Maximal Matching Matters: Preventing Representation Collapse for Robust Cross-Modal Retrieval

Hani Alomari
Virginia Tech
hani@vt.edu

Anushka Sivakumar
Virginia Tech
anushkas01@vt.edu

Andrew Zhang
Virginia Tech
azhang42@vt.edu

Chris Thomas
Virginia Tech
chris@cs.vt.edu

Abstract

Cross-modal image-text retrieval is challenging because of the diverse possible associations between content from different modalities. Traditional methods learn a single-vector embedding to represent semantics of each sample, but struggle to capture nuanced and diverse relationships that can exist across modalities. Set-based approaches, which represent each sample with multiple embeddings, offer a promising alternative, as they can capture richer and more diverse relationships. In this paper, we show that, despite their promise, these set-based representations continue to face issues including sparse supervision and set collapse, which limits their effectiveness. To address these challenges, we propose **Maximal Pair Assignment Similarity** to optimize one-to-one matching between embedding sets which preserve semantic diversity within the set. We also introduce two loss functions to further enhance the representations: **Global Discriminative Loss** to enhance distinction among embeddings, and **Intra-Set Divergence Loss** to prevent collapse within each set. Our method achieves state-of-the-art performance on MS-COCO and Flickr30k without relying on external data.

1 Introduction

Cross-modal retrieval methods aim to align different modalities, such as images and text, by learning shared semantic representations (Frome et al., 2013). The challenge lies in bridging the semantic gap between modalities: each modality contributes unique information, and aligning them requires identifying which features to preserve or discard. Moreover, the relationship between modalities is often one-to-many (*e.g.*, a single image can be described by multiple captions, each emphasizing different aspects). Handling this complexity requires learning robust embeddings that capture the diverse semantic relationships that exist across modalities (Song and Soleymani, 2019; Chun et al., 2021).

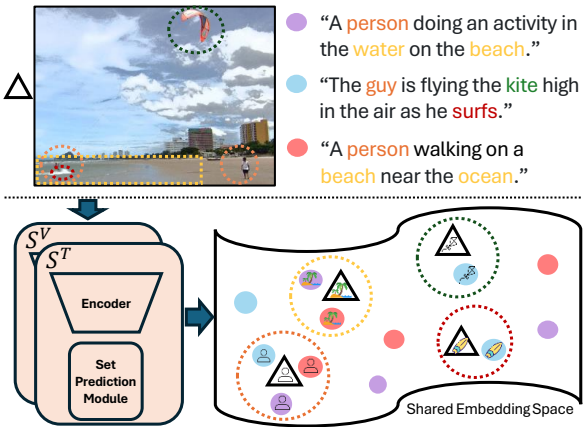

Figure 1: We illustrate how the same image can be paired with multiple captions. Color-coded regions link visual elements to corresponding textual descriptions. Our method produces multiple embeddings for each sample (image or caption) in a shared space where similar visual and textual elements cluster (indicated by colored groups and dashed boundaries). Icons on embeddings illustrate the model's capacity to map multiple image regions to relevant text descriptions.

Most existing approaches rely on learning a single shared embedding space for images and text (Faghri et al., 2018; Fu et al., 2023; Pham et al., 2024) performing well on benchmarks like Flickr30k and COCO (Plummer et al., 2015; Lin et al., 2014) but struggling with abstract or nuanced cross-modal relationships. To address this, some methods use cross-attention networks that predict similarity between images and text by attending to both modalities simultaneously (Wei et al., 2020b; Diao et al., 2021; Lee et al., 2018; Miech et al., 2021), yet are computationally intensive, requiring joint processing of image-text pairs for each query, which limits their scalability. A common approach is to use separate visual and textual encoders, which allow for efficient search on precomputed embeddings (Gu et al., 2018; Messina et al., 2021). This method scales well to large datasets but often fails

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 31769–31785
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

to capture the diversity of cross-modal relationships because each input is reduced to a single embedding vector limiting the representation's ability to handle one-to-many relationships between queries and potential matches.

A few recent works have proposed architectures that aim to capture these diverse relationships using separate encoders (Song and Soleymani, 2019; Chun, 2024; Kim et al., 2023). The goal is to reduce ambiguity and improve alignment between modalities by representing multiple facets of each input with a set of embedding vectors. Figure 1 shows how these embedding sets cluster similar features from both modalities in a shared space.

While set-based embedding models offer a promising solution for capturing diverse cross-modal relationships, the loss and similarity functions to train these methods lead to issues like sparse supervision (where some embeddings are unused) and set collapse (where embeddings collapse into a single point). For example, PVSE (Song and Soleymani, 2019) uses a distance function that relies on maximum similarity within the set, leaving many embeddings undertrained. Our analysis shows that Smooth-Chamfer similarity in SetDiv (Kim et al., 2023), proposed as a solution to this issue, leads to low variance across the embeddings within the set, reducing the ability to capture diverse relationships in the data.

In this paper, we address these limitations by introducing MaxMatch, a new Maximal Pair Assignment Similarity mechanism and diversity-promoting losses. Unlike previous work, our approach optimally matches pairs within the embedding sets using a *permutation-based* similarity, ensuring that every embedding in the set contributes to the final objective. In addition, we design a global discriminative loss that encourages each embedding to diverge from a global reference vector, and an intra-set divergence constraint that pushes embeddings apart within the set. These strategies effectively prevent set collapse and improve alignment across modalities. Specifically, our contributions include:

- We introduce a new similarity measure for measuring the distance between sets of embeddings. MaxMatch uses permutation-based assignments and the Hungarian algorithm to match embeddings between sets and compute their distance. We show that MaxMatch prevents degenerate embedding sets unlike previous methods.
- We propose a novel loss which enforces a margin between residual embeddings and a global reference embedding to encourage semantic diversity. We also introduce intra-set divergence constraints to further discourage set collapse. We show that these functions promote semantic diversity and improve the model's discriminative capability.
- We conduct a comprehensive experimental analysis of our proposed technique against a number of state-of-the-art methods, including embedding space visualizations and analysis. Our results show that MaxMatch learns richer, more diverse embedding sets while outperforming similar methods.

2 Related Work

Cross-modal Retrieval: Methods fall into two main categories: independent-embedding and interactive-embedding approaches. Independent-embedding approaches use separate encoders for each modality to learn a shared embedding space (Faghri et al., 2018; Yan and Mikolajczyk, 2015; Gu et al., 2018), enabling efficient retrieval through precomputed embeddings. While researchers have explored various improvements to similarity metrics (Wei et al., 2020a; Kim et al., 2023), loss functions (Thomas and Kovashka, 2020; Chun et al., 2021), and architectures (Gu et al., 2018), these methods still struggle with complex cross-modal relationships (Song and Soleymani, 2019). Recent work like CORA (Pham et al., 2024) and 3SHNet (Ge et al., 2024) achieve improvements by incorporating external knowledge (scene graphs, segmentation), but this makes direct comparisons with methods that do not use external data more difficult.

Interactive-embedding approaches instead use cross-attention networks for direct similarity estimation (Wei et al., 2020b; Diao et al., 2021), particularly when they are built on large pretrained architectures like CLIP (Radford et al., 2021). While effective, these methods are computationally intensive as they require joint processing of each query-candidate pair at inference time. In contrast, MaxMatch focuses on improving independent embeddings to maintain efficiency while better capturing complex relationships.

Multi-embedding Representations: Recently, a few papers have emphasized the limitations of representing each sample as a single embedding vector for cross-modal retrieval (Pham et al., 2024; Faghri et al., 2018; Chen et al., 2021; Fu et al., 2023;

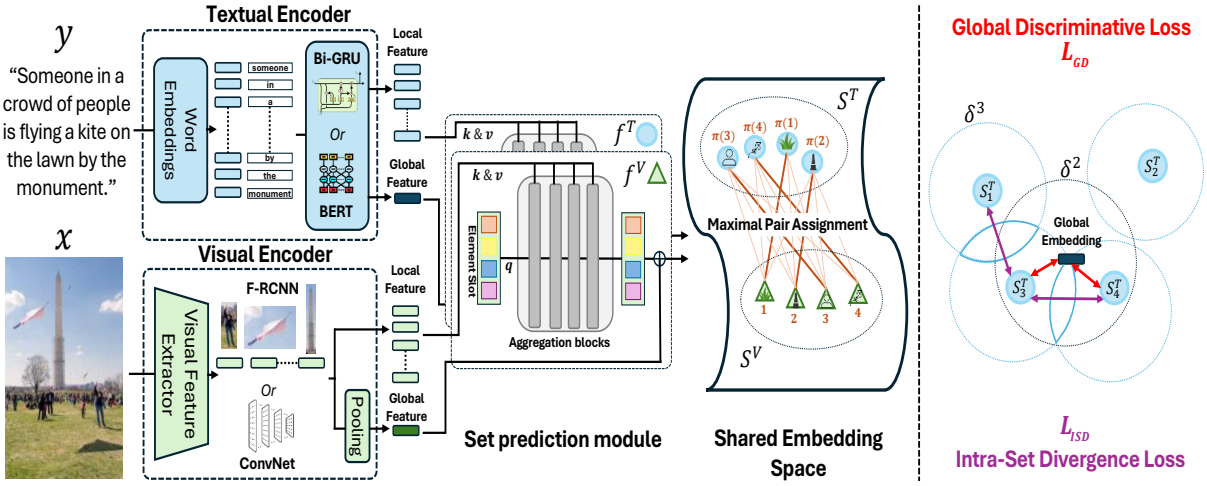

Figure 2: Left: An overview of the model architecture based on Kim et al. (2023) consisting of a visual encoder, textual encoder, and set prediction modules for f^V and f^T. Local and global features are extracted from each modality and input into a set prediction module to generate embedding sets S^V and S^T. We contribute three key components which enable the model to learn diverse embedding sets: a Maximal Pair Assignment Similarity function, Global Discriminative Loss, and Intra-Set Divergence Loss. Right: Our Global Discriminative Loss pushes embeddings within each set away from the global embedding, preventing set collapse, while the Intra-Set Divergence Loss ensures that individual embeddings within each set are distinct, promoting intra-set diversity.

