92.8	80.0
Shuffle	41.7	35.3	92.8	81.1

Table S4. **Ablation on sentence sampling methods for text-to-image (T2I) retrieval**. We find that there is relatively little difference between different sampling methods. The *Ordered* method improves short retrieval at the cost of long retrieval, while the *Shuffle* method improves long retrieval on DOCCI at the cost of short retrieval. We find that our default *Random* sampling offers the best tradeoff between short and long retrieval.

C.3. Ablation on Sentence Sampling Method

Our DeBias-CLIP method generates short captions by first generating a random number of sentences $n_{\text{sampled}} = \mathcal{U}\{1, 2, \ldots, n_{\text{sents}} - 1\}$ from a uniform distribution, and then sampling that number without replacement from the original set of sentences $C^{\text{no\_sum}} = [s_2, \ldots, s_{n_{\text{sents}}}]$ (i.e., with the first sentence excluded), and finally concatenating the sentences together (*Random* method). In this section, we consider four alternative sampling methods:

- *Ordered*: We select a contiguous block of sentences starting from the second sentence up to a randomly selected end sentence, preserving the original narrative order.
- *Independent*: We independently select each sentence in $C^{\text{no\_sum}}$ with a probability $p = 0.5$, creating a more uniform number of sentences.
- *Keep 4*: We fix the sample size to $n_{\text{sampled}} = 4$ while sampling randomly without replacement.
- *Shuffle*: We use the full set of sentences (no subsampling) but randomize their order.

Results are presented in Tab. S4. We find that performance differences between most sampling methods are marginal. Sampling contiguous sentences (*Ordered*) improves short-text retrieval ($+1.0\%$ on Flickr) at the expense of long-text retrieval (-1.3% on Urban1k). This is similar to the SmartCLIP strategy, differing only in that SmartCLIP includes the initial summary sentence (s_1). Conversely, using all sentences (*Shuffle*) boosts long retrieval ($+1.4\%$ on DOCCI) but causes significant degradation in short retrieval (-1.3% on Flickr). Ultimately, we find that our default *Random* sampling offers the best balance between short and long retrieval without requiring additional parameter tuning.

C.4. Ablation on Padding Method

In DeBias-CLIP, we add a random number of padding tokens before the tokenized text tokens from the caption, which depends on the post-caption padding length and leads to

Method	COCO T2I	Flickr T2I	Urban1k T2I	DOCCI T2I
No pad	41.9	35.8	92.5	80.8
Random	43.0	36.6	93.0	79.7
Pad 20	43.1	36.7	92.8	80.3
Pad 40	43.0	36.7	92.8	80.3
Pre-SOT	43.0	36.4	92.1	80.1

Table S5. **Ablation on token padding methods for text-to-image (T2I) retrieval**. All padding methods generally improve short-text performance. Replacing our default random padding (*Random*) with fixed padding length of 20 (*Pad 20*) or 40 (*Pad 40*) yields comparable performance, with only marginal gains on DOCCI T2I ($+0.6\%$), while adding the padding before the `SOT` token leads to slightly worse performance.

padding of varying lengths. Tab. 5 showed that our padding strategy (*Random*) improves on using no padding (*No pad*), and in Sec 4.4, we hypothesize that one of the key benefits comes from shifting the short caption away from the initial token positions. As a result, early-position tokens are mostly trained with the full long caption, and particularly its summary (first) sentence. In this ablation, we consider alternative fixed padding lengths of 20 and 40 tokens (*Pad 20* and *Pad 40*) and present results in Tab. S5. Additionally, while our default method adds padding *after* the `SOT` token (preserving its position), we also evaluate adding padding *before* the `SOT` token (*Pre-SOT*).

We find that all considered padding approaches improve short-text retrieval. Replacing random padding with fixed-length padding generally yields comparable performance, with only a marginal gain on DOCCI T2I ($+0.6\%$). Given this negligible difference, and to avoid introducing an arbitrary fixed hyperparameter, we retain the base random sampling approach. Finally, we find that adding padding after the `SOT` token generally outperforms padding before it (*Pre-SOT*), justifying our default implementation. In Sec. E.3 below, we discuss attention across tokens and highlight that the attention to the `SOT` token is much larger than for other tokens. Because the `SOT` token is present across all training datasets, is text-independent, and is always the first token, it plays a distinct role in the attention mechanism compared to regular text tokens. Moving it away from the first position could disrupt this, leading to worse performance. Recent work has shown that non-text positional tokens can serve as attention sinks in LLMs [1], reducing the magnitude of cross-attention between tokens when it isn't required. A similar effect could be present for CLIP text encoders.

