Improved Text-Image Matching by Mitigating Visual Semantic Hubs

Anonymous EMNLP-IJCNLP submission

Abstract

The *hubness problem* widely exists in high-dimensional embedding space and is a fundamental source of error for cross-modal matching tasks. In this work, we study the emergence of hubs in *Visual Semantic Embeddings* (VSE) with application to text-image matching. We introduce novel methods that mitigate hubs during both training and inference. For training, we analyze the pros and cons of two widely adopted optimization objectives and propose a novel hubness-aware loss function. The loss is self-adaptive in the sense that it utilizes local statistics to scale up the weights of "hubs" within a mini-batch. For inference, we propose a heuristic algorithm that imposes hard constraints on the existence of hubs in the predicted graph. It can be combined with previously proposed cross-modal retrieval criterion which together achieve even better performance. We experiment our methods with various configurations of model architectures and datasets. Both the loss function and the heuristic algorithm exhibit surprisingly good robustness and bring consistent improvement on the task of text-image matching across all settings. Specifically, we report results on Flickr30k and MS-COCO datasets that are above the state-of-the-art.

1 Introduction

The hubness problem is a general phenomenon in high-dimensional space where a small set of source vectors, dubbed hubs, appear too frequently in the neighborhood of target vectors (Radovanović et al., 2010). As embedding learning going deeper, it has been a concern in various contexts including object classification (Tomašev et al., 2011), image feature matching (Jegou et al., 2008) in Computer Vision and word embedding evaluation (Schnabel et al., 2015; Faruqui et al., 2016), word translation (Dinu et al., 2015; Lazaridou et al., 2015) in NLP. It is described as "a new aspect of the dimensionality curse"(Bellman, 1961; Schnitzer et al., 2012).

In this work, we study the hubness problem in the task of text-image matching. In recent years, deep neural models have gained a significant edge over non-neural methods in cross-modal matching tasks (Wang et al., 2016). Text-image matching has been one of the most popular ones among them. Most deep methods involve two phases: 1) training: two neural encoders (one for image and one for text) are learned end-to-end, mapping texts and images into a joint space, where items (either texts or images) with similar meanings are close to each other; 2) inference: for a query vector in modality A, a nearest neighbor search is performed to match the query vector against all item vectors in modality B. As the embedding space is learned through jointly modeling vision and language, it is often referred as *Visual Semantic Embeddings* (VSE). Recent work on VSE has shown a clear trend of growing dimensions in order to obtain better embedding quality (Wehrmann, 2018). With embeddings going deeper, visual semantic hubs increase dramatically. This property is undesired as we firmly know that a one-to-one mapping exists among text and image points during both training (within a mini-batch) and inference (within the validation/test set).

However, the hubness problem is not well addressed by current methods neither in training nor inference. For training, current VSE models use either sum-margin (SUM, Eq. (2)) or max-margin (MAX, Eq. (3)) ranking loss to cluster the positive pairs and push away the negative pairs. SUM is robust across various settings but does not utilize information from hard samples and does not address the hubness problem at all. MAX excels at mining hard samples and achieves state-of-the-art on MS-COCO (Faghri et al., 2018). However, it also does not explicitly consider the hubness problem, nor does it resist noise well. To combine robustness with information from hard samples and hubs, we propose a self-adjustable hubness-aware loss called HAL. It is inspired by Zelnik-Manor and Perona who used local statistics to reweight affini-

ties among two sets of points to automate spectral clustering. HAL leverages information of hubs to automatically adjust weights of negative samples. It learns from hard samples and is robust to noise at the same time by taking multiple samples into account. Specifically, we exploit a sample's relationship with its k-nearest neighbor queries within a mini-batch to decide its weight. The larger a hub is, the more it contributes to the loss. Through a thorough empirical comparison, we show that our method outperforms SUM and MAX loss on various datasets and architectures.

The inference phase of text-image matching mainly refers to the process of obtaining an actual matching from the text and image embeddings. Dinu et al. showed how naive nearest neighbor search (NNS) is flawed for this need. However, to the best of our knowledge, it has been the only strategy adopted by deep text-image matching methods in recent years. While it receives little attention in text-image matching, it can be formulated as a well-studied problem in Combinatorial Optimization (CO): the *assignment problem*. And most recently, within the NLP community, it is extensively studied in the context of Bilingual Lexicon Induction (BLI) which aims to produce a one-to-one mapping among two sets of word vectors. Smith et al. used Inverted Softmax (IS) to reweight similarity scores leveraging an item's distance with all queries. Lample et al. proposed Cross-modal Local Scaling (CSLS) to reduce the scores of items that appear frequently in the neighborhood of multiple queries. Both IS and CSLS are targeting the cross-modal hubs with *soft* criteria. As we do have the strong prior that the final text-image correspondence is a bipartite matching[1], we impose a *hard* constraint on the predicted graph. We propose a heuristic algorithm called Relaxed Greedy Matching (RGM). It is adapted from Greedy Matching (GM) algorithm proposed by Kollias et al. but exhibits much better empirical performance by a small modification which we will explain in detail in section 3.3.3. We will show in experiments how RGM can be combined with IS/CSLS and achieve R@K scores that are well above the widely used NNS.

The two major contributions of this work are:

- a self-adaptive hubness-aware loss function (HAL) that achieves the state-of-the-art across different datasets and model architectures;
- a heuristic algorithm that produces refined prediction from an embedding similarity matrix which further advances the state-of-the-art.

[1]In CO, a *matching* in a bipartite graph is a set of edges chosen in such a way that no two edges share an endpoint. In our context, it means no item should be the nearest neighbor of more than one query.

2 Related Work

In this section, we introduce works from two fields which are highly-related to our work: 1) text-image matching and VSE; 2) tackling the hubness problem in various contexts.