Thomas and Kovashka, 2022). A single vector often fails to capture the rich and diverse semantics present in the data. One strategy to address this challenge is set-based representation learning, where multiple embeddings are generated for each input, like in SetDiv, PVSE, and PCME (Chun et al., 2021). These approaches aim to reduce ambiguity by capturing the varied semantics of each modality using a set-based embedding representation. SetDiv introduces smooth-Chamfer similarity to overcome sparse supervision issues, but it leads to a new problem of set collapse, where embeddings lose diversity. To address sparse supervision and set collapse, we introduce a Maximal Pair Assignment Similarity function that explicitly preserves diversity. Even though PVSE and SetDiv use diversity loss to encourage distinct representations by penalizing pairwise similarity but lack global context, focusing only on residual embeddings before fusion. To address this, we propose novel pushing losses to enhance intra-set diversity and better capture modality relationships, achieving improved cross-modal retrieval performance compared to prior methods.

3 Method

3.1 Feature Extraction

MaxMatch architecture consists of two encoders, shown in Figure 2, each connected to a set prediction model. For vision, a visual encoder $f^{\mathcal{V}}$ takes an image as input x and produces the image embedding vector $v = f^{\mathcal{V}}(x) \in \mathbb{R}$, and a text encoder $f^{\mathcal{T}}$ that takes in the text caption y and produces its embedding $t = f^{\mathcal{T}}(y) \in \mathbb{R}$. Each encoder has two branches that compute local features and global features from the input sample. The extracted local features are given as input to the set prediction modules, each of which fuses the local and global features to encode them into an embedding set. We follow the settings of previous works (Song and Soleymani, 2019; Kim et al., 2023) for the visual and textual feature extractors.

Visual Feature Extractor: We use one of two types of visual feature extractors following previous works like PVSE and SetDiv: (1) Flattened convolutional feature maps as local features $\psi^{\mathcal{V}}(x) \in \mathbb{R}^{N \times D}$ and their average pooled features as the global feature $\phi^{\mathcal{V}}(x) \in \mathbb{R}^D$. (2) ROI features from a pretrained object detector as local features $\psi^{\mathcal{V}}(x) \in \mathbb{R}^{N \times D}$ and their max-pooled representation as the global feature $\phi^{\mathcal{V}}(x) \in \mathbb{R}^D$.

Textual Feature Extractor: We use one of two types of text feature extractors: (1) Bi-GRU: An input caption y of L words is encoded using GloVe (Pennington et al., 2014), producing 300-dimensional word vectors as local features $\psi^{\mathcal{T}}(y) \in \mathbb{R}^{L \times 300}$. A Bi-GRU with H hidden units processes these features, with the final hidden state representing the global features $\phi^{\mathcal{T}}(y) \in \mathbb{R}^D$. (2) BERT (Devlin et al., 2019): The hidden state outputs serve as local features $\psi^{\mathcal{T}}(y) \in \mathbb{R}^{L \times D}$, and their max-pooled representation forms the global

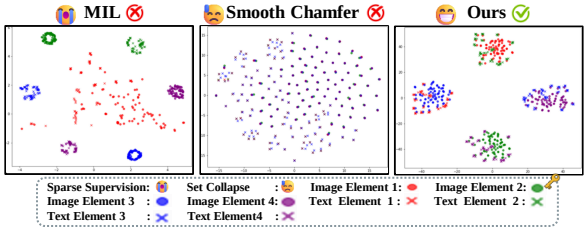

Figure 3: t-SNE visualization of learned embedding spaces, where colors indicate different elements within each set, and markers differentiate modalities (dots for image embeddings, crosses for text embeddings). MIL produces scattered embeddings, Smooth Chamfer collapses them, while MaxMatch preserves semantic distinctiveness and strong alignment.

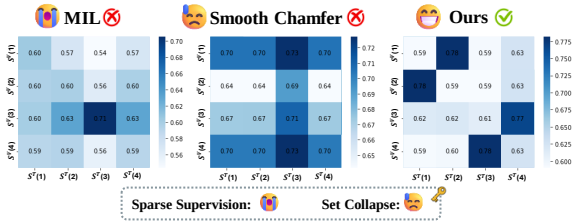

Figure 4: Heatmaps showing average pairwise similarities between image-text embeddings across all test samples for models trained with different similarity functions: MIL shows sparse supervision, leading to isolated high similarities (only one embedding in the set is used), Smooth Chamfer causes set collapse, resulting in uniformly high similarities, and MaxMatch maintains distinct alignment, and fully utilizing embeddings set.

feature $\phi^{\mathcal{T}}(y) \in \mathbb{R}^D$.

3.2 Embedding Set Prediction Module

We next describe the set prediction module adapted from Kim et al. (2023) that we utilize to generate set embeddings. As shown in Figure 2, slots in the module iteratively compete to attend to input features. Each of the L aggregation blocks in the module contains a cross-attention layer followed by a feed-forward network. K learnable queries ("slots") cross-attend to keys and values from $\psi^{\mathcal{V}}(x)$, denoted $E^l = AggBlock(E^{l-1}) = MLP(\bar{E}^l) + \bar{E}^l \in \mathbb{R}^{K \times D}$, where $\bar{E}^l$ represents the slot states after cross-attention. The final embedding set S is computed as $S = LN(E^L) + [LN(\phi(y))...^{\times K}]$, where LN is a normalization layer, and $[LN(\phi(y))...^{\times K}] \in \mathbb{R}^{K \times D}$ denotes K repetitions of global features $\phi(y)$. Conceptually, each slot represents an offset from the global feature $\phi^{\mathcal{V}}(x)$. However, embeddings in standard SetDiv have a tendency to cluster closely around the global feature (i.e. set collapse). To encourage diversity, MaxMatch introduces a Maximal Pair Assignment Similarity, Global Discriminative Loss, and Intra-Set Divergence Loss.

3.3 Maximal Pair Assignment Similarity

MaxMatch aims to overcome issues related to set-based embeddings and set-based similarity measures, particularly set collapse and sparse supervision. To better understand these limitations, we first examine existing set-based similarity functions such as Multiple Instance Learning (MIL) similarity, introduced in PVSE, and smooth-Chamfer similarity, introduced by SetDiv.

Given two sets of embeddings $S1 = \{x_1, x_2, ..., x_K\}$ and $S2 = \{y_1, y_2, ..., y_K\}$, and a similarity function $c(x, y)$ between vectors $x \in S_1$ and $y \in S_2$, the MIL loss objective is defined as $S_{MIL}(S_1, S_2) = \max_{x \in S_1, y \in S_2} c(x, y)$; while MIL provides a simple mechanism for similarity calculation between embedding sets, it suffers from sparse supervision because the max function passes gradients to a single element in each set. By focusing only on maximum similarity during training, it underutilizes the remaining elements in the sets and leads to embedding sets where only one element contributes during retrieval. Kim et al. (2023) proposed smooth-Chamfer similarity to address the sparse gradient issue. This averages the similarity scores between all pairs from S_1 and S_2, which allows all embeddings to receive gradients:

$$S_{SC}(S_1, S_2) = \frac{1}{2\alpha|S_1|} \sum_{x \in S_1} \log \Big(\sum_{y \in S_2} e^{\alpha c(x,y)} \Big) + \frac{1}{2\alpha|S_2|} \sum_{x \in S_2} \log \Big(\sum_{y \in S_1} e^{\alpha c(x,y)} \Big) \quad (1)$$

However, despite providing gradient signals for the entire set, we show in Figures 3 and 4 that Smooth-Chamfer similarity leads to set collapse (i.e. all embeddings become the same), while MIL leads to unused embeddings. Even with a high α value, intended to emphasize stronger similarities, the log function smooths out contributions and leads to uniformly high similarity scores across all elements (i.e. collapse). See our Appendix for additional analysis. In sum, both existing techniques for training set-based representations lead to degenerate conditions. We propose Maximal Pair Assignment Similarity, which learns optimal matching between elements of embedding sets. By no longer pulling all embeddings towards the mean, sets are no longer encouraged to collapse, yet all embeddings remain active in the matching process.

MaxMatch leverages permutation-based assignments, which we efficiently implement using block processing and masking techniques. It finds the optimal pairing of embeddings from S_1 and S_2, where the overall matching score of the sets is maximized. We describe our approach in detail below.

