

STAIR: Learning Sparse Text and Image Representation in Grounded Tokens

Chen Chen, Bowen Zhang, Liangliang Cao, Jiguang Shen,
Tom Gunter, Albin Madappally Jose, Alexander Toshev, Yantao Zheng,
Jonathon Shlens[†], Ruoming Pang, Yinfei Yang
Apple AI/ML

{chen_chen999, bowen_zhang4, lcao27, yinfeiy}@apple.com

Abstract

Image and text retrieval is one of the foundational tasks in the vision and language domain with multiple real-world applications. State-of-the-art contrastive approaches, *e.g.* CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021), represent images and texts as dense embeddings and calculate the similarity in the dense embedding space as the matching score. On the other hand, sparse semantic features like bag-of-words models are inherently more interpretable, but believed to suffer from inferior accuracy than dense representations. In this work, we show that it is possible to build a sparse semantic representation that is as powerful as, or even better than, dense presentations. We extend the CLIP model and build a sparse text and image representation (STAIR), where the image and text are mapped to a sparse token space. Each token in the space is a (sub-)word in the vocabulary, which is not only interpretable but also easy to integrate with existing information retrieval systems. STAIR model significantly outperforms a CLIP model with +4.9% and +4.3% absolute Recall@1 improvement on COCO-5k text→image and image→text retrieval respectively. It also achieved better performance on both of ImageNet zero-shot and linear probing compared to CLIP.¹

1 Introduction

Learning high-quality and performant representations from large-scale image-text data has been intensively studied in recent years. Improved vision-language representation benefits many applications, including image-text retrieval (Chen et al., 2015; Plummer et al., 2015), VQA (Antol et al., 2015; Johnson et al., 2017), and image captioning (Vinyals et al., 2015). The resurgence of interest in contrastive learning has fueled recent advances. State-of-the-art models like CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) employ

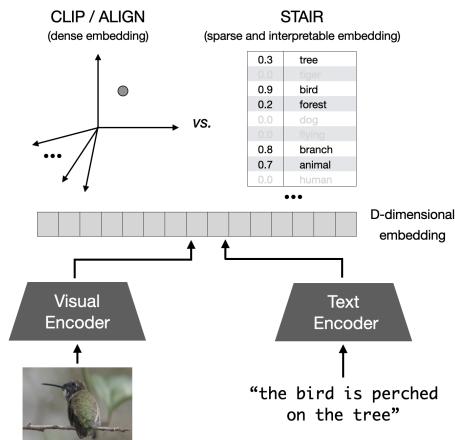


Figure 1: **Learning a sparse and interpretable embedding for a vision-language model.** CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) learn a compact dense embedding, while STAIR aims to learn a sparse and interpretable embedding in a high-dimensional space. Each dimension in the sparse embedding represents a (sub-)word from a large dictionary, with a non-negative importance weight.

dedicated encoders for images and text, enabling joint embedding in a shared space. By enhancing the cosine similarity for aligned pairs and dissimilarity for unmatched pairs, these contrastive models achieve remarkable performance in fine-tuning and zero-shot generalization for image-text retrieval, VQA, and image classification tasks.

Despite the impressive benchmark performance, the dense embedding space is usually considered as a black box and challenging to interpret. The meaning of a dense embedding is determined by its vector relationships with other embeddings, rather than corresponding directly to any human-understandable concept. This lack of direct correspondence with interpretable concepts hinders the transparency and interpretability. Moreover, deploying a retrieval model trained with a contrastive objective to real-world large-scale image-text retrieval is a non-trivial task. Despite approximated

¹Work done while at Apple

nearest neighbor search (Guo et al., 2020; Johnson et al., 2019a) can be used to retrieve from a dense embedding space, the cost can be high when scaling up to billions of images. Additionally, traditional approaches like inverted indexes, commonly employed by search engines, cannot be straightforwardly applied to the dense embeddings. The lack of interoperability further complicates the integration of dense embeddings with other retrieval features without additional training. In comparison, interpretable sparse embeddings possess distinct advantages across numerous applications. More detailed discussion can be found in Appendix D.

The community has explored sparse and interpretable image representations extensively. Early approaches, *e.g.* bag-of-words and topic models (Csurka et al., 2004; Fei-Fei and Perona, 2005; Lazebnik et al., 2006) were widely coupled with SIFT descriptor (Lowe, 2004) but found to exhibit inferior performance compared to dense vectors (Lin et al., 2011; Sánchez et al., 2013). Other endeavors, like deep visual-semantic embedding (Frome et al., 2013) using ImageNet topics yielded more interpretability but still failed to outperform dense representations (Faghri et al., 2017).

We hypothesize that the gap between sparse semantic and dense embedding stems from two primary factors: (1) Previous works on semantic embedding inadequately explored large-scale training to capture the rich semantics in the image-text domain. (2) Most existing semantic embedding methods are built on a fixed vocabulary (*e.g.* thousand of concepts), which cannot handle out-of-vocabulary concepts. In this paper, we present a new model, named **STAIR**, and a multi-stage training recipe to learn a **Sparse Text And Image Representation** to tackle the aforementioned challenges. Our sparse representation not only matches but also outperforms the dense image-text representations.

Inspired by the recent success of sparse representation in information retrieval field (Bai et al., 2020; Formal et al., 2021b), STAIR encodes the image and text into a sparse and grounded token space, as illustrated in Figure 1. Specifically, images and text are mapped to sparse and interpretable in a high-dimensional space, where each dimension is associated with an actual (sub-)word from a large dictionary, with non-zero weights. Notably, the proposed multi-stage training recipe plays a critical role in the grounding of the sparse embedding.

We conduct a comparative analysis between

the STAIR model and a CLIP model, sharing the same architecture and dataset. Experiment results a significant improvement of STAIR model over CLIP on image-text retrieval tasks, with +4.9% and +4.3% recall@1 on COCO-5K (Chen et al., 2015) text→image and image→text retrieval respectively. STAIR models also demonstrate similar or better performance on zero-shot classification and linear probing tasks including ImageNet. To quantify the interpretability of the embeddings, we define an interpretable space using BERT vocab. Experiments suggest STAIR is substantially more interpretable, achieving a Top-1 accuracy 32.9% on ImageNet, outperforming CLIP’s accuracy of 13.7%.

2 Approach

We begin with an overview of the dual-encoder architecture to establish the basis of our research.

2.1 Dual-Encoder Contrastive Learning

Given a dataset of image-text pairs $D = \{(x_i, y_i)\}$, where x_i and y_i represent the image and text respectively, a dual encoder learns a similarity $M(x, y)$, such that the aligned pair (x_i, y_i) is assigned higher similarity score compared to the unmatched pairs sampled from a negative sample set D'_i .

A dual-encoder architecture comprises an image encoder E_{IMAGE} and a text encoder E_{TEXT} . Each encoder E consists of two components: 1) a standard input-specific neural network $f(\cdot)$ and 2) a projection head $g(\cdot)$ that maps the features into embeddings in a joint dense embedding space:

$$E(\cdot) = g(f(\cdot)) \quad (1)$$

where $f(\cdot)$ is typically a transformer model and $g(\cdot)$ is a pooling layer.

The similarity between an image x and text y is measured by cosine similarity:

$$M(x, y) = \frac{E(x)^T E(y)}{\|E(x)\| \|E(y)\|} \quad (2)$$

The dual encoder is trained by minimizing the contrastive loss \mathcal{L}_{CON} :

$$\mathcal{L}_{\text{CON}} = -\frac{1}{|D|} \sum_i \log \frac{e^{M(x_i, y_i)/T}}{\sum_{(x', y') \in D_i} e^{M(x', y')/T}} \quad (3)$$

where $D_i = D'_i \cup \{(x_i, y_i)\}$ denotes the positive pair and its negative set for each pair, and T is the softmax temperature.

