STORI: Semantic Token Reweighting for Interpretable and Controllable Text Embeddings in Vision-Language Models

Anonymous ACL submission

Abstract

A text encoder within Vision-Language Models (VLMs) plays a crucial role in translating textual input into an embedding space shared with images, thereby facilitating the interpretative analysis of vision tasks through natural language. Despite varying significance of different textual elements within a sentence depending on the context, efforts to account for variation of importance when constructing text embeddings have been lacking. This paper proposes Semantic Token Reweighting to build Interpretable text embeddings (SToRI), which incorporates controllability as well. SToRI refines the text encoding process in VLMs by differentially weighting semantic elements based on contextual importance, enabling finer control over emphasis responsive to user preferences and data-driven insights. The efficacy of SToRI is demonstrated through comprehensive experiments, showcasing its strength in image retrieval tailored to user preferences and its capability in few-shot image classification tasks.

1 Introduction

As artificial intelligence (AI) systems based on deep learning models grow in application in our daily lives, their black box nature raises issues of transparency, resulting in a demand for enhanced interpretability to promote trust in AI systems (Murdoch et al., 2019; Li et al., 2022). Consequently, research efforts have been focused on making the systems' decision-making processes more human-understandable through various explanatory methods (Simonyan et al., 2014; Kim et al., 2018; Goyal et al., 2019; Wu and Mooney, 2019). Among the various forms of explanation, natural language has emerged as an excellent medium due to its human-friendly nature and adeptness in managing high-level abstractions (Kayser et al., 2021; Sammani et al., 2022). These advantages have led to a growing interest in research that utilizes natural language for interpretative analysis, extending even to domain of vision tasks (Hendricks et al., 2021; Yang et al., 2023). To facilitate the use of natural language in vision tasks, Vision-Language Models (VLMs) like CLIP (Radford et al., 2021) are commonly deployed to bridge visual information and its linguistic interpretation (Yuksekgonul et al., 2023; Yang et al., 2023; Oikarinen et al., 2023). Two encoders of VLMs translate an input image and text into image and text embeddings, respectively, which take vectorized forms and coexist in a shared embedding space.

Natural language sentences often carry multiple implications, with varying levels of significance that can change based on the desired outcome, even if the text remains unchanged. For instance, when searching for images using the query 'a castle surrounded by trees,' a standard text query might bring up relevant images, but the preference on 'trees' relative to 'a castle' could differ based on user intent (see examples of retrieved images in Figure 1). Texts rich in detail may benefit from selectively emphasizing certain information relevant to the task. While there have been attempts to modulate focus in image and text generation (Ge et al., 2023; Zhang et al., 2024), there remains a lack of efforts to fine-tune the importance given to specific pieces of information within text embeddings from VLMs. This paper endeavors to create text embeddings that can incorporate a varying controlled importance of each semantic element within a sentence.

To meet our objective, we introduce **SToRI** (Semantic Token Reweighting for Interpretable text embeddings), which refines the focus on individual semantic components during text embedding extraction in VLMs. Each semantic element is assigned a numerical weight, denoting its significance, and these weights modulate the self-attention mechanism in text encoding. The proposed method makes it possible for the final text embedding vector to naturally include the desired emphasis on specific semantic elements, allowing

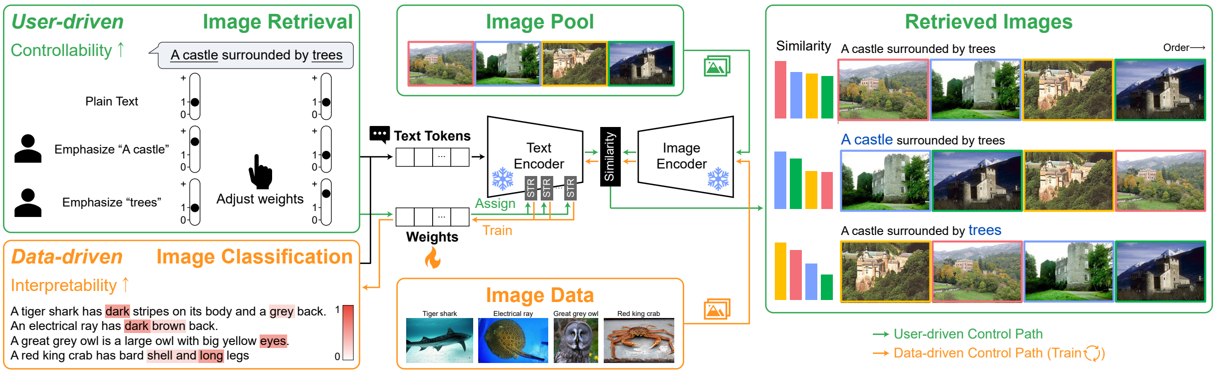

Figure 1: System diagram of SToRI. SToRI enables *user-driven control* over the multiple images by allowing fine-grained manipulation of the text prompts. It also facilitates *data-driven control* through interpretable weight optimization in the semantic space, enhancing the classification performance of the image data. Weight affects text embeddings via semantic token reweighting (STR).

for controllability. Moreover, the emphasis on particular semantic meanings remains within the realm of interpretability. SToRI efficiently produces text embeddings that reflect the desired focus without necessitating the training of new modules.

Our framework enables text embeddings to be tailored in two ways: user-driven and data-driven. In the user-driven approach, individuals can set the weight for each semantic token, allowing them to emphasize the elements they consider most relevant and customize the model to fit their preferences, as shown by the green path in Figure 1. On the other hand, the data-driven method derives token weights from training on dataset, facilitating the creation of text embeddings that are optimized for specific tasks like image classification and offer interpretable insights into the classifiers derived from texts, as shown by the orange path in Figure 1. These enhancements have been substantiated through evaluation across various image recognition tasks, including image retrieval and few-shot classification.

Our main contributions are outlined as follows:

- We propose a novel framework of semantic token reweighting, which differentiates the importance of textual information during the construction of text embeddings in VLMs.
- Our approach facilitates the customization of emphasis on specific semantics, and we demonstrate its usefulness in image retrieval tasks with a new metric for controllability.
- We demonstrate that our methodology not only builds improved text classifier in few-shot learning tasks but also unlocks a new dimension of interpretability.

2 Preliminary: Text embeddings in CLIP

The text encoder of CLIP (Radford et al., 2021), which utilizes a transformer-based architecture, transforms a given text prompt into a single vector through the following process. Initially, a given text prompt is converted into a sequence of text tokens $\{x_i\}_{i=1}^{N}$, where N represents the number of the text tokens. Tokens indicating the start and end, `[SOS]` and `[EOS]` tokens, are appended at the beginning and the end of the sequence of tokens, resulting in the extended series $\{x_i\}_{i=0}^{N+1}$, with x_0 and x_{N+1} representing the `[SOS]` and `[EOS]`, respectively. Each text token is then converted into an embedded input token, and positional embedding is added, resulting in the input embedding for the first transformer block $\{z_i^0\}_{i=0}^{N+1}$. For the l-th block of the encoder, the input tokens can be represented as $Z^{l-1} = [z_0^{l-1}, ..., z_{N+1}^{l-1}]$. The output tokens from the l-th block is given by:

$$Z^l = \text{Block}^l(Z^{l-1}), \tag{1}$$

where $l \in [1, L]$ with the encoder consisting of L blocks. Each block contains a multi-head self-attention mechanism. First, Z^{l-1} is projected into the query Q, key K, and value V. Then, the attention process is performed as follows:

$$\begin{aligned} &\text{Attention}(Q, K, V) = AV, \\ &\text{s.t. } A = \text{softmax}(QK^T). \end{aligned} \tag{2}$$

Scaling and masking operations are omitted for simplicity. Through the attention mechanism, tokens influence each other, and the values of A represent the extent to which they influence one another (Vaswani et al., 2017). In general, the final output text embedding of the [EOS] token encapsulates the full semantic meaning of the text prompt. This embedding is compared with image embeddings to assess the degree of correspondence with images once it has been projected into a multimodal embedding space.

