MAFA: Managing False Negatives for Vision-Language Pre-training

Supplementary Material

1. Data and Implementation Details

In this section, we provide information about the software and the dataset used in our study. We conducted experiments with four NVIDIA A100 GPUS with Python 3.8 and Pytorch with CUDA 11.1.

To construct the 4M-Clean dataset, we download the dataset corpus (CC3M+CC12M+SBU) from the official github of BLIP. The dataset corpus is generated and filtered by a model equipped with VIT-B/16 as its image transformer. From this corpus, we selectively utilize image-text pairs from the (CC3M+SBU) to align with the 4M-Noisy dataset. Note that the dataset refinement process (generating clean captions, and filtering noisy image-text pairs) is exclusively done with the web-crawled dataset (CC3M+SBU), and (COCO+VG) dataset is incorporated in the 4M-Clean dataset without any modification.

As described in the manuscript (Section 5), we mainly use a synthetically generated and filtered set (I_w, T_s) from this dataset corpus. To ensure that the size of the 4M-Clean dataset is nearly identical to the 4M-Noisy dataset, we additionally use the small number of (I_w, T_w) among the (CC3M+SBU) dataset due to the synthetically generated and filtered set (I_w, T_s) being slightly smaller than the 4M-Noisy dataset. The total number of image-text pairs in the 4M-Noisy set and the 4M-Clean set is $4,999,065$ and $4,933,639$, respectively.

2. Details on counting false negatives in Section 3 (manuscript)

Here, we first clarify the purpose of using the pre-trained BLIP (129M) model in quantitative analysis in Section 3 (manuscript) for better understanding. The quantitative analysis represented in Table 1 and Figure 2 (manuscript) includes the estimated count of false negative pairs during the ITM task of training. The main goal is to compare the number of false negatives arising from randomly constructed mini-batches (typical VLP setting) to those from GRIT-sampled mini-batches that group similar pairs in each mini-batch. Here, to identify false negative pairs with perfect accuracy, it is essential to employ a human evaluation process, requiring manual examination of each individual negative pair (constructed for performing ITM task) to determine if it is matched or not. However, given the infeasibility of manually checking all negative pairs, we have opted to leverage the pre-trained BLIP (129M) model, a strong ITM model, as an alternative to human evaluation. While the BLIP (129M) does not always classify with perfect accuracy, it is deemed sufficiently reliable for approximating the tendency of the number of false negatives.

To validate this, we additionally conduct a human evaluation on randomly sampled 200 false negatives in the 4M-Clean dataset during training which are filtered by BLIP (129M) filter. For this, two ML researchers manually check whether each false negative is genuinely a false negative or not. In this analysis, among the samples classified as false negatives by BLIP (129M), over 83% are also determined to be false negatives upon human evaluation. While it's not 100% accurate, we believe that using BLIP (129M) is reasonable for approximating the number of false negatives during training and, consequently, for comparing the occurrence of false negatives between random sampling and GRIT sampling. To report representative numbers, the values provided in Table 1 (manuscript) are obtained by multiplying the actual values from the BLIP (129M) model by 0.83, assuming that human correction is statistically significant. The raw values prior to multiplication are presented in rows 1 and 2 of Table 1.

Additionally, we examine the number of false negatives using other discriminators, as detailed in Table 1. Unlike the values in Table 1 (manuscript), these values are not multiplied by 0.83. ALBEF (14M) denotes the ALBEF model pre-trained on the 14M dataset and fine-tuned on the MS-COCO dataset, while BLIP-2 represents the BLIP-2 model (equipped with ViT-g) pre-trained on 129M and fine-tuned on the MS-COCO dataset. These results also indicate that the tendency of false negative ratios remains consistent across different discriminators.

Table 1. The number of false negatives (FNs) counted by various discriminators. Unlike Table 1 (manuscript), these values represent the raw counts without any adjustments, as the human correction factor of 0.83 has not been applied.

Discriminator	Dataset	Sampling	FN w.r.t. image	FN w.r.t. text
BLIP (129M)	4M-Noisy	Random	146,127 (2.9%)	142,265 (2.8%)
		GRIT	985,531 (19.7%)	977,283 (19.5%)
	4M-Clean	Random	184,345 (3.7%)	179,191 (3.6%)
		GRIT	1,383,752 (28.0%)	1,321,066 (26.8%)
ALBEF (14M)	4M-Clean	Random	216,179 (4.4%)	210,976 (4.3%)
		GRIT	1,565,023 (31.7%)	1,477,597 (29.9%)
BLIP-2	4M-Clean	Random	118,118 (2.4%)	115,281 (2.3%)
		GRIT	939,797 (19.0%)	859,356 (17.4%)

3. Additional analyses on false negative ratio (FNR)

We quantify the FNR *w.r.t.* batch size under a *random* sampling scenario. As shown in Figure 1, increasing the batch size on 4M-Clean apparently results in a higher FNR. This finding suggests the presence of a significant number of false negatives even in a *random* sampling scenario with a large

batch size, which is a common practice in recent VLP training schemes such as BLIP-2. Consequently, this result also underscores the applicability of MAFA in settings where hard negative sampling like GRIT is not employed.