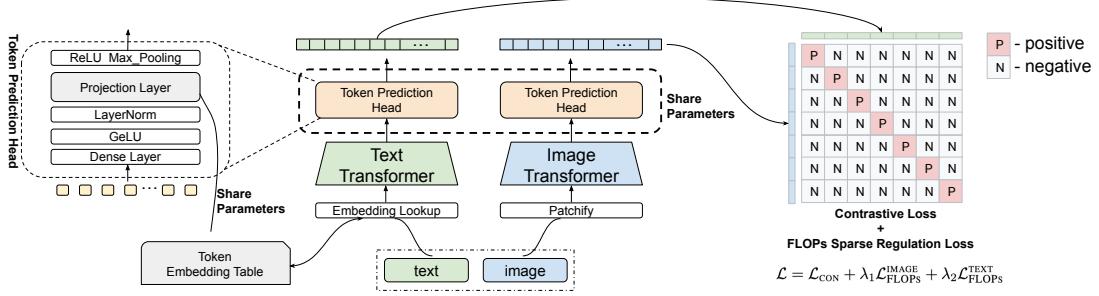


Figure 2: **Diagram of STAIR architecture.** It employs a dual-encoder architecture. Different than dense models like CLIP or ALIGN, STAIR maps the dense embeddings to a sparse latent embedding space via a token prediction head. In addition to a regular contrastive loss to minimize an image-text matching loss, a FLOPs (Paria et al., 2020) loss is added to encourage sparsity of image and text embeddings.

2.2 STAIR

Following CLIP (Radford et al., 2021), STAIR also adopts a dual-encoder architecture. As depicted in Figure 2, it consists of an image encoder and a text encoder, both comprising a feature extraction network $f(\cdot)$ and a projection head $g(\cdot)$. In particular, the dense project head $g(\cdot)$ is replaced with a **Token Projection Head**, which maps the representation to a sparse embedding space. A vocabulary V is used as the basis of embedding space for interpretability purpose.

The token projection head $g(\cdot)$ comprises two components: (1) a mapping function that maps the input sequence h to a sequence of weights for each token j in the vocabulary space V , and (2) a pooling layer that summarizes the sequence into a sparse embedding in the vocabulary space V .

For the mapping function, we leverage the BERT (Devlin et al., 2019) masked language model (MLM) prediction head $p(\cdot)$:

$$p(h_j) = e \cdot \text{TRANSFORM}(h_j) + b \quad (4)$$

where $h_j = f(\cdot)$ corresponds to the j^{th} token in the sequence of the feature extraction network output. The TRANSFORM function comprises a FC layer with GELU activation and a LAYER NORM layer, and e and b are the linear mapping and bias in MLM prediction head that maps the output to vocabulary space. The weights of e is tied with token embedding lookup table in the text encoder.

For the pooling layer, we adopt the approach from Formal et al., 2021b,a to aggregate the weight of token j to sparse embedding ENC :

$$\text{ENC} = \log(1 + \text{RELU}(\max_j(p(h_j)))) \quad (5)$$

RELU activation non-negativity of token weights

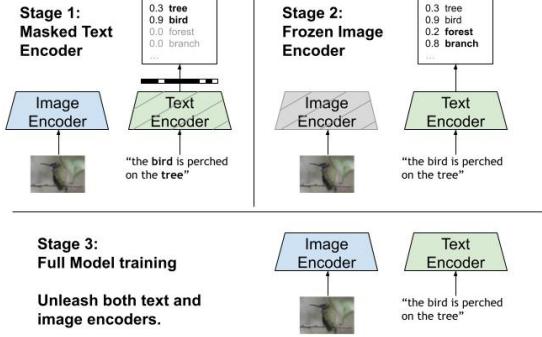


Figure 3: **Training strategy for STAIR model.** (1) The text output is masked to only predict the weights of tokens that occur in the input. (2) The image encoder is frozen and the mask is removed from the text encoder. (3) All constraints are removed.

and adding the log function empirically improves performance by suppressing overwhelmingly large weights (Zhao et al., 2021). Consequently, the image embedding $\text{ENC}^{\text{IMAGE}}$ and text embedding ENC^{TEXT} are sparse vectors residing in a $|V|$ -dimensional space defined by the vocabulary.

In practice, relying solely on the ReLU operator in Eq. (5) often leads to insufficient sparsity. To obtain better control over sparsity, we incorporate FLOPs regularization loss (Paria et al., 2020) to encourage a small number of non-zeros token embeddings in V :

$$\mathcal{L}_{\text{FLOPS}} = \sum_{k \in V} \left(\frac{1}{N} \sum_{i=1}^N \text{ENC}_k^{(i)} \right)^2 \quad (6)$$

By combining the contrastive loss and the FLOPs loss, the STAIR model is optimized by:

$$\mathcal{L} = \mathcal{L}_{\text{CON}} + \lambda_1 \mathcal{L}_{\text{FLOPs}}^{\text{IMAGE}} + \lambda_2 \mathcal{L}_{\text{FLOPs}}^{\text{TEXT}} \quad (7)$$

where λ_1 and λ_2 are FLOPs regularization weights

Table 1: **Model Size (Unit: Millions).** The number of parameters of STAIR is comparable to CLIP.

	Text Encoder	Image Encoder
CLIP	53.7	88.6
STAIR	53.8	86.8

for image and text embeddings, respectively. Notably, by adjusting these weights, STAIR embeddings can achieve a high level of sparsity, resulting in an effective embedding size even smaller than CLIP. Details are discussed in Section 6.3.

3 Training Details

The STAIR model aims to achieve two goals: 1) aligning text and images in the sparse embedding space; 2) grounding the sparse embedding dimension with human-understandable (sub-)word in the vocabulary. In other words, the image and text embeddings should reside in a space defined by basis vectors that correspond to interpretable tokens. However, in practice, we found that simply replacing the dense projection head with a token prediction head alone does not guarantee the 2nd goal outright. This is because the images and text are distinct modalities with inherent semantic gaps, and contrastive loss alone only encourages text/image alignment. We show in Section 5.2 that the model learns a shortcut by relying on less common tokens to bridge the modality gap.

To address this challenge, we propose a multi-stage training approach that sequentially bridges the discrepancy between the sparse embedding space and the interpretable space defined by the vocabulary as illustrated in Figure 3.

Stage 1: Training image embedding with masked tokens In the first stage, we co-train both encoders while applying a binary mask to the text embedding. Formally, given the text input y , the masked text embedding is formulated as:

$$\text{ENC}_{\text{MASK}}^{\text{TEXT}} = g(f(y)) \cdot \text{MASK}(y) \quad (8)$$

where $\text{MASK}_i = 1$ if the i^{th} token exists in y , and $g(f(y))$ predicts the weights of the non-masked tokens. In other words, the text embedding is forced to activate only the tokens appearing in the original text input, while disregarding others. By matching with the masked text embedding, the image encoder learns to ground its image embedding on the tokens from the paired text. Consequently, after the stage 1, it anchors the image embeddings

to meaningful tokens in the interpretable space defined by V .

Stage 2: Training with frozen image encoder

In this stage, our focus shifts to grounding the text embedding to the same interpretable space. The key idea is to leverage the image encoder to teach the text encoder as a teacher model to guide the training of the text encoder. Specifically, we freeze the image encoder while training the text encoder to align its image embedding using contrastive loss. After stage 2 training, both image and text embeddings are positioned within the same human-interpretable embedding space constructed by V .