2.1 Text-image Matching and VSE

Since the dawn of deep learning, works have emerged using a two-branch architecture to connect language and vision. In 2010, Weston et al. trained a *shallow* neural network to map word-image pairs into a joint space for image annotation. In 2013, Frome et al. brought up the term VSE and trained joint embeddings for sentence-image pairs. Later works extended VSE for the task of text-image matching (Hodosh et al., 2013; Kiros et al., 2015; Gong et al., 2014; Vendrov et al., 2016; Hubert Tsai et al., 2017; Faghri et al., 2018; Wang et al., 2019), which is also our task of interest. Notice that text-image matching is different from generating novel captions for images (Lebret et al., 2015; Karpathy and Fei-Fei, 2015) but to retrieve existing descriptive texts or images in a database.

While many of these works improve model architectures for training VSE, few have tackled the shortcomings in learning objectives. Faghri et al. made the latest attempt to reform the long being used SUM loss. Their proposed MAX loss is indeed a much stronger baseline than SUM in most data and model configurations. But it fails significantly when the dataset is small or noise is contained. Shekhar et al.; Shi et al. raised concerns over this issue. They mainly focused on creating better training data while we target the training objective itself.

2.2 Tackling the Hubness Problem

We have stated what the hubness problem is in the introduction. Now we introduce works from 1) Bilingual Lexicon Induction (BLI) and 2) Combinatorial Optimization that could be used to tackle the problem.

BLI is the task of inducing word translations from monolingual corpora in two languages (Irvine and Callison-Burch, 2017). The bilingual word vectors are usually trained from methods based on Distributional Semantics like (Mikolov et al., 2013). The word translation problem thus converts to finding the appropriate matching among two sets of vectors (which is similar to our task of interest). Smith et al.; Lample et al.

proposed to first conduct a direct Procrustes Analysis and then use criteria that heavily punish hubs during inference to reduce the hubness problem. Joulin et al. integrated the inference criterion CSLS from (Lample et al., 2018) into a least-square loss and trained a transformation matrix end-to-end. Though this work has a similar philosophy to ours, it is specifically designed for BLI and only trains one linear layer over two sets of word vectors. When CSLS is appended to a triplet loss like ours, it is merely a resampling of hard samples, making it non-special in terms of both form and intuition.

As mentioned in the introduction, the inference phase of text-image matching can also be formulated as an assignment problem. We can completely eliminate the existence of hubs by solving the matching problem this way as all queries would be matched to a fixed number of items (and vice versa). Kuhn proposed the famous Hungarian algorithm for producing maximum weight matching in the 1950s. Murty extended the idea to the k-best assignment algorithm to obtain a list of k-best candidates for all queries (so a ranking is accessible). However, (Kuhn, 1955; Murty, 1968) run in $O(n^3)$ and are thus inapplicable for large scale embeddings. Our method (RGM) is modified from a recent heuristic algorithm proposed by Kollias et al. and is capable of real-time inference.

3 Method

We first introduce the basic formulation of VSE model in section 3.1. In section 3.2 and 3.3, we review several existing methods that we will compare to and also combine with; then propose our intended methods for training and inference respectively. In the end, we state the tools used for measuring hubs in section 3.4.

3.1 Basic Formulation

The bidirectional text-image matching framework consists of a text encoder and an image encoder. The text encoder is composed of word embeddings, a GRU (Chung et al., 2014) (or other sequential models) layer and a temporal pooling layer. The image encoder is usually a deep CNN and a linear layer. We use ResNet152 (He et al., 2016), Inception-ResNet-v2 (IRv2) (Szegedy et al., 2017) and VGG19 (Simonyan and Zisserman, 2014) pre-trained on ImageNet (Deng et al., 2009) in our models. We denote them as functions f and g, which map text and image to some vectors of size d respectively.

For a text-image pair (t,i), the similarity of t and i is measured by cosine of their normalized encodings:

$$s(i,t) = \left\langle \frac{f(t)}{\|f(t)\|_2}, \frac{g(i)}{\|g(i)\|_2} \right\rangle : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}. \quad (1)$$

During training, a margin based triplet ranking loss is adopted to cluster positive pairs and push negative pairs away from each other. There are mainly two prevalent choices which are SUM and MAX. We introduce them in the next section along with our newly proposed loss HAL.

3.2 Training Objectives

In section 3.2.1 and 3.2.2 we restate the two popular loss functions that have been adopted for training VSE and analyze their pros and cons. In section 3.2.3 we introduce our proposed Hubness-Aware Loss (HAL).

3.2.1 Sum-margin Loss (SUM)

SUM has been used for training VSE since the start of this line of work (Frome et al., 2013; Kiros et al., 2015). Its early form can be found in (Weston et al., 2010) which was used for training joint word-image embeddings. SUM is defined as:

$$\begin{aligned} \min_\theta & \sum_{i \in I} \sum_{\bar{t} \in T \setminus \{t\}} [\alpha - s(i,t) + s(i,\bar{t})]_+ \\ & + \sum_{t \in T} \sum_{\bar{i} \in I \setminus \{i\}} [\alpha - s(t,i) + s(t,\bar{i})]_+, \end{aligned} \quad (2)$$

where $[\cdot]_+ = \max(0,\cdot)$; α is a preset margin; T and I are all text and image encodings in a mini-batch; t is the descriptive text for image i and vice versa; $\bar{t}$ denotes non-descriptive texts for i while $\bar{i}$ denotes non-descriptive images for t.

