Appendix

A. Experimental Details for Sec. 3

For the "VLM" mentioned in Sec. 3, we use the architecture designed in METER [8]. Similar to the ablation study in experiments, the model is pre-trained on the 4M dataset with an image-text matching loss and the (cross-modal) masked language modeling loss, but the input image is resized to 224×224 instead of 288 as in the normal setting of METER. We replace the image encoder CLIP-16 [26] with CLIP-32 for less memory usage and train the model for 100k steps. For "LM", we use roberta-base directly loaded from HuggingFace [34]. When training LM, the corpus come from all the sentences in the 4M dataset by excluding images. The model is finetuned with the masked language modeling loss. Other hyper-parameters are set to identical to that in "VLM". For evaluation, the accuracy is computed over the validation set of the COCO dataset [18].

Identifying Modality Bias. We train VLM and LM on the CMLM/MLM task simultaneously, that is, for each mini-batch in the dataloader (including images, sentences, masked tokens), the VLM takes the masked sentences and images but the LM takes only the masked sentences. The mask probability is set to 15%. We report the per-epoch accuracy.

Under-utilization of Unmasked Tokens.

This experiment does not involve any training as the pre-trained models of VLM and LM come from Sec. 3. The corruption of unmasked tokens are similar to the methods described in Sec. 4.3. Specifically, in the first round, we mask 15% of the tokens for the CMLM/MLM task. In the second round, we additionally randomly mask $\tau\%$ of the tokens from the tokens that are *NOT* masked in the first round. Then we use a pre-trained roberta-base model loaded from [34] to recover the tokens that are masked in the second round. Some tokens are successfully recovered by roberta-base, while some are not. The corruption ratio is an average of $\frac{\text{\# tokens are not recovered}}{\text{\# all the tokens in the sentence}}$ among all the sentence in the validation set. Finally, this corrupted masked sentence is fed into the VLM/LM for the CMLM/MLM inference. We record the CMLM/MLM accuracy over corrupted masked sentences and compute the relative performance drop compared to CMLM/MLM accuracy on the clean masked sentence. We gradually increase τ and record the resulting corruption ratio (i.e., x-axis) and relative performance drop (i.e., y-axis).

B. Complete Algorithm of EPIC

The algorithm for the EPIC model is shown in Algorithm 1, where $\tilde{f}_{\mathrm{VL}(\cdot)}$ denotes the pre-trained vision-language model to obtain $\boldsymbol{q}_k^V$ and $\boldsymbol{K}_k^W$.

Algorithm 1 EPIC

```
Input: Token sequence $\boldsymbol{w}$, raw image patches/regions $\boldsymbol{v}$, expected number of inconsistent tokens $m$;
Output: $\mathcal{L}_{\mathrm{ITC}}, \mathcal{L}_{\mathrm{GEN}}$
  // Obtain salient masking positions (Sec. 4.4)
  Obtain $\boldsymbol{q}_k^V$ and $\boldsymbol{K}_k^W$ from $\tilde{f}_{\mathrm{VL}}(\boldsymbol{w}, \boldsymbol{v})$;
  Compute $\boldsymbol{\alpha}$ based on Eq. (8) and sample $\mathcal{M}$ based on $m$;
  // Generate inconsistent samples (Sec. 4.3)
  Obtain $\boldsymbol{H}^L$ from $f_{\mathrm{L}}(\boldsymbol{w}^{\mathrm{mask}})$;
  Generate new sentence $\bar{\boldsymbol{w}}$ based on Eqs. (4) and (5);
  Obtain inconsistent tokens positions $\mathcal{T}$ via Eq. (6);
  Compute $\mathcal{L}_{\mathrm{GEN}}$ based on Eq. (7);
  // Compute $\mathcal{L}_{\mathrm{ITC}}$ (Sec. (4.2))
  Obtain $\left\{\bar{\boldsymbol{h}}_i^{\mathrm{VL}}\right\}_{i=1}^n$ from $f_{\mathrm{VL}}(\bar{\boldsymbol{w}}, \boldsymbol{v})$;
  Compute $\mathcal{L}_{\mathrm{ITC}}$ based on Eqs. (1) and (2);
return $\mathcal{L}_{\mathrm{GEN}}, \mathcal{L}_{\mathrm{ITC}}$
```

C. Details of Teacher VLM

In our implementation, we use the publicly-available checkpoints of the corresponding baselines as the teacher model. The teacher model is frozen and is only leveraged for saliency-based masking during pre-training.