Stage 3: Joint fine-tuning of both encoders

The initial two-stage training establishes a solid foundation for both encoders to produce human-interpretable embeddings. In stage 3, we enhance the performance of image-text matching by jointly finetuning both encoders.

Experiments show that this multi-stage training recipe is critical for the embedding interpretability. A qualitative and quantitative comparison between training with multi-stage recipe and without can be found in Section 6.1 and Appendix C.

4 Experiments

4.1 Datasets

Our dataset is a combination of internal and public datasets, totaling 1.1B image-text pairs. The public datasets consist of Conceptual Caption 3M (Sharma et al., 2018) and Conceptual Captions 12M (Changpinyo et al., 2021). The internal dataset consists of 1B image-text pairs, including a 134M clean licensed dataset (see Appendix A for details) and a 971M noisy web-crawled dataset. The web-crawled dataset is mined using a similar approach as described in ALIGN (Jia et al., 2021). We further filter the data using a public CLIP model², removing pairs with a CLIPScore (Hessel et al., 2021) below 0.2.

4.2 Configurations

In the experiment, we train a CLIP model and a STAIR model for comparison. For both models, we utilize transformer (Vaswani et al., 2017) as the backbone with modified CLIP-B/16 (Radford et al., 2021) configurations. The text encoder is a 12-layer Transformer with 512 hidden dimensions

²<https://huggingface.co/openai/clip-vit-base-patch16>

Table 2: **Zero-shot text/image retrieval.** Reporting recall@K on Flickr30K and COCO.

	COCO 5K						Flickr30K					
	text → image			image → text			text → image			image → text		
	R@1	R@5	R@10									
CLIP	36.2	62.2	72.2	53.4	78.3	85.6	63.0	86.7	92.5	79.6	95.5	98.1
STAIR	41.1	65.4	75.0	57.7	80.5	87.3	66.6	88.7	93.5	81.2	96.1	98.4

Table 3: **Zero-shot classification accuracy.** Reporting the top-1 accuracy (%) across 9 datasets.

	ImageNet	Caltech-101	CIFAR-100	SVHN	DTD	OxPet	OxFlowers	Eurosat	RESISC45
CLIP	65.1	82.3	63.2	42.0	53.6	85.8	67.7	52.4	64.3
STAIR	65.6	82.5	63.4	53.0	56.3	85.9	68.2	51.0	62.8

and 8 attention heads, while the image encoder is a 12-layer ViT with 768 hidden dimensions and 12 attention heads. The text input is tokenized using BERT WordPiece tokenizer (Devlin et al., 2019) with a vocabulary of 30,522 tokens. The maximum text input sequence length is set to 76.

The CLIP model is trained for 600K steps using the LAMB optimizer (You et al., 2020) with a learning rate of $1e^{-3}$ and a weight decay of $1e^{-2}$. The learning rate is warmed up for the first 10k steps and then linear decay to 0. The STAIR model goes through 3 stages, with each stage trained for 300K, 300K, and 600K steps, using the same configuration as CLIP. To mitigate catastrophic forgetting, a smaller max learning rate of $1e^{-4}$ is adopted in stage 3. FLOPs regularization weights are set to $\lambda_1 = \lambda_2 = 1e^{-3}$ by default, following the quadratic growth in Formal et al., 2021a. Unless specified otherwise, all models are trained using a global batch size of 16,384. All our experiments were conducted using 64 A100 GPUs.

It is noted that the proposed STAIR model does not introduce additional parameters compared to CLIP, because the linear projection e for both the image and text modalities parameters with token embedding lookup table as described in Section 2.2. The detailed number of learnable parameters for each model are summarized in Table 1.

4.3 Zero-Shot Text Image Retrieval

Table 2 shows the recall@K (K=1, 5, 10) performance of image/text retrieval tasks on Flickr30K (Plummer et al., 2015) and COCO-5K (Chen et al., 2015). The metrics are reported with a prompt of “a photo of ” added before the original caption, following Radford et al. (2021). We observe a significant improvement of STAIR models compared to the CLIP baseline, with a 4.9% and 4.3% en-

hancement in recall@1 for COCO-5K text→image and image→text retrieval, respectively. Similar improvements are observed on the Flickr30K.

4.4 Zero-Shot Visual Classification

In Table 3, we report zero-shot image classification top-1 accuracy on 9 datasets using the same prompts set from Radford et al. (2021). The results indicate that STAIR either outperforms or performs competitively with CLIP on most datasets. Particularly, STAIR shows significantly better performance on SVHN that requires an exact match, thanks to its token grounding capability. However, we observe that STAIR also struggles with specialized out-of-distribution tasks, such as Eurosat (Helber et al., 2019) and RESISC45 (Cheng et al., 2017), which is consistent with the observations from Radford et al. (2021).

4.5 Linear Probe of Visual Classification

We also compare the linear probe performance using the same 9 datasets in Table 4. We observe that the STAIR model consistently achieves superior results compared to CLIP, even on Eurosat and RESISC45, where it exhibits weaker performance in zero-shot. This improvement can be attributed to the larger embedding dimensionality of 30,522 in the STAIR model, compared to 512 in CLIP. Moreover, with sparsity enforced in STAIR, there is no extra cost on storage and computation compared to the dense embeddings (further discussed in Section 6.3).

5 Interpretability

5.1 Quantitatively analysis

Interpreting high-dimensional embeddings poses a challenge because their representations in the embedding space R^d do not naturally correspond to

Table 4: **Linear probe classification accuracy.** Reporting the top-1 accuracy (%) across 9 datasets.

	ImageNet	Caltech-101	CIFAR-100	SVHN	DTD	OxPet	OxFlowers	Eurosat	RESISC45
CLIP	77.0	94.9	76.7	65.8	80.6	88.8	97.6	95.1	92.6
STAIR	78.1	96.1	77.7	72.5	82.7	91.7	98.2	95.9	93.9

Input		
CLIP	##zoo: 0.34, ##oso: 0.33, ##zhou: 0.31, ##jia: 0.31, ##jiang: 0.31, ##feng: 0.3, ##mei: 0.3, ##hua: 0.3, ##bei: 0.3, ##chen: 0.3	##trip: 0.33, ##hopper: 0.33, ##tsa: 0.32, ##boarding: 0.32, ##jet: 0.31, ##ghan: 0.31, ##eki: 0.31, ##avia: 0.31, ##cat: 0.31, ##cot: 0.31
STAIR	panda: 2.38, bear: 1.63, zoo: 1.52, snow: 1.19, wildlife: 1.16, bears: 1.13, china: 1.1, snowfall: 1.01, giant: 1.01, mammals: 0.98	cat: 1.39, suitcase: 1.14, luggage: 1.08, cats: 1.01, amused: 0.83, muttering: 0.81, traveling: 0.79, pets: 0.78, packing: 0.77, kitten: 0.77

Figure 4: **Examples of most relevant (sub-)words in the lexicon space and the similarities in CLIP and STAIR.** STAIR provides more interpretable visual concepts compared to CLIP. $\#\#$ indicates subword from the vocabulary.

easily understandable human concepts. To overcome this challenge, Kim et al. (2018) proposed leveraging an **interpretable space** R^H spanned by vectors c representing human-interpretable concepts. From this standpoint, interpreting the embedding can be seen as a mapping $\mathcal{F} : R^d \rightarrow R^H$.