3.2.2 Max-margin Loss (MAX)

Faghri et al. proposed MAX fairly recently (2018). Though MAX was not used in the context of VSE before, it was thoroughly exploited in other embedding learning tasks (Wu et al., 2017). MAX differs from SUM by considering only the hardest negative sample within the mini-batch (instead of summing over all margins).

$$\begin{aligned} \min_\theta & \sum_{i \in I} \max_{\bar{t} \in T \setminus \{t\}} [\alpha - s(i,t) + s(i,\bar{t})]_+ \\ & + \sum_{t \in T} \max_{\bar{i} \in I \setminus \{i\}} [\alpha - s(t,i) + s(t,\bar{i})]_+, \end{aligned} \quad (3)$$

***Pseudo* hardest negatives.** The existence of pseudo hardest negative in training data is a major problem for MAX. During training, as only the

hardest sample in a mini-batch is considered, if that sample happens to be incorrectly labeled or inaccurate, misleading gradients would be imposed on the network. Notice that SUM eases such noise in labels by taking all samples in a mini-batch into account. When a small set of samples are with false labels, their false gradients would be canceled out by other correct negatives within the mini-batch, preventing the model from an optimization failure or overfitting to incorrect labels. That being said, SUM fails to make use of hard samples and does not address the hubness problem at all. It thus performs poorly on a well-labeled dataset like MS-COCO.

3.2.3 The Hubness-Aware Loss (HAL)

On the one hand, we desire a certain degree of robustness through considering multiple samples; on the other hand, we wish the samples being considered are hard enough - so that the training is effective. We tackle this problem leveraging information from visual semantic hubs. Inspired by Zelnik-Manor and Perona, we propose a self-adaptive loss that reweights samples within a mini-batch according to local statistics. More specifically, HAL assigns more weights to negatives which appear to be hubs (being close neighbors to multiple queries).

We define a function $\mathrm{kNN}(x,M,k)$ to return the k closest points (measured by l_2 distance) in point set M to x and HAL can be formulated as:

$$s'(i,t)=s(i,t)\cdot e^{\sum_{\bar{t}\in K_1}\frac{s(i,\bar{t})}{|K_1|}+\sum_{\bar{i}\in K_2}\frac{s(\bar{i},t)}{|K_2|}} \quad (4)$$

where $K_1=\mathrm{kNN}(i,T\backslash\{t\},k), K_2=\mathrm{kNN}(t,I\backslash\{i\},k)$. Then we normally apply SUM on the similarity matrix reweighted by Eq. (4). Though the default configuration of HAL is Eq. (4)+(2), MAX could also be combined by switching Eq. (2) to Eq. (3).

HAL vs MAX. As pointed out by Lazaridou et al., MAX actually implicitly mitigates the hubness problem by targeting the hardest sample only. A hub, by definition, is a close (potentially nearest) neighbor to multiple queries and would thus be punished by MAX for multiple times (in different batches). Lazaridou et al.'s experiments also verified such theory empirically. However, it is a risky choice as the hardest sample within a mini-batch can easily be a pseudo hardest negative as analyzed in section 3.2.2. As we would show in experiments, HAL prevails in a broader range of data and model configurations. In some specific circumstances where both training data and encoders are of ideal quality, we could combine MAX with HAL to reach optimal performance.

Also, HAL is essentially leveraging more information than MAX. SUM considers the anchor's relation with the positive and all negatives; MAX goes one step further to also exploit relations among the negatives; HAL digs into negatives' relations with other queries (besides the anchor) to decide the importance of negatives.

3.3 Inference Objectives

The standard procedure for text-image matching inference is a naive nearest neighbor search (NNS). This, however, easily leads to severe hubness problem as suggested by Dinu et al.; Lazaridou et al.. We thus leverage the prior that "one item should not be a close neighbor to *too* many queries" to improve the predicted matching. In the following, we will introduce a class of cross-modal matching algorithms that punish the existence of hubs during inference. In section 3.3.1 and 3.3.2 we briefly introduce two soft criteria: IS and CSLS proposed by Smith et al. and Lample et al. for the task of BLI (however have never been used for text-image matching). In section 3.3.3, we explain our proposed Relaxed Greedy Matching (RGM) that post-processes a similarity matrix and produces a refined matching.

3.3.1 Inverted Softmax (IS)

IS estimates the confidence of a prediction $i\rightarrow t$ not by similarity score $s(i,t)$, but the score reweighted by t's similarity with other queries:

$$s'(i,t)=\frac{e^{\beta s(i,t)}}{\sum_{\bar{i}\in I\backslash\{i\}}e^{\beta s(\bar{i},t)}} \quad (5)$$

where β is a temperature. Intuitively, it scales down the similarity if t is also very close to other queries.

3.3.2 Cross-modal Local Scaling (CSLS)

CSLS aims to decrease a query vector's similarity to item vectors lying in *dense* areas while increase similarity to *isolated*[2] item vectors. Specifically, we update the similarity scores with the formulas:

$$\begin{aligned} s'(i,t)&=2s(i,t)-\frac{1}{k}\sum_{i_t\in K_1}s(i_t,t);\\ s'(i,t)&=2s(i,t)-\frac{1}{k}\sum_{t_i\in K_2}s(i,t_i) \end{aligned} \quad (6)$$

where $K_1=\mathrm{kNN}(t,I,k)$ and $K_2=\mathrm{kNN}(i,T,k)$; first and second line are for text→image and image→text inference respectively.

[2] *Dense* and *isolated* are in terms of query.

3.3.3 Relaxed Greedy Matching (RGM)

Kollias et al. proposed a heuristic algorithm called Greedy Matching (GM) for producing a one-to-one mapping from a similarity matrix. It is a simple iterative approach: 1) sort elements in a similarity matrix S; 2) the highest score $s(i,j)$ is located, the pairing (i,j) is recorded and scores involving either index i or j are deleted; 3) repeat step 2) until one of the two sets gets all its points paired.