Cosine Similarity Calculation: We begin by computing the cosine similarity between the L2-normalized set of image embeddings $\mathbf{V}_i = \{v_{i1}, v_{i2}, \cdots, v_{iK}\} \in \mathbb{R}^{K \times D}$ and the set of text embeddings $\mathbf{T}_j = \{t_{j1}, t_{j2}, \cdots, t_{jK}\} \in \mathbb{R}^{K \times D}$, where K represents the number of embeddings per sample. The similarity score between each pair of sets is $Sim(V_i, T_j) = S(V_i, T_j)$, where $S(V_i, T_j) \in \mathbb{R}^{K \times K}$ is the similarity matrix for the sets: $S_{mn}(V_i, T_j) = \frac{v_{im} \cdot t_{jn}}{||v_{im}|| ||t_{jn}||}$, where $S_{mn}(V_i, T_j)$ denotes the element in the m-th row and n-th column of $S(V_i, T_j)$.

Hungarian Algorithm: To obtain the optimal matching, we apply the Hungarian algorithm to blocks of the similarity matrix $S(V_i, T_j)$ between embedding sets V_i and T_j which yields the permutation π that maximizes the total similarity by selecting the most semantically meaningful pairs:

$$\pi^* = \arg\max_{\pi}(\text{tr}(S(V_i, \pi(T_j)))) \tag{2}$$

where π is a permutation of indices representing the optimal matching between the image and text embeddings and tr denotes trace. After that, we apply the optimal assignment on S_{ij} that maximizes the total similarity, where the outputs are a binary mask M_{ij} indicating the selected matches:

$$M_{ij} = \begin{cases} 1, & \text{if } (V_i, \pi^*(T_j)) \text{ is an optimal match} \\ 0, & \text{otherwise} \end{cases}$$

The mask M_{ij} is applied to the similarity matrix to extract only the optimal matches:

$$MaxSim(S_i^{\mathcal{V}}, S_j^{\mathcal{T}}) = \sum_{k=1}^{K} M_{ij} \odot S_{ij} \tag{3}$$

where M_{ij} is a binary mask matrix that indicates whether a particular pair $(V_i, \pi^*(T_j))$ is part of the optimal one-to-one matching between the sets $S_i^{\mathcal{V}}$ and $S_j^{\mathcal{T}}$. Finally, we scale and sum them to calculate the final similarity score S_H for each image-text pair:

$$S_H(S_i^{\mathcal{V}}, S_j^{\mathcal{T}}) = \frac{1}{K} \sum_{i=0}^{K} \sum_{j=0}^{K} \left(\left(\exp\left(MaxSim(S_i^{\mathcal{V}}, S_j^{\mathcal{T}}) \right) \right) - 1 \right) \tag{4}$$

where the exponential function exp is used to amplify the influence of pairs with higher similarity. This approach aligns with the intuition that the closest pairs, those that best capture the semantic relationship between the image and text, should have more weight in the final similarity score. For example, if a particular image-text pair is closely related, their similarity score after exponential scaling will dominate, ensuring that these critical pairs contribute more significantly to the overall score.

3.4 Training and Inference

We train MaxMatch to minimize standard objectives presented in previous works (Kim et al., 2023; Pham et al., 2024; Song and Soleymani, 2019): triplet loss (TRI), diversity regularizer (Div), Maximum Mean Discrepancy (MMD) regularizer, and Contrastive loss (CON). In addition, we introduce a Global Discriminative Loss (GD) and Intra-Set Divergence Loss (ISD) to prevent set collapse and encourage the model to learn diverse embeddings.

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{TRI} + \lambda_{GD}\mathcal{L}_{GD} + \lambda_{ISD}\mathcal{L}_{ISD} \\ + \lambda_{MMD}\mathcal{L}_{MMD} + \lambda_{Div}\mathcal{L}_{Div} + \lambda_{CON}\mathcal{L}_{CON} \end{aligned} \tag{5}$$

Triplet loss with hardest negatives: Following prior work (Kim et al., 2023; Song and Soleymani, 2019), we incorporate the a hinge-based Triplet Loss with Hardest Negatives mining. Given a batch of embeddings $B = \{(S_i^{\mathcal{V}}, S_i^{\mathcal{T}})\}_{i=1}^{N}$, the triplet loss is formulated as follows:

$$\begin{aligned} \mathcal{L}_{TRI} = \sum_{i=1}^{N} \max_j \left[\delta_1 + S_H(S_i^{\mathcal{V}}, S_j^{\mathcal{T}}) \right] - S_H(S_i^{\mathcal{V}}, S_i^{\mathcal{T}}) \Big]_+ \\ + \sum_{i=1}^{N} \max_j \left[\delta_1 + S_H(S_i^{\mathcal{T}}, S_j^{\mathcal{V}}) \right] - S_H(S_i^{\mathcal{T}}, S_i^{\mathcal{V}}) \Big]_+ \end{aligned} \tag{6}$$

where $\delta_1 > 0$ denotes the margin. For each positive pair $(S_i^{\mathcal{V}}, S_i^{\mathcal{T}})$, the model identifies the hardest negative sample $S_j^{\mathcal{T}}$ closest to $S_i^{\mathcal{V}}$, and similarly the hardest negative $S_j^{\mathcal{V}}$ closest to $S_i^{\mathcal{T}}$.

Regularization and Stabilization Losses: To ensure robust and stable training, we employ a set of losses following prior work (Kim et al., 2023). The Maximum Mean Discrepancy (MMD) (Gretton et al., 2006) minimizes the MMD between embedding sets of images and text, helping to prevent early divergence between modalities. The diversity regularizer (Song and Soleymani, 2019) penalizes similar element slots. Consistent with previous findings (Chen et al., 2021), we observed that using the hardest triplet loss can lead to instability

during early training. Therefore, following (Pham et al., 2024), we also leverage contrastive loss to align all matching image and text representations.

Global Discriminative Loss: pushes each embedding away from the corresponding global reference embedding (within a margin and with a scaling factor) within each set to ensure the embedding set captures meaningful semantics beyond what is already captured in the global representation. The key idea is to ensure that embeddings within each set are gradually pushed away from their respective global, encouraging distinct representations within the set. The loss is defined as

$$\mathcal{L}_{GD} = \frac{1}{2N}\sum_{i=1}^{N}\Big(\exp\Big(s\cdot\Big(\cos(v_i,\phi^{\mathcal{V}}(y_i))-\delta_2\Big)\Big) + \exp\Big(s\cdot\Big(\cos(t_i,\phi^{\mathcal{T}}(y_i))-\delta_2\Big)\Big)\Big) \tag{7}$$

where N is the batch size, δ_2 is the separation margin, and s is the scaling factor that controls gradient smoothness. Here, $\cos(v_i, \phi^{\mathcal{V}}(y))$ and $\cos(t_i, \phi^{T}(y))$ represent the similarities between each visual embedding v_i or textual embedding t_i and their respective global references, $\phi^{\mathcal{V}}(y)$ and $\phi^{T}(y)$. The exponential function applied to the scaled similarity encourages embeddings to spread apart in the feature space if they exceed the margin δ_2, which promotes a well-distributed feature space across both modalities.

Intra-Set Divergence Loss: is designed to encourage diversity among embeddings *within* each set by penalizing high similarity between embeddings. This loss reduces redundancy within the set to encourage sets to learn unique representations which capture different aspects of the data. The loss is formulated as:

$$\mathcal{L}_{ISD} = \frac{2}{M(M-1)}\sum_{j=1}^{M-1}\sum_{k=j+1}^{M}\big(\exp\left(s\cdot\left(\cos(v_{i,j},v_{i,k})-\delta_3\right)\right) + \exp\left(s\cdot\left(\cos(t_{i,j},t_{i,k})-\delta_3\right)\right)\big) \tag{8}$$

where M is the number of embeddings per set, δ_3 is the similarity margin, and s is a scaling factor that smooths the gradients. The cosine similarity terms $\cos(v_{i,j}, v_{i,k})$ and $\cos(t_{i,j}, t_{i,k})$ measure the alignment between each pair of non-identical embeddings within the visual and textual sets, respectively. By applying the exponential function with scaling, the loss pushes embeddings further apart in the feature space if they exceed the margin δ_3, encouraging a diverse distribution. The normalization factor $\frac{2}{M(M-1)}$ averages the loss across all non-matching pairs.

Inference: During inference, we select the $top\text{-}k$ most similar embeddings from the predicted sets without applying the one-to-one matching process used during training. This approach reduces the computational complexity and significantly speeds up the inference phase, making the model efficient while still leveraging the diverse embeddings.