For image-text dual encoder models, a suitable interpretable space that connects multimodalities is a lexicon space spanned by basis vectors c representing human-understandable tokens, words, phrases, and/or sentences. The interpretation can then be determined by the similarity between the embedding and each basis vector, $Sim(\cdot, c)$. The lexicon space is crucial for comparing interpretations between different models. For example, if we have an embedding from an image of a dog but the lexicon space does not include a concept for "dog," it becomes challenging to understand the image embedding itself.

Zero-shot image classification is a restricted form of functionally-grounded interpretability evaluation (Doshi-Velez and Kim, 2017) as its interpretable space is predefined by its classes and dataset specific. However, in practice, the interpretable space can be both lack of human labels and unlimited (Ghorbani et al., 2019). To lift the constraint, we expand our interpretable space as the

Table 5: **Improved interpretability of the STAIR model.** We report the top-K accuracy (%) of the label among all of the vocabulary in the tokenizer on the ImageNet, CIFAR-100, and CalTech-101 datasets.

ImageNet	Top-1	Top-10	Top-50	Top-100
STAIR	32.9	69.0	83.8	87.7
CLIP	13.7	34.3	47.0	51.9
CIFAR-100	Top-1	Top-10	Top-50	Top-100
STAIR	10.3	56.8	75.4	80.7
CLIP	8.0	28.9	44.7	50.5
CalTech-101	Top-1	Top-10	Top-50	Top-100
STAIR	29.3	45.4	56.0	64.8
CLIP	8.1	24.2	38.9	43.8

vocabulary of the BERT WordPiece Tokenizer (Devlin et al., 2019), approximating a lexicon space that covers any human-interpretable concepts. Note that, under this definition, the embedding space becomes the same as the interpretable space for STAIR and \mathcal{F} reduces to an identity function. Similar to zero-shot image classification, we consider an image embedding easier to interpret if it is closer to its ground-truth class in the lexicon space. This task is generally more challenging than the traditional zero-shot classification because the candidates now become the entire vocabulary, which is much more than the predefined classes.

We compare the interpretability of CLIP and STAIR on three datasets, ImageNet, CIFAR-100, and Caltech-101 and use the Top-K accuracy as quantitative metrics. Specifically, if a ground-truth class is tokenized into separated sub-words c_k in the vocabulary, we consider the maximum similarity over the sub-words, $\max_k(Sim(\cdot, c^k))$ as the final similarity. As shown in Table 5, our results demonstrate that STAIR exhibits significantly superior interpretability compared to CLIP in the interpretable space.

5.2 Qualitatively analysis

We also conduct a qualitative examination to compare the interpretability of embeddings from the STAIR and CLIP models. In Figure 4, we present the top $Sim(\cdot, c^k)$ (sub-)words in the interpretable



wave flick stillness bird version
gui beach subspecies turbulent
ashore waves reintroduced
coasts photograph vague republished
plumage schleswig beaches

(a) Caption: A seagull standing on the sand of a beach.



weddings parties tierfruits lighted
berries fruit flick receptions
cake lily marriage cheese
slicing cakes traditionally
grapes grapes Wedding marriages

(b) Bride and grooms arms cutting the wedding cake...



opera inhabit race waterfront
Sydney aquatics rowing
prosperous dingy canoes regatta
swimming wider australia pink
paddle australian competitors

(c) A couple of people on kayak boats in the middle...



lax view traveled
mildly nexus window
concourse gate rainy
airport flights delta
airports airline
traveling airplane
airlines

(d) An airport filled with planes sitting on tarmacs.



loch livestock
tyrol cumbria goat
inaccessible sky heading
lamb herd scotland
driving sheep
flock goats flick
roadside animals

(e) Sheared sheep on roadway taken from vehicle.



creditors liver airline spotting
boeing ana nippon
airlines busiest 777 el
aircraft japan airport departing
flights republished

(f) A plane on the runway is being led by a tow cart.



minister ab sign 3 iwa
roads agreed westminster peterborough
minster nw north west
abbey west ey road abby

(g) There is a sign in front of a brick house.



synchronized lake mating
flick subspecies canada
awhile geese brant wildlife
photograph goose version
duck courting ducks sired
swans pond oldies

(h) Two ducks floating together on water.



republished mmm televised
feast pie christmas
thanksgiving dinner
chicken stuffing turkey roast
tasted cans holidays
meal bread

(i) A table set for a traditional Thanksgiving dinner.

Figure 5: **Predictions from STAIR model.** Top 20 tokens predicted by STAIR for an image. Font size indicates prediction weight. The original caption is shown below the image. Detailed weights are provided in Appendix F.

space defined by the BERT WordPiece vocabulary for each image. We see that STAIR is better at capturing visual concepts that humans can easily understand than CLIP, which is consistent with our quantitative analysis. Additionally, we observe that the top tokens from CLIP tend to have similar matching scores while STAIR avoids the problem by adopting Eq. (5) in the Token Projection Head. Figure 5 shows more examples of (sub-)words with the highest weights from STAIR embeddings given each image. The results suggest that STAIR is capable of grounding images to tokens that are semantically related. For example, it can infer “wedding” from the picture depicting a bride and groom cutting a cake. This grounding and interpretability ability of STAIR is highly valuable for debugging and understanding the model’s behavior. For example, we observe a bias towards activating the “https” token in many predicted embeddings. We find that mainly stems from a substantial portion of web-mined content in our training data, where “https” frequently appears in associated texts.

6 Analysis and Discussion

6.1 Necessity of Multi-stage training

Multi-stage training strategy plays crucial role in ensuring STAIR embedding grounded to meaningful tokens. To demonstrate its necessity, we train a single-stage model, denoted as STAIR_{SINGLE-STAGE}

Input	STAIR _{SINGLE-STAGE}	STAIR
	→ #ilitated 1.55 #igh 1.14 norris 1.29 funded 1.13 stella 1.25 ud 1.08 scarcely 1.22 gonzales 1.06 r: 1.18 clue 1.06	sking 1.62 ##riding 1.14 hd 1.59 gripped 1.12 daytime 1.29 crippled 1.11 skier 1.18 silhouette 1.11 flick 1.17 sku 1.07
A person riding downhill on skis on a snowy hill, with large mountains in the background.	scarcely 1.62 thugs 1.35 paralyzed 1.86 #rona 1.18 papal 1.51 reunite 1.09 ##mable 1.49 ##vot 1.07 clue 1.42 kitchens 1.03	mountains 2.11 snowy 1.06 downhill 1.33 capped 0.95 ski 1.31 https 0.86 riding 1.25 deals 0.74 skiing 1.06 skier 0.73
	→ torque 1.50 ##piration 1.00 eminent 1.34 lower 0.99 messied 1.26 ##ilated 0.95 nottingham 1.10 harta 0.94 skirmish 1.08 necessity 0.91	teddy 1.64 ##oh 1.10 bear 1.48 table 1.07 toys 1.19 fair 1.02 bears 1.18 stuffed 1.02 pumpkin 1.12 flick 0.98
A group of different colored teddy bears sitting on top of a blue table.	##eration 2.52 ennial 1.37 eminent 1.99 nottingham 1.32 distant 1.57 ##mable 1.26 bombed 1.56 religions 1.26 invariably 1.46 intuition 1.16	teddy 2.83 buddies 0.90 blue 2.04 colourful 0.88 table 1.67 toys 0.83 https 1.29 plush 0.78 bears 1.22 entertained 0.78

Figure 6: **Examples of top predictions and weights from STAIR models.** STAIR_{SINGLE-STAGE} predictions are disconnected from the actual input meaning, while STAIR predictions from multi-stage training are grounded to input semantics. ## indicates subword from the vocabulary.