A pre-trained CLIP model is commonly employed for image classification, where given an image, it computes similarity scores with class names, which become logits. To adapt the model to a specific dataset, fine-tuning is performed by minimizing the cross-entropy loss as follows:

$$\mathcal{L} = L_{\text{CE}}(y, \text{sim}(\phi_T, \phi_I)/\tau), \tag{3}$$

where ϕ_T and ϕ_I represent output text and image embeddings from two encoders, respectively, and τ is a temperature factor.

3 Method

We propose SToRI, a novel framework that adjusts the importance of various textual elements while encoding a given text prompt into a single text embedding vector within VLMs. The weights are determined through user-driven and data-driven controls. In Section 3.1, we elaborate semantic token reweighting, which involves modifying the attention given to individual tokens within the text encoding process based on their respective weights. In Section 3.2, we present two methods for determining these weights. Figure 2 presents an overview of our comprehensive framework.

3.1 Semantic Token Reweighting

In natural language processing, a given text is tokenized prior to encoding, resulting in one or more tokens. Consequently, to emphasize or de-emphasize a particular semantic element, one must focus on the corresponding tokens. Henceforth, our discussion will center on the process of reweighting in terms of these tokens.

Given a sequence of text tokens $\{x_i\}_{i=1}^N$, we first define a sequence of weights $\{w_i\}_{i=1}^N$, where w_i is the level of significance of token x_i. Note that $w_i = 1$ indicates a typical weight in common situations, where x_i is neither emphasized nor de-emphasized. Our goal is to modulate the impact each token has on the final output embedding of the text prompt. As elaborated in Section 2, tokens

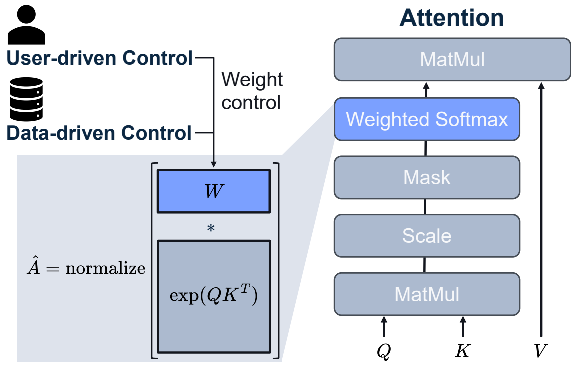

Figure 2: Overview of semantic token reweighting. The weights can be determined through either user-driven or data-driven control. The weight vector is represented as $W = [w_1, ..., w_N]$.

interact with each other through attention mechanisms. Each token generates its embedding by referencing other tokens, including itself, in proportion to the attention scores. Consequently, as the attention score of a specific token increases, its influence on the text embedding becomes more substantial. Therefore, we directly multiply the weights $\{w_i\}_{i=1}^N$ to amplify original attention values proportionally. From Eq. (2), the weighted attention scores can be reformulated as follows:

$$\hat{a}_{m,n} = \frac{w_n \exp\left(q_m k_n^T\right)}{\sum_j w_j \exp\left(q_m k_j^T\right)}, \tag{4}$$

where $\hat{a}_{m,n}$ represents attention value for n-th value token to be attended by m-th query token. q_m and k_n represent vector elements of Q and K, respectively. Through this process, we can selectively enhance the influence of particular tokens during the attention process by simply changing the corresponding weights.

The reweighting process is applied to all blocks following a certain block. Experimentally, we confirm that the effects are similar regardless of starting from any intermediate block. Please refer to Appendix B.6 for further details.

3.2 Strategies to Control

There are two approaches to determine weights for tokens: user-driven and data-driven controls.

User-driven control applies to scenarios where the user assigns weights to each token. This method allows user to determine a particular textual information to be emphasized or de-emphasized according to their intentions, thereby influencing the resulting text embeddings. The green path in Figure 1 presents examples of preference-based image retrieval, an application in the user-driven control. Users may initially set a text prompt and then progressively amplify the weight of keywords perceived as more crucial, assess the resulting arrangement, and refine their selection accordingly.

Data-driven control determines weights by learning from data. This approach is suitable when data is available and we want to obtain text embeddings that align closely with the data. An illustrative task where this can be effectively applied is image classification (see the orange path in Figure 1). In image classification, weights are trained using Eq. (3), where ϕ_T is obtained with $\hat{a}_{i,j}$, allowing only $\{w_i\}_{i=1}^{N}$ to be updated. Since the weights are trained to build text embeddings that correspond well to image belonging to their corresponding classes, we can interpret which textual information prominently stands out in the image data with the weights.

4 Experiments

We evaluate SToRI under two scenarios: user-driven and data-driven controls. In the user-driven scenario, we demonstrate its application in preference-based image retrieval. In the data-driven scenario, we show its effectiveness in training an enhanced classifier for few-shot image classification and interpreting the classifier through its weights.

4.1 User-driven Control

To assess the effectiveness of SToRI in emphasizing or de-emphasizing specific information based on applied weights, we compare the ordering of retrieved images using text embeddings.

4.1.1 Experimental Setup

Dataset. We use CelebA (Liu et al., 2015) and CUB (Wah et al., 2011) datasets. The CelebA dataset contains over 200K face images, each annotated with 40 attributes. The CUB dataset contains over 11K bird images, which are annotated with 312 attributes. Three attributes are chosen to create eight categories based on their presence or absence. For the CelebA dataset, each category comprises 100 randomly selected images, resulting in a total of 800 images. For the CUB dataset, all images are used. For more details, please refer to Appendix A.1.

Image Retrieval with Preference. We construct a text prompt containing the selected attributes.

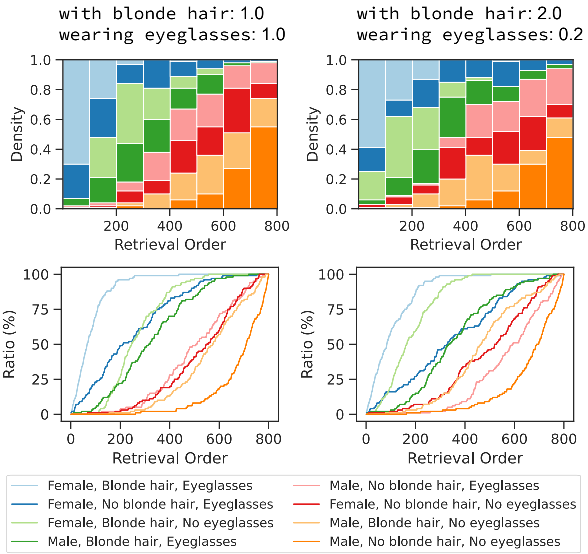

Figure 3: Results of preference retrieval using the text prompt `'a photo of a woman with blonde hair, wearing eyeglasses'`. The first row shows density plots with the retrieval order, and the second row visualizes the ratio of retrieved samples within each category. The left column shows results from a plain text prompt, whereas the right column depicts the results when the weights are adjusted. Best viewed in color.