We also measure the FNR using a considerably larger, web-crawled CC12M-Clean dataset with a default batch size of 96. In the GRIT sampling scenario, the FNR *w.r.t.* images and text is 22.2% and 20.6%, respectively, remaining notably high. This result can serve as a proxy for the FNR on larger-scale datasets (*e.g.,* LAION400M [8]) since they share a similar dataset construction pipeline: *i.e.,* randomly sourced from the web. Therefore, these findings highlight the significance of addressing false negatives in recent large-scale models and the broad applicability of MAFA.

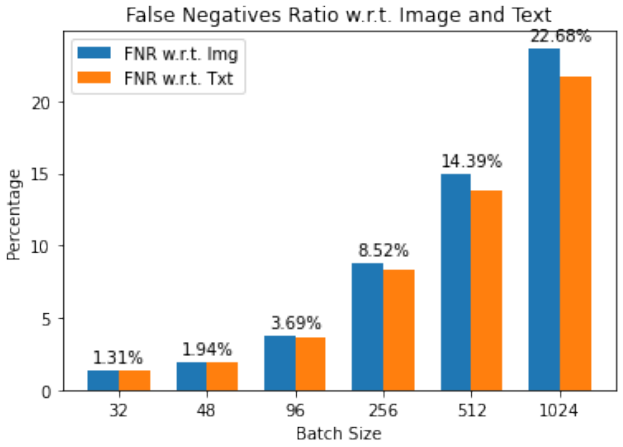

Figure 1. FNR on 4M-Clean dataset in random sampling scenario. Each annotated number is a mean of two ratios, w.r.t. image and text.

4. Detailed explanation on Table 2 (manuscript)

In Table 2 (manuscript), we present the shape of soft labels of three soft labeling methods: momentum distillation, consistency loss, and S-ITC. In this section, we explain the detailed process of constructing this table. The values in the table are obtained in the last epoch of the training with the following procedure.

1. For each anchor image i (resp. text t), a soft label is given to each text t_k (resp. image i_k) in a batch. For the batch-size B, k is from 1 to B. For momentum distillation with queue size Q, k is from 1 to $B+Q$. The set of these soft labels can be represented as a B (or $B+Q$)-dimensional vector, which is denoted as $\boldsymbol{y}_a$ for anchor a.
2. B (or $B+Q$) values of $\boldsymbol{y}_a$ are sorted in descending order, and it is denoted as $\boldsymbol{y}_a^{\text{sorted}}$.
3. Given that the model sees A anchors during an epoch of training, all $\boldsymbol{y}_a^{\text{sorted}}$s are averaged, and it is denoted as $\bar{\boldsymbol{y}}^{\text{sorted}} = \frac{1}{A}\Sigma_a \boldsymbol{y}_a^{\text{sorted}}$.
4. Now, B (or $B+Q$) values in $\bar{\boldsymbol{y}}^{\text{sorted}}$ are categorized into three groups: Top1 $\sim$ 5, Top6 $\sim$ B, and Top$B+1$ $\sim$ Q.
5. The sum of the values within each category is written in the table.

5. Pseudocode of MAFA

We provide pseudocode in Algorithm 1 for a thorough understanding of the MAFA framework. Although the pseudocode mostly follows the notation of the manuscript, there are some modified notations to explain the process in more detail. Specifically, $\tilde{y}^{\text{I2T-ITC}}$ and $\tilde{y}^{\text{T2I-ITC}}$ are used instead of $\tilde{\boldsymbol{y}}^{\text{ITC}}$. Also, in the manuscript, $\tilde{\boldsymbol{y}}^{\text{ITM}}$ is a 2-dim one-hot vector for an image-text sample. If the sample is treated as a positive in ITM loss, $\tilde{\boldsymbol{y}}^{\text{ITM}}$ is equal to $(1,0)$. If not, it is equal to $(0,1)$. On the other hand, in the pseudocode, $y_{\text{pos}}^{\text{ITM}}$ and $\tilde{y}_{\text{neg}}^{\text{I2T-ITM}}$ are B-dim vectors in which the b-th element represents whether the corresponding sample is treated as positive (1) or not (0) in ITM loss.