for comparison. Figure 6 illustrates the predictions from STAIR_{SINGLE-STAGE} and the multi-stage STAIR models, respectively. We observed that STAIR_{SINGLE-STAGE} tends to redefine the semantic meaning of tokens and repurpose them to match images and text. For instance, tokens such as “eminent” and “nottingham” are redefined as topics related to teddy bears, as they consistently appear for teddy bear related images. Similarly, “clue” is reinterpreted as a concept related to skiing. While we can deduce the new semantic meaning through reverse engineering, this task is non-trivial. In con-

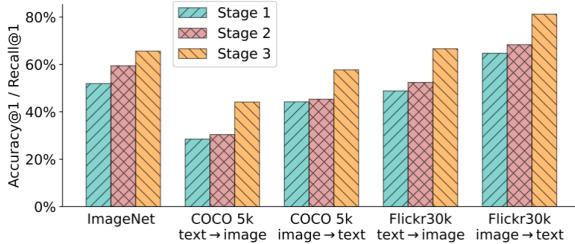


Figure 7: **Ablation on multi-stage training.** Stage 1 establishes a solid foundation for zero-shot performance. Stage 2 enhances performance primarily on ImageNet. Stage 3 provides additional improvements on both ImageNet and COCO/Flickr30K.

trast, the inclusion of multi-stage training ensures that tokens retain their original meaning, resulting in a human-readable embedding³.

6.2 Ablation on Multi-stage training

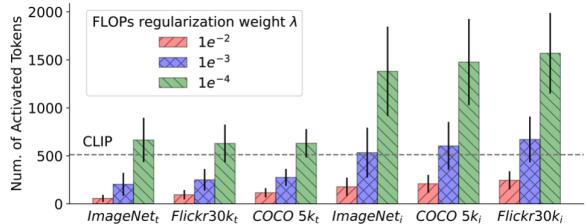
In this section, we qualitatively study the effect of the multi-stage training strategy on zero-shot transfer. Figure 7 shows the top-1 accuracy of ImageNet classification and recall@1 on COCO-5K and Flickr30K image-text retrieval for each stage separately. We observe that even in stage 1, reasonable performance can already be achieved, despite the restricted activation of text tokens. Stage 2 provides additional improvements, particularly in the classification task. Stage 3 significantly enhances the text-image matching capability, as evidenced by the improvements in all metrics.

6.3 Ablation on Sparsity

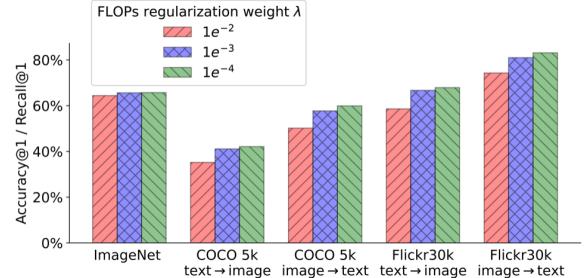
Embedding sparsity is crucial for ensuring the efficiency of similarity computation and retrieval speed. In STAIR, the strength is controlled by the FLOPs regularization weights. To study its impact, we train three STAIR models with regularization weights $\lambda = \lambda_1 = \lambda_2 \in \{1e^{-2}, 1e^{-3}, 1e^{-4}\}$. We examine their text and image embedding sparsity, *i.e.* the number of tokens with non-zero weights in the predictions, as well as their zero-shot transfer performance on ImageNet, COCO-5k, and Flickr30k as summarized in Figure 8.

The results suggest that, when $\lambda \geq 1e^{-3}$, the effective number of tokens in STAIR is significantly lower than the dense embedding dimension of 512 used in CLIP for text embeddings. Since the time

³Interestingly, the STAIR_{SINGLE-STAGE} model is still able to achieve comparable performance to multi-stage training on various retrieval and classification tasks. Further details on STAIR_{SINGLE-STAGE} training and quantitative results can be found in Appendix C.



(a) **Number of Activated Tokens.** Subscripts t and i denote results for text and image, respectively. The horizontal line represents an embedding size of 512 of CLIP baseline. The effective text embedding size in STAIR is significantly smaller than CLIP when $\lambda \geq 1e^{-3}$, enabling faster retrieval.



(b) **Performance on zero-shot transfer.** More sparsity in embedding lowers the zero-shot transfer performance.

Figure 8: **Ablation on FLOPs regularization weights.**

complexity of sparse embeddings dot product is linear to the smaller number of non-zero units in two embeddings, STAIR models are more efficient in conducting similarity computation during retrieval compared to CLIP. Moreover, we observe that more tokens are activated in the image embeddings than in the text embeddings. One explanation is that the image semantics is usually broader and more general while the text meaning is more specific. We also notice that when λ is large, text embedding tends to be more sparse especially when the inputs are shorter. On the other hand, the regularization weights show a negative impact on zero-shot performance, particularly in the case of retrieval tasks.

7 Related Work

Image and text retrieval Image-text retrieval approaches can be categorized into dual-encoder and cross-encoder approaches. In the dual-encoder approach, images and text are encoded separately as dense embeddings. DeViSE (Frome et al., 2013) was one of the pioneering dual-encoder models. With the advent of transformers, Radford et al. (2021) proposed CLIP that leverages large-scale pretraining datasets and established new state-of-the-art across multiple benchmarks. Finetuning the visual model (Dosovitskiy et al., 2021) and lan-

guage model (Devlin et al., 2019) further improves performance. On the other hand, the cross-encoder approach like UNITER (Chen et al., 2020) employs a single encoder for both image and text inputs. While cross-encoder models capture fine-grained alignment between image and text but are slower at inference, dual-encoder models offer faster retrieval speed with precomputed embeddings.

STAIR follows the dual-encoder approach but introduces a novel aspect by using *sparse embeddings* instead of dense embeddings. This choice of sparse embeddings in STAIR leads to improved interpretability and better retrieval performance compared to dense embeddings.

Document retrieval via sparse embedding
Sparse embedding has been widely used in information retrieval (Dai and Callan, 2019; Bai et al., 2020; Jang et al., 2021). By hashing the sparse embeddings, retrieval can be efficiently performed using an inverted index system. Our work draws inspiration from SPLADE (Formal et al., 2021b,a) employing sparse embeddings. However, unlike SPLADE, our approach tackles retrieval across modalities. The semantic gap between images and text makes designing a joint sparse embedding space challenging. Additionally, grounding images and text to meaningful tokens is a non-trivial task. In STAIR, we propose a streamlined approach that enables fast retrieval speed, interpretability, and high retrieval accuracy in the multimodal setting.

8 Conclusion

In this paper, we introduced STAIR (Sparse Text And Image Representation), an approach that encodes image and text inputs into sparse embeddings within a sparse token space. We also employed a multi-stage training strategy to ensure that the embeddings are grounded in meaningful tokens. By comparing STAIR with the CLIP model, we observed that STAIR achieved superior performance on image-text retrieval tasks and demonstrated better results on various zero-shot and linear probing classification tasks. Furthermore, through quantitative and qualitative analysis, we illustrated the interpretability advantage of our sparse embeddings over dense embeddings, making them more easily understandable for humans.

Limitations

While STAIR demonstrates several strengths in image-text representation and interpretation, it is

important to acknowledge its limitations. We discuss the following aspects as areas for future improvement and exploration.