C.5. Transferring to SmartCLIP

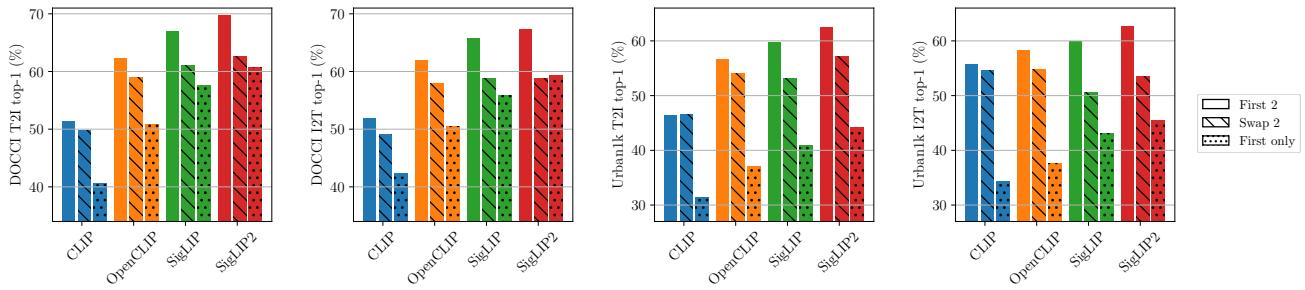
In the main results, we show that the text sampling used in DeBias-CLIP substantially outperforms the Long-CLIP baseline. Here, we apply similar sampling methods to SmartCLIP [7], which was consistently one of the best-performing methods for both long and short retrieval (see Sec. 5.3). SmartCLIP trains with a single caption obtained by truncating the original long caption after a random number of sentences. Thus, it always contains the summary sentence. As alternatives, we consider two sampling methods: one identical to SmartCLIP except that we always remove the summary sentence by starting from the second sentence (*Start at 2*), and one where we keep the whole caption except the first sentence (*Only drop 1*). We train with SmartCLIP's released code, use a ViT-B/16 backbone, and run for 3 epochs. Results are presented in Tab. S6. We find that removing the first sentence consistently improves long-text retrieval performance (*Original* vs *Start at 2*, $+2.7\%$ on Urban1k T2I), with mixed results on short-text retrieval, with$+0.7\%$ on COCO T2I but -0.5% for I2T (not in the table), and better performance on Flickr. We can further improve long retrieval by not subsampling the caption sentences (*Only drop 1*), at the cost of some short-text performance. However, the final short retrieval results remain similar to the original SmartCLIP while being much better ($+5.1\%$ on Urban1k T2I) for long-text retrieval. Overall, our results on SmartCLIP highlight the importance of not training with the summary sentence.

Method	COCO T2I	Flickr T2I	Urban1k T2I	DOCCI T2I
Original	42.4	36.3	87.4	78.0
Start at 2	43.1	37.4	90.1	79.7
Only drop 1	42.2	36.3	92.5	81.8

Table S6. **Ablation on caption sampling with the SmartCLIP training on text-to-image (T2I) retrieval.** We obtain consistently better performance compared to the original SmartCLIP by not sampling the first (summary) sentence. There are clear tradeoffs between short-text (COCO and Flickr) and long-text (Urban1k and DOCCI) retrieval when subsampling the sentences instead of keeping the full original caption.

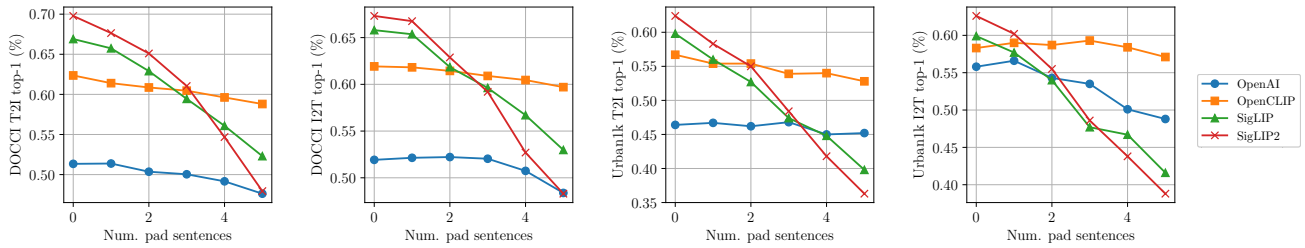
D. Additional Discussion on CLIP Biases

We extend the empirical analysis from Sec. 3 with additional results on DOCCI and Urban1k. We continue our analysis of retrieval with the first two sentences of long-caption datasets (see Sec. 3.2), comparing three configurations: using both sentences in order (*First 2*), both sentences with the order swapped (2 before 1, *Swap 2*), and using only the first sentence (*First only*). For DOCCI, the average token count for the first two sentences is 46.6, 43.8, and 40.4 for CLIP, SigLIP, and SigLIP2 tokenizers, respectively. For Urban1k, these averages are 49.5, 46.5, and 42.1, respectively. Results are presented in Fig. S1. When comparing models, we broadly observe the same trends, with significant performance drops in the *Swap 2* case, particularly for SigLIP and SigLIP2. We observe slightly different trends