D. Downstream Tasks

Image-Text Retrieval. Image-text retrieval includes two subtasks: (1) retrieving images for given text (Image Retrieval (IR)) and (2) retrieving text for given images (Text Retrieval (TR)). We conduct two different scenarios for evaluations: "zero-shot" (ZS) retrieval task and "after-finetuning" retrieval task. We conduct experiments on the MSCOCO [18] and Flickr30K [25] datasets. For evaluation, we use the recall at K (R@K) metric, which considers top-K predictions as candidates for correct predictions, and K is chosen from $\{1, 5, 10\}$. Note that METER, ViLT, and X-VLM directly conduct zero-shot inference with the pre-trained model, while ALBEF uses the model fine-tuned on the MSCOCO dataset for inference.

Visual Question Answering (VQA). VQA [1] requires the model to predict an answer given an image and a corresponding question. We conduct experiments on the VQA 2.0 dataset [9]. For ALBEF and X-VLM, we treat VQA as a language generation task; for METER and ViLT, we follow the common practice to convert the task into a classification problem. The final evaluation scores test-dev (dev) and test-std (std) are obtained from an official evaluation server.

Natural Language for Visual Reasoning (NLVR2). This task is to determine whether a natural language caption is true about a pair of photographs. We use the dataset from [31]. Following the practice of baselines, ALBEF and X-VLM conduct further pre-training on the extended pre-trained model and fine-tuned it on the dataset afterwards. METER and ViLT directly fine-tune the pre-trained model on the dataset.

Visual Entailment (VE). Visual entailment is a visual reasoning task to predict whether the relationship between an image and text is entailment, neutral or contradictory. Of the four baselines, only METER and ViLT conducted fine-tuning on this task, so we follow their procedure to treat this task as a three-way classification problem and report classification accuracy on the SNLI-VE dataset [35].

For all the downstream tasks, we follow original implementations and evaluations of baselines.

E. Pre-training Dataset

In Table 6, we provide the statistics of the 4M and 16M datasets. For details about the 4M$^+$ and 16M$^+$ dataset, please refer to [39].

	Dataset	# Images	# Captions
4M	COCO	0.11M	0.55M
	VG	0.10M	5.7M
	SBU	0.86M	0.86M
	CC3M	2.9M	2.9M
16M	4M	4.0M	10M
	CC12M	11.1M	11.M

Table 6. Statistics of the pre-training dataset.

F. Implementation Details of Baselines

For, X-VLM and ALBEF, their names refer to the architecture/method proposed in the corresponding paper. The implementation of X-VLM and ALBEF are identical to the official repository.

METER. We reproduced the pre-training of METER-CLIP-ViT$_{\text{BASE}}$. During pre-training, we additionally apply RandAugment to input images (we remove color changes from RandAugment because the text often contains color information, otherwise we may generate inconsistent tokens unintentionally) as we found that this technique is widely used by a lot of existing vision-language pre-training methods [12, 14, 39] and that it improves the generalization of the pre-trained model. Further, we found that using a tri-stage learning rate scheduler is beneficial for the performance of downstream tasks. Specifically, we linearly warm up the learning rate to its peak value in the first 10% steps, hold this value for the next 80% of the steps, and finally decay it exponentially to 1% of the peak value in the remaining steps.

ViLT. We tried to reproduce the pre-training of ViLT-B/32ⓐ⊕ [12], but we encountered "nan" in the middle of the training process. In response, we switched off half-precision (fp16) and continued the training with full-precision (fp32).

Note that the discrepancy in implementations between the original paper and ours does not pose unfairness in comparisons as all the results in Sec. 5 are obtained based on our reproductions.

G. Previous Checkpoints of VLM

At the beginning, the auxiliary VLM is the same as the newly-initialized VLM. After that, at the end of each k-th epoch, we save the current model's checkpoint and use it as the auxiliary VLM. Note that during pre-training, the auxiliary VLM is frozen. Therefore, the auxiliary VLM has the same architecture as the main VLM and is able to do the CMLM inference. Different from the LM as the generators for inconsistent tokens, the auxiliary VLM considers the visual inputs when trying to recover the masked tokens.