Interpretability Loss of Word Order The vocabulary-based representation employed by STAIR, utilizing a unigram vocabulary for sparse embeddings, demonstrates strong performance in various scenarios. However, it may encounter challenges in interpretability for humans when dealing with phrases or sentences that rely on the specific order of words, such as distinguishing between "dog chasing cat" and "cat chasing dog". While the sparse embeddings still maintain competitive performance in terms of text-image semantic matching, humans may find it difficult to infer the intended semantics solely from individual activated tokens. To address this limitation, future research could explore incorporating bigram representations or combining sparse and dense embeddings to capture a wider range of linguistic nuances and enhance interpretability.

Prediction Bias Another limitation pertains to potential prediction biases that can arise from the training data, particularly due to web-mined content (Chang et al., 2019; Dev et al., 2021). The prevalence of certain concepts or biases in web data can influence the model’s predictions, leading to unintended biases in image-text retrieval or classification tasks, as well as impacting interpretability. Addressing this issue requires careful consideration of the training data sources and strategies to mitigate biases during model training.

Computation Overhead of Multi-Stage Training We acknowledge that the multi-stage training process in STAIR requires more iterations than the baseline CLIP model, resulting in increased computation overhead. It is worth noting that the additional computational cost through the multi-stage training is necessary to achieve interpretability. We provide a fair comparison by reporting the performance of a single-stage STAIR model trained under the same computation budget in Appendix C. The results demonstrate that even the single-stage STAIR model performs competitively with or outperforms CLIP, showcasing the efficacy of the approach within given computational constraints.

Acknowledgements

We thank our teammates from Visual Intelligence, foundation model, and other Apple groups for their

feedback and suggestion. Sepcial thanks goes to Navdeep Jaitly, Brandon McKinzie, Xianzhi Du for the early review of the paper, Floris Weers, Vaishaal Shankar for linear probing experiments setup and Cybertron team for infrastructure supports.

References

- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.
- Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. Sparterm: Learning term-based sparse representation for fast text retrieval. *arXiv preprint arXiv:2010.00768*.
- Kai-Wei Chang, Vinodkumar Prabhakaran, and Vicente Ordonez. 2019. Bias and fairness in natural language processing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): Tutorial Abstracts*.
- Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568.
- Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer.
- Gong Cheng, Junwei Han, and Xiaoqiang Lu. 2017. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883.
- Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. 2004. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague.
- Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687*.
- Sunipa Dev, Masoud Monajatipoor, Anaelia Ovalle, Arjun Subramonian, Jeff Phillips, and Kai-Wei Chang. 2021. Harms of gender exclusivity and challenges in non-binary representation in language technologies. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1994.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.
- Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.
- Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*.
- Li Fei-Fei and Pietro Perona. 2005. A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 2, pages 524–531. IEEE.
- Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021a. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.
- Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021b. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.
- Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc’Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26.
- Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. 2019. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32.
- Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. 2020. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, pages 3887–3896. PMLR.

- Shayan Hassantabar, Zeyu Wang, and Niraj K Jha. 2021. Scann: Synthesis of compact and accurate neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(9):3012–3025.
- Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226.
- Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528.
- Kyoung-Rok Jang, Junmo Kang, Giwon Hong, Sung-Hyon Myaeng, Joohee Park, Taewon Yoon, and Heecheol Seo. 2021. Ultra-high dimensional sparse representations with binarization for efficient text retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1016–1029.
- Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019a. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019b. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910.
- 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.
- Carlos Lassance and Stéphane Clinchant. 2022. An efficiency study for splade models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2220–2226.
- Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR’06)*, volume 2, pages 2169–2178.
- Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. 2022. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975.
- Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021a. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362.
- Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021b. Contextualized query embeddings for conversational search. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015.
- Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Ming Yang, T Cour, Kai Yu, Liangliang Cao, and T Huang. 2011. Large-scale image classification: Fast feature extraction and svm training. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1689–1696.
- David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.
- Biswajit Paria, Chih-Kuan Yeh, Ian EH Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. 2020. Minimizing flops to learn efficient sparse representations. In *International Conference on Learning Representations*.
- Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.
- 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.
- Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens. 2022. Perceptual grouping in vision-language models. *arXiv preprint arXiv:2210.09996*.

Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. 2013. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105:222–245.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Xiaolong Huang, Binxing Jiao, Linjun Yang, and Dixin Jiang. 2022. Lexmae: Lexicon-bottlenecked pre-training for large-scale retrieval. *arXiv preprint arXiv:2208.14754*.

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.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*.

Tiancheng Zhao, Xiaopeng Lu, and Kyusong Lee. 2021. Sparta: Efficient open-domain question answering via sparse transformer matching retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 565–575.

A Clean Licensed Dataset

The High Quality Image Text Pairs (a.k.a. HQITP-134m) dataset consists of approximately 134m diverse and high quality images paired with descriptive captions and titles. Images range in spatial resolution from 320 to 2048 pixels on the short side. All images are JPEG format and most are RGB. Each example image is associated with a title, and a list of several captions. On average, these captions consist of 20.1 tokens, with the shortest caption consisting of just one token and the longest spanning over 1000 tokens. This dataset was licensed to our industrial research lab by a third party for commercial use.

B Impact of training data and global batch size

In Section 3, both the CLIP and STAIR models were trained on the 1.1B dataset with a batch size of 16,384, which was determined by our computation budget. To investigate the impact of training data and global batch size, we conducted additional experiments. Specifically, we trained a CLIP model with a batch size of 32,768, denoted as CLIP_{32K}, and compared its performance with the OpenAI CLIP-B/16 benchmark (Radford et al., 2021). The benchmark model was trained on a 400M dataset with a batch size of 32,768, as shown in Table 6. Our implemented CLIP_{32K} performs competitively on ImageNet and achieves significantly better results on retrieval tasks when trained on our larger 1.1B dataset.

Furthermore, we observed that the choice of global batch size has a noticeable impact on both zero-shot classification and retrieval benchmarks. Despite using a smaller batch size, the STAIR model outperforms other models on the COCO image/text retrieval task. This highlights the effectiveness of the STAIR approach even with a reduced batch size.

It is important to emphasize that our primary focus is on introducing a novel approach to image-text representation learning. Despite smaller batch size, fair comparisons can be made when models are trained under similar condition. The STAIR approach is not limited to a specific model and can be applied to other dual-encoder architectures such as ALIGN (Jia et al., 2021) and GLIP (Li et al., 2022).

C Single-stage training

In this section, we investigate the performance of the STAIR model when trained without the multi-stage training strategy described in Section 3. We denote this variant as STAIR_{SINGLE-STAGE}. The STAIR_{SINGLE-STAGE} model shares the same architecture as the multi-stage STAIR model depicted in Figure 1, and it is trained using the same configurations for 600K steps as the CLIP model described in Section 4.2. The FLOPs regularization weights λ_1 and λ_2 are set to the same values as the STAIR model, i.e., $\lambda_1 = \lambda_2 = 1e^{-3}$.

Table 7 presents the zero-shot image-text retrieval and classification performance of the two versions of the STAIR model and the baseline CLIP model. Remarkably, the STAIR_{SINGLE-STAGE}

Table 6: **Zero-shot transfer of open-source CLIP vs our CLIP using 32K batch size.** We report recall@K on Flickr30K and COCO, and top-1 accuracy (%) on ImageNet. We attach our 16K batch-size STAIR metrics for reference.