(a) DOCCI T2I (b) DOCCI I2T (c) Urban1k T2I (d) Urban1k I2T

Figure S1. **Effects of permutation of the first two sentences on the top-1 text-to-image (T2I) and image-to-text (I2T) retrieval performance for CLIP models.** We present results on DOCCI in (a) and (b), and on Urban1k in (c) and (d). We analyze three setups: the first two sentences in the correct order (*First 2*), the same two sentences swapped (*Swap 2*), and the first sentence only (*First only*). Sentence swapping generally leads to worse retrieval performance and tends to be more severe for the SigLIP and SigLIP2 models. Dropping the second sentence leads to significantly larger degradation on Urban1k compared to DOCCI.



(a) DOCCI T2I (b) DOCCI I2T (c) Urban1k T2I (d) Urban1k I2T

Figure S2. **Effects of padding sentences on the top-1 text-to-image (T2I) and image-to-text (I2T) retrieval performance for CLIP models.** We present results on DOCCI in (a) and (b), and on Urban1k in (c) and (d). One to five padding sentences `'This is a photo.'` are added before the truncated original caption (we keep the first two sentences only). Increased padding consistently leads to worse retrieval performance, but this effect is much more severe for the SigLIP and SigLIP2 models.

between retrieval on DOCCI and Urban1k. OpenAI CLIP is less sensitive to sentence permutation on Urban1k ($+0.1\%$ and -1.3% for Urban1k T2I and I2T, respectively), but this could be due to lower performance in general. Furthermore, performance for the *First only* setting on Urban1k is substantially worse than the *First 2* baseline (-9.5% on DOCCI I2T and -21.5% on Urban1k I2T for OpenAI CLIP), indicating that additional text is critical for retrieval on this dataset. This could be due to Urban1k images being mostly limited to ground-level urban scenes, compared to the higher diversity of DOCCI. Consequently, many Urban1k captions likely share similar first (summary) sentences, making the second sentence essential for disambiguation.

Fig. S2 presents additional results for the padding effects on DOCCI and Urban1k. We see similar trends across models and datasets, with CLIP models (OpenAI and OpenCLIP) exhibiting significantly less sensitivity to padding compared to SigLIP models. While SigLIP models perform better than CLIP variants in the no-padding setup, this advantage does not hold for longer padding sequences. This aligns with our results for sentence transposition and implies that SigLIP models have a very narrow effective context window.

E. Additional Discussion on Long-CLIP Biases

E.1. Long-Text Retrieval with Sentence Permutation and Removal

In sections 3.3 and 5.3, we showed that our DeBias-CLIP method achieves better long-text retrieval performance and is more robust than Long-CLIP to sentence transposition or to erasing the first sentence. Here, we extend the comparison to SmartCLIP, which is a consistently strong model for both long- and short-text retrieval. We repeat the text permutation analysis from Sec. 5.3.3 but now consider ViT-L models and additionally evaluate on both DOCCI and Urban1k to show that these results generalize broadly. We consider four text augmentation cases: the original full caption (*Keep*), transposing the first and second sentences (*Move 2*), transposing the first and fourth sentences (*Move 4*), and removing the first sentence (*Remove*). If there are fewer than four sentences, we transpose the last sentence with the first. We present

results in Fig. S3. First, we note that DeBias-CLIP is consistently better in the nominal case (*Keep*) and substantially more robust to sentence permutations. On DOCCI T2I, the *Move 4* case leads to a reduction of -8.7% for Long-CLIP and -6.9% for SmartCLIP, but only -2.0% for DeBias-CLIP. Even when removing the first sentence, DeBias-CLIP loses only -12.1%, while Long-CLIP and SmartCLIP lose -18.3% and -15.9%, respectively, on DOCCI T2I. On Urban1k T2I, DeBias-CLIP loses only -10.8% compared to -20.2% for Long-CLIP and -20.1% for SmartCLIP, showing that our model is significantly better at using details in the later sentences of the caption to match images without relying on the summary sentence information.

E.2. Encoder Design to Resolve the Early Token Bias

Our method resolves the early token bias issue from a data perspective by avoiding training with the summary captions. In this section, we investigate two possible encoder design solutions to reduce the bias: using relative positional embedding instead of the absolute positional embedding, and using token average pooling as the output token instead of aggregating information to a single token.