Method	CA	Flickr30k Test Images						
		Image-to-Text			Text-to-Image			
		R@1	R@5	R@10	R@1	R@5	R@10	RSUM
Resnet152 + Bi-GRU								
VSE++ (Faghri et al., 2018)	✗	52.9	80.5	87.2	39.6	70.1	79.5	409.8
PVSE (Song and Soleymani, 2019)	✗	59.1	84.5	91.0	43.4	73.1	81.5	432.6
PCME (Chun et al., 2021)	✗	58.5	81.4	89.3	44.3	72.7	81.9	428.1
Set Div (Kim et al., 2023)	✗	61.8	85.5	91.1	46.1	74.8	83.3	442.6
MaxMatch	✗	68.6	89.6	94.6	51.5	78.9	86.8	**469.9**
Faster R-CNN + Bi-GRU								
SCAN† (Lee et al., 2018)	✓	67.4	90.3	95.8	48.6	77.7	85.2	465
VSRN† (Li et al., 2019)	✗	71.3	90.6	96.0	54.7	81.8	88.2	482.6
CAAN (Zhang et al., 2020)	✓	70.1	91.6	97.2	52.8	79	87.9	478.6
SGRAF† (Diao et al., 2021)	✓	77.8	94.1	97.4	58.5	83.0	88.8	499.6
VSE∞ (Chen et al., 2021)	✗	76.5	94.2	97.7	56.4	83.4	89.9	498.1
NAAF† (Zhang et al., 2022)	✓	81.9	96.1	98.3	61.0	85.3	90.6	513.2
CHAN (Pan et al., 2023)	✓	79.7	94.5	97.3	60.2	85.3	90.7	507.8
HREM† (Fu et al., 2023)	✗	81.4	96.5	98.5	60.9	85.6	91.3	514.2
CORA†‡ (Pham et al., 2024)	✗	82.3	96.1	98.0	63.0	87.4	92.8	**519.6**
Set Div (Kim et al., 2023)	✗	77.8	94.0	97.5	57.5	84.0	90.0	500.8
MaxMatch	✗	80.8	95.9	97.4	59.3	84.7	90.9	509.1
MaxMatch †	✗	82.1	95.6	98.3	61.6	86.3	91.9	515.8
Faster R-CNN + BERT								
DSRAN† (Wen et al., 2021)	✗	77.8	95.1	97.6	59.2	86.0	91.9	507.6
VSE∞ (Chen et al., 2021)	✗	81.7	95.4	97.6	61.4	85.9	91.5	513.5
MV-VSE† (Diao et al., 2021)	✗	82.1	95.8	97.9	63.1	86.7	92.3	517.5
CHAN (Pan et al., 2023)	✓	80.6	96.1	97.8	63.9	87.5	92.6	518.5
CODER (Wang et al., 2022)	✓	83.2	96.5	98.0	63.1	87.1	93.0	520.9
HREM† (Fu et al., 2023)	✗	84.0	96.1	98.6	64.4	88.0	93.1	524.2
CORA†‡ (Pham et al., 2024)	✗	83.4	95.9	98.6	64.1	88.1	93.1	523.3
Set Div (Kim et al., 2023)	✗	81.3	95.5	97.7	62.4	86.5	91.4	514.8
MaxMatch	✗	84.2	96.1	97.9	63.2	87.3	92.2	520.8
MaxMatch †	✗	86.2	95.7	98.4	64.8	88.8	93.2	**527.1**

Table 1: Recall@K (%) and RSUM on the Flickr30k dataset. The best RSUM scores are marked in **bold**, and the second-best scores are underlined. CA, ‡, and † indicate models using cross-attention, models that use external data, and ensemble models of two hypotheses.

4 Experiments

4.1 Datasets and Evaluation Metric

We evaluate MaxMatch on Flickr30k and COCO datasets. We follow the standard train, validation, test splits and evaluation practices from prior work. For COCO, we average performance over five folds of 1K test images and on the full set of 5K test images, while for Flickr30k, we use the test set. In both datasets, each image has five matching captions. Performance is measured using Recall@K ($K \in \{1, 5, 10\}$) and the RSUM metric, which sums Recall@K scores for image-to-text and text-to-image retrieval (Faghri et al., 2018). Implementation details are provided in the Appendix.

Method	CA	COCO 1K Test Images							COCO 5K Test Images						
		Image-to-Text			Text-to-Image				Image-to-Text			Text-to-Image			
		R@1	R@5	R@10	R@1	R@5	R@10	RSUM	R@1	R@5	R@10	R@1	R@5	R@10	RSUM
Resnet152 + Bi-GRU															
VSE++ (Faghri et al., 2018)	✗	64.6	90.0	95.7	52.0	84.0	92.0	478.6	41.3	71.1	81.2	30.3	59.4	72.4	355.7
PVSE (Song and Soleymani, 2019)	✗	69.2	91.6	96.6	55.2	86.5	93.7	492.8	45.2	74.3	84.5	32.4	63.0	75.0	374.4
PCME (Chun et al., 2021)	✗	68.8	-	-	54.6	-	-	-	44.2	-	-	31.9	-	-	-
Set Div (Kim et al., 2023)	✗	70.3	91.5	96.3	56.0	85.8	93.3	493.2	47.2	74.8	84.1	33.8	63.1	74.7	377.7
MaxMatch	✗	74.54	93.98	97.54	58.35	87.63	94.31	**506.75**	51.84	79.86	87.62	36.35	66	77.28	**398.95**
Faster R-CNN + Bi-GRU															
SCAN † (Lee et al., 2018)	✓	72.7	94.8	98.4	58.8	88.4	94.8	507.9	50.4	82.2	90	38.6	69.3	80.4	410.9
VSRN † (Li et al., 2019)	✗	76.2	94.8	98.2	62.8	89.7	95.1	516.8	53	81.1	89.4	40.5	70.6	81.1	415.7
CAAN (Zhang et al., 2020)	✓	75.5	95.4	98.5	61.3	89.7	95.2	515.6	52.5	83.3	90.9	41.2	70.3	82.9	421.1
SGRAF† (Diao et al., 2021)	✓	79.6	96.2	98.5	63.2	90.7	96.1	524.3	57.8	84.9	91.6	41.9	70.7	81.3	428.2
VSE$_\infty$ (Chen et al., 2021)	✗	78.5	96	98.7	61.7	90.3	95.6	520.8	56.6	83.6	91.4	39.3	69.9	81.1	421.9
NAAF† (Zhang et al., 2022)	✓	80.5	96.5	98.8	64.1	90.7	96.5	527.2	58.9	85.2	92.0	42.5	70.9	81.4	430.9
CHAN (Pan et al., 2023)	✓	79.7	96.7	98.7	63.8	90.4	95.8	525.0	60.2	85.9	92.4	41.7	71.5	81.7	433.4
HREM † (Fu et al., 2023)	✗	80.0	96.0	98.7	62.7	90.1	95.4	522.8	58.9	85.3	92.1	40.0	70.6	81.2	428.1
CORA †‡ (Pham et al., 2024)	✗	81.7	96.7	99.0	66.0	92.0	96.7	**532.1**	63.0	86.8	92.7	44.2	73.9	84.0	**444.6**
Set Div (Kim et al., 2023)	✗	79.8	96.2	98.6	63.6	90.7	95.7	524.6	58.8	84.9	91.5	41.1	72.0	82.4	430.7
MaxMatch	✗	80.4	96.4	98.4	64.6	91.0	96.0	526.6	59.5	85.8	94.0	42.3	72.8	83.0	436.1
MaxMatch †	✗	81.7	96.5	98.7	65.4	91.6	96.5	530.3	61.9	86.86	93.1	43.1	73.8	83.9	443.1
Faster R-CNN + BERT															
VSE∞ (Chen et al., 2021)	✗	79.7	96.4	98.9	64.8	91.4	96.3	527.5	58.3	85.3	92.3	42.4	72.7	83.2	434.2
CODER (Wang et al., 2022)	✓	82.1	96.6	98.8	65.5	91.5	96.2	530.7	62.6	86.6	93.1	42.5	73.1	83.3	441.2
MV-VSE† (Diao et al., 2021)	✗	80.4	96.6	99.0	64.9	91.2	96.0	528.1	59.1	86.3	92.5	42.5	72.8	83.1	436.3
CHAN (Pan et al., 2023)	✓	81.4	96.9	98.9	66.5	92.1	96.7	532.5	59.8	87.2	93.3	44.9	74.5	84.2	443.9
HREM † (Fu et al., 2023)	✗	82.9	96.9	99.0	67.1	92.0	96.6	534.5	64.0	88.5	93.7	45.4	75.1	84.3	451.0
CORA †‡ (Pham et al., 2024)	✗	82.8	97.3	99.0	67.3	92.4	96.9	535.6	64.3	87.5	93.6	45.4	74.7	84.6	450.1
Set Div (Kim et al., 2023)	✗	82.1	95.9	95.9	65.1	91.2	96.2	529.1	62.3	86.2	92.6	42.8	72.8	82.8	439.6
MaxMatch	✗	83.0	96.9	98.9	66.4	91.9	96.6	533.8	63.3	87.9	93.2	44.2	73.9	83.9	446.5
MaxMatch †	✗	83.8	97.0	98.9	67.3	92.4	96.9	**536.3**	65.4	88.7	93.8	45.1	74.9	84.8	**452.6**

Table 2: COCO Recall@K (%) and RSUM results on both the 1K test setting (average of 5-fold test dataset) and 5K test setting are presented. The best RSUM scores are marked in **bold**, and the second-best scores are underlined. CA, ‡, and † indicate models using cross-attention, models that use external data, and ensemble models, respectively.

4.2 Comparisons with Other Methods

We report results for the Flickr30k and COCO datasets in tables 1 and 2, respectively. We evaluate our method across two visual feature extractors: ImageNet pretrained ResNet152 (He et al., 2016), and Faster R-CNN with pre-extracted region-of-interest (ROI) features (Ren et al., 2015). We use either Bi-GRU or BERT for the text feature extraction. Following PVSE, SetDiv, CORA, *etc*., we report ensemble results by averaging similarities from two checkpoints with different seeds. Additionally, the results for the Faster R-CNN + BERT configuration were obtained using the official codebase, as this setting was not reported in the original paper (Kim et al., 2023).