Notice that their approach only produces one single matching instead of a k-best matching set which can be used to calculate R@k scores. We extend it to a generalized form by allowing one index to be recorded for k times. And instead of constraining each index to be matched for k times only, we associate k with a relaxation factor λ. So that a total number of λk times of records are allowed for producing a k-best matching[3]. We thus call our algorithm Relaxed Greedy Matching (RGM). Essentially, RGM is compulsively limiting the size of hubs. Notice that when $\lambda \to \infty$, RGM is equivalent to NNS as no constraint is imposed anymore. We will show in experiments that GM actually harms inference performance in most cases. But RGM, instead, can improve it by allowing the existence of "small" hubs.

RGM runs in $O(n^2 \log n)$. While RGM can do real-time inference for a test set of size $5,000$ or $25,000$, exact matching algorithms like Hungarian (Kuhn, 1955) and Murty's (Murty, 1968) which run in $O(n^3)$ couldn't. In fact, they are hundreds and thousands of times slower when applied on problems of this scale: when $n = 5,000$, $\frac{n}{\log n} \approx 587$; when $n = 25{,}000$, $\frac{n}{\log n} \approx 2{,}469$.

3.4 Measuring Hubness

As our methods stress the idea of mitigating the hubness problem in VSE, we desire certain quantitative tools to actually measure the degree of hubness (before and after applying our methods).

We use similar tools as Radovanović et al.; Zhang et al.. Suppose there are two sets of points T, I. For some $t \in T$, N_k describes the frequency of sample t being a top-k neighbour over all points in I. Let N_k be the distribution of $N_k(\cdot)$. Then the skewness of N_k characterizes the existence of "popular" neighbors in I for points in T (if the hubness problem is severe, N_k skews to the right). Formally, skewness of N_k can be formulated as

$$\text{skew}(N_k) = \frac{\sum_{i=1}^{n} (N_k(i) - \text{E}[N_k])^3}{n \cdot \text{Var}[N_k]^{\frac{3}{2}}}. \quad (7)$$

[3] A listing of pseudocode of RGM can be found in appendices.

In experiments, we include this indicator for all models trained and investigate its relation to other metrics. We also plot the change of distribution N_k to intuitively show how visual semantic hubs are affected by our proposed methods.

4 Experiments

We list our experimental setups in section 4.1. Then we compare and analyze training objectives in section 4.2 and inference objectives in section 4.3.

4.1 Experimental Setups

Dataset. We use MS-COCO (Lin et al., 2014) and Flickr30k (Young et al., 2014) as our experimental datasets. For MS-COCO, there have been several different splitting protocols being used in the community. We use the same split as (Karpathy and Fei-Fei, 2015): 113,287 images for training, 5,000 for validation and 5,000 for testing[5]. During testing, scores are computed as the average of 5 folds of 1k images. As many of the previous works report test results on a 1k test set (a subset of the 5k one), we would experiment with both protocols. We refer to the 1k test set as $c1$ and the 5k test set as $c2$. Flickr30k has 30,000 images for training; 1,000 for validation; 1,000 for testing.

Evaluation metrics. We use R@Ks (recall at K), Med r, Mean r, rsum and hs-sum to evaluate the results. R@K: the ratio of "# of queries that the ground-truth item is ranked in top K" to "total # of queries" (we use $K \in \{1,5,10\}$); Med r: the median of the ground-truth ranking; Mean r: the mean of the ground-truth ranking; rsum: the sum of R@$\{1,5,10\}$ for both text→image and image→text; hs-sum: the sum of skew(N_k) where $k = \{1,5,10\}$ for both text→image and image→text. R@Ks and rsum are the higher the better while Med r, Mean r and hs-sum are the lower the better. We compute all metrics for both text→image and image→text retrieval. During training, we follow the convention of taking the model with the maximum rsum on validation set as the best model for testing.

Model, training and inference details. We use 300-d word embeddings and 1024 internal states for GRU text encoder (all randomly initialized with Xavier init. (Glorot and Bengio, 2010)); all image encodings are obtained from image encoders pre-trained on ImageNet (for fair comparison, we don't finetune any image encoders); $d = 1024$ for both text and image embeddings; margin $\alpha = 0.2$ for all loss functions.

[5] Note that 1 image in MS-COCO and Flickr30k has 5 captions, so 5 text-image pairs are used for every image.

Table 1: Quantitative results on Flickr30k (Young et al., 2014).

#	architecture	loss	image→text					text→image						
			R@1	R@5	R@10	Med r	Mean r	R@1	R@5	R@10	Med r	Mean r	rsum	hs-sum
1.1	GRU+VGG19	SUM	30.0	59.6	67.7	4.0	34.7	22.8	49.4	61.4	6.0	47.5	291.0	10.77
1.2		MAX	30.1	56.3	67.9	4.0	30.5	21.3	47.1	58.7	6.0	40.2	281.4	10.83
1.3		HAL	**32.3**	**61.0**	**71.7**	**3.0**	**29.2**	**24.8**	**50.6**	**62.8**	**5.0**	**38.9**	**303.2**	**9.03**
1.4	Order (VGG19, ours[4]) (Vendrov et al., 2016)	SUM	29.3	56.1	68.0	4.0	**25.6**	22.7	49.5	**62.0**	6.0	**34.7**	287.7	**13.26**
1.5		MAX	23.9	51.4	62.3	5.0	33.6	19.3	45.6	57.5	7.0	37.1	259.9	22.08
1.6		HAL	**30.2**	**58.6**	**70.4**	**3.0**	27.0	**23.1**	**50.8**	**62.0**	**5.0**	36.7	**295.1**	17.98

Table 2: Quantitative results on MS-COCO (Lin et al., 2014). First three blocks (line 2.1-2.12) are using protocol $c2$ (5k test set); last block (line 2.13-2.23) is using $c1$ (1k test set) in convenience of comparing with results reported in previous works.