	Data Size	Batch Size	ImageNet		COCO 5K			Flickr30K		
			Acc@1	R@1	text → image	image → text	text → image	image → text	R@1	R@1
CLIP _{OPENAI}	400M	32,768	68.6	33.3		54.1		63.5		82.3
CLIP _{32K}	1.1B	32,768	68.3	39.0		57.4		67.0		85.0
CLIP	1.1B	16,384	65.1	36.2		53.4		63.0		79.6
STAIR	1.1B	16,384	65.6	41.1		57.7		66.6		81.2

Table 7: **Zero-shot text/image retrieval and classification of STAIR_{SINGLE-STAGE}.** We report recall@K and top-1 accuracy (%). Bold indicates the best overall performance.

	COCO 5K						Flickr30K					
	text → image			image → text			text → image			image → text		
	R@1	R@5	R@10									
CLIP	36.2	62.2	72.2	53.4	78.3	85.6	63.0	86.7	92.5	79.6	95.5	98.1
STAIR _{SINGLE-STAGE}	40.5	64.9	74.9	57.7	80.3	87.4	66.7	88.6	93.7	81.0	96.7	98.5
STAIR	41.1	65.4	75.0	57.7	80.5	87.3	66.6	88.7	93.5	81.2	96.1	98.4

	ImageNet	Caltech-101	CIFAR-100	SVHN	DTD	OxPet	OxFlowers	Eurosat	RESISC45
CLIP	65.1	82.3	63.2	42.0	53.6	85.8	67.7	52.4	64.3
STAIR _{SINGLE-STAGE}	64.0	81.5	63.7	39.4	56.7	85.9	67.0	49.5	62.2
STAIR	65.6	82.5	63.4	53.0	56.3	85.9	68.2	51.0	62.8

model achieves similar performance to the multi-stage trained model in terms of zero-shot text/image retrieval and classification tasks. Both STAIR_{SINGLE-STAGE} and STAIR models outperform the CLIP model across most of the metrics.

However, it is important to note that the interpretability of the STAIR_{SINGLE-STAGE} model is considerably worse, as indicated in Table 8. As discussed in Section 5.2, the STAIR_{SINGLE-STAGE} model functions as a multi-modal clustering algorithm that repurposes words as weighted token centroids. The contrastive objective trains the model to prioritize matching the aligned image and text using these token centroids. However, it lacks the capability to effectively constrain the predicted tokens to their original human-readable meanings compared to multi-stage training.

D Token Based vs Embedding Based Search

The development of embedding-based retrieval systems has gained significant attention in recent years (Hassantabar et al., 2021; Johnson et al., 2019a; Lin et al., 2021b). While these systems show promising progress, deploying large-scale embedding-based search systems still poses several challenges. Firstly, embedding-based systems

often rely on approximated neighbor search techniques, such as k-means clustering and product quantization, to handle large-scale scenarios. These operations can be computationally costly and may require additional approximations or quantization to reduce memory usage, leading to a loss of precision. Additionally, updating the index with new data often necessitates extra computation or even re-computation, as operations like k-means clustering and product quantization are data-dependent.

In contrast, token-based retrieval systems in STAIR do not face these challenges, as the indexing is based directly on the tokens themselves. Furthermore, the tokens in STAIR are optimized with a FLOPs regularizer to encourage a uniform distribution among sparse tokens, which benefits the retrieval process. Moreover, the interpretable nature of the token-based system provides additional advantages, such as the ability to build customized query trees using logical operators, leverage token-based blacklists/whitelists on both the query and index side, and combine with other token-based features in the inverted index system. This benefits applications like web image search, where supplementary signals such as image alt text and surrounding textual content play important role besides the image content itself.

Table 8: **Interpretability of STAIR_{SINGLE-STAGE}**. We report the top-K accuracy (%) of the label among all of the vocabulary. Bold indicates the best overall performance.

	ImageNet		CIFAR-100		CalTech	
	Top-1	Top-100	Top-1	Top-100	Top-1	Top-100
CLIP	13.7	51.9	8.0	50.5	8.1	43.8
STAIR _{SINGLE-STAGE}	0.0	0.6	0.0	0.3	0.0	0.2
STAIR	32.9	87.7	10.3	80.7	29.3	64.8

Table 9: **Retrieval Speed Comparison**. STAIR achieves better retrieval speed compared to CLIP with comparable retrieval accuracy (%).

	QPS	R@10	R@25
CLIP	6.02	42.1	55.2
STAIR	14.1	42.5	55.7

D.1 Retrieval Speed

Retrieval speed between dense and sparse embedding has been extensively studied in the information retrieval community (Shen et al., 2022; Lassance and Clinchant, 2022). For instance, the Shen et al. (2022) demonstrates that a sparse model can achieve similar or better efficiency and accuracy trade-offs compared to a dense model. One can tune the retriever, such as using the top N tokens, which still results in strong performance. However, latency and accuracy depend heavily on the implementation and configuration of retrieval systems.

We conducted a preliminary comparison of CLIP and STAIR models using the popular retrieval libraries FAISS (Johnson et al., 2019b) and Pyserini (Lin et al., 2021a) on a 1M-scale dataset sampled from COCO (Chen et al., 2015). The preliminary results are summarized in Table 9 showing that STAIR is capable of achieving 2x speed over CLIP while maintaining comparable retrieval accuracy. Comprehensive comparison would require more extensive experimentation, which we will leave as future work.

E Text Encoder Free Applications

The token grounding capability of the STAIR model opens up possibilities for more efficient approaches to existing tasks. In this section, we explore two potential applications: 1) *Text Encoder-Free Localization*, and 2) *Text Encoder-Free Image-Text Retrieval*.

E.1 Text Encoder Free Localization

The STAIR model is capable of localizing image regions related to an arbitrary query (e.g., an object) from the vocabulary without relying on the text side for inference ⁴.

Recall that the image encoder uses a vision transformer that divides the original image into grids. Equation 4 introduces the mapping function $p(\cdot)$, which projects each token/grid representation to the vocabulary space, where each dimension represents the activation of the corresponding token from the vocabulary. Building on this, we can identify the regions correlated with a query by examining the activation scores of the input tokens at each grid. Figure 9 illustrates examples of images given arbitrary text queries and their corresponding activation heatmaps. In the first example, we visualize the activation maps for the queries “German Shepherd Dog”, “Dog”, and “Cat”. Both “German Shepherd Dog” and “Dog” align well with the actual dog in the image. In contrast, the activations for the “Cat” query spread across the entire image. Additionally, we observe that the activation heatmap for “Dog” aligns better compared to the heatmap for “German Shepherd Dog”. This is because the multi-token query is decomposed into multiple tokens from the pre-defined vocabulary, and tokens “German” and “Shepherd” often represent other concepts rather than a dog. As mentioned in Section 8, a better-tuned vocabulary, which includes “German Shepherd Dog” as a single token, would help address this issue.

We hypothesize that the localization capability of STAIR models stems from the max pooling operation in Eq. 5. According to Ranasinghe et al., 2022, simply changing the image and text encoder poolers to max pooling can significantly enhance the localization and segmentation capability of a CLIP model. While the CLIP model still requires text encoding to align with the patch representa-

⁴In cases where a query is not in the vocabulary or consists of multiple tokens, the query are tokenized into in-vocabulary tokens, and their average is used for localization.