For instance, the text prompt becomes `'a photo of a woman with blonde hair, wearing eyeglasses'` for the attributes *female*, *blonde hair*, and *eyeglasses*. Using the text prompt and attribute weights, we obtain a corresponding text embedding through SToRI, followed by sorting the images in descending order of similarity between their image embeddings and the text embedding.

Model. Most experiments are conducted using CLIP ViT-L/14 (Radford et al., 2021), unless otherwise specified. Experiments are also conducted using various VLMs, including OpenCLIP (Cherti et al., 2023) and MetaCLIP (Xu et al., 2023). Reweighting is applied from the 7th block.

4.1.2 Metric for Preference Retrieval

Our primary focus is on observing how adjusting weights for specific semantic elements affects the image retrieval order. To facilitate this comparison, we report the average precision score (AP) and precision at rank k (P_k) for images with the attributes influenced by the adjusted weights. For instance, when we modify the weight on `'eyeglasses'`, we consider images with eyeglasses as positive samples and calculate AP and P_k.

Additionally, we introduce a novel metric to quantify priority in preference retrieval. We gener-

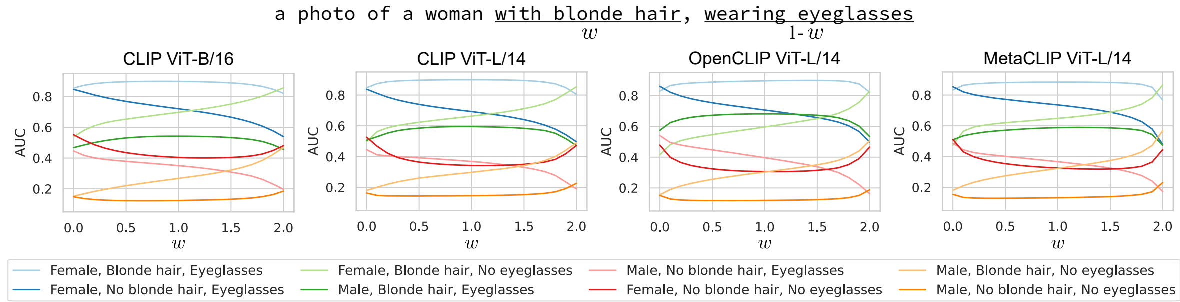

Figure 4: AUC scores from preference retrieval with varying weights. The text prompt is `a photo of a woman with blonde hair, wearing eyeglasses`. The weights on `with blonde hair` and `wearing eyeglasses` are w and $(1 - w)$, respectively, which are adjusted simultaneously in opposite direction. Best viewed in color.

	CelebA AP	CelebA P_{400}	CUB AP
Plain ($w = 1.0$)	0.752±0.089	0.679±0.084	0.154±0.070
Emphasized ($w = 1.5$)	0.773±0.084 **Δ0.021±0.011**	0.697±0.068 **Δ0.017±0.009**	0.183±0.079 **Δ0.029±0.018**
De-emphasized ($w = 0.5$)	0.709±0.096 **Δ-0.043±0.021**	0.648±0.072 **Δ-0.031±0.031**	0.116±0.057 **Δ-0.038±0.021**

Table 1: Retrieval performance on attributes of the CelebA and CUB datasets with CLIP ViT-L/14. The results show mean values with standard deviation across multiple controlled attributes.

ate a line plot illustrating the proportion of images retrieved for each attribute combination up to the n-th retrieved image (see Figure 4), and calculate the Area Under the Curve (AUC) for each plotted curve. A higher AUC value suggests a faster retrieval of associated visual attribute set, indicating a higher priority in the retrieval process.

4.1.3 Results

Initially, we select three attributes, *female*, *blonde hair*, and *eyeglasses*, and observe the ordering of image retrieval as shown Figure 3. With the plain text embedding, the initial bin predominantly contains images featuring all selected attributes, followed by a prevalence of images from the 'female, no blonde hair, eyeglasses' category. When the weight on `with blonde hair` increases and on `wearing eyeglasses` decreases, images belonging to 'female, blonde hair, no eyeglasses' are retrieved more prominently. This suggests that the 'blonde hair' gains more representation in the text embedding through reweighting. The groups with two or more mismatched attributes still rank lower, indicating that our method preserves the meanings of the original text while appropriately reflecting the intention of emphasis and de-emphasis.

We conduct quantitative validation across various text prompts. Table 1 presents AP and P_{400} scores while controlling weights on attributes. We generate image pools and text prompts from three selected attributes. The reported scores are based on adjusting the weight for one specific attribute, considering the images containing that attribute as positive samples. Various combinations of attributes, totaling 20 text prompts for the CelebA dataset and 58 text prompts for the CUB dataset, are used to obtain scores, and their averages and standard deviations are reported. Further details are in Appendix A.1. The results show that modifying the weight of tokens corresponding to a specific attribute in the text prompt results in faster retrieval of images with that attribute (both scores become higher) when the weight increases and slower retrieval when decreases (both scores become lower). This shows that adjusting the weight influences the creation of text embeddings, effectively highlighting or downplaying the corresponding attribute. Additional results on more complex scenarios, including those with MetaCLIP, are in Appendix B.2.

Figure 4 demonstrates the effects of weight control on the AUC scores for the retrieval of each category. As the weight assigned to the `with blonde hair` increases and the weight for `wearing eyeglasses` decreases, there is a noticeable rise in the AUC scores for the two categories that have blonde hair but no eyeglasses. In contrast, categories characterized by the absence of blonde hair and the presence of eyeglasses see a reduction in their AUC scores. When the weight assigned to `with blonde hair` is set to zero, the differentiation between the 'female, blonde hair, eyeglasses' and 'female, no blonde hair, eyeglasses' categories is effectively eliminated, resulting in re-