TULIP [4] proposes to learn relative positional embedding, particularly rotary positional embeddings (RoPE) [6], instead of linearly interpolating the original CLIP absolute positional embeddings like Long-CLIP. RoPE is commonly used in language models and has been shown to have better interpolation and extrapolation properties. TULIP shows significant improvements over Long-CLIP, but it is unclear whether this is because the method resolves the early token bias or because the model is generally better. Thus, we evaluate TULIP ViT-L in the sentence permutation setting considered in Sec. 5.3.4 (*Move* permute first and fourth sentences, *Remove* remove first sentence from caption) and show results in Fig. S4. While TULIP is consistently better than Long-CLIP, performance is significantly worse when the first sentence is moved (-7.3%) or removed (-16.1%) when compared to our DeBias-CLIP (-2.0% and -12.1% respectively). This shows that the bias problem persists even when using relative positional embeddings.

Next, we consider using token average pooling to generate the final text output token, compared to aggregating information in a single token (the `EOT` token for CLIP). This would force the final text output to be explicitly dependent on all the encoded tokens, potentially making it more sensitive to information later in the caption. We train both Long-CLIP and DeBias-CLIP ViT-B for 3 epochs with token averaging (*Avg*) and again consider the sentence permutation of Sec. 5.3.4, comparing them to the original CLIP single token aggregation (*Orig*) in Fig. S5. We see that average pooling generally reduces performance by a small amount (about -2% for Long-CLIP, -1% for DeBias-CLIP) and does not improve the issue of early token bias. For the original Long-CLIP, we see a drop of -10.4% when permuting the first and fourth sentences (*Move*), and a drop of -11.0% with average pooling.

E.3. Additional Details and Results on the Attention Weights Distribution

In Fig. 1 (b), we present a plot of the average attention weights over tokens for both Long-CLIP and our DeBias-CLIP method. We provide additional details here. Our goal is to show that Long-CLIP models are biased towards early tokens in the caption. To this end, we measure the token self-attention values in the attention matrix *before* the softmax, and consider specifically the attention from the output token (analogous to the `CLS` token for the image encoder and it is the `EOT` token for the CLIP text encoder) to all other positional tokens in the last transformer layer, which captures how much information from the other positions is used to update the output token. We use the pre-softmax values as these directly capture the token-to-token affinity without the effects of the softmax normalization.

For a single caption, the query $\mathbf{Q} \in \mathbb{R}^D$ and key $\mathbf{K} \in \mathbb{R}^D$ matrices are used in self-attention (before softmax) as

$$\mathbf{M} = \left(\frac{\mathbf{Q}\mathbf{K}^\mathsf{T}}{\sqrt{D}} \right), \tag{1}$$

where $\mathbf{M} \in \mathbb{R}^{D \times D}$ is the self-attention weight matrix across all token positions and D is the hidden dimension. We are interested in the row of matrix $\mathbf{M}$ corresponding to the output token (i.e., `EOT` position) $\mathbf{m}_{\texttt{EOT}}$ given by

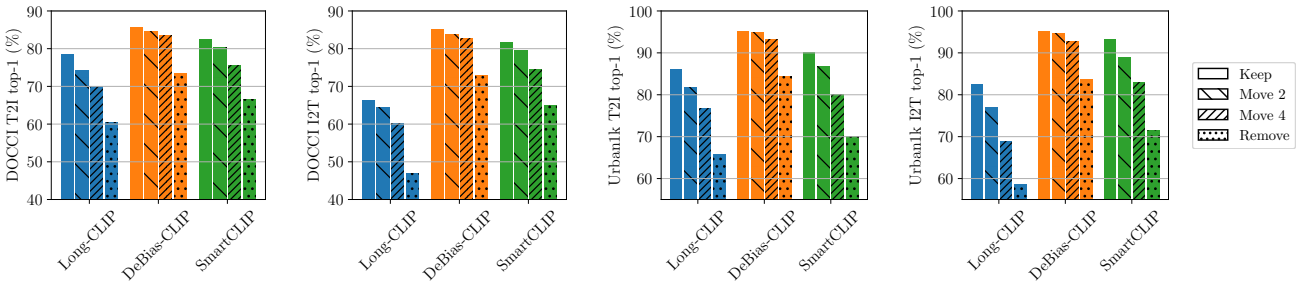
$$\mathbf{m}_{\texttt{EOT}} = \frac{\mathbf{q}_{\texttt{EOT}}\mathbf{K}^\mathsf{T}}{\sqrt{D}}, \tag{2}$$

which is the vector of attention weights from the output token to all other tokens. Because CLIP models use a causal mask in the text encoder, padding tokens that appear after the `EOT` token have an attention weight of $-\infty$ (this becomes a weight of 0 after the softmax), so $\mathbf{m}_{\texttt{EOT}}$ has the form $\mathbf{m}_{\texttt{EOT}} = [v_{\texttt{SOT}}, v_1, \ldots, v_k, v_{\texttt{EOT}}, -\infty, \ldots, -\infty]$.

