Direct Metric Optimization for Image Captioning through Reward-Weighted Augmented Data Utilization

Takumi Takada[1], Yuma Suzuki[1], Hiroki Takushima[1], Hayato Tanoue[1], Haruki Sato[1], Aiswariya Manoj Kumar[1], Hiroki Nishihara[1], Takayuki Hori[1], Kazuya Ueki[2]

[1]SoftBank Corp. [2]Meisei University

Abstract

While image captioning is an essential field of vision language models (VLM), a lack of continuity between the learning objective and final performance metrics of VLMs complicates their training and optimization. Reinforcement learning (RL) can directly optimize such metrics, but it is accompanied by a significant computational cost, making it difficult to apply to recent large-scale VLMs. In this paper, we propose *Direct Metric Optimization* (DMO), which is a lightweight final-metric-optimizing training method. We replace the computationally expensive exploration process in RL with an offline, diverse text data augmentation and show that self-supervised training on reward-weighted augmented data leads to direct and stable metric optimization. Our experiments demonstrate that DMO achieves performance comparable to those of the state-of-the-art RL method while saving hundreds of times more model forwarding iterations and greater amounts of computation time. This suggests that DMO constitutes a promising alternative for metric optimization in the era of large-scale VLMs.

1 Introduction

With the advent of CLIP (Radford et al., 2021), the boundaries between vision and language modalities in machine learning have been dissolved, leading to rapid advancements in research involving these areas. Furthermore, the rise of large language models (LLM) has led to the emergence of large-scale vision language models (VLM), extending their influence to practical applications. For example, models such as ChatGPT (Achiam et al., 2023) and Gemini (Team et al., 2023) generate detailed natural language descriptions from visual information. With the increasing prevalence of VLMs, methods for customizing and fine-tuning these models for specific domains or individuals are attracting significant interest and attention (Sun et al., 2023; Zhao et al., 2023).

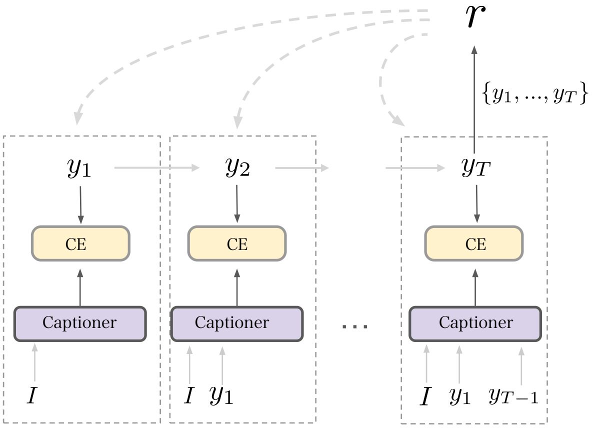


Figure 1: An overview of our *Direct Metric Optimization* (DMO). Image and tokens are denoted as I and y_i and CE stands for cross entropy function. Precomputed rewards r assign different weights to each sample in a textually augmented dataset, effectively enhancing the targeted performance metrics.

Recent standard captioning models adopt self-supervised learning for training purposes (Wang et al., 2022; Yu et al., 2022; Alayrac et al., 2022; Li et al., 2023). This method treats the ground truth captions both as inputs and labels, and the model predicts only the next token from the given image and preceding tokens. Specifically, recent transformer-based encoder-decoder models can conduct the next token prediction of each step in parallel, significantly enhancing their computational efficiency. However, this approach is subject to certain limitations. Typically, the performance of image captioning is evaluated using metrics such as BLEU (Papineni et al., 2002) or CIDEr (Vedantam et al., 2015); however, self-supervised learning of language modeling does not necessarily optimize those metrics. To target final metrics directly, reinforcement learning (RL) methods have been employed (Ranzato et al., 2015; Zhang et al., 2017b; Rennie et al., 2017; Gao et al., 2019). Reinforce-

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8333–8346
August 11-16, 2024 ©2024 Association for Computational Linguistics

ment learning is a powerful method capable of optimizing even non-differentiable metrics; however, it has certain drawbacks, such as learning instability and significant time and computational costs. With the growing trend of using large pre-trained models, those challenges have become increasingly serious. Conducting RL with models containing billions of parameters demands extensive computational time and resources, making the application of RL methods impractical.