6. Details on Downstream tasks

In the Supplementary Materials, the term "total batch size" refers to the overall mini-batch size. Specifically, it represents the product of the "number of GPUs" and the "mini-batch size per GPU," which is calculated as $4 \times B$. We primarily adhere to the implementation details of GRIT-VLP when performing fine-tuning. During fine-tuning, we employ randomly cropped images with a resolution of 384 × 384. Conversely, during the inference stage, we resize the images without cropping. Additionally, we apply the exact same RandAugment, optimizer selection, cosine learning rate decay, and weight decay with GRIT-VLP. Following GRIT-VLP, we do not utilize a momentum encoder in the pre-training phase. Consequently, the momentum distillation (MD) technique is not employed for all downstream tasks,

[Image-Text Retrieval (IRTR)] IRTR aims to find the most similar image to a given text or text to a given image. Following GRIT-VLP, we do not use the momentum distillation for ITC, but use the queue and negatives from the momentum encoder when calculating ITC loss in the fine-tuning step. For model fine-tuning, we use the COCO and Flickr-30K datasets. Specifically, the COCO dataset, comprising 113,000 training images, 5,000 for validation, and another 5,000 for testing, is fine-tuned over 5 epochs. Conversely, the Flickr-30K dataset, with 29,000 training images, 1,000 for validation, and 1,000 for testing, undergoes a longer fine-tuning phase of 10 epochs. For evaluation, we use a 5K COCO test set and Flickr-1K set following previous works. During the fine-tuning phase, we use a total batch size of 256 and an initial learning rate of 1e-5 for both datasets. Following ALBEF, during evaluation, we employ a two-step process. First, we retrieve the top-k candidates by calculating image-text contrastive similarities only using uni-modal encoders. Then, we re-rank them with ITM scores. Here, k

Algorithm 1: MAFA

Inputs : Image-text paired dataset D, Initialized model f_0
Output : Trained model f

$f \leftarrow f_0$
Con-D $\leftarrow$ Train-Discriminator(D, f_0)
for $epoch = 1, 2, ..., E$ **do**
  /* 1.Similar samples are grouped by GRIT (Line 4) */
  $\tilde{D} \leftarrow$ GRIT(D, f)
  foreach $\{(I_b, T_b)\}_{b=1}^{B} \in \tilde{D}$ **do**
    $\{i_b\}_{b=1}^{B}, \{t_b\}_{b=1}^{B} \leftarrow f_{\text{encoder}}^{\text{image}}(\{I_b\}_{b=1}^{B}), f_{\text{encoder}}^{\text{text}}(\{T_b\}_{b=1}^{B})$
    $\tilde{y}^{\text{I2T-ITC}}, \tilde{y}^{\text{T2I-ITC}} \leftarrow \mathbf{Id}_B, \mathbf{Id}_B$ // $\mathbf{Id}_B$:$B \times B$ Identity matrix
    $y_{\text{pos}}^{\text{ITM}}, \tilde{y}_{\text{neg}}^{\text{I2T-ITM}}, \tilde{y}_{\text{neg}}^{\text{T2I-ITM}} \leftarrow \mathbf{1}_B, \mathbf{0}_B, \mathbf{0}_B$ // $\mathbf{1}_B$:B-dim one vector, $\mathbf{0}_B$:B-dim zero vector
    $D_{\text{ITM}}^{\text{pos}}, D_{\text{ITM}}^{\text{neg}}, D_{\text{MLM}} \leftarrow \{(i_b, t_b)\}_{b=1}^{B}, \{\}, \{(i_b, T_b)\}_{b=1}^{B}$
    for $b = 1, ..., B$ **do**
      /* 2.A hard negative is picked per anchor (Line 11) */
      $k \leftarrow \arg\max_{j \neq b} s(i_b, t_j)$
      /* 3.A new connection is established (Line 12-20) */
      $p_{\text{con}} \leftarrow$ *Con-D*(i_b, t_k)
      if $p_{con} > 0.8$ **then**
        $\tilde{y}_b^{\text{I2T-ITC}}[k] \leftarrow 1$ // $\tilde{y}_b^{\text{I2T-ITC}}$:$b$-th row of $\tilde{y}^{\text{I2T-ITC}}$
        $\tilde{y}_{\text{neg}}^{\text{I2T-ITM}}[b] \leftarrow 1$
        $D_{\text{MLM}} \leftarrow D_{\text{MLM}} \cup \{(i_b, T_k)\}$
      else if $0.5 < p_{con} < 0.8$ **then**
        $k \leftarrow \arg\max_{j \neq b,k} \text{sim}_f(i_b, t_j)$
      end
      $D_{\text{ITM}}^{\text{neg}} \leftarrow D_{\text{ITM}}^{\text{neg}} \cup \{(i_b, t_k)\}$
      $\tilde{y}_b^{\text{I2T-ITC}} \leftarrow \frac{\tilde{y}_b^{\text{I2T-ITC}}}{\Sigma_j \tilde{y}_b^{\text{I2T-ITC}}[j]}$
      /* (Line 11-22) are for an image anchor i_b. Similarly, execute for a text anchor t_b */
    end
    $L_{\text{S-ITC}}^{ECM} \leftarrow$ S-ITC-Loss($f, \tilde{y}^{\text{I2T-ITC}}, \tilde{y}^{\text{T2I-ITC}}, \{(i_b, t_b)\}_{b=1}^{B}$)
    $L_{\text{ITM}}^{ECM} \leftarrow$ ITM-Loss($f, y_{\text{pos}}^{\text{ITM}}, \tilde{y}_{\text{neg}}^{\text{I2T-ITM}}, \tilde{y}_{\text{neg}}^{\text{T2I-ITM}}, D_{\text{ITM}}^{\text{pos}} \cup D_{\text{ITM}}^{\text{neg}}$)
    $L_{\text{MLM}}^{ECM} \leftarrow$ MLM-Loss(f, D_{MLM})
    $f \leftarrow$ Backward-Update($f, L_{\text{S-ITC}}^{ECM} + L_{\text{MLM}}^{ECM} + L_{\text{ITM}}^{ECM}$)
  end
end
return f