	Method	Text	ImageNet	DTD	Flowers102	SUN397	Caltech101	Food101	AVG
1shot	TaskRes	Base	75.95±0.03	55.40±0.27	81.16±0.44	68.10±0.16	94.28±0.11	90.30±0.10	77.53
	TaskRes	Base+CuPL	74.69±0.04	65.66±0.82	90.07±0.79	73.52±0.49	95.89±0.57	90.35±0.36	<u>81.70</u>
	SToRI (Ours)	Base+CuPL	76.68±0.15	65.82±0.98	89.05±0.58	72.88±0.20	96.27±0.67	91.34±0.12	**82.01**
2shot	TaskRes	Base	76.03±0.00	55.52±0.48	81.50±0.62	69.53±0.14	94.54±0.05	90.49±0.05	77.93
	TaskRes	Base+CuPL	75.55±0.04	66.45±1.57	92.38±0.69	75.69±0.29	96.96±0.27	90.64±0.38	<u>82.95</u>
	SToRI (Ours)	Base+CuPL	77.36±0.23	66.37±1.01	91.56±0.60	75.75±0.04	97.15±0.13	91.49±0.24	**83.28**
4shot	TaskRes	Base	76.16±0.02	55.85±0.12	81.65±0.28	71.15±0.09	94.58±0.09	90.44±0.05	78.31
	TaskRes	Base+CuPL	76.42±0.03	70.76±1.12	93.22±0.37	77.20±0.08	97.40±0.21	91.45±0.15	**84.41**
	SToRI (Ours)	Base+CuPL	77.90±0.05	69.03±1.48	92.46±0.09	76.89±0.02	97.39±0.08	91.68±0.07	<u>84.22</u>
8shot	TaskRes	Base	76.87±0.05	58.14±0.07	86.82±0.19	74.52±0.07	96.17±0.08	91.12±0.07	80.60
	TaskRes	Base+CuPL	77.97±0.02	73.42±0.86	98.17±0.25	77.54±0.16	97.00±0.28	91.27±0.11	**85.89**
	SToRI (Ours)	Base+CuPL	78.38±0.13	72.03±0.60	97.51±0.43	78.34±0.13	96.98±0.29	90.50±0.05	<u>85.62</u>
16shot	TaskRes	Base	77.34±0.03	61.47±0.16	90.85±0.21	76.01±0.24	96.75±0.07	91.30±0.10	82.29
	TaskRes	Base+CuPL	79.18±0.10	77.05±0.65	99.07±0.11	78.98±0.10	97.65±0.23	91.49±0.08	**87.24**
	SToRI (Ours)	Base+CuPL	79.03±0.13	74.94±0.10	98.55±0.23	79.61±0.11	97.43±0.20	91.18±0.10	<u>86.79</u>

Table 2: Accuracy (%) on few-shot classification with CLIP ViT-L/14. The results include mean values with standard deviation across three runs. The results of TaskRes are reproduced. The best performance is indicated in bold, while the second-best performance is underlined.

markably similar AUC scores. The effect of weight control is consistent across different CLIP models, such as CLIP ViT-B/16, CLIP ViT-L/14, OpenCLIP (Cherti et al., 2023), and MetaCLIP (Xu et al., 2023). This shows that SToRI enables the emphasis or de-emphasis of specific semantics within a text when constructing text embeddings across various models, showcasing its versatility.

4.2 Data-driven Control

We train weights that best represent each dataset for the image classification task.

4.2.1 Experimental Setup

Datasets. We use various benchmarks for few-shot learning *i.e.*, ImageNet (Deng et al., 2009), DTD (Cimpoi et al., 2014), SUN397 (Xiao et al., 2010), Flowers102 (Nilsback and Zisserman, 2008), Caltech101 (Fei-Fei et al., 2004), and Food101 (Bossard et al., 2014). We use CUB (Wah et al., 2011) dataset for analysis on interpretation.

Text Prompts. We use text descriptions for each class which are provided by CuPL (Pratt et al., 2023). For the ImageNet and SUN397 datasets, due to the large number of total prompts, we use 10 text prompts for each class, selected based on their similarity with training set. We average the text embeddings from multiple text prompts to build one text embedding for each class. We refer the text embedding for image classifier as a text classifier.

Model. The experiments are conducted using CLIP and MetaCLIP ViT-L/14, with reweighting applied from the 7th block onward.

Implementation Details. We set the logarithm of the weight as the parameter to be trained in order to constrain the weights to non-negative values. Each text prompt has its own individual set of weights.

4.2.2 Few-shot Classification

Experimental Details. Following TaskRes (Yu et al., 2023), we evaluate our method by training with 1/2/4/8/16 examples (shots) per class from the training sets, respectively, and testing on the comprehensive test sets. For further details, please refer to Appendix A.2.

Comparison. To evaluate the capability of the text classifier obtained through SToRI to perform few-shot image classification, we conduct a comparative analysis of the prediction performance between SToRI and TaskRes (Yu et al., 2023). TaskRes is a recent method for few-shot image classification, which trains class-specific residual embedding x_c added to initial text embedding t_c to create new classifier $t_c + \alpha x_c$ for each class c. Here, t_c denotes the text embedding derived from a given text prompt for class c, and α is a hyperparameter for scaling. x_c is trained with cross-entropy loss (refer to Eq. (3)). Such residual embeddings exist in uninterpretable space, rendering the final classifier also uninterpretable. In contrast, SToRI trains only weights, indicating the degree to which each semantic element within a given sentence should be emphasized, thus maintaining interpretability.

Ensuring interpretability, SToRI achieves performance comparable to TaskRes, as presented in Table 2. "Base" refers to custom text prompts in-

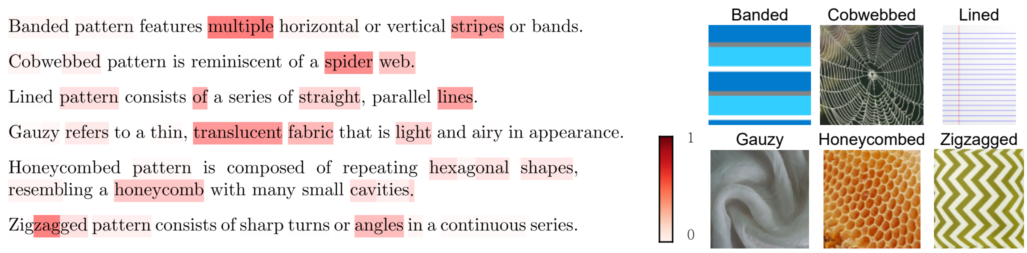

Figure 5: Text prompts and corresponding weights are provided as examples after training. The intensity of the red shading reflects the weight assigned, with darker shades indicating higher weights. For visualization, the weights are normalized to sum up 1. The figures on the right display an example image for each class.

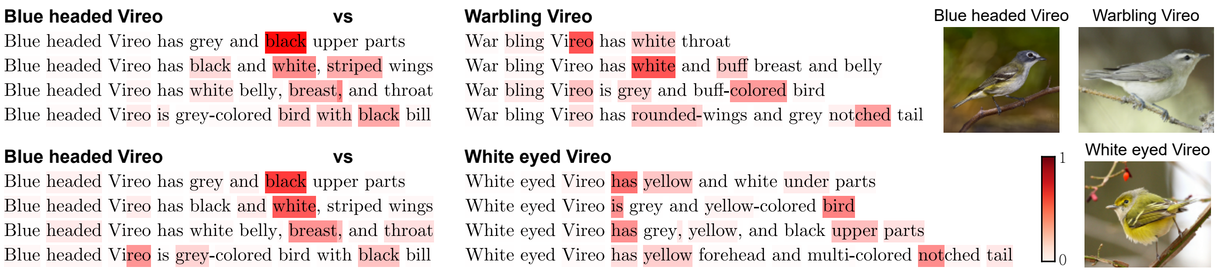

Figure 6: Text prompts and their corresponding weights are presented after training with the CUB dataset. The more intense the shade of red, the greater the weight assigned. In each scenario, the text classifier is trained to discriminate two classes. The weights for the same text prompts vary depending on the class to be distinguished.

cluding class names, which are generally used in few-shot image classification tasks with CLIP (Yu et al., 2023). We use both base and CuPL text prompts, with weights trained exclusively on CuPL. In the 1/2-shot setting, SToRI generally outperforms TaskRes across most datasets. In the 4/8/16-shot setting, it exhibits only a marginal difference, achieving nearly similar performance. This indicates that SToRI provides substantial flexibility to text embeddings, enabling it to be an enhanced text classifier that effectively represents image data. Please refer to Appendix B.3 for the MetaCLIP results, which align closely with those from CLIP.