We observe that MaxMatch outperforms state-of-the-art methods by an impressive margin except when using Bi-GRU as a semantic concept encoder with the Faster R-CNN visual features. However, MaxMatch achieves the second-best results in this setting. Note that the top-performing method, CORA enhances text representations through a graph generated by an external large language model-based parser (Li et al., 2023), introducing an additional model and text processing step. Using ResNet152 extracted visual features, MaxMatch substantially outperforms all other methods with a jump in RSUM by $+27.3$ on Flickr30k dataset and an increase in RSUM by $+24.55$ on COCO 5K. With BERT as the semantic concept encoder and Faster-RCNN as the visual feature extractor, MaxMatch outperforms all state-of-the-art methods by a significant margin. In particular, MaxMatch achieves $+2.9$ RSUM improvement over HREM on Flickr30k, and $+1.6$ RSUM on COCO 5K. It is also worth mentioning that MaxMatch outperforms CHAN by $+2.3$ RSUM on Flickr30k, and $+2.6$ RSUM on COCO 5K which requires cross-attention during inference time, making CHAN more computationally expensive.

4.3 Ablation Study

Table 3 ablates the different similarity functions (MIL, smooth-Chamfer, and Maximal Pair Assign-

Similarity	Div	MMD	GD	ISD	RSUM
MIL	✓	✓			438.98
Smooth-Chamfer	✓	✓			439.57
Smooth-Chamfer	✓	✓	✓		441.76
Smooth-Chamfer	✓	✓		✓	442.43
Smooth-Chamfer			✓	✓	443.11
Smooth-Chamfer	✓	✓	✓	✓	444.10
Maximal Pair Assignment	✓	✓			441.23
Maximal Pair Assignment			✓		442.67
Maximal Pair Assignment				✓	443.49
Maximal Pair Assignment			✓	✓	444.49
Maximal Pair Assignment	✓	✓	✓	✓	**446.53**

Table 3: Ablation study of the proposed similarity function and different similarity measures (MIL and Smooth Chamfer) combined with various loss settings, including Div, MMD, GD, and ISD, on the overall performance, measured by RSUM on COCO.

ment) in combination with various loss functions (Div, MMD, GD, and ISD) to analyze their impact on overall performance, as measured by RSUM on the 5k MS-COCO testing dataset. The Maximal Pair Assignment function achieves the highest RSUM score of 446.53 when combined with all four loss settings (Div, MMD, GD, and ISD), resulting in the best performance among all tested configurations. This highlights Maximal Pair Assignment as the most effective similarity function when combined with the appropriate losses. In contrast, other combinations, such as MIL or smooth-Chamfer with selected loss functions, achieve lower RSUM scores, suggesting reduced effectiveness at capturing the desired relationships. Notably, Maximal Pair Assignment with GD and ISD losses exhibits strong performance, achieving RSUM scores above 441, showing the importance of these loss functions in improving the model's performance.

4.4 Embedding Set Element Analysis

We analyze the role of individual embedding set elements in both $S^{\mathcal{V}}$ and $S^{\mathcal{T}}$ modalities in Table 4. Applying a max operation to pair embeddings from both modalities yields identical results, indicating a robust alignment mechanism. Retaining all embeddings *consistently achieves the best performance*, with the highest RSUM score of 446.53, highlighting their collective contribution. When using only $S^{\mathcal{V}}(3)$ or $S^{\mathcal{T}}(1)$, the RSUM drops to 437.91, emphasizing the critical role of all components. These findings underscore the advantage of leveraging complete embedding sets, unlike prior methods (e.g., PVSE and SetDiv). The bottom section reports circular variance $= 1 - \left\| \frac{\sum_{x \in S} x}{|S|} \right\|^2$, where lower log(Var.) values indicate stronger set

Evaluation				RSUM		
$S^{\mathcal{V}}(1)$	$S^{\mathcal{V}}(2)$	$S^{\mathcal{V}}(3)$	$S^{\mathcal{V}}(4)$	MIL	Smooth Chamfer	MaxMatch
✓	✓	✓	✓	**438.98**	**439.57**	**446.53**
✓				439.08	212.12	440.89
	✓			389.97	439.03	443.78
		✓		400.06	439.50	437.91
			✓	302.14	438.84	442.39
$S^{\mathcal{T}}(1)$	$S^{\mathcal{T}}(2)$	$S^{\mathcal{T}}(3)$	$S^{\mathcal{T}}(4)$			
✓	✓	✓	✓	**438.98**	**439.57**	**446.53**
✓				438.84	439.24	437.91
	✓			393.10	438.12	440.89
		✓		293.50	438.12	442.39
			✓	412.32	439.14	443.78
Circular variance of embedding set				MIL	Smooth Chamfer	MaxMatch
RSUM				483.3	500.8	509.1
log(Var.)				-7.35	-2.13	-1.68

Table 4: RSUM on COCO dataset for $S^{\mathcal{V}}$ (top) and $S^{\mathcal{T}}$ (middle). This highlights the impact of selectively removing components on overall performance and embedding stability across three methods: MIL, Smooth Chamfer, and MaxMatch . The circular variance of the embedding set (bottom) on Flickr30k, where lower log(Var.) values indicate stronger collapse.

collapsing, where MaxMatch maintains a higher log(Var.), indicating less set collapse compared to other method, ensuring better disentanglement and a more diverse embedding space.

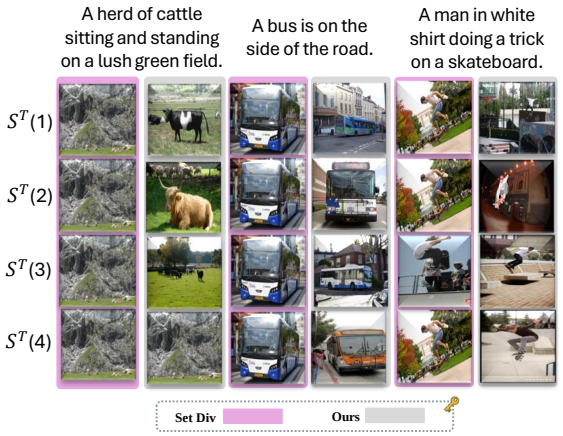

Figure 5: Visual comparison of image retrieval results between our method and Set Div. For each embedding in the set, we show the top retrieved image. Our method generates embeddings that retrieve diverse images, while Set Div's embeddings tend to collapse, resulting in retrieval of identical or highly similar images.

4.5 Qualitative Analysis

To compare the semantic diversity of our learned embeddings with prior work, we perform cross-modal retrieval using each embedding in our embedding set independently. Given a set of embeddings for a query, we retrieve the closest test sample for each query embedding by selecting the best-matched embedding from the test sample's set. This evaluates how well our embeddings capture

diverse semantics. Qualitative results in Figure 5 show that SetDiv representations collapse, retrieving identical or highly similar samples, whereas our embeddings retrieve diverse yet semantically consistent results. The left side of the figure indicates which query embedding was used for retrieval.

5 Conclusion

We introduced a novel set-based embedding framework for cross-modal retrieval with Maximal Pair Assignment Similarity and a combination of new loss functions. The Maximal Pair Assignment Similarity, utilizing permutation-based assignments, resolves challenges of sparse supervision and set collapsing, improving accurate image-text similarity scoring. Global Discriminative Loss further enhances the model's ability to differentiate between embeddings, while the Intra-Set Divergence Loss mitigates set collapsing by encouraging diversity within each set. Our approach achieves superior performance on standard benchmarks compared to previous methods, and we plan to extend it to other modalities and tasks in future work.

6 Acknowledgements

We acknowledge Advanced Research Computing at Virginia Tech for providing computational resources and technical support that have contributed to the results reported within this paper. We also thank all reviewers for their comments which helped improve the paper.

7 Limitation

While our proposed MaxMatch demonstrates strong performance for cross-modal retrieval and overcomes many important limitations related to representation set collapse, several important limitations remain unaddressed. While MS-COCO and Flickr30k are standard benchmarks, they represent a relatively constrained subset of real-world cross-modal scenarios. Most image captions in these datasets focus on describing concrete objects and actions, rather than abstract concepts, emotional content, or diverse cultural interpretations that might benefit from multiple distinct representations. Future work should explore more diverse and challenging datasets that better reflect the complexity of human visual and linguistic understanding. In addition , while effective for image-text pairs, the current formulation of MaxMatch is specific to bi-modal retrieval. Extending the framework to handle additional modalities or simultaneous alignment across three or more modalities would require substantial modifications to both the matching mechanism and loss functions. This limits its immediate applicability in broader multimodal scenarios involving audio, video, or other data types. These limitations point to several promising directions for future research, including more efficient matching algorithms, adaptive set size mechanisms, and generalization to broader multimodal contexts.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Jiacheng Chen, Hexiang Hu, Hao Wu, Yuning Jiang, and Changhu Wang. 2021. Learning the best pooling strategy for visual semantic embedding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15789–15798.

Sanghyuk Chun. 2024. Improved probabilistic image-text representations. In *International Conference on Learning Representations (ICLR)*.

Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio De Rezende, Yannis Kalantidis, and Diane Larlus. 2021. Probabilistic embeddings for cross-modal retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8415–8424.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu. 2021. Similarity reasoning and filtration for image-text matching. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(2):1218–1226.

Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. Vse++: Improving visual-semantic embeddings with hard negatives. In *Proceedings of the British Machine Vision Conference (BMVC)*.

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. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Zheren Fu, Zhendong Mao, Yan Song, and Yongdong Zhang. 2023. Learning semantic relationship among instances for image-text matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xuri Ge, Songpei Xu, Fuhai Chen, Jie Wang, Guoxin Wang, Shan An, and Joemon M Jose. 2024. 3shnet: Boosting image–sentence retrieval via visual semantic–spatial self-highlighting. *Information Processing & Management*, 61(4):103716.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. 2006. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19.

Jiuxiang Gu, Jianfei Cai, Shafiq R Joty, Li Niu, and Gang Wang. 2018. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7181–7189.

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.

Dongwon Kim, Namyup Kim, and Suha Kwak. 2023. Improving cross-modal retrieval with set of diverse embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23422–23431.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *Proceedings of the European conference on computer vision (ECCV)*, pages 201–216.

Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. 2019. Visual semantic reasoning for image-text matching. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4654–4662.

Zhuang Li, Yuyang Chai, Terry Yue Zhuo, Lizhen Qu, Gholamreza Haffari, Fei Li, Donghong Ji, and Quan Hung Tran. 2023. Factual: A benchmark for faithful and consistent textual scene graph parsing. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6377–6390.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Nicola Messina, Giuseppe Amato, Andrea Esuli, Fabrizio Falchi, Claudio Gennaro, and Stéphane Marchand-Maillet. 2021. Fine-grained visual textual alignment for cross-modal retrieval using transformer encoders. *ACM Trans. Multimedia Comput. Commun. Appl.*, 17(4).

Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2021. Thinking fast and slow: Efficient text-to-visual retrieval with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9826–9836.

Zhengxin Pan, Fangyu Wu, and Bailing Zhang. 2023. Fine-grained image-text matching by cross-modal hard aligning network. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19275–19284.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W: Workshop on Automatic Differentiation*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Khoi Pham, Chuong Huynh, Ser-Nam Lim, and Abhinav Shrivastava. 2024. Composing object relations and attributes for image-text matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14354–14363.

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.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Yale Song and Mohammad Soleymani. 2019. Polysemous visual-semantic embedding for cross-modal retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1979–1988.

Christopher Thomas and Adriana Kovashka. 2020. Preserving semantic neighborhoods for robust cross-modal retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 317–335. Springer.

Christopher Thomas and Adriana Kovashka. 2022. Emphasizing complementary samples for non-literal cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4632–4641.

Haoran Wang, Dongliang He, Wenhao Wu, Boyang Xia, Min Yang, Fu Li, Yunlong Yu, Zhong Ji, Errui Ding, and Jingdong Wang. 2022. Coder: Coupled diversity-sensitive momentum contrastive learning for image-text retrieval. In *European Conference on Computer Vision*, pages 700–716. Springer.

Jiwei Wei, Xing Xu, Yang Yang, Yanli Ji, Zheng Wang, and Heng Tao Shen. 2020a. Universal weighting metric learning for cross-modal matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13005–13014.

Xi Wei, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. 2020b. Multi-modality cross attention network for image and sentence matching. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10938–10947.

Keyu Wen, Xiaodong Gu, and Qingrong Cheng. 2021. Learning dual semantic relations with graph attention for image-text matching. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(7):2866–2879.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.

Fei Yan and Krystian Mikolajczyk. 2015. Deep correlation for matching images and text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kun Zhang, Zhendong Mao, Quan Wang, and Yongdong Zhang. 2022. Negative-aware attention framework for image-text matching. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qi Zhang, Zhen Lei, Zhaoxiang Zhang, and Stan Z. Li. 2020. Context-aware attention network for image-text retrieval. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3533–3542.

A Appendix

This Appendix provides additional details and results that complement the main paper. Section B outlines our comprehensive implementation for various model configurations. In Section C, we show that smooth Chamfer similarity encourages set collapse. Methodological considerations regarding external knowledge integration are discussed in Section D. Section E evaluates our approach using stronger visual backbones, and Section F examines its performance on the PVSE architecture. Finally, Section H presents pseudocode for our key components.

B Implementation Details

Following prior work (Kim et al., 2023; Song and Soleymani, 2019), we use two types of feature extractors and set the embedding dimension $D = 1024$. We obtain convolutional visual features by applying a 1×1 convolution to the final feature map of a CNN. Then, we use the pre-extracted 2048-dimensional region features from Faster R-CNN (Ren et al., 2015) as BUTD (Anderson et al., 2018). Both feature types were transformed to the embedding space via a two-layer MLP with residual connections. We set $K = 4$ and $D_h = 1024$ for convolutional features and $D_h = 2048$ for region features (Kim et al., 2023). We process textual features using a BiGRU with either GloVe (Pennington et al., 2014) or BERT (Devlin et al., 2019) embeddings.

Our implementation uses PyTorch (Paszke et al., 2017) v2.0.1, leveraging the model architecture from the SetDiv codebase (Kim et al., 2023). We employ automatic mixed precision to improve training speed and efficiency. Each input batch consists of either 128 or 200 images along with their corresponding captions. We train our model using the AdamW optimizer, with both MMD loss (λ_{MMD}) and diversity loss (λ_{Div}) weighted at 0.01 across all configurations. While the main paper describes the key implementation details, specific training parameters vary based on the feature extractors used.

ResNet152 + BiGRU: We train the model for 120 epochs using initial learning rates of 1e-3 for MS-COCO (Lin et al., 2014) and 2e-3 for Flickr30k (Plummer et al., 2015). For the set prediction module, these learning rates are scaled by factors of 0.1 and 0.01 for MS-COCO and Flickr30k, respectively. Following (Song and Soleymani, 2019; Kim

et al., 2023), we apply step-wise learning rate decay with a factor of 0.1 every 20 epochs. During the first 50 epochs, the CNN weights are kept frozen. Training is performed on dual NVIDIA A40 GPUs with a batch size of 200. For the loss configuration, the triplet margin (δ_1) is set to 0.2, while both Global Discriminative Loss (λ_{GD}) and Intra-Set Divergence Loss (λ_{ISD}) use weights of 0.1. The margin parameters ($\delta_{2,3}$) and scaling parameter (s) are set to 0.6 and 0.5, respectively. Contrastive loss is not included in this configuration.

Faster R-CNN + BiGRU: The model is trained for 80 epochs with an initial learning rate of 1e-3, employing cosine annealing (Loshchilov and Hutter, 2017) and weight decay of 1e-4. For the set prediction module, the learning rate is scaled by 0.1 for both datasets. Following (Chen et al., 2021), we apply a dropout of 20% to both ROI features and word embeddings during training. Training is conducted on a single NVIDIA A40 GPU with a batch size of 200. For the loss configuration, the triplet margin (δ_1) is set to 0.3, while both Global Discriminative Loss (λ_{GD}) and Intra-Set Divergence Loss (λ_{ISD}) use weights of 0.1 for MS-COCO and 0.05 for Flickr30K. The margin parameters ($\delta_{2,3}$) and scaling parameter (s) are set to 0.6 and 0.5, respectively. The contrastive loss weight is set to 0.001.

Faster R-CNN + BERT: The model trains for 45 epochs with an initial learning rate of 1e-3, employing cosine annealing and weight decay of 5e-4. For the set prediction module and BERT, the learning rates are scaled by 0.5 and 0.1, respectively. Following previous configurations, we apply dropout of 20% to both ROI features and word embeddings during training. Training is conducted on dual NVIDIA A40 GPUs with batch sizes of 200 for MS-COCO and 128 for Flickr30k. For the loss configuration, the triplet margin (δ_1) is set to 0.15, while both Global Discriminative Loss (λ_{GD}) and Intra-Set Divergence Loss (λ_{ISD}) use weights of 0.1. The margin parameters ($\delta_{2,3}$) and scaling parameter (s) are set to 0.8 and 0.5, respectively. The contrastive loss weight is set to 0.001.

C Smooth Chamfer Leads to Set Collapse

We consider the smooth Chamfer similarity defined as

$$S_{SC}(S_1, S_2) = \frac{1}{2\alpha|S_1|} \sum_{x \in S_1} \log\left(\sum_{y \in S_2} e^{\alpha c(x,y)} \right) + \frac{1}{2\alpha|S_2|} \sum_{y \in S_2} \log\left(\sum_{x \in S_1} e^{\alpha c(x,y)} \right). \quad (9)$$

In what follows, we show, under suitable assumptions (e.g. when $c(x, y)$ is affine in x, as in $c(x, y) = x^\top y$), that the objective is minimized when all elements in each set are identical. For clarity, we detail the argument for S_1, and a symmetric argument applies to S_2.

Let $S_1 = \{x_1, x_2, \ldots, x_n\}$. Assume that S_2 is fixed. Define the function $L(x) \triangleq \log\left(\sum_{y \in S_2} e^{\alpha\, c(x,y)} \right)$. Then, the contribution of S_1 to the overall objective is $f(S_1) = \frac{1}{2\alpha n} \sum_{i=1}^{n} L(x_i)$. (We omit the constant prefactor $1/(2\alpha)$ in what follows, as it does not affect the location of the minimum.)