In Fig. 1 (b), we plot this value aggregated over the DOCCI dataset. For each token position, we sum the corresponding entries of $\mathbf{m}_{\texttt{EOT}}$ over all captions and divide by the number of times that position corresponds to a text token (rather than post-text padding), giving a normalizing vector $\mathbf{n}$:

$$\begin{aligned} \mathbf{n} &= [n_1, n_2, \ldots, n_{248}] \\ &= \sum_{i=1}^{N} \mathbf{m}_{\texttt{EOT},i} > -\infty, \end{aligned} \tag{3}$$

where N is the number of samples in the dataset, and where the inequality is applied independently for each element of $\mathbf{m}_{\texttt{EOT},i}$. This yields a vector of binary elements for each caption in the dataset. The normalization excludes the padding



(a) DOCCI T2I ViT-L (b) DOCCI I2T ViT-L (c) Urban1k T2I ViT-L (d) Urban1k I2T ViT-L

Figure S3. **Comparison of long-text top-1 retrieval with sentence permutations for Long-CLIP [8], SmartCLIP [7] and our proposed method (DeBias-CLIP).** We consider four cases: the original full caption (*Keep*), transposing the first and second sentences (*Move 2*), transposing the first and fourth sentences (*Move 4*), and removing the first sentence (*Remove*). We present results on DOCCI in (a) and (b), and on Urban1k in (c) and (d). Sentence permutation generally leads to worse retrieval performance, but our method demonstrates greater robustness compared to either Long-CLIP or SmartCLIP.

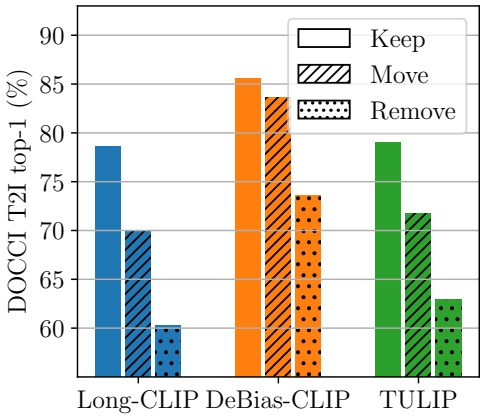


Figure S4. **Effects of permutations of the first two sentences on top-1 text-to-image retrieval for TULIP ViT-L.** While TULIP is consistently better than Long-CLIP, it still shows significant performance degradation when sentences are permuted (*Move*) or the first sentence is removed (*Remove*).

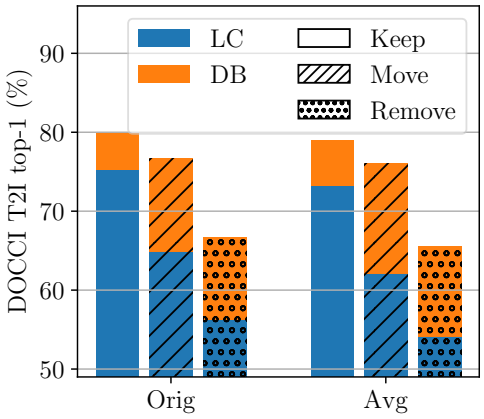


Figure S5. **Effects of permutations of the first two sentences on top-1 text-to-image retrieval with token average pooling for Long-CLIP (LC) and DeBias-CLIP (DB).** Token average pooling generally does slightly worse, and does not reduce early token bias.

tokens, which are more frequent for later embedding positions, and captures the average value of attention weights when they correspond to text tokens. In Fig. 1 (b) and in the following figures, we exclude the weight with respect to the SOT token $v_{\texttt{SOT},i}$, which is generally much larger (on the order of 2.5), and corresponds to a softmax probability between 0.35 and 0.40.

Similar to how our method improves Long-CLIP by assigning more consistent attention weights to later text tokens, Fig. S6 repeats the analysis for SmartCLIP, showing it has a less steep drop in attention over token positions compared to Long-CLIP, which is consistent with its improved long-text retrieval performance. However, there is still substantially less attention on later tokens in SmartCLIP. Finally, when evaluated on DOCCI, where the first ∼25 tokens correspond to the summary sentence, we see that the earliest text positions consistently have larger attention weights on average, even for DeBias-CLIP. This could be related to a positional bias toward earlier tokens and/or a bias towards the summary sentence.