is set to 256 and 128 for COCO and Flickr, respectively

[Visual Reasoning (NLVR2)] NLVR2 is a classification task based on one caption and two images. Since the model architecture should be changed to get two images as an input, the fine-tuning step of NLVR2 requires an additional pre-training phase with the 4M-Noisy dataset for 1 epoch. For this pre-training phase, we employ a batch size of 256 and set the learning rate to $2e-5$, and the image resolution is set as (256×256). After the single epoch pre-training phase, we fine-tune the model for 10 epochs while using a total batch size of 64.

[Visual Question Answering (VQA)] VQA is a task to obtain an answer given image and question pair. We perform experiments on the VQA2.0 dataset [4], which is divided into training, validation, and test sets with 83,000, 41,000, and 81,000, respectively. Both the training and validation set are utilized for training. Following GRIT-VLP and ALBEF, we also include additional pairs from Visual Genome. Fine-tuning is conducted for 8 epochs, employing a total batch size of 128 and an initial learning rate of $2e-5$. For a fair comparison, the decoder only generates answers from 3192 candidates.

Table 2. Comparison with state-of-the-art: fine-tuned results of IRTR on Flickr30K and MSCOCO datasets

Method	Pre-train # Images	MSCOCO (5K test set)						Flickr30K (1K test set)					
		TR			IR			TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
UNITER	4M	65.7	88.6	93.8	52.9	79.9	88.0	87.3	98.0	99.2	75.6	94.1	96.8
VILLA	4M	-	-	-	-	-	-	87.9	79.30		76.3	73.67	
OSCAR	4M	70.0	91.1	95.5	54.0	80.8	88.5	-	-	-	-	-	-
ALBEF	4M	73.1	91.4	96.0	56.8	81.5	89.2	94.3	99.4	99.8	82.8	96.7	98.4
TCL	4M	75.6	92.8	96.7	59.0	83.2	89.9	94.9	99.5	99.8	84	**96.7**	**98.5**
BLIP* (4M-Clean)	4M	76.5	93.2	96.8	58.9	83.1	89.6	94.3	99.4	99.9	82.6	96.2	98.3
GRIT-VLP*	4M	76.6	93.4	96.9	59.6	83.3	89.9	95.5	99.6	99.8	82.9	96.2	97.9
MAFA	4M	78.0	94.1	97.2	61.2	84.3	90.3	96.1	99.8	**100**	**84.9**	96.5	98.0
MAFA (4M-Clean)	4M	**79.4**	**94.4**	**97.5**	**61.6**	**84.5**	**90.4**	**96.2**	**99.9**	**100**	84.6	96.4	98.1
ALBEF	14M	77.6	94.3	97.2	60.7	84.3	90.5	95.9	99.8	100	85.6	97.5	98.9
BLIP	14M	80.6	95.2	97.6	63.1	85.3	91.1	96.6	99.8	100	87.2	97.5	98.8

Table 3. Ablation study on the proposed method: fine-tuned results of IRTR on Flickr30K and MSCOCO datasets.