To bypass the prohibitive computational cost of RL, we propose to replace the expensive exploration process in RL with diverse text data augmentation and reduce RL to simple importance-weighted self-supervised learning. The approach that utilizes previously collected data for RL is known as offline-RL (Levine et al., 2020). Particularly in our approach, datasets are augmented by various methods and the augmentation diversity brings a variety of samples of different rewards, enabling the efficient estimation of the optimal caption for the image. We call this metric-optimizing self-supervised training *Direct Metric Optimization* (DMO). Our experiments demonstrate that DMO achieves performance on par with state-of-the-art (SOTA) RL methods in standard image captioning metrics while retaining lightweight computational efficiency and learning stability. This highlights DMO's significant practical advantages in metric optimization, especially considering the increasing need to tune and customize large-scale VLMs.

2 Preliminaries

2.1 Self-Supervised Learning for Image Captioning

The standard approach for image captioning in recent years has been to employ an encoder-decoder model, where the encoder maps the image into the latent space and extracts features from the image, and the text decoder autoregressively generates tokens for the next step from extracted features and previously generated tokens. In the self-supervised learning of language models (LM), the model is often trained by teacher-forcing (Williams and Zipser, 1989). This method increases the likelihood of ground-truth sentences by aligning the model's conditional distribution $p_\theta(y_i|I, y_{<i})$ with the label distribution $q(y_i)$ using cross-entropy (CE) for each step $i \in \{1, \dots, T\}$. Here, y_i represents a text token at step i and I is the given image. The label distribution $q(y_i)$ is a one-hot vector or label-smoothed vector (Szegedy et al., 2016) from ground truth label y_i. The objective function of self-supervised learning for language modeling $\mathcal{L}_{\text{LM}}$ is expressed as follows:

$$\mathcal{L}_{\text{LM}} = \sum_{i}^{T} \text{CE}(q(y_i), p_\theta(y_i|I, y_{<i})). \qquad (1)$$

Especially in the recent Transformer-based architecture, the predictions of the next tokens at each time step can be performed in parallel (Vaswani et al., 2017). Thus this training method is extremely time and computationally efficient because it does not require recursive operations, as is the case with conventional RNN-based methods (Vinyals et al., 2015; Xu et al., 2015).

2.2 Reinforcement Learning for Image Captioning

Typically, the performance of captioning models is assessed using metrics such as BLEU or CIDEr. However, self-supervised learning does not necessarily optimize these metrics. Furthermore, these evaluation metrics are often non-differentiable, making it impossible to apply the gradient descent directly. One approach for directly optimizing those non-differentiable metrics is to employ RL. Recent studies of image captioning have applied the various RL algorithms including REINFORCE and Actor-Critic (Ranzato et al., 2015; Zhang et al., 2017b; Liu et al., 2017; Rennie et al., 2017; Zhang et al., 2021), to captioning tasks by regarding the captioning models as agents and the final evaluation metrics (such as CIDEr) as rewards. The objective function of the captioning model in the RL framework is expressed as follows:

$$\mathcal{L}_{\text{RL}} = -\mathbb{E}_{w_i \sim p_\theta}[r(\{w_1, \dots, w_T\})], \qquad (2)$$

where w_i is the token sampled from the model's distribution p_θ at the time step i and T is the total length of the token sequence. The partial derivatives of $\mathcal{L}_{\text{RL}}$ can be determined using the REINFORCE algorithm (Williams and Zipser, 1989). The calculation of the expected values is approximated by Monte Carlo sampling within a mini-batch as follows:

$$\nabla_\theta \mathcal{L}_{\text{RL}} \approx -r(\{w_1, \dots, w_T\}) \cdot \nabla_\theta \log p_\theta(\{w_1, \dots, w_T\}|I). \qquad (3)$$

In RL, the reward metric need not be confined to automatic evaluation scores. It can accept any user-defined score including ambiguous evaluations from humans (Christiano et al., 2017; Ouyang et al., 2022), thus demonstrating significant versatility and adaptability to various tasks.