4.2.3 Interpretability

Interpretation with Trained Weights. After training for an image classification task, we analyze the trained weights. Figure 5 presents examples of text prompts and the corresponding trained weights for each token within the DTD dataset. We have crafted the text prompts. We can discern that *banded* is associated with an emphasis on words like `multiple` and `stripes`. For *gauzy*, terms such as `translucent` and `light` are emphasized, and *cobwebbed* are notably associated with the word `spider web`. As illustrated by the images corresponding to each category, high weight values are assigned to important semantic tokens. This shows that SToRI can learn text embeddings that effectively represent the data in a data-driven control context, and the trained weights can offer novel insights for interpretation.

Does Optimization Occur in Interpretable Space? To ensure interpretability of text embeddings through data-driven control optimization, we conduct two experiments: an analysis on trained classifiers with different class compositions and an assessment of the effect of nonsensical text tokens.

The role of classifier is to distinguish one class from others. Thus, even for classifiers within the same class, the critical distinguishing features can vary depending on the alternative categories being compared. Figure 6 shows two text classifiers trained on the CUB dataset for two distinct pairs: *Blue headed Vireo* versus *Warbling Vireo*, and *Blue headed Vireo* versus *White eyed Vireo*. The text prompts for each class are generated with the attribute labels from the dataset. When contrasting *Blue headed Vireo* with the *Warbling Vireo*, `striped` is attributed a high weight. However, when distinguished from the *White eyed Vireo*, the weight on `striped` becomes low and `grey` is attributed a high weight. Note that *White eyed Vireo* also has striped wings. These terms highlight the

Text	Caltech101	SUN397
CuPL	97.42±0.23	79.54±0.12
CuPL+Nonsensical tokens	97.30±0.15	79.11±0.10

Table 3: Accuracy (%) on 16-shot image classification.

key distinctions between each pair of classes.

Table 3 reports the 16-shot classification performance when nonsensical text tokens are added. We randomly sample five tokens from the set of three rare tokens (Ruiz et al., 2023), namely `'sks'`, `'pll'`, and `'ucd'`, and add them to the end of all the original texts from CuPL. The inclusion of rare tokens does not contribute meaningful information to build a text classifier; it simply extends the number of tokens and trainable parameters. As a result, the performance when rare tokens are added did not surpass that without their addition. This demonstrates that adoption of the tokens without semantic meaning does not contribute to performance improvement. These findings support that data-driven control, achieved through attention modulation for tokens with semantic meaning, facilitates the creation of text embeddings that effectively represent the data, thereby ensuring the interpretability of text embeddings.

5 Related Works

VLMs and Interpretability. In recent vision tasks, interpretative analysis in natural language becomes popular rather than relying solely on visual form. For this purpose, VLMs have commonly been employed to connect the image feature space with the text feature space used for explanation. Kim et al. (2023) utilized VLMs to get concept activation vector (Kim et al., 2018) in vision model. Yuksekgonul et al. (2023) and Oikarinen et al. (2023) leveraged VLMs to determine whether concepts defined in text are present in images. Menon and Vondrick (2023) formulated text prompts for image classes using Large Language Models and employed them for zero-shot classification with VLMs. These approaches simply utilize the shared embedding space of existing VLMs. In contrast, our method introduces a new dimension of interpretability by providing controllability over the focus of textual information, thereby enhancing its interpretative utility.

Few-shot Image Classification. VLMs exhibit promising performance in image recognition tasks, leading to the development of various few-shot learning approaches. CoOp (Zhou et al., 2022b) and CoCoOp (Zhou et al., 2022a) are representative methods based on prompt tuning. Tip-Adapter (Zhang et al., 2022) integrates an extra adapter unit following the encoders. TaskRes (Yu et al., 2023) involves training task-specific residual text embeddings for each category. These approaches incorporate extra trainable parameters outside an interpretable framework, thereby not ensuring interpretability.

Enrich Textual Representation. In text-to-image generation, several approaches have been developed to enrich textual representation. Prompt weighting[1] is a common technique in Stable Diffusion (Rombach et al., 2022), which multiplies weights to individual output token embeddings prior to supplying them to the image generation model. Prompt-to-Prompt controls cross-attention between noise images and text embeddings (Hertz et al., 2022). Additionally, Ge et al. (2023) proposed a richer text editor that allows users to define various input conditions for image generation, such as coloring and footnotes. A similar approach has been explored in text generation. Zhang et al. (2024) introduced a method that enables large language models to process text with user-defined emphasis by reducing attention to unspecified parts of the text. While prior works have focused on image and text generation, typically using only user-defined attention, our work innovates by developing enriched textual representations for image recognition and proposing an approach for deriving these representations from data. This distinctive approach establishes a new avenue for incorporating linguistic context in visual understanding.

6 Conclusion

We propose SToRI, a framework that builds interpretable text embeddings by reweighting semantic tokens in VLMs. This approach is a novel means of adapting the explanatory power of natural language in vision tasks. Our user-driven and data-driven controls empower users to dictate the emphasis on specific terms and facilitate the tuning of text embeddings for classification while ensuring interpretability. Our approach can be easily applied to any model based on attention, and has potential scalability in various vision tasks and multi-modal tasks, given the widespread use of VLMs.

[1] https://huggingface.co/docs/diffusers/using-diffusers/weighted_prompts

7 Limitations

Our method is focusing on controlling the attention of each semantic element within a given natural language sentence, rather than generating new textual information. Therefore, one of the limitations of our method is its dependence on the richness and quality of the given texts. For example, when using data to train a classifier, if the given text lacks sufficient rich information, adjusting the attention may not sufficiently enlarge the text embedding space. This difficulty in expanding the embedding space makes it challenging to establish a basis for improving classification performance and explaining data.

Additionally, we do not consider the inherent black box characteristics of VLMs. However, if this model has undergone sufficient testing and is deemed reliable, the advantage of our method lies in additional optimization and control being in a reliable and controllable space.

8 Ethics Statement

Our goal is to employ contollability when building text embeddings. This enables for users to emphasize or deemphasize a certain part of textual information and improving text embeddings for vision tasks, ensuring interpretability. We believe this work can be used to build trustful AI systems by providing natural language interpretation.

If the VLMs in use are biased towards the attributes targeted for reweighting, it may also affect other related attributes. The best approach to address this issue is to use VLMs that have been trained to reduce bias. However, if a biased VLM must be used, designing text prompts that can help mitigate the bias could be a potential strategy to consider.

References

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. 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*, pages 446–461. Springer.

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

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Li Fei-Fei, Rob Fergus, and Pietro Perona. 2004. 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*, pages 178–178. IEEE.

Songwei Ge, Taesung Park, Jun-Yan Zhu, and Jia-Bin Huang. 2023. Expressive text-to-image generation with rich text. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7545–7556.

Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR.

Lisa Anne Hendricks, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Zeynep Akata. 2021. Generating visual explanations with natural language. *Applied AI Letters*, 2(4):e55.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*.

Maxime Kayser, Oana-Maria Camburu, Leonard Salewski, Cornelius Emde, Virginie Do, Zeynep Akata, and Thomas Lukasiewicz. 2021. e-vil: A dataset and benchmark for natural language explanations in vision-language tasks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1244–1254.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pages 2668–2677. PMLR.