Assume that $c(x, y)$ is affine in x for every fixed y; for example, if $c(x, y) = x^\top y$, then for fixed y the map $x \mapsto \alpha\, c(x, y)$ is affine. Since the exponential is convex and increasing, the composition $x \mapsto e^{\alpha\, c(x,y)}$ is convex. Moreover, the sum over y of convex functions is convex, and the logarithm of a sum of exponentials (the log-sum-exp function) is also convex. Thus, the function $L(x) = \log\left(\sum_{y \in S_2} e^{\alpha\, c(x,y)} \right)$ is convex in x.

Let the arithmetic mean of the embeddings in S_1 be $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$. Then by Jensen's inequality we have:

$$\frac{1}{n} \sum_{i=1}^{n} L(x_i) \geq L\left(\frac{1}{n} \sum_{i=1}^{n} x_i \right) = L(\bar{x})\,. \quad (10)$$

Equality holds if and only if all x_i are equal. That is, for any fixed mean $\bar{x}$, the minimum of

$$f(S_1) = \frac{1}{n} \sum_{i=1}^{n} L(x_i) \quad (11)$$

is achieved if and only if

$$x_1 = x_2 = \cdots = x_n = \bar{x}. \quad (12)$$

Thus, any deviation (i.e., any diversity among the x_i) increases the objective.

This also becomes obvious when analyzed from the perspective of the gradient. Consider the gradient of $L(x)$ with respect to x. By the chain rule,

$$\nabla_x L(x) = \frac{\sum_{y \in S_2} \alpha \, e^{\alpha \, c(x,y)} \, \nabla_x c(x,y)}{\sum_{y \in S_2} e^{\alpha \, c(x,y)}}. \quad (13)$$

Thus, each x_i receives a gradient that is a softmax-weighted average of the vectors $\alpha \nabla_x c(x,y)$ (for example, when $c(x,y) = x^\top y$, one has $\nabla_x c(x,y) = y$).

Now, consider a small perturbation of the set S_1 that preserves the overall mean.

$$x_i = \bar{x} + \delta_i, \qquad \text{with} \qquad \sum_{i=1}^{n} \delta_i = 0. \quad (14)$$

Because L is convex, its Hessian denoted by

$$H(x) = \nabla_x^2 L(x), \quad (15)$$

is positive semidefinite:

$$H(x) \succeq 0. \quad (16)$$

A standard result in convex analysis is that the average

$$\frac{1}{n} \sum_{i=1}^{n} L(\bar{x} + \delta_i) \quad (17)$$

is minimized (with respect to the deviations $\{\delta_i\}$ under the constraint $\sum_i \delta_i = 0$) when all $\delta_i = 0$. In other words, any nonzero differences δ_i (i.e., any diversity among the embeddings) increases the objective's value. Hence, the Hessian is strictly positive in the directions that would cause dispersion in S_1. Thus, the gradients computed above will push any x_i that deviates from the mean $\bar{x}$ back toward $\bar{x}$. Hence, during training the optimal (or at least a stationary) point is one where all embeddings in S_1 are equal. A symmetric argument applies to S_2.

D Discussion on CORA and 3SHNet External Knowledge Integration

Recent work by CORA (Pham et al., 2024) achieves impressive results through the integration of an external LLM as a scene graph parser (Li et al., 2023), particularly when using Bi-GRU encoders. While this approach demonstrates strong performance, it introduces important considerations for fair comparison. CORA's key innovation lies in enhancing text representations by augmenting raw captions with structured scene graphs derived from an LLM parser, effectively incorporating external semantic knowledge into the representation learning process. This external knowledge particularly benefits simpler text encoders like Bi-GRU, where the additional semantic structure helps bridge the gap in language understanding capabilities. CORA achieves state-of-the-art results even with embeddings trained from scratch (as reported in their supplementary material), highlighting how the LLM parser significantly improves textual representations.

In contrast, 3SHNet (Ge et al., 2024) takes a different approach by integrating external visual knowledge through semantic segmentation features. Rather than relying on a language-based external module, 3SHNet incorporates structured spatial and object-level information using UPSNet to enhance object-region representations. This structured semantic-spatial self-highlighting method improves image representation, making retrieval more robust without requiring external textual scene graphs.

These approaches raise important methodological considerations. Unlike CORA and 3SHNet, our method operates directly on raw captions using a standard text encoder (GloVe or BERT) without incorporating external knowledge or additional preprocessing, and does not rely on UPSNet segmentation beyond the basic steps used in prior work (Song and Soleymani, 2019; Kim et al., 2023; Fu et al., 2023; Faghri et al., 2018; Chen et al., 2021). This distinction becomes particularly evident when using stronger language models like BERT, where the benefits of external parsing diminish as the encoder itself becomes more capable of understanding complex semantic relationships. While CORA's paper acknowledges that BERT excels at encoding longer text sequences, their approach primarily uses it to encode short phrases, which loses the global context of the sentence. Our method, in contrast, leverages BERT's full capability to process complete captions, achieving competitive performance without requiring additional scene graph parsing or external knowledge integration. This demonstrates the effectiveness of our approach in maintaining semantic richness while keeping the architecture simpler and more efficient.

E Testing with a Larger Backbone

In alignment with prior work (Kim et al., 2023), we select ResNeXT101 (Xie et al., 2017) pretrained

	Image-to-Text			Text-to-Image			RSUM
	R@1	R@5	R@10	R@1	R@5	R@10	
MS-COCO 1K Test Images							
VSE∞	84.5	98.1	99.4	72	93.9	97.5	545.4
Set Div	86.3	97.8	99.4	72.4	94	97.6	547.5
MaxMatch	86.9	98.12	99.38	72.22	94.12	97.6	**547.84**
MS-COCO 5K Test Images							
VSE∞	66.4	89.3	94.6	51.6	79.3	87.6	468.9
Set Div	69.1	90.7	95.6	52.1	79.6	87.8	474.9
MaxMatch	69.62	90.76	95.54	52.18	79.46	87.58	**475.14**
Flickr30k Test Images							
VSE∞	88.4	97.3	99.5	74.2	93.7	96.8	550.9
Set Div	88.8	98.5	99.6	74.3	94	96.7	551.9
MaxMatch	90.5	99	99.8	74.96	94.2	96.78	**555.24**

Table 5: Recall@K (%) and RSUM on MS-COCO and Flickr30k dataset with ResNeXT101 and BERT backbones. The best RSUM scores are marked in **bold**, and the second-best scores are underlined.

on the Instagram dataset (Li et al., 2019) as the larger visual extractor paired with BERT (Devlin et al., 2019). The model trains for 50 epochs with an initial learning rate of 1e-4, which decays by a factor of 0.1 every 20 epochs. Consistent with (Faghri et al., 2018), the CNN's learning rate is scaled down by 0.1, and the CNN remains frozen during the first epoch. During this initial phase, we employ triplet loss without hard negative mining, then transition to hardest negative mining in subsequent epochs. For the loss configuration, both Global Discriminative Loss (λ_{GD}) and Intra-Set Divergence Loss (λ_{ISD}) use weights of 0.1. The margin parameters ($\delta_{2,3}$) and scaling parameter (s) are set to 0.6 and 0.5, respectively, with a triplet margin (δ_1) of 0.1. Contrastive loss is not included in this configuration. Training is performed on dual A100 PCIe GPUs with a batch size of 128. This visual backbone is also used in VSE (Chen et al., 2021) and Set Div (Kim et al., 2023). The results on the MS-COCO and Flickr30K test sets are displayed in Table 5, demonstrating performance improvements with our method.

The results in Table 5 demonstrate the effectiveness of our method when combined with stronger visual backbones. On Flickr30K, our method achieves significant improvements over both VSE∞ and Set Div baselines, with notable gains in Image-to-Text retrieval (R@1 improves from 88.8% to 90.5%). Similar improvements are observed on MS-COCO, where our method consistently outperforms previous approaches across both 1K and 5K test sets. Particularly in the more challenging 5K setting, we improve the RSUM from 474.9 to 475.14, demonstrating our method's robustness to larger retrieval spaces.

	Image-to-Text			Text-to-Image			RSUM
	R@1	R@5	R@10	R@1	R@5	R@10	
VSE++	52.9	80.5	87.2	39.6	70.1	79.5	409.8
PVSE	59.1	84.5	91	43.4	73.1	81.5	432.6
PCME	58.5	81.4	89.3	44.3	72.7	81.9	428.1
Set Div	61.8	85.5	91.1	46.1	74.8	83.3	442.6
MaxMatch + PVSE	**65.4**	**88.1**	**93.7**	**45.4**	**74.7**	**83.2**	**450.6**
MaxMatch	68.6	89.6	94.6	51.5	78.9	86.8	469.9

Table 6: Recall@K (%) and RSUM on Flickr30k dataset on PVSE architicture. The scores for PVSE with our method is in **bold**.