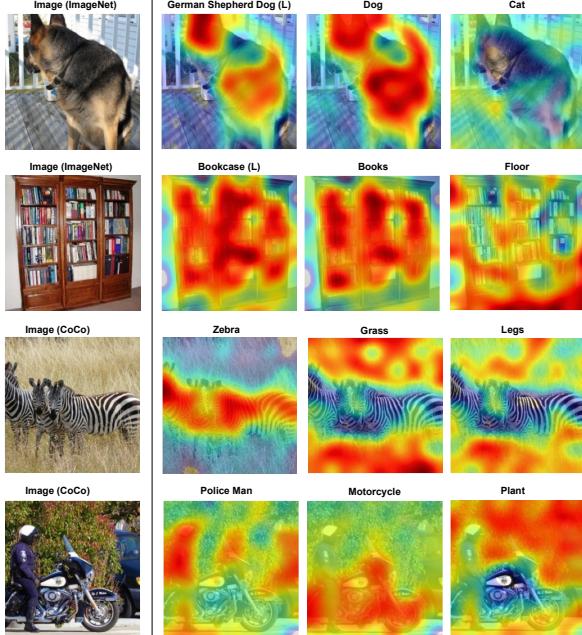


Figure 9: **Qualitative examples of text inference free localization in STAIR**. Left side shows the original images from ImageNet or COCO 5K. Right side shows the queries and their activation heatmaps. (L) indicates the query is the original ImageNet label of the image. (The heatmap colors are normalized for better visualization, not representing real values.)

tions from the image, STAIR does not rely on any inference from the text side. These findings highlight the potential of STAIR models for localization, segmentation, and open vocabulary detection tasks, which we leave as future work.

E.2 Text Encoder Free Image-Text Retrieval

Another notable advantage of the STAIR model is its ability to enable text-encoder-free retrieval systems. Figure 10 compares the dual-encoder architecture with the text-encoder-free architecture. Specifically, we utilize the image encoder of STAIR to generate sparse image embeddings, while the texts are directly converted into MASK in the vocabulary space after tokenization, serving as the sparse text embeddings. Unlike the dual-encoder approach that requires inference from both the image and text encoder, this architecture only relies on the former, making it suitable for applications with restricted latency requirements. Note that it also differs from retrieval systems built with fixed taxonomies using inverted indexes since it can handle any free text inputs.

Table 10 summarizes the zero-shot performance of the text-encoder-free STAIR model, denoted

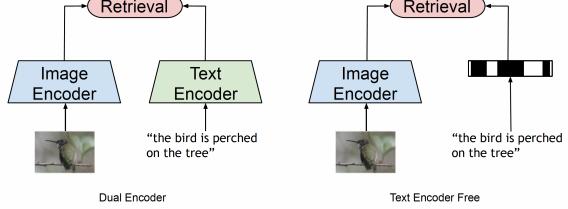


Figure 10: **Dual encoder vs text encoder free architecture**. Text-encoder-free architecture contains only image encoder and uses MASK as text embedding.

as STAIR_{IMAGE}, compared to the original STAIR. While it may not match the performance of the dual encoder baseline, the results are encouraging given its potential. We observe that the text-encoder-free STAIR demonstrates relatively stronger performance in ImageNet classification compared to text/image retrieval tasks. This discrepancy is due to the fact that ImageNet classes are generally more concise. In contrast, captions from COCO-5k and Flickr30k often contain more stop words, which are treated with equivalent importance as semantically meaningful tokens by MASK. This indicates significant room for improvement in text-encoder-free performance, which we leave as future work.

F Image Prediction Weights

We provide the detailed weights of predicted tokens for each image from Figure 5 in the Table 11 for reference.

Table 10: **Zero-shot of STAIR with and without text encoder inference.** We report recall@K on Flickr30K and COCO, and top-1 accuracy (%) on ImageNet.

	ImageNet Acc@1	COCO 5K		Flickr30K	
		text → image R@1	image → text R@1	text → image R@1	image → text R@1
STAIR	65.6	41.1	57.7	66.6	81.2
STAIR _{IMAGE}	50.0	21.0	27.2	40.4	47.0

Table 11: **Detailed weights of STAIR predicted tokens.** Top 20 tokens predicted by STAIR for an image. ## indicates subword from the vocabulary.

		
##gul: 2.09, ##ll: 1.56, photograph: 1.35, plumage: 1.19, waves: 1.18, ashore: 1.13, beaches: 1.12, beach: 1.1, bird: 1.09, reintroduced: 1.08, version: 1.07, subspecies: 1.04, schleswig: 0.97, flick: 0.89, vague: 0.84, wave: 0.81, stillness: 0.79, turbulent: 0.79, republished: 0.78, coasts: 0.78	cake: 1.75, wedding: 1.47, cakes: 1.31, flick: 1.26, tier: 1.25, grapes: 1.23, fruit: 1.18, weddings: 1.15, receptions: 1.09, marriage: 1.0, ##lr: 0.99, ##illy: 0.98, slicing: 0.98, fruits: 0.88, berries: 0.87, marriages: 0.86, traditionally: 0.86, cheese: 0.86, lighted: 0.84, parties: 0.82	sydney: 1.98, australia: 1.92, australian: 1.42, opera: 1.33, kay: 1.32, ##dling: 1.27, paddle: 1.17, canoe: 1.16, harbour: 1.1, rowing: 1.04, competitors: 1.01, swimming: 0.97, regatta: 0.96, prosperous: 0.95, aquatics: 0.88, race: 0.85, inhabit: 0.85, waterfront: 0.79, pink: 0.79, wider: 0.78
		
airport: 1.6, airline: 1.35, airlines: 1.22, window: 1.22, airports: 1.21, delta: 1.21, nexus: 1.06, lax: 0.99, concourse: 0.97, ##fk: 0.92, gate: 0.87, flights: 0.86, traveled: 0.84, view: 0.82, airways: 0.79, aircraft: 0.76, mildly: 0.76, ##lr: 0.75, rainy: 0.75, traveling: 0.75	sheep: 1.74, herd: 1.54, goats: 1.36, cumbria: 1.32, flick: 1.29, livestock: 1.29, driving: 1.27, goat: 1.24, scotland: 1.09, traffic: 1.04, lamb: 1.02, loch: 1.01, tyrol: 0.94, skye: 0.92, unusual: 0.92, flock: 0.91, animals: 0.9, heading: 0.89, roadside: 0.87, inaccessible: 0.85	nippon: 2.26, ana: 2.03, airline: 1.7, aircraft: 1.4, airlines: 1.37, 77: 1.36, airport: 1.3, el: 1.16, flights: 1.09, departing: 1.06, ##7: 1.01, republished: 0.92, busiest: 0.91, boeing: 0.9, livery: 0.88, aviation: 0.87, creditors: 0.85, spotting: 0.8, japan: 0.75, commons: 0.72,
		
abbey: 2.7, westminster: 2.43, nw: 2.03, ##minster: 1.87, northwest: 1.82, west: 1.78, ab: 1.67, ##bey: 1.62, roads: 1.6, ##3: 1.59, ##ey: 1.56, road: 1.51, abby: 1.38, sign: 1.23, minister: 1.22, peterborough: 1.21, ##mist: 1.13, nwa: 1.0, agreed: 0.97, n: 0.94	geese: 1.86, goose: 1.75, canada: 1.4, photograph: 1.31, mating: 1.18, ducks: 1.18, ##oides: 1.14, brant: 1.13, subspecies: 1.12, wildlife: 1.11, awhile: 1.1, courtship: 1.08, version: 1.04, swans: 0.97, lake: 0.97, pond: 0.95, sired: 0.95, synchronized: 0.94, duck: 0.94, flick: 0.93	thanksgiving: 1.72, turkey: 1.46, roast: 1.26, very: 1.25, pie: 1.23, christmas: 1.04, ##cans: 1.02, mmm: 1.01, chicken: 0.99, holidays: 0.88, feast: 0.82, republished: 0.81, stuffing: 0.78, dinner: 0.76, bread: 0.76, tasted: 0.75, convinces: 0.75, televised: 0.74, meal: 0.73