Siwon Kim, Jinoh Oh, Sungjin Lee, Seunghak Yu, Jaeyoung Do, and Tara Taghavi. 2023. Grounding counterfactual explanation of image classifiers to textual concept space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10942–10950.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR.

Xuhong Li, Haoyi Xiong, Xingjian Li, Xuanyu Wu, Xiao Zhang, Ji Liu, Jiang Bian, and Dejing Dou. 2022. Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *Knowledge and Information Systems*, 64(12):3197–3234.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

Sachit Menon and Carl Vondrick. 2023. Visual classification via description from large language models. In *The Eleventh International Conference on Learning Representations*.

W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. 2019. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080.

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

Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. 2023. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. 2023. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15691–15701.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510.

Fawaz Sammani, Tanmoy Mukherjee, and Nikos Deligiannis. 2022. Nlx-gpt: A model for natural language explanations in vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8322–8332.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset.

Jialin Wu and Raymond Mooney. 2019. Faithful multimodal explanation for visual question answering. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 103–112, Florence, Italy. Association for Computational Linguistics.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE.

Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. 2023. Demystifying clip data. *arXiv preprint arXiv:2309.16671*.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. 2023. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197.

Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. 2023. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10899–10909.

Mert Yuksekgonul, Maggie Wang, and James Zou. 2023. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*.

Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. 2024. Tell your model where to attend: Post-hoc attention steering for LLMs. In *The Twelfth International Conference on Learning Representations*.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2022. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European Conference on Computer Vision*, pages 493–510. Springer.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348.

A Experimental Details

A.1 User-driven Control

CelebA. We initially select 11 attributes with a zero-shot classification performance of AUROC 0.75 or higher with CLIP on test set. For zero-shot classification, we create text prompt for each attribute and calculate AUROC using the similarity between the test set images and the text prompt. For example, when evaluating the attribute *smiling*, we use the text prompt `'a photo of a smiling person'`. Among the identified 11 attributes, we create combinations of three and five attributes, each including either *female* or *male*. For the combinations of three attributes, we filter out the combinations where all eight categories contain fewer than 100 images. We conduct image retrieval with total 20 numbers of text prompts based on the combinations of attributes, as shown in Table 8. Details on combinations of five attributes can be found in Appendix B.2.

CUB. Following the filtering process described by Koh et al. (2020), we initially retain 112 attributes. We then select 15 attributes that achieve a zero-shot classification performance with AUROC 0.75 or higher using CLIP. Notably, the attribute labels in the CUB dataset are finely detailed and related to various parts of birds, which poses a challenge for CLIP in differentiation. With the chosen attributes, we form combinations of three attributes that do not share the same color, yielding 58 combinations. The text prompt we use is `'a photo of a bird, which has [text for attribute1], has [text for attribute2], and has [text for attribute3]'`. Table 9 presents 15 attributes and their corresponding texts.

A.2 Data-driven Control

We follow the data split outlined in CoOp (Zhou et al., 2022b), conducting tests on the official test set of each dataset and the validation set of the ImageNet dataset. We use Adam optimizer with the cosine learning rate scheduler (Loshchilov and Hutter, 2017) following the training scheme of TaskRes (Yu et al., 2023). For CLIP, the learning rate is set to 1×10^{-2} for the ImageNet and SUN397 datasets, 0.1 for the Food101 dataset and for 8/16-shot scenarios on the DTD and Flower102 datasets, and 5×10^{-2} for the other datasets. For MetaCLIP, the learning rate is set to 1×10^{-2} for the ImageNet and SUN397 datasets, 0.1 for Flower102 dataset, and 5×10^{-2} for the other datasets. The weight

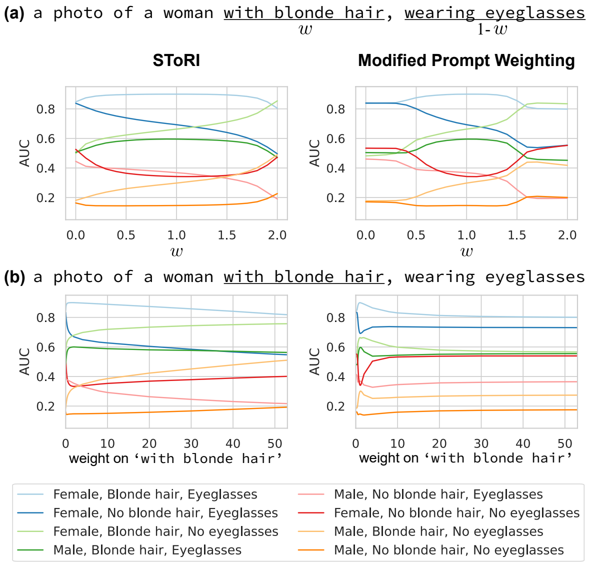

Figure 7: AUC scores from preference retrieval with varying weights. The text prompt is '`a photo of a woman with blonde hair, wearing eyeglasses`'. (a) The weights on '`with blonde hair`' and '`wearing eyeglasses`' are w and $(1 - w)$, respectively, which are adjusted simultaneously in opposite direction. (b) Only the weight on '`with blonde hair`' is adjusted. Best viewed in color.

decay is set to 0 for both models. When reproducing TaskRes, the learning rate is set to 2×10^{-5} for the ImageNet dataset and 2×10^{-4} for the other datasets. The weight decay is set to 0.005 and α is set to 0.5. 1/2/4-shot training is done with 100 epoch and the other is done with 200 epoch for all datasets. The training is conducted with a batch size of 256. All experiments are implemented using PyTorch (Paszke et al., 2017), and we use official code base released by Yu et al. (2023) to reproduce TaskRes.

We use all the datasets and models solely for academic research purposes and do not employ them for improper intentions.

B Additional Experimental Results

B.1 Comparison to Prompt Weighting

We compare SToRI with prompt weighting, a technique often used in text-to-image generation via Stable Diffusion (Rombach et al., 2022). Prompt weighting multiplies weights by the difference in output token embeddings when provided with a text prompt versus an empty one. Unlike Stable Diffusion, which utilizes all output token embeddings, we aim to build a vector form of text embedding from `[EOS]` token. Therefore, we modify

		AP	P_{400}
Plain ($w = 1.0$)		0.752±0.089	0.679±0.070
Emphasized	Attribute with $w = 1.5$	0.754±0.085 **Δ0.003±0.017**	0.681±0.064 **Δ0.002±0.016**
	Attribute with $w = 2.0$	0.776±0.082 **Δ0.024±0.019**	0.698±0.064 **Δ0.019±0.016**

Table 4: Retrieval performance on attributes of the CelebA dataset when two attributes are assigned different weights. The results show mean values with standard deviation across multiple controlled attributes.

prompt weighting for use at an intermediate layer, which we refer to as modified prompt weighting, and compare it with SToRI on preference-based image retrieval.

As depicted in Figure 7(a), the modified prompt weighting influences the significance of tokens similarity to SToRI. However, the change in AUC is not gradual; it remains nearly static when weights fall below 0.5 or above 1.5. As shown in Figure 7(b), even when the weight for '`with blonde hair`' increases significantly, SToRI consistently raises the AUC for the category 'female, blonde hair, no eyeglasses'. In contrast, the AUC with modified prompt weighting initially increases but subsequently decreases, indicating augmented weight fails to heighten emphasis. This could stem from the scaling of intermediate embeddings which, when overextended, surpasses the scale that the text encoder is pre-trained to deal with, lessening the intended effect of emphasis. SToRI, on the other hand, adjusts normalized attention scores within the self-attention mechanism, ensuring that as weight escalates, the relevant tokens consistently obtain attention scores approaching 1, thus preserving the desired impact.