#	architecture	loss	image→text					text→image						
			R@1	R@5	R@10	Med r	Mean r	R@1	R@5	R@10	Med r	Mean r	rsum	hs-sum
2.1	GRU+VGG19	SUM	46.9	79.7	89.5	2.0	5.9	37.0	73.1	85.3	**2.0**	11.1	411.5	15.73
2.2		MAX	51.8	82.1	90.5	**1.0**	5.1	39.0	73.9	84.7	**2.0**	12.0	421.9	14.54
2.3		HAL	51.3	**82.2**	**91.0**	**1.0**	**4.8**	38.9	**74.7**	**86.4**	**2.0**	**8.1**	**424.5**	**13.64**
2.4		MAX+HAL	**52.0**	81.8	90.5	**1.0**	5.3	**39.1**	73.4	84.6	**2.0**	9.8	421.5	13.70
2.5	GRU+IRv2	SUM	50.9	82.7	92.2	1.4	4.1	39.5	75.8	87.2	**2.0**	9.4	428.3	16.04
2.6		MAX	57.0	86.2	93.8	**1.0**	3.5	43.3	77.9	87.9	**2.0**	8.6	446.0	13.87
2.7		HAL	54.5	85.1	93.1	**1.0**	3.8	42.4	77.2	**88.2**	**2.0**	**7.4**	440.5	14.89
2.8		MAX+HAL	**58.6**	**87.2**	**94.2**	**1.0**	**3.4**	**44.3**	**78.3**	**88.2**	**2.0**	7.5	**450.9**	**13.86**
2.9	GRU+ResNet152	SUM	53.2	85.0	93.0	**1.0**	3.9	41.9	77.2	88.0	**2.0**	8.7	438.3	15.15
2.10		MAX	58.7	88.2	94.8	**1.0**	3.2	45.0	78.9	88.6	**2.0**	8.6	454.2	13.36
2.11		HAL	58.1	87.6	94.5	**1.0**	3.3	44.1	**79.1**	**89.0**	**2.0**	**6.9**	452.4	12.57
2.12		MAX+HAL	**61.9**	**88.8**	**95.2**	**1.0**	**3.0**	**46.4**	79.0	88.9	**2.0**	7.6	**460.2**	**12.42**
2.13	(Kiros et al., 2015) (ours)		49.9	79.4	90.1	2.0	5.2	37.3	74.3	85.9	**2.0**	10.8	416.8	-
2.14	(Vendrov et al., 2016)		46.7	-	88.9	2.0	5.7	37.9	-	85.9	**2.0**	8.1	-	-
2.15	(Huang et al., 2017)		53.2	83.1	91.5	**1.0**	-	40.7	75.8	87.4	**2.0**	-	431.8	-
2.16	(Liu et al., 2017)		56.4	85.3	91.5	-	-	43.9	78.1	88.6	-	-	443.8	-
2.17	(You et al., 2018)		56.3	84.4	92.2	**1.0**	-	45.7	**81.2**	90.6	**2.0**	-	450.4	-
2.18	(Wehrmann, 2018) (d=1024)		57.8	87.9	95.6	**1.0**	3.3	44.2	80.4	**90.7**	**2.0**	5.4	456.6	-
2.19	(Faghri et al., 2018)		58.3	86.1	93.3	**1.0**	-	43.6	77.6	87.8	**2.0**	-	446.7	-
2.20	(Faghri et al., 2018) (ours)		60.5	89.6	94.9	**1.0**	3.1	46.1	79.5	88.7	**2.0**	8.5	459.3	-
2.21	GRU+ResNet152, HAL		59.6	**90.4**	96.3	**1.0**	3.0	47.0	80.9	**90.7**	**2.0**	6.4	464.9	-
2.22	GRU+ResNet152, MAX+HAL		62.5	89.9	96.0	**1.0**	3.0	47.4	81.0	89.6	**2.0**	**6.2**	466.4	-
2.23	GRU+ResNet152, MAX+HAL (k=1)		**64.2**	90.3	**97.0**	**1.0**	**2.6**	**48.4**	80.6	89.7	**2.0**	7.2	**470.2**	-

During training, we start with a learning rate of 0.001 and decay it by 10 times after every 10 epochs. Except that for all that use MAX, we follow the original configuration proposed by Faghri et al. and start with a learning rate of 0.0002, decaying it by 10 every 15 epochs. We train all models for 30 epochs with a batch size of 128. All models are optimized using an Adam optimizer (Kingma and Ba, 2015).

For inference during testing, we use $\beta = 30$ for IS; $k = 10$ for CSLS across all models. For RGM we use the λs that maximize R@K scores on validation set.

4.2 Comparison of Training Objectives

Comparing HAL, SUM and MAX. Table 1 and 2 present our quantitative results on Flickr30k and MS-COCO respectively[6]. On Flickr30k, we experiment two models and HAL achieves significantly better performance than MAX and SUM on both. On MS-COCO $c2$, HAL beats SUM but is slightly worse than MAX. Though MAX is very competitive on MS-COCO, it fails badly on Flickr30k. This serves as an evidence of MAX easily overfitting to small datasets. Faghri et al. showed that data augmentation techniques like random crop applied on input images can improve MAX's performance over small datasets.

But notice that HAL can actually be combined with MAX by first modifying the scoring matrix then choosing only the hardest sample. The hardest sample would still self-adjust its scale according to the density of its neighborhood. Though there would be no difference of scales within a mini-batch, HAL enforces an order to scales of hardest samples among different batches. Relatively *easier* hardest samples result in smaller gradients than those *harder* hardest samples. The combined loss MAX+HAL achieves better performance than both MAX and HAL. However, this combination might not always work, especially on noisy or small datasets where the base loss MAX is flawed by its nature. On the contrary, HAL is likely to maintain its good performance regardless of the data distribution and should be the loss of choice in most settings.