Pre-train dataset	MAFA		MSCOCO (5K test set)						Flickr30K (1K test set)					
			TR			IR			TR			IR		
	S-ITC	ECM	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
4M-Noisy	✗	✗	76.6	93.4	96.9	59.6	83.3	89.9	95.5	99.6	99.8	82.9	96.2	97.9
	✗	✓	77.4	93.9	96.9	60.2	83.8	90.4	95.5	99.5	99.8	83.7	96.3	98.0
	✓	✗	77.5	94.3	**97.2**	60.5	83.7	90.2	96.1	99.8	99.9	84.2	96.3	**98.1**
	✓	✓	**78.0**	**94.1**	**97.2**	**61.2**	**84.3**	**90.3**	**96.1**	**99.8**	**100.0**	**84.9**	**96.5**	98.0
4M-Clean	✗	✗	77.7	93.5	96.9	60.7	83.3	90.1	95.2	99.6	99.9	84.2	95.8	98.0
	✓	✓	**79.4**	**94.4**	**97.5**	**61.6**	**84.5**	**90.4**	**96.2**	**99.9**	**100.0**	**84.6**	**96.4**	**98.1**

Table 4. Comparison of computational costs. MD represents momentum distillation.

Model	Time per epoch	Parameters	Queue for MD	Queue for GRIT
ALBEF	3h 10m	210M (MD: 210M)	65536	-
BLIP	3h 30m	252M (MD: 252M)	57600	-
GRIT-VLP	2h 30m	210M	-	48000
MAFA	3h 06m	210M (Con-D: 210M)	-	48000

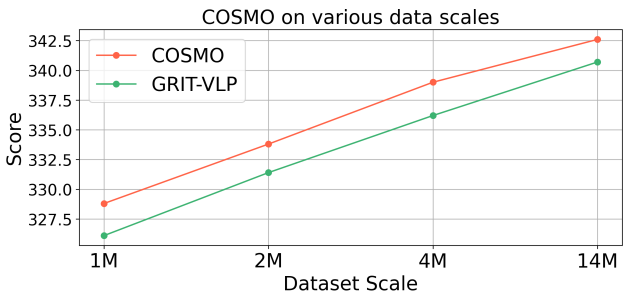

Figure 2. The score denotes the sum of the average results of four tasks (COCO-IRTR, Flickr-IRTR, NLVR2, and VQA).

7. Additional Experimental Results

7.1. Experiments on diverse data scales

We carry out additional experiments on various data scales under GRIT-sampling. For a large-scale dataset, we utilize the 14M-Clean dataset (CC12M-Clean + 4M-Clean), and for small-scale datasets, we use 1M and 2M subsets randomly selected from the 4M-Clean dataset. To compare the overall performance, we compute the accuracy of each of the four tasks (COCO-IRTR, Flickr-IRTR, NLVR2, VQA) by averaging their respective metrics: TR/R@1, TR/R@5, TR/R@10, IR/R@1, IR/R@5, IR/R@10 for IRTR, dev and test-P for NLVR2, and text-dev and test-std for VQA. Then, we sum up the four averaged values. In Figure 2, we observe that MAFA consistently achieves considerable performance improvements across all data scales compared to GRIT-VLP, again demonstrating the robustness of MAFA *w.r.t.* data scale variations.

7.2. Computation comparison

We describe the computational cost for pre-training. Table 4 shows the pre-training time per epoch, the number of parameters of the model, and queue size. Here, the "MD" denotes additional model parameters for the momentum model, and "Con-D" denotes additional model parameters for *Con-D* in the pre-training phase. As pointed out in the manuscript (Section 4), although MAFA becomes relatively slower than the GRIT-VLP, it is still competitive with ALBEF, and faster than BLIP.

7.3. Detailed experiments on overall framework

We present a comprehensive comparison of various baselines on image-text retrieval tasks in Table 2. We observe a simi-

lar tendency in the manuscript (Section 5). Namely, MAFA mostly surpasses other baselines in performance when pre-trained on the 4M-Noisy dataset, outperforming even ALBEF pre-trained on the much larger dataset (14M). Moreover, in the case of the 4M-Clean dataset, MAFA again shows a significant improvement over BLIP, which is pre-trained on the same dataset.