Notwithstanding these advantages, the RL methodology presents significant challenges. First, for the method to be effective, extensive exploration is needed, which makes RL time-consuming. Second, the learning process tends to be unstable, especially during the early stage of training, when the model poorly samples the high-rewarded sequences. (Ranzato et al., 2015; Rennie et al., 2017). In addition, the sampling process in RL is computationally inefficient because the gradient computation requires the last token w_T (see Equation 3), but token generation is done in a left-to-right manner, undermining the computational parallelism of Transformer architecture.

2.3 Text Data Augmentation (TDA)

Data augmentation is a traditional yet effective method that is used to enhance a language model's performance (Li et al., 2023; Fan et al., 2023; Yang et al., 2023). Previous studies have shown that text data augmentation (TDA) strategies can be broadly categorized into three types: paraphrasing, noising and sampling (Li et al., 2022). Paraphrasing is a method that generates data that convey very similar information as the original data with restrained changes. Noising adds noise to datasets to improve the robustness of the model. Sampling produces a set of new data from the model that masters the distribution of the original data (Li et al., 2022). The primary objective of these strategies is to introduce diversity into the dataset. This is particularly crucial in scenarios with limited datasets, where models are prone to overfitting. Augmenting the dataset and smoothing the distribution can effectively prevent this overfitting.

In the fields of image and audio processing, data augmentation has traditionally achieved significant success (DeVries and Taylor, 2017; Zhang et al., 2017a). In contrast, it has not been explored as extensively in the field of natural language processing. This disparity can be attributed to the challenges inherent in text augmentation. Unlike images and audio, which comprise continuous data, tokenized text data is discrete and even minor alterations can lead to significant semantic shifts. Implementing superficial changes while controlling these semantic variations is not straightforward, and universally effective methods for achieving this are yet to be established (Feng et al., 2021).

3 Proposed Framework

3.1 Overview of Method

In RL, the sequences are sampled from the model's distribution p_θ, but this raises problems of its high computational cost and instability in the early stages of training. We propose the following perspective shift: What if we were to consider the sequences drawn from the given dataset as the sequences sampled from the model itself? This approach allows us to obtain the gradients of RL objective function with ground truth data in a self-supervised manner. Furthermore, since sequences are pre-sampled, the bottleneck in RL, specifically the recursive generation process, is resolved. This significantly enhances computational efficiency.

The approach that utilizes previously collected data for RL is known as "offline-RL" (Levine et al., 2020), and is commonly used to bypass the computationally expensive exploration in RL (Chen et al., 2021; Jang et al., 2022; Shi et al., 2023; Baheti et al., 2023). Applying the offline-RL to image captioning, however, is not straightforward. First, because of the nature of major captioning metrics (such as BLEU, METEOR, and ROUGE-L) that measure the overlap of n-grams, words or subsequences with a set of ground-truth captions, ground truth captions always receive rewards of 1 in the offline-RL framework. Because the rewards are indicators of the quality of samples, receiving a constant value of rewards gives no clue about how good each caption is, and consequently, there is no advantage compared with standard self-supervised training. The second problem is data suboptimality, a common challenge in offline-RL (Levine et al., 2020). The reliance on limited static data restricts exposure to high-reward samples, thereby capping the model's performance improvements. We address those obstacles by introducing diverse text data augmentation (TDA). With TDA, we expose models to a variety of expressions with different rewards outside the original dataset, providing a greater number of clues about the optimal caption for the images. Furthermore, substituting TDA for exploration improves the stability of the learning process in the early stage. This is because, unlike RL, TDA can consistently provide reasonable quality samples and training does not rely on the model's capability of sampling high-reward sequences. This metric-optimizing self-supervised training on textually augmented datasets, which we call *Direct Metric Optimization*, offers the following two significant advantages.

1. It allows direct optimization of metrics in a self-supervised manner, significantly enhancing computational efficiency.
2. Training is stable even at an early stage because it does not rely on the model's capability of generating captions of high rewards.

3.2 Direct Metric Optimization

In the proposed DMO method, sequences are sampled from textually augmented dataset D_{aug}. Because the dataset is known, scores for each ground truth sample can be calculated in advance. Let the score function (for example, the BLEU and CIDEr scorer) be the reward function $r(\cdot)$. Once ground truth data $d = \{y_1, \dots, y_T\}$ from dataset D_{aug} is sampled, the gradient of our DMO objective $\mathcal{L}_{\text{DMO}}$ is defined as follows:

$$\nabla_\theta \mathcal{L}_{\