[6]To make the comparison fair, all results in this section are using naive nearest neighbor search (NNS) for inference. We discuss better methods for inference in section 3.3.

The impact of k in HAL. HAL has one hyperparameter k, which characterizes the scope of neighborhood being considered for local statistics. HAL achieves comparable results regardless of the choice of k as suggested in Figure 1. We experiment GRU+VGG19 on Flickr30k with different loss functions. And we plot rsum against k ranging from 1 to 10: HAL is noticeably better than the two baselines all the time. We choose $k = 3$ (which is the best among all) for all other experiments if not explicitly mentioned.

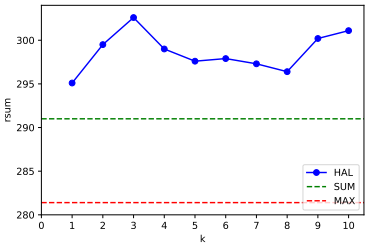

Figure 1: Plotting two baselines and HAL's rsum against k ranging from 1 to 10. All loss functions are using GRU+VGG19 as the base model and are trained & tested on Flickr30k.

HAL vs. State-of-the-art. Table 2 line 2.13-2.23 list quantitative results of both our proposed methods (2.21-2.23) and numbers reported in previous works (2.13-2.20). Though we only use routine encoder architectures (GRU and ResNet152), with HAL/MAX+HAL, our model reaches much better rsum than the ones reported before. On this specific configuration, as the dataset is clean and image embeddings (obtained from ResNet152) are of high quality, we are safe to provide more precise supervision to the model - by setting HAL's hyperparameter k to 1 (only consider the top-1 neighbor when deciding weight of a negative sample), the model on line 2.23 reaches to a rsum of 470.2.

Hub's relation to performance. HAL exploits the information from hubs to reweight negative samples and it achieves outstanding empirical performance. But metrics regularly used for text-image matching do not tell what exactly happens to hubs. In Table 1 and 2, we list a new metric hs-sum, which characterizes the existence of hubs as described in both the method and evaluation metric section. Higher hs-sum means more severe hubness problem in embeddings. The tables show that except line 1.4-1.6, all embeddings trained with HAL/MAX+HAL have lower hs-sum than the ones trained with SUM/MAX. And there is also strong inverse correlation between rsum and hs-sum, suggesting that better embeddings do have less hubness problem. We plot an embedding's N_1 distribution (text→image) in Figure 2 as an example. It is quite clear that HAL successfully regularizes the "outlier"s (large hubs) and the tail is pulled back to the left comparing to SUM and MAX.

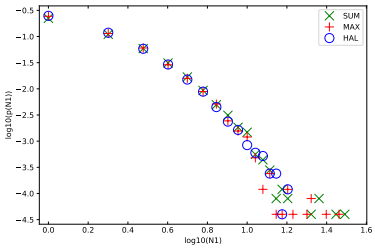

Figure 2: Comparing the N_1 distributions (text→image) of embeddings obtained from model line 2.1-2.3 (N_k defined in section 3.4). We take $\log_{10}$ of both N_1 and $p(N_1)$ for better visualization effect. All are using test set MS-COCO $c2$.

4.3 Comparison of Inference Objectives

Table 3: Comparing inference methods on models from Table 1 line 1.3 and Table 2 line 2.1, 2.12. All are using MS-COCO $c2$.

#	model & dataset	inference	rsum
4.1	GRU+VGG19 HAL Flickr30k (line 1.3)	NNS	303.2
4.2		GM	298.8
4.3		RGM	307.1
4.4		IS	308.2
4.5		CSLS	307.3
4.6		IS+RGM	308.5
4.7		CSLS+RGM	**309.6**
4.8	GRU+VGG19 SUM MS-COCO (line 2.1)	NNS	411.5
4.9		GM	411.2
4.10		RGM	415.9
4.11		IS	422.2
4.12		CSLS	427.6
4.13		IS+RGM	424.2
4.14		CSLS+RGM	**429.1**
4.15	GRU+ResNet152 MAX+HAL MS-COCO (line 2.12)	NNS	460.2
4.16		GM	455.6
4.17		RGM	462.0
4.18		IS	471.2
4.19		CSLS	472.0
4.20		IS+RGM	**472.2**
4.21		CSLS+RGM	472.0

Comparing and combining RGM, IS and CSLS. We first quantitatively compare NNS, GM (Kollias et al., 2012), RGM, IS(Smith et al., 2017), CSLS (Lample et al., 2018) and also RGM+IS/CSLS in Table 3[7]. We pick three embeddings of MS-COCO $c2$ (Table 1 line 1.3 and Table 2 line 2.1, 2.12) trained in section 4.2 for this comparison. When used alone, RGM generally performs worse than IS and CSLS. On model line 1.3, RGM and CSLS are comparable (~307) while IS is slightly better (~308). On model line 2.1 and 2.12, IS and CSLS beat RGM by a large margin. However, RGM can be integrated with IS/CSLS

[7]We only report rsum here. But full table with all R@Ks can be found in appendices Table 4.

as a post-processing procedure and refines their results. RGM advances the best results from IS/CSLS further (+∼2) across all three models. The combined methods improve rsum by 6.4,17.6,12.0 comparing to NNS. We also include GM in comparison. As suggested in the table, eliminating hubs actually harms inference performance (−4.3, −0.3, −4.6 comparing to NNS). We will further discuss how size of hubs affects empirical results in section 4.3.