7.4. Analysis on search space

When applying GRIT sampling, the degree of similarity between samples within a batch is influenced by the search space since GRIT groups together similar samples within that specific search space. Consequently, an increase in the search space leads to a higher number of false negatives as shown in Figure 2 (manuscript). We observe that the average IRTR performance of GRIT-VLP is decreased as the search space increases, as shown in Table 5. In contrast, MAFA shows a performance improvement with larger search spaces and consistently outperforms GRIT-VLP regardless of the search space. We believe this result shows the effectiveness of MAFA in managing *false negatives*.

Table 5. Performance comparison of GRIT-VLP and MAFA on COCO using different search space sizes.

Method	M	MSCOCO (5K test set)						Avg
		TR			IR			
		R@1	R@5	R@10	R@1	R@5	R@10	
GRIT-VLP	960	76.7	93.9	97.2	59.6	83.5	90.1	83.5
	4800	76.6	93.4	96.9	59.6	83.3	89.9	83.3
	48000	77.0	93.1	96.6	59.2	83.1	89.8	83.1
MAFA	960	77.6	**94.4**	97.1	60.7	84.1	90.2	84.0
	4800	78.0	94.1	97.2	61.2	84.3	90.3	84.2
	48000	**78.3**	94.2	**97.3**	**61.4**	**84.4**	**90.5**	**84.4**

7.5. Detailed experiments on ECM

[Comparative analysis with ECM variants] We conduct a comparative analysis of different uses of new missing positives constructed from ECM. Namely, we assess the performance of models when new positives are exclusively incorporated in ITC, ITM, and MLM, respectively. As can be verified in Table 6, the usage of new positives is effective for each of these objectives. We believe the relatively marginal impact of new positives in ITC arises from the possible inclusion of additional false negatives in the mini-batch, which highlights the necessity of S-ITC. Moreover, their combined usage across three objectives leads to more improvements. We believe this result clearly shows the benefits of using new missing positives constructed from ECM, across all objectives. Note that all variants do not use S-ITC as objectives here.

[Robustness of MAFA for varying threshold τ] We evaluate the impact of choosing different thresholds τ for *Con-D*. Table 7 demonstrates that ECM process exhibits stability across different thresholds. Here, if τ is set to 0.5, the resampling strategy for ITM is omitted.

Table 6. Comparison of COCO performance with ECM-variants

ECM			MSCOCO (5K test set)						
			TR			IR			
ITC	ITM	MLM	R@1	R@5	R@10	R@1	R@5	R@10	Avg
✗	✗	✗	76.6	93.4	96.9	59.6	83.3	89.9	83.3
✓	✗	✗	77.1	93.5	96.8	59.7	83.4	90.1	83.4
✗	✓	✗	77.1	93.5	**97.3**	59.9	83.7	90.0	83.6
✗	✗	✓	**77.4**	93.7	97.0	59.9	83.2	89.9	83.5
✓	✓	✓	**77.4**	**93.9**	96.9	**60.2**	**83.8**	**90.4**	**83.8**

Table 7. Comparison of COCO performance using different threshold τ

τ	COCO R@1		NLVR2		VQA	
	IR	TR	dev	test-P	test-dev	test-std
0.5	**78.0**	**61.2**	82.06	**82.35**	**75.60**	**75.78**
0.8	**78.0**	**61.2**	**82.52**	82.08	75.55	75.77

[Comparison with oracle *Con-D*] We measure the performance of MAFA using different *Con-D*: one pre-trained on the 4M-Noisy dataset and the other obtained from BLIP (pre-trained with 129M dataset). In Table 8, we observe that MAFA exhibits only a little performance gap with the MAFA (Oracle). We believe this result shows that the *Con-D* constructed with a 4M-Noisy dataset is sufficient to reliably identify *false negatives* within that particular dataset. This aligns with the findings of BLIP, where it was shown that a filter trained with a noisy dataset of the same scale can effectively handle *false positives* within that specific dataset.

Table 8. Comparison with MAFA (Oracle) which uses the strong *Con-D* pre-trained with 129M data from BLIP

Dataset	Method	COCO R@1		Flickr R@1	
		TR	IR	TR	IR
4M-Noisy	GRIT	76.6	59.6	95.5	82.9
	MAFA	78.0	**61.2**	**96.1**	**84.9**
	MAFA (Oracle)	**78.5**	**61.2**	95.8	84.2

7.6. Additional analyses on ITC

[Effect of Queue in Momentum Distillation] In this section, we present additional experiments and analyses that delve deeper into the topics discussed in the manuscript. Given the significance of larger batch sizes in contrastive learning, it is standard practice to maintain a sufficient number of negative samples by employing a queue. To investigate the impact of the queue, we compare the performance of momentum distillation with a queue (MD) and without a queue (MD_{NQ}).