To disentangle these effects, we replicate the experiment under the *Move 4* setting, where we transpose the first and fourth sentences. Results in Fig. S7 show that (i) a positional bias remains, with significantly larger attention values for early tokens, and (ii) the summary sentence continues to receive more attention (as it likely contains more information) even when moved to a later position. In fact, we see an increase of about 0.2 in the attention values from token positions 50 to 100 in the *Move 4* setting. Additionally, we

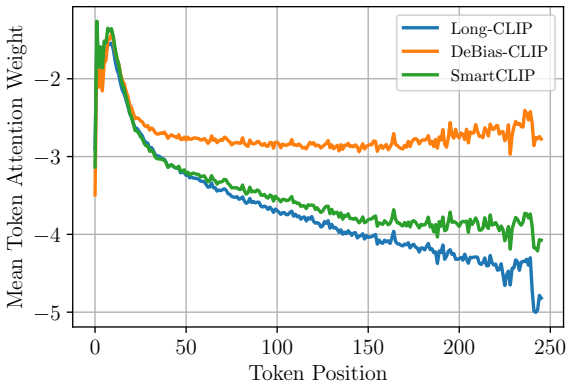


Figure S6. **Mean *pre-softmax* self-attention weight from the output token to the text tokens as a function of the token position on DOCCI.** For both Long-CLIP and SmartCLIP, the attention magnitude consistently decays beyond 50 tokens, while for DeBias-CLIP it remains essentially flat (ignoring high-variance values at the very end).

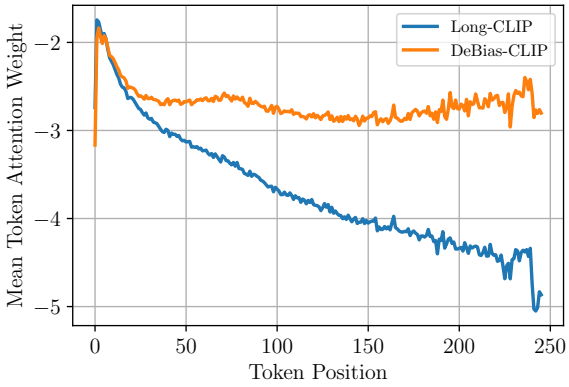


Figure S7. **Mean *pre-softmax* self-attention weight from the output token to the text tokens as a function of the token position on DOCCI, with the first and fourth sentences transposed.** We see similar trends as in the untransposed case: there is a significant bias toward early tokens even if the summary sentence is moved to a later position. The summary sentence visibly receives more attention even after being moved (around tokens 50-100).

present the post-softmax attention weights in Fig. S8. After softmax normalization, the larger weights on later tokens in DeBias-CLIP lead to a redistribution of the attention from the first few tokens to the rest of the caption.

F. Visualizations and Results for Image Generation

We claim that DeBias-CLIP is more sensitive to caption details than Long-CLIP, particularly in later sentences. Fig. S9 presents examples on Urban1k, DCI, and DOCCI, where Long-CLIP fails to retrieve the correct image given the long caption, but where DeBias-CLIP succeeds. In general, these

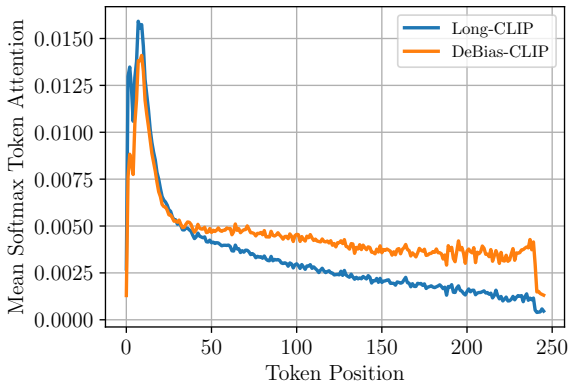


Figure S8. **Mean *post-softmax* self-attention weight from the output token to the text tokens as a function of the token position on DOCCI.** DeBias-CLIP reduces the bias on early tokens and has a relatively even attention spread on later tokens, while we observe a decay for Long-CLIP.

are cases where the first summary caption applies equally to either image, and where specific details in the caption (e.g., presence of umbrellas, oil containers, the color of the cat's paws) are required to identify the matching image.

We also investigate how our DeBias-CLIP models perform on downstream tasks. In particular, we replace the CLIP text encoder from SDXL [5] with DeBias-CLIP. Fig. S10 compares text-to-image generation with the original CLIP, Long-CLIP, and DeBias-CLIP, with prompts from ShareGPT4V and Urban1k captions. The generation results show that CLIP struggles with long captions and often fails to capture key details, such as the Japanese text in the first example, or the color of the dinosaurs in the second. When comparing DeBias-CLIP to Long-CLIP, we find that DeBias-CLIP preserves more fine-grained details (e.g., the green traffic light in the third example). However, generation with long captions also highlights a broader limitation of all CLIP models: they suffer from poor relational understanding, with limited ability to encode relative position (left/right), accurate object counts, and consistent object-attribute matches. For example, the blue dinosaur generated by DeBias-CLIP in row 2 is an incorrect amalgamation of an orange dinosaur holding a blue toothbrush. Quantitatively, we compare the performance of DeBias-CLIP with Long-CLIP by measuring the similarity between the original images and images generated with the 1000 paired long caption from the ShareGPT4v validation split. We evaluate Fréchet inception distance (FID) [3] and the DINO score [2]. Tab. S7 shows that DeBias-CLIP improves both metric, showing we generate images from the long captions that are semantically closer to the paired images.