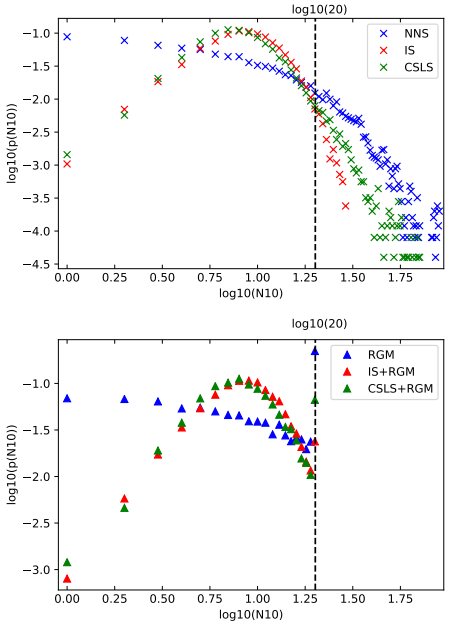

Figure 3: Comparing different inference methods' N_{10} distributions (text→image) (N_k defined in section 3.4). Using model in Table 2 line 2.1. All embeddings produced on MS-COCO $c2$ test set. Notice that on the second figure, with RGM ($\lambda = 2$), all hubs are rigidly regularized to to be as small as $\log_{10}20$.

To demonstrate how different inference methods have affected the existence of hubs more intuitively, we plot the N_{10} distributions for all methods listed in Table 3 as an example. The results are in Figure 3. Again, large hubs (characterized by $k = 10$) exist in the form of "long tail" trailing to the right of the main body. The more hubness there is, the more the graph skews to the right. The first figure shows that IS and CSLS successfully pull the long tail backward. The second figure shows that our proposed RGM "clips" the long tails compulsively. Specifically, in this figure, the size of hubs is limited to be $\leq \log_{10}20$. Through this plotting, we can also qualitatively tell why RGM barely improves a high-quality model like line 4.20, 4.21: when IS and CSLS have already made tail retract enough, there is little room for RGM to function.

Small hubs improve performance. The intuition for introducing a relaxed GM is, empirically speaking, tolerance of small hubs might in reverse raise R@Ks. This might be a bit counter-intuitive as we thought all hubs are harmful. But tolerating small hubs actually makes the algorithm more robust, especially when an item is quite certain a query should be in its neighborhood but uncertain its exact ranking. We notice a clear trend in Figure 4 that the R@10s first increase as relaxation factor λ grows, then decrease after a certain point and finally converge with the unconstrained versions', which is equivalent to not having RGM at all. RGM estimates a suitable λ on validation set and takes advantage of the performance boost from having small hubs.

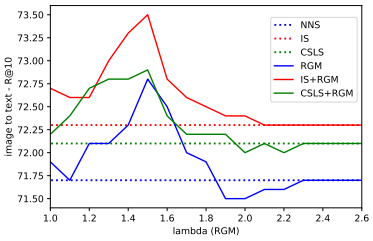

Figure 4: Comparing multiple inference methods combined with RGM. R@10 (image→text) against k ranging from 1 to 10 is plotted. Using model in Table 1 line 1.3.

5 Conclusion

We introduce novel tools for mitigating visual semantic hubs in both training and inference phase of text-image matching. For training, we propose a self-adaptive loss (HAL) that leverages information of hubs, giving considerations to both robustness and hard negative mining. For inference, we propose the Relaxed Greedy Matching algorithm (RGM) which can be combined with existing cross-modal mapping criteria that punish hubs. Both methods are comparable or better than the state-of-the-art across different datasets and model architectures. Despite empirical results, we also offer insights on how the existence of hubs has affected models' performance. Though our methods have only experimented on the task of text-image matching, they can be presumably applied on all cross-modal matching tasks.

References

Richard Bellman. 1961. Adaptive control processes: a guided tour princeton university press. *Princeton, New Jersey, USA*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of

gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 248–255. IEEE.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. *ICLR worshop*.

F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler. 2018. Vse++: Improving visual-semantic embeddings with hard negatives.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35.

A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, and T. Mikolov. 2013. Devise: A deep visual-semantic embedding model. In *NIPS*, pages 2121–2129.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. 2014. Improving image-sentence embeddings using large weakly annotated photo collections. In *European Conference on Computer Vision (ECCV)*, pages 529–545. Springer.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899.

Yan Huang, Wei Wang, and Liang Wang. 2017. Instance-aware image and sentence matching with selective multimodal lstm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2310–2318.

Yao-Hung Hubert Tsai, Liang-Kang Huang, and Ruslan Salakhutdinov. 2017. Learning robust visual-semantic embeddings. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3571–3580.

Ann Irvine and Chris Callison-Burch. 2017. A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics*, 43(2):273–310.

Herve Jegou, Cordelia Schmid, Hedi Harzallah, and Jakob Verbeek. 2008. Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):2–11.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. *EMNLP (short paper)*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *ICLR*.

R. Kiros, R. Salakhutdinov, and R. S. Zemel. 2015. Unifying visual-semantic embeddings with multimodal neural language models. *Transactions of the Association for Computational Linguistics (TACL)*.

Giorgos Kollias, Shahin Mohammadi, and Ananth Grama. 2012. Network similarity decomposition (nsd): A fast and scalable approach to network alignment. *IEEE Transactions on Knowledge and Data Engineering*, 24(12):2232–2243.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Herv Jgou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 270–280.

Rémi Lebret, Pedro O Pinheiro, and Ronan Collobert. 2015. Phrase-based image captioning. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37 (ICML)*, pages 2085–2094. JMLR. org.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and L. Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision (ECCV)*, pages 740–755. Springer.

Yu Liu, Yanming Guo, Erwin M Bakker, and Michael S Lew. 2017. Learning a recurrent residual fusion network for multimodal matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4107–4116.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In

Advances in neural information processing systems, pages 3111–3119.