Table 9 shows the results of our observations regarding the effect of the queue in the GRIT sampling scenario. We observe that the queue has a detrimental effect on performance in this particular setting. This can be attributed to the substantial increase in the number of negatives, which consequently leads to significantly smaller labels assigned to negatives (compared to the case without a queue). The

impact of the queue highlights the crucial requirement of non-zero soft labels in the GRIT sampling scenario.

Table 9. Comparison of smoothing for ITC.

Method	Label smoothing	COCO R@1		Flickr R@1	
		TR	IR	TR	IR
ALBEF	X	**74.4**	57.6	**93.5**	81.7
	MD_{NQ}	73.8	**57.9**	93.3	81.5
	MD	74.2	57.4	**93.5**	**81.9**
	S-ITC	73.5	56.1	92.9	79.9
GRIT-VLP	CS	76.6	59.6	95.5	82.9
	MD_{NQ}	77.1	59.8	95.5	83.8
	MD	76.1	58.9	94.4	82.7
	S-ITC	**77.5**	**60.5**	**96.1**	**84.2**

[Effect of Mixing parameter α] As depicted in Table 2 (manuscript), the failure of MD and CS to provide soft labels can be attributed to different reasons. MD fails due to the large size of the queue (48000), while CS excessively concentrates only on a few similar samples. Adjusting the parameter α alone does not resolve this issue, as the underlying problem lies in the inclination of the model to favor the closest few samples. Moreover, naively increasing the value of α poses another challenge in training the model which amplifies the portion of uncertain labels of the momentum model (or the model itself) rather than ground-truth labels.

On the contrary, as shown in Table 10, the S-ITC method demonstrates robustness across a wide range of α values from 0.1 to 0.7. This highlights the distinct approach of S-ITC, which deviates from the inclination of the model towards the closest samples and effectively assigns non-zero labels to majorities of the negatives in the GRIT sampling scenario.

Table 10. Comparison of COCO performance using varying α of S-ITC in the GRIT sampling scenario, with a fixed training epoch of 10.

	MSCOCO (5K test set)						
	TR			IR			
α	R@1	R@5	R@10	R@1	R@5	R@10	Avg
0.1	75.5	**93.3**	**96.9**	58.4	82.7	89.3	**82.7**
0.3	**75.8**	93.0	96.4	58.8	82.7	89.4	**82.7**
0.5	75.6	92.9	96.6	**59.0**	**82.8**	**89.6**	**82.7**
0.7	75.3	93.1	96.7	58.3	82.6	89.3	82.6
0.9	73.0	92.1	96.2	56.4	81.3	88.6	81.3

7.7. Details on experiments with BLIP-2

BLIP-2 aims to propose an efficient vision-language pre-training framework that connects off-the-shelf frozen pre-trained image encoders and frozen large language models. To bridge the modality gap, BLIP-2 adopts a lightweight Querying Transformer (Q-Former), which is pre-trained in two stages. In *stage 1*, Q-Former is pre-trained to extract visual features relevant to text from a frozen image encoder with ITC, ITM, and (autoregressive) LM objectives. Thus, the training objectives are almost the same as those of ALBEF [5] and BLIP [6]. In *stage-2*, Q-Former is connected to a frozen LLM and pre-trained with LM loss to generate the text conditioned on the visual representation from Q-Former.

We mainly follow the default implementation setting of BLIP-2. Namely, we use ViT-G/14 from EVA-CLIP [3] as the vision encoder and OPT-2.7B [9] as a language decoder. Moreover, Q-Former is initialized with BERT-base [2] and has 32 learnable queries per single representation. For training and evaluation of BLIP-2, we use 8 A100 GPUs. For *stage 1*, we use the total batch size as 384 which is the same as the previous setting (in the manuscript). For *stage 2*, we use the total batch size as 512. For image-text retrieval fine-tuning, we use 112 as the total batch size. For other hyper-parameters, we use exactly the same as the default setting from BLIP-2 [7]. For GRIT sampling, we choose 3840 as search space, which is 10 times of the batch size. Since BLIP-2 uses 32 queries per image, we extract a single query representation that has a maximum similarity with its corresponding text and then use this representation to find similar examples for GRIT sampling.

For MAFA, to accelerate the efficiency of the experiment, we omit the re-sampling strategy in ITM and usage of additional positives, and we set α as 0.2 for S-ITC. Note that GRIT sampling and MAFA are only applied in *stage-1*. In both two stages, we use the 4M-Noisy dataset. Moreover, as mentioned in Section 5 (manuscript), since the exclusive use of GRIT sampling causes the failure of learning, we omit the results of "BLIP-2+GRIT" in *stage 2*. In addition, to adopt MAFA, we utilize the original BLIP-2 model for *Con-D*, which is pre-trained with the 4M-Noisy dataset and then fine-tuned with the COCO dataset. For zero-shot VQA tasks, following BLIP-2, we utilize the prompt "Question: Answer:" and beam search with beam width 5 and set length-penalty to -1 for all models.