B.2 Additional Results for Preference-based Retrieval

We assess SToRI in the context of preference-based retrieval by assigning different weights to multiple attributes to explore how varying weight magnitudes affect emphasis. We create combinations of three attributes and assign them different weights: one attribute is assigned a weight of 2.0, another a weight of 1.5, and the remaining one a weight of 1.0. We then compare the retrieval performance for attributes with weights of 1.5 and 2.0. Table 4 demonstrates that the retrieval performance of the attribute with a weight of 1.5 increases, while the

	CelebA		CUB
	AP	P_{400}	AP
Plain ($w = 1.0$)	0.753±0.088	0.681±0.062	0.148±0.055
Emphasized ($w = 1.5$)	0.774±0.086 **Δ0.021±0.011**	0.699±0.063 **Δ0.018±0.009**	0.195±0.074 **Δ0.047±0.026**
De-emphasized ($w = 0.5$)	0.709±0.087 **Δ-0.044±0.022**	0.647±0.057 **Δ-0.035±0.016**	0.098±0.035 **Δ-0.051±0.026**

Table 5: Retrieval performance on attributes of the CelebA and CUB datasets with MetaCLIP ViT-L/14. The results show mean values with standard deviation across multiple controlled attributes.

		AP	P_{80}
CLIP	Plain ($w = 1.0$)	0.684±0.097	0.627±0.062
	Emphasized ($w = 1.5$)	0.705±0.099 **Δ0.021±0.009**	0.643±0.069 **Δ0.015±0.012**
	De-emphasized ($w = 0.5$)	0.643±0.086 **Δ-0.041±0.019**	0.601±0.054 **Δ-0.026±0.012**
MetaCLIP	Plain ($w = 1.0$)	0.689±0.074	0.631±0.062
	Emphasized ($w = 1.5$)	0.713±0.078 **Δ0.023±0.008**	0.646±0.062 **Δ0.015±0.011**
	De-emphasized ($w = 0.5$)	0.644±0.064 **Δ-0.045±0.020**	0.602±0.057 **Δ-0.029±0.014**

Table 6: Retrieval performance on the CelebA dataset with CLIP and MetaCLIP ViT-L/14 when five attributes are combined. The results show mean values with standard deviation across multiple controlled attributes.

attribute with a weight of 2.0 shows an even greater increase in retrieval performance. This indicates that when semantic tokens are assigned different weights, the emphasis effect increases proportionally with the assigned weights compared to plain text. This highlights the significance of the magnitude of weights.

Table 5 presents the results on MetaCLIP ViT-L/14 when adjusting the weight of one attribute among three within combinations of three attributes (as outlined in Section 4.1). The results demonstrate that emphasizing or de-emphasizing an attribute in MetaCLIP leads to increased or decreased retrieval performance for images with the specified attribute, showcasing the scalability of SToRI across models.

To evaluate SToRI in more complex attribute combinations, we perform retrieval using combinations of five attributes. Only the following five attributes result in images for all 32 possible categories formed by combinations of the five attributes: *male* or *female*, *smiling*, *bangs*, *gray hair*, and *eyeglasses*. We use two text prompts for *male* and *female*. We randomly select five images for each category, resulting in a total of 160 images. Table 6 presents the results on CLIP and MetaCLIP ViT-L/14 when adjusting the weight of one attribute among five. These findings underscore a consistent trend of increasing retrieval scores when attributes are emphasized and decreasing scores when attributes are de-emphasized, across different attribute combinations.

Method	Plain Text Embeddings	SToRI
Relative Run Time	1.00	1.02

Table 7: Relative compuational cost

B.3 Additional Results for Few-shot Classification

Table 10 compares few-shot classification performances of SToRI and TaskRes (Yu et al., 2023) on MetaCLIP ViT-L/14. Similar to the results on CLIP, the results show that SToRI achieves performance comparable to TaskRes, which uses uninterpretable classifiers. These experiments further support our findings, demonstrating our approach's effectiveness across models and highlighting its adaptability and scalability.

B.4 Additional Examples for Interpretation

Figures 8 and 9 present examples of text prompts and the corresponding trained weights for each token within the ImageNet and DTD datasets, respectively. Higher weights are assigned to word tokens that effectively represent images.

B.5 Computational Cost

We calculate runtime for applying SToRI compared to plain text embeddings, as reported in Table 7. The experiment is done on RTX A5000 and the reported values are mean values from 28K runs. Since SToRI only multiplies predefined weights when calculating attention scores, the runtime is not significantly different from that of plain text embeddings.

B.6 Position for Reweighting

Figure 10(a) compares the changes in AUC scores when we start reweighting at various positions. The reweighting process is applied to all blocks following a specific block. There is not a significant difference when we initiate token reweighting at intermediate positions. However, when token reweighting is applied to all blocks (from 1st block), a sharp bend is observed at 0.1 when the weight decreases. This is unlike other cases, which show a smooth decrease or increase in all scenarios. It is presumed that this abrupt occurrence is due to tokens in the specified position being completely disregarded when the weight becomes 0, leading to sudden gaps in those areas.

Figure 10(b) illustrates that when reweighting is applied only within a single specific intermediate block, the effects of emphasis or de-emphasis are scarcely observed. This suggests that if reweighting is confined within a single intermediate block, its effects in the subsequent blocks are counteracted, indicating that it should be applied in the subsequent blocks to emphasize or de-emphasize semantic tokens.

C Demonstration of Preference-based Image Retrieval

Figure 11 shows a practical demo application of SToRI. It enables users to actively adjust image rettrieval results by tweaking weights in real time.

Selected Attributes	Text prompts
Female/Male, Smiling, Bangs	a photo of a smiling [woman/man] with bangs
Female/Male, Smiling, Blond Hair	a photo of a smiling [woman/man] with blond hair
Female/Male, Smiling, Gray Hair	a photo of a smiling [woman/man] with gray hair
Female/Male, Smiling, Wearing Hat	a photo of a smiling [woman/man] wearing hat
Female/Male, Smiling, Eyeglasses	a photo of a smiling [woman/man] wearing eyeglasses
Female/Male, Bangs, Wearing Hat	a photo of a [woman/man] with bangs, wearing hat
Female/Male, Bangs, Eyeglasses	a photo of a [woman/man] with bangs, wearing eyeglasses
Female/Male, Blond Hair, Eyeglasses	a photo of a [woman/man] with blond hair, wearing eyeglasses
Female/Male, Gray Hair, Eyeglasses	a photo of a [woman/man] with gray hair, wearing eyeglasses
Female/Male, Wearing Hat, Eyeglasses	a photo of a [woman/man] wearing hat and eyeglasses

Table 8: All combinations of attributes and corresponding text prompts on the CelebA dataset.