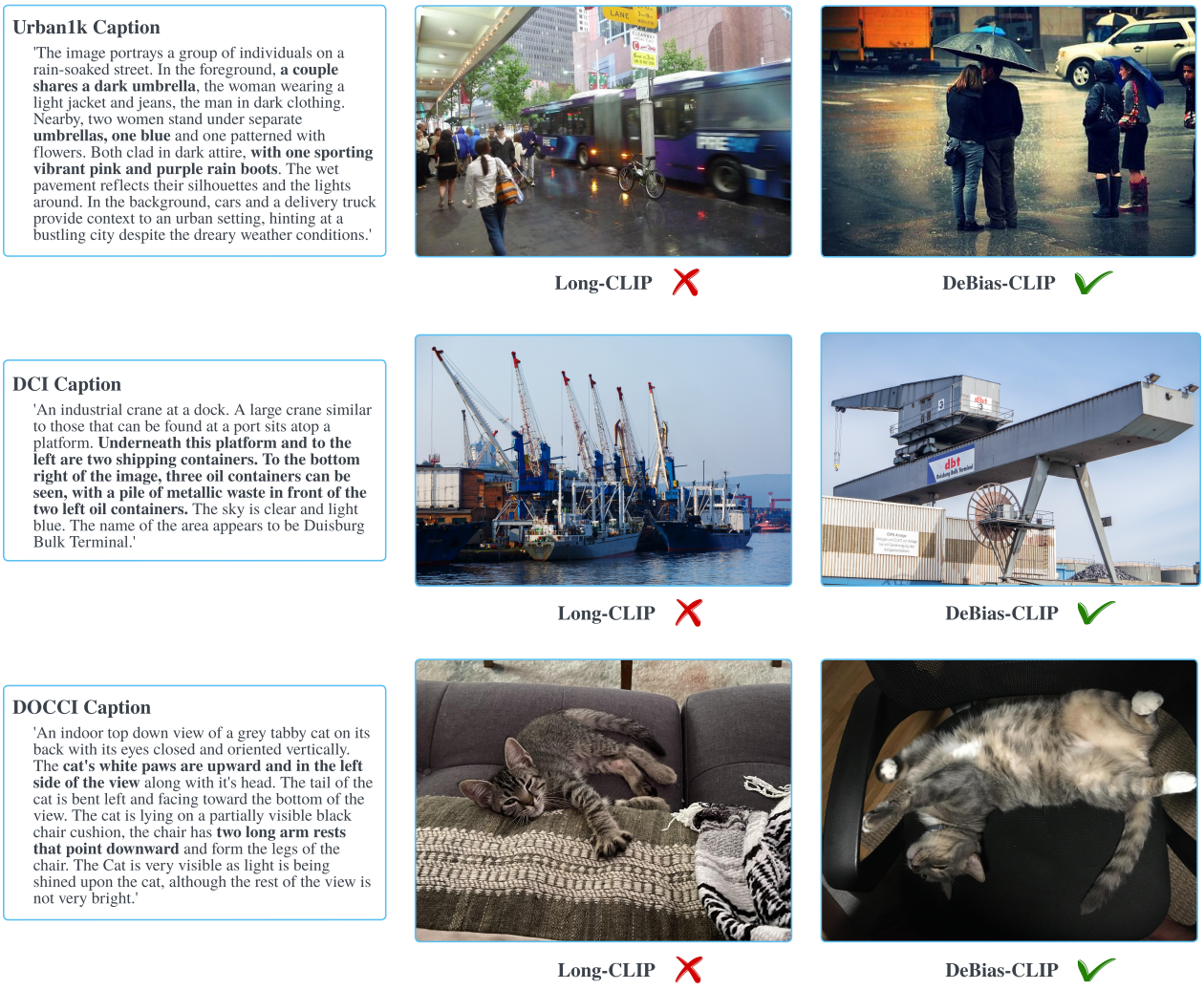


Figure S9. **Comparison between Long-CLIP and DeBias-CLIP on long text-to-image retrieval.** We observe that DeBias-CLIP more consistently recovers images that are correctly matched to the text. Text in bold marks the caption details that are required to retrieve the right image.

Model	FID↓	DINO Score↑
Long-CLIP	0.335	0.515
DeBias-CLIP (ours)	0.317	0.528

Table S7. **Quantitative image generation comparison**. We replace the CLIP text encoder in SDXL with either Long-CLIP or DeBias-CLIP generate images from the 1000 image-caption pairs of the ShareGPT4V validation set. DeBias-CLIP generates images that are semantically more similar to the original images.