7.8. Additional results with BLIP-2

[Results on fine-tuned image captioning] We fine-tune models on COCO with 5 epochs with a total batch size of 128. We use the prompt "a photo of" as the initial input for the LLM decoder (OPT-2.7B model) and train with autoregressive LM loss. For all other hyper-parameters, we use the exact same hyper-parameters as BLIP-2. In fine-tuning, the parameters of the Q-Former and image encoder are only updated while those of LLM are kept frozen. We evaluate models on both the Karpathy test split of MSCOCO and zero-shot transfer ability to NoCaps dataset [1]. The results, which can be observed in Table 11, indicate a similar trend in zero-shot ability. Namely, MAFA significantly enhances the captioning ability. Note that MAFA is exclusively used in *stage 1* and not used in *stage 2* and the fine-tuning stage.
[Results of *stage-2* with extra positives from ECM] To fur-

Table 11. Results on NoCaps and COCO Caption. All methods are fine-tuned. C: CIDEr, S: SPICE, B@4: BLEU@4.

Model	NoCaps Zero-shot (validation set)								COCO Fine-tuned Karpathy test	
	in-domain		near-domain		out-domain		overall			
	C	S	C	S	C	S	C	S	B@4	C
BLIP-2	109.55	14.46	105.83	14.26	106.33	13.51	106.47	14.14	40.7	139.7
BLIP-2+MAFA	**114.67**	**15.19**	**111.74**	**14.88**	**113.29**	**14.27**	**112.48**	**14.80**	**41.6**	**142.5**

Table 12. Effectiveness of extra positives from ECM (*stage 2*).

Model	VQAv2	OK-VQA	GQA	COCO zero-shot Karpathy test		Sum
	val	test	test-dev	BLEU@4	CIDEr	
4M	46.6	23.8	29.1	35.6	118.8	253.9
repeated-6M	**47.9**	24.9	30.8	35.8	118.3	257.7
ECM-6M	47.4	**27.0**	**30.9**	**37.5**	**123.6**	**266.4**

ther investigate the wider applicability of ECM in VLP models (w/o ITC and ITM), we conduct additional experiments on BLIP-2 *stage-2* model with extra positives generated by ECM. Here, in contrast to the results in Tables 7, 8 and 11 (manuscript), *stage-1* model is trained without MAFA. Namely, before training *stage-2 model*, by applying GRIT-sampling and using the frozen BLIP-2 stage-1 model as our *Con-D*, we generate 2M additional positives with a single forward pass, augmenting the original 4M-Noisy dataset. As reported in Table 12, we observe the *stage-2* model, which is trained on this "ECM-6M" (4M-Noisy + ECM-generated 2M) dataset, significantly outperforms the baseline trained on "repeated-6M" (4MNoisy + 2M sampled from the same 4M-Noisy) dataset. The scores are 266.4 vs. 257.7 in which the zero-shot results across VQA, OK-VQA, GQA, and COCO captioning tasks are summed. We believe this result validates the effectiveness of ECM-generated positives and underscores the general applicability of our framework.

8. Additional examples of new positive connections by ECM in training

We provide additional examples of new positive connections by ECM during training. Figures 3 and 4 show anchors ([Anchor]), their corresponding pairs ([Positive]), and new positives ([False Negative]) constructed by ECM during training. The number in parentheses indicates the ITM score between the anchor and the false negative computed by *Con-D*.

Figure 3. Examples of new positive connections constructed by ECM with respect to the texts during training.

[Positive]

[False Negative (ITM score)]

the cat in the stunted hat

cat in hat (0.8750)

shoots the ball against sports team

basketball player driving to the hoop in an one on game (0.9304)

ideas for bathroom tiles on the way to your dream bathroom

a very nice looking walk in shower with glass walls (0.9419)

a model walks the runway at the summer fashion show during event

a model showcases designs on the runway during show (0.8153)

3d rendered elegant black and gold clock face isolated on a white background

an image of a nice clock (0.8784)

Figure 4. Examples of new positive connections constructed by ECM with respect to the images during training.

References

[1] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object captioning at scale. In *ICCV*, 2019. 6

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 6

[3] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023. 6

[4] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA Matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 3

[5] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before Fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 6

[6] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 6

[7] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 6

[8] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 2

[9] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 6