H. Ablations on different mask ratios

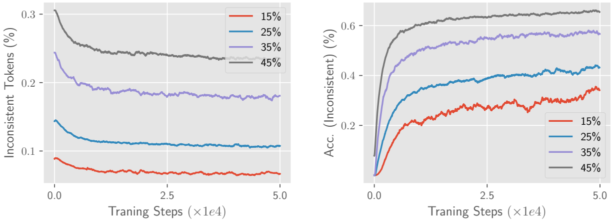
As shown in Table 7, when the mask ratio increases, more inconsistent tokens can be generated, which expedites the learning of vision-language associations. However, when too many tokens are masked, the meaning of the sentence can be completely different. As a result, the model simply predicts all tokens as inconsistent, harming the pre-training process. Therefore, the mask ratio is set to 35% as it brings the best (or nearly the best) results under all settings.

I. Analysis of ITC Task

Here we empirically analyze the ITC task. In Fig. 6a, we show the ratio of inconsistent samples. This ratio quickly drops in the first few epochs because the language model learns to fit to the text corpus. However, it converges in later stages as the language model is incapable of recovering the masked salient tokens, thereby producing inconsistent tokens for the ITC task. Fig. 6b shows that the inconsistent tokens are quite challenging for the VLM trained by EPIC to identify. Of all the inconsistent tokens, only about half (depending on mask rates) could be identified by the VLM because some image-token inconsistency can be subtle. Such challenge distinguishes ITC from CMLM in that ITC requires sophisticated cross-modal reasoning for each task while only language reasoning could help CMLM to achieve good performance as shown in Fig. 2a.

J. Integration to Baselines

To integrate our method with an existing VLP baseline, one only need to additionally load an auxiliary BERT-like language model and optimize a union of the original objectives and ours. For example, in the case of METER [8] that



(a) Ratio of Incons. tokens.

(b) ITC Acc. on Incons. tokens.

Figure 6. Left: The ratio of inconsistent samples (y-axis). Right: the accuracy of the ITC task over the inconsistent samples (y-axis). Colors of the curves indicate different mask rates.

Corrupt Rate	NLVR2	Flickr30K-ft		Flickr30K-zs		MSCOCO-ft	
	dev	TR1	IR1	TR1	IR1	TR1	IR1
15%	80.9	91.4	78.9	83.2	67.7	73.1	55.5
25%	**81.0**	92.0	**79.0**	83.8	71.5	73.5	55.4
35%	**81.0**	**92.9**	**79.0**	84.3	**72.5**	**74.1**	55.5
45%	**81.0**	91.6	**79.0**	**84.4**	71.6	73.0	**55.8**

Table 7. Ablations on different mask ratios.

is trained by $\mathcal{L}_{\text{ITM}}$ and $\mathcal{L}_{\text{MLM}}$, we now optimize the objective function for the integration of EPIC and METER as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ITM}} + \mathcal{L}_{\text{MLM}} + \lambda\mathcal{L}_{\text{ITC}} + \mathcal{L}_{\text{GEN}}, \qquad (9)$$

where λ is a hyperparameter to balance the ITC loss and losses in METER.

K. Improving CMLM using Saliency-based Masking

The proposed saliency-based masking can also be applied to CMLM to alleviate the modality bias problem. After we obtained the masking positions $\mathcal{M}$ as described in Sec. 4.4, we mask input sentences accordingly. Then we conduct CMLM on masked sentences. We term the improved CMLM as †CMLM. As shown in Table 8, when salient tokens are masked in CMLM, the pre-trained model demonstrates improved results on downstream tasks. This validates our claim in Sec. 3 that the modality biasproblem of prevents CMLM from learning sufficient vision-language associations. We also tried to replace the original CMLM in EPIC with †CMLM but we did not observe additional performance gain on downstream tasks. We surmise that the improvement of †CMLM overlaps with that of EPIC.

Method	NLVR2	Flickr30K-ft		Flickr30K-zs		MSCOCO-ft	
	dev	TR1	IR1	TR1	IR1	TR1	IR1
vanilla METER	79.6	89.2	76.6	83.2	67.7	71.0	52.5
†CMLM	**79.7**	**90.8**	**77.9**	**84.6**	**69.3**	**72.6**	**53.7**

Table 8. Replacing the CMLM in vanilla METER with †CMLM brings improvement to downstream tasks.