F Evaluating Method Performance on PVSE

We evaluate the versatility and effectiveness of our proposed method by applying it to the PVSE architecture, demonstrating its ability to enhance various baseline models. While PVSE offers more modest performance compared to recent methods like SDE, it provides an excellent test case for assessing our method's generalizability. The results in Table 6 show substantial improvements across all metrics when our method is integrated with PVSE. Specifically, our method improves PVSE's Image-to-Text R@1 performance from 59.1% to 65.4%, a significant gain of 6.3 percentage points, surpassing even the more recent Set Div model (61.8%). In Text-to-Image retrieval, we achieve consistent improvements, with R@1 increasing from 43.4% to 45.4%. The overall RSUM metric improves by 18 points (from 432.6 to 450.6), demonstrating the comprehensive enhancement our method brings to the baseline model.

Notably, our method's performance with PVSE not only surpasses the original PVSE results but also outperforms several more recent approaches including PCME and approaches the performance of Set Div. This is particularly impressive given PVSE's relatively simpler architecture. A key distinction of our approach is its ability to utilize the entire embedding set during training and inference, whereas traditional Multiple Instance Learning (MIL) approaches in PVSE only select a single embedding from the set. This comprehensive utilization enables our method to better capture diverse nuanced meanings from the samples, leading to more robust representations. When further combined with Set Div, our method achieves even more substantial gains, reaching state-of-the-art performance with an RSUM of 469.9, highlighting its complementary nature to existing advanced techniques. These results convincingly demonstrate that our method is architecture-agnostic and can

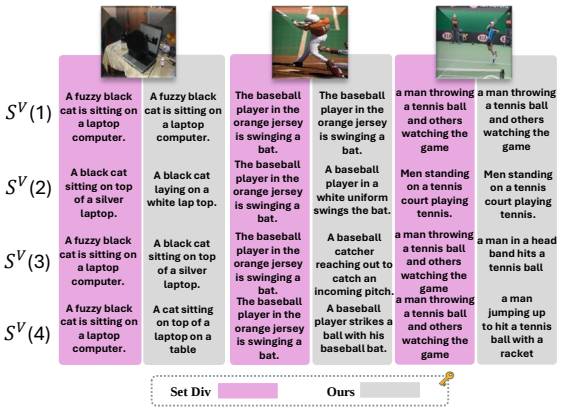

Figure 6: Visual comparison of image retrieval results between our method and Set Div. For each embedding in the set, we show the top retrieved image. Our method generates embeddings that retrieve diverse images, while Set Div's embeddings tend to collapse, resulting in retrieval of identical or highly similar images.

Figure 7: Failure cases in image-to-text retrieval using our method. While our approach typically retrieves semantically diverse captions, some examples particularly those with visually homogeneous content result in repeated or overly similar captions across the set. This highlights a limitation in capturing fine-grained diversity for certain input images.

effectively enhance both simple and sophisticated models, validating its broad applicability in cross-modal retrieval tasks.

G Qualitative Analysis for Image-to-Text Retrieval

Following Section 4.5, we extend our qualitative analysis to image-to-text retrieval. We perform cross-modal retrieval using each embedding in our set independently, selecting the closest test sample for each query embedding. This evaluates the semantic diversity captured by our embeddings compared to prior work. As shown in Figure 6, SetDiv representations tend to collapse, retrieving nearly identical samples, whereas our embeddings yield diverse yet semantically coherent results. The left side of the figure indicates the query embedding used for retrieval.

Failure Cases and Limitations: While our method generally produces more semantically diverse and accurate retrievals compared to SetDiv, we also observe certain failure cases where diversity is reduced or retrievals collapse. As shown in the rightmost column of Figure 7, our embeddings occasionally yield highly similar or even repeated captions across different slots, especially for images with limited semantic variation or dominant foreground objects. This suggests that, in scenarios where visual content is overly focused or lacks contextual richness, the learned embeddings may converge to similar representations. To further characterize these limitations, we include additional qualitative examples in the appendix, including cases where our model retrieves semantically plausible yet not part of the ground truth captions, a common occurrence in image-text datasets.

H Code Availability

We provide pseudocodes of our key components illustrating the Global Discriminative Loss, Intra-Set Divergence Loss, and Maximal Pair Assignment Similarity functions. These code snippets demonstrate the conceptual implementation of our method in PyTorch notation, highlighting both the efficiency and simplicity of our approach. We will release the codebase on GitHub.

Maximal Pair Assignment Similarity

```python
def mask_max_similarity(sim_matrix,
    text_indices, img_indices):
    """
    Generate a binary mask to identify
        maximum similarity pairs in a
        similarity matrix.

    Args:
        sim_matrix (Tensor): Similarity
             matrix between text and
            image embeddings.
        text_indices (Tensor): Indices
            corresponding to text
            embeddings.
        img_indices (Tensor): Indices
            corresponding to image
            embeddings.

    Returns:
        Tensor: Binary mask with 1s at
            maximum similarity pairs.
    """
    # Step 1: Extract similarities at
        specified indices
    selected_similarities = sim_matrix[
        text_indices, img_indices]

    # Step 2: Find indices of maximum
        similarity for each pair
    max_indices = selected_similarities
        .argmax(dim=1)

    # Step 3: Create binary mask with 1
        s at maximum similarity indices
    mask = torch.zeros_like(sim_matrix)
    mask[text_indices[max_indices],
        img_indices[max_indices]] = 1

    return mask
```

Listing 1: Generate a mask for maximum similarity pairs in a similarity matrix.

```python
def create_index_permutations(
    num_embeddings, row_size, col_size):
    # Generate permutations of indices
        for matrix operations.
    # Step 1: Generate all permutations
         of the embeddings
    permutations = torch.tensor(list(
        itertools.permutations(range(
        num_embeddings))), dtype=torch.
        long)
    # Step 2: Repeat permutations to
        match the required row size
    row_indices = permutations.repeat(
        row_size // num_embeddings, 1)
    # Step 3: Create column indices to
        match the required column size
    col_indices = torch.arange(col_size
        ).repeat(row_indices.size(0), 1)
    # tuple: Two tensors representing
        row and column indices.
    return row_indices, col_indices
```

Listing 2: Create index permutations for matrix operations.

```python
def maximal_pair_assignment_similarity(
    img_embs, txt_embs):
    """
    Compute assignment of maximum
        similarity between image and
        text embeddings.

    Args:
        img_embs (Tensor): Image
            embeddings.
        txt_embs (Tensor): Text
            embeddings.

    Returns:
        Tensor: Maximum similarity
            tensor.
    """
    # Step 1: Compute cosine similarity
        between normalized embeddings
    dist = cosine_sim(l2norm(img_embs),
        l2norm(txt_embs))

    # Step 2: Determine row and column
        sizes based on batch and set
        sizes
    row_size = image_batch_size *
        img_set_size  # Number of image
        embeddings
    col_size = text_batch_size *
        txt_set_size  # Number of text
        embeddings

    # Step 3: Generate all possible
        index permutations for matching
    text_index_all, image_index_all =
        create_index_permutations(
        img_set_size, row_size, col_size
        )

    # Step 4: Create a mask for maximum
        similarity pairs
    mask = mask_max_similarity(dist.
        detach(), text_index_all,
        image_index_all)

    # Step 5: Calculate maximum
        similarity scores
    max_similarity = mask * dist

    # Step 6: Apply exponential scaling
        and average pooling
    max_similarity = avg_pool(torch.exp(
        max_similarity.unsqueeze(0)) -
        1) * img_set_size

    return max_similarity.squeeze()
```

Listing 3: Function to compute assignment of maximum similarity between embeddings.

Intra-Set Divergence Loss

```
def Intra_Set_Divergence_Loss(set_emb,
    margin, scale):
    # Compute pairwise similarity
        between all embeddings in the
        set
    A = torch.bmm(set_emb, set_emb.
        transpose(1, 2))  # [bs, K, K]

    # Create a mask to exclude self-
        similarities and include only
        upper triangular matrix elements
    num_embeddings = set_emb.size(1)
    mask = torch.triu(torch.ones(
        num_embeddings, num_embeddings,
        device=set_emb.device), diagonal
        =1).bool()
    mask = mask.unsqueeze(0)  # Expand
        mask to match batch dimension

    # Apply the mask to extract unique
        pairwise similarities
    A_masked = A.masked_select(mask)

    # Compute the loss using the smooth
        exponential function with margin
         and scale
    loss = torch.exp((A_masked - margin)
         * scale)

    # Normalize the loss over the batch
        and number of unique pairs
    return loss.sum() / (set_emb.shape
        [0] * (set_emb.shape[1] * (
        set_emb.shape[1] - 1) / 2))
```

Listing 4: Intra-Set Divergence Loss

Global Discriminative Loss

```
def Global_Discriminative_Loss(Set_emb,
    global_emb, margin, scale):
    # Add a singleton dimension to
        global_embedding for batch
        matrix multiplication
    global_emb= global_emb.unsqueeze(1)
        #[bs,1,D]

    # Transpose embeddings to match
        dimensions for batch matrix
        multiplication
    Set_emb= Set_emb.transpose(1, 2) #[
        bs,D,K]

    # Compute pairwise similarity
        between global embedding and the
         set of embeddings
    A= torch.bmm(global_emb, Set_emb).
        squeeze() #[bs,K]

    # Compute the loss by applying the
        margin and scale, then take the
        mean
    loss= torch.mean(torch.exp(scale * (
        A - margin)))
    return loss
```

Listing 5: Global Discriminative Loss