Attributes	Texts
has_bill_shape::hooked_seabird	hooked seabird bill
has_shape::duck-like	duck-like shape
has_crown_color::blue	blue crown
has_forehead_color::blue	blue forehead
has_wing_color::yellow	yellow wing
upperparts_color::yellow	yellow upperparts
has_underparts_color::yellow	yellow underparts
has_back_color::yellow	yellow back
has_breast_color::yellow	yellow breast
has_throat_color::yellow	yellow throat
has_forehead_color::yellow	yellow forehead
has_nape_color::yellow	yellow nape
has_belly_color::yellow	yellow belly
has_primary_color::yellow	yellow color
has_crown_color::yellow	yellow crown

Table 9: Candidates of attributes and corresponding texts on the CUB dataset.

	Method	Text	ImageNet	DTD	Flowers102	SUN397	Caltech101	Food101	AVG
1shot	TaskRes	Base	79.38±0.02	67.91±0.26	83.75±0.16	74.89±0.08	97.21±0.15	90.63±0.04	82.29
	TaskRes	Base+CuPL	79.59±0.22	72.79±0.54	92.26±0.10	76.16±0.2	97.59±0.19	90.28±0.15	**84.78**
	SToRI (Ours)	Base+CuPL	79.44±0.17	72.66±0.73	92.38±0.75	76.05±0.38	97.46±0.23	90.12±0.22	<u>84.68</u>
2shot	TaskRes	Standard	79.46±0.01	67.93±0.18	84.03±0.13	75.71±0.13	97.48±0.07	90.83±0.03	82.57
	TaskRes	Base+CuPL	80.23±0.14	74.27±1.08	94.42±0.08	77.64±0.28	98.20±0.08	90.68±0.22	<u>85.91</u>
	SToRI (Ours)	Base+CuPL	79.98±0.16	73.76±1.38	95.09±0.45	78.21±0.27	98.04±0.02	90.57±0.18	**85.94**
4shot	TaskRes	Standard	79.58±0.00	68.34±0.22	84.07±0.12	76.66±0.06	97.44±0.06	90.82±0.02	82.82
	TaskRes	Base+CuPL	80.68±0.04	76.91±1.24	94.94±0.18	78.88±0.11	98.16±0.11	90.85±0.07	<u>86.74</u>
	SToRI (Ours)	Base+CuPL	80.53±0.09	75.91±0.39	96.28±0.31	79.38±0.14	98.01±0.33	90.73±0.13	**86.81**
8shot	TaskRes	Standard	80.03±0.08	69.7±0.45	90.12±0.07	78.87±0.04	97.84±0.10	91.30±0.03	84.64
	TaskRes	Base+CuPL	81.30±0.12	78.88±0.10	98.55±0.17	78.87±0.17	98.22±0.07	90.81±0.18	**87.77**
	SToRI (Ours)	Base+CuPL	81.01±0.18	78.39±0.27	98.04±0.05	80.24±0.09	98.23±0.10	90.71±0.16	**87.77**
16shot	TaskRes	Standard	80.46±0.01	72.03±0.46	93.72±0.13	79.92±0.13	98.00±0.08	91.47±0.05	85.93
	TaskRes	Base+CuPL	81.78±0.02	81.28±0.82	99.22±0.12	79.92±0.17	98.47±0.08	91.19±0.11	**88.65**
	SToRI (Ours)	Base+CuPL	81.40±0.02	79.89±0.70	98.58±0.06	81.43±0.16	98.47±0.12	91.25±0.04	<u>88.50</u>

Table 10: Accuracy (%) on few-shot classification with MetaCLIP ViT-L/14. The results include mean values with Standard deviation across three runs. The results of TaskRes are reproduced. The best performance is indicated in bold, while the second-best performance is underlined.

A tiger shark is one of the largest shark.
A tiger shark has dark stripes on its body and a grey back.

An electrical ray has dark brown back.
An electrical ray is wide, flat-shaped fish that can give off an electric shock.

A great grey owl has grey and white features.
A great grey owl is a large owl with big yellow eyes.

A red king crab has hard shell and long legs.
A red king crab is brownish-red in color.

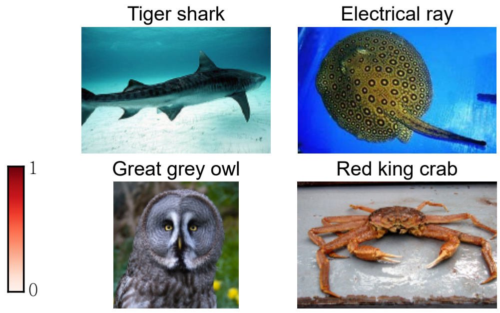

Figure 8: Text prompts and corresponding weights on the ImageNet dataset are provided as examples after training with data. For visualization, the weights are normalized to sum up 1. The figures on the right display an example image for each class.

Bubbly pattern contains a series of small bubbles or resembles a piece of foam.
Dotted pattern includes a series of regular or irregular dots.
Cracked surface displays a network of lines where it has fractured or been damaged.
Polka-dotted pattern displays a series of uniform, rounded dots.
Swirly pattern is full of swirling shapes and curved lines.
Perforated pattern is made up of small, evenly spaced holes.

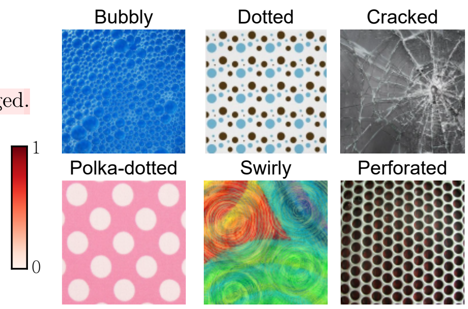

Figure 9: Text prompts and corresponding weights on the DTD dataset are provided as examples after training with data. For visualization, the weights are normalized to sum up 1. The figures on the right display an example image for each class.

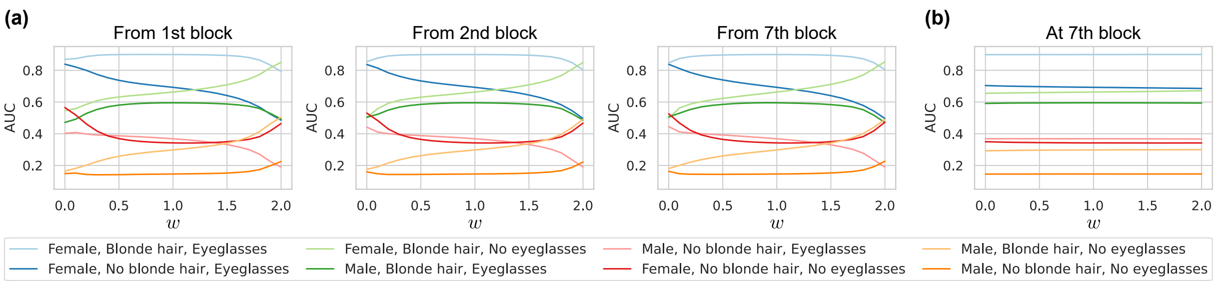

Figure 10: The change of AUC scores for preference retrieval with weight control when diversifying blocks that semantic token reweighting is applied. (a) The results when reweighting is applied within the subsequent blocks as well. (b) The result when reweighting is applied within a single block.

Figure 11: Demonstration of a real-world, functioning demo application using OpenCLIP alongside SToRI, where users can dynamically manipulate image retrieval outcomes through targeted weight adjustments. The application effectively showcases how identical textual prompts can yield substantially different visual results based on user-specified weight modifications.