References

[1] Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. Why do llms attend to the first token? *arXiv preprint arXiv:2504.02732*, 2025. 3

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 7

[3] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7

[4] 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. 5

[5] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach.

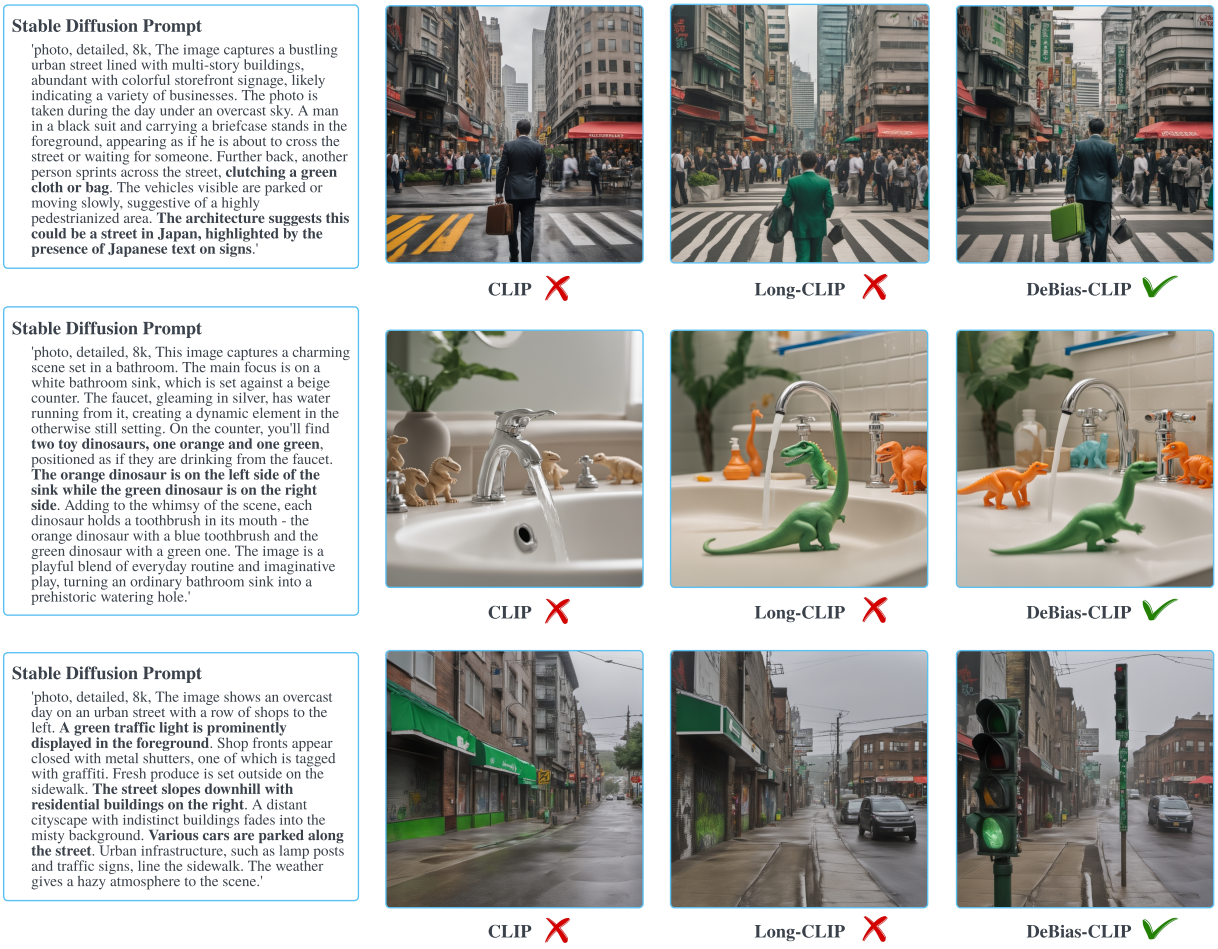


Figure S10. **Stable Diffusion text-to-image generation results for CLIP, Long-CLIP, and DeBias-CLIP.** DeBias-CLIP can leverage more details that appear later in the caption and suffers less from inaccurate localization of details (e.g., the green color of the bag being transferred to the suit in the Long-CLIP first row image). Text in bold highlights the caption details that are correctly represented in our DeBias-CLIP method.

Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024. 7

[6] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 5

[7] Shaoan Xie, Lingjing Lingjing, Yujia Zheng, Yu Yao, Zeyu Tang, Eric P Xing, Guangyi Chen, and Kun Zhang. Smart-clip: Modular vision-language alignment with identification guarantees. In *CVPR*, pages 29780–29790, 2025. 3, 6

[8] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip. In *ECCV*, pages 310–325. Springer, 2024. 1, 6