Katta G Murty. 1968. Letter to the editoran algorithm for ranking all the assignments in order of increasing cost. *Operations research*, 16(3):682–687.

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(Sep):2487–2531.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307.

Dominik Schnitzer, Arthur Flexer, Markus Schedl, and Gerhard Widmer. 2012. Local and global scaling reduce hubs in space. *Journal of Machine Learning Research*, 13(Oct):2871–2902.

Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurélie Herbelot, Moin Nabi, Enver Sangineto, and Raffaella Bernardi. 2017. Foil it! find one mismatch between image and language caption. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 255–265.

Haoyue Shi, Jiayuan Mao, Tete Xiao, Yuning Jiang, and Jian Sun. 2018. Learning visually-grounded semantics from contrastive adversarial samples. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3715–3727.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *ICLR*.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Nenad Tomašev, Raluca Brehar, Dunja Mladenić, and Sergiu Nedevschi. 2011. The influence of hubness on nearest-neighbor methods in object recognition. In *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing*, pages 367–374. IEEE.

I. Vendrov, R. Kiros, S. Fidler, and R/ Urtasun. 2016. Order-embeddings of images and language. *ICLR*.

Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang. 2016. A comprehensive survey on cross-modal retrieval. *arXiv preprint arXiv:1607.06215*.

Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. 2019. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):394–407.

Barros-Rodrigo C Wehrmann, Jônatas. 2018. Bidirectional retrieval made simple. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7718–7726.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2010. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 81(1):21–35.

Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. 2017. Sampling matters in deep embedding learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*.

Quanzeng You, Zhengyou Zhang, and Jiebo Luo. 2018. End-to-end convolutional semantic embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5735–5744.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Lihi Zelnik-Manor and Pietro Perona. 2005. Self-tuning spectral clustering. In *Advances in neural information processing systems*, pages 1601–1608.

Li Zhang, Tao Xiang, and Shaogang Gong. 2017. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2021–2030.

A Appendices

The appendices include pseudocode of RGM and the full version of Table 3.

Algorithm 1 Relaxed Greedy Matching (RGM)

Input: X: similarity matrix; k: size of neighborhood considered for computing R@k; λ: relaxation coefficient

Output: a stack S that records all matchings

$a,b \leftarrow$ shape of X
$N \leftarrow a \times b$
$M \leftarrow \texttt{min}(a,b)$
$R_u \leftarrow$ a zero vector of size m
$C_u \leftarrow$ a zero vector of size n
$S \leftarrow \texttt{stack}()$ // an empty stack
$x \leftarrow \texttt{flatten}(X)$ // matrix to a list of elements
$ix \leftarrow \texttt{argsort}(x)$ // indices of x being sorted in descent order
$id \leftarrow 0$
$matched \leftarrow 0$
while $matched < M \times k$ **do**
    $i \leftarrow ix[id]$ **div** n
    $j \leftarrow ix[id]$ **mod** n
    if $R_u[i] < k$ **and** $C_u[j] < \texttt{round}(\lambda \times k)$ **then**
        $S.\texttt{push}([i,j])$
        $R_u[i] \leftarrow R_u[i]+1$
        $C_u[j] \leftarrow C_u[j]+1$
        $matched \leftarrow matched+1$
    end if
    $id \leftarrow id+1$
end while
return S

Table 4: Comparing various inference methods on models from Table 1 line 1.3 and Table 2 line 2.1, 2.12 (full table).

#	model & dataset	Inference	image→text			text→image			
			R@1	R@5	R@10	R@1	R@5	R@10	rsum
4.1	GRU+VGG19 HAL Flickr30k (line 1.3)	NNS	32.3	61.0	71.7	24.8	50.6	62.8	303.2
4.2		GM	29.3	60.6	71.9	24.2	50.5	62.3	298.8
4.3		RGM	33.3	62.2	72.8	24.9	50.8	63.1	307.1
4.4		IS	33.9	62.1	72.3	24.9	**51.6**	**63.4**	308.2
4.5		CSLS	33.9	61.4	72.1	**25.3**	51.5	63.0	307.3
4.6		IS+RGM	33.7	**62.4**	**73.2**	24.9	51.0	63.3	308.5
4.7		CSLS+RGM	**34.3**	**62.4**	72.7	**25.3**	**51.6**	63.3	**309.6**
4.8	GRU+VGG19 SUM MS-COCO (line 2.1)	NNS	46.9	79.7	89.5	37.0	73.1	85.3	411.5
4.9		GM	44.4	81.6	89.4	37.2	73.3	85.3	411.2
4.10		RGM	48.0	81.0	90.6	37.2	73.6	85.5	415.9
4.11		IS	**53.2**	82.2	90.9	38.4	73.2	84.2	422.2
4.12		CSLS	52.3	82.8	91.0	**40.4**	75.0	86.1	427.6
4.13		IS+RGM	**53.2**	82.2	91.1	38.6	73.8	85.3	424.2
4.14		CSLS+RGM	52.8	**83.0**	**91.4**	**40.4**	**75.2**	**86.3**	**429.1**
4.15	GRU+ResNet152 MAX+HAL MS-COCO (line 2.12)	NNS	61.9	88.8	95.2	46.4	79.0	88.9	460.2
4.16		GM	56.5	89.6	95.0	46.7	79.2	88.6	455.6
4.17		RGM	62.7	89.1	95.4	46.4	79.4	89.0	462.0
4.18		IS	67.0	90.9	**96.0**	47.5	80.3	89.6	471.2
4.19		CSLS	65.6	90.8	95.9	48.6	**81.2**	**89.9**	472.0
4.20		IS+RGM	**67.1**	**91.1**	**96.0**	47.8	80.6	89.6	**472.2**
4.21		CSLS+RGM	65.9	90.7	95.9	**48.7**	81.1	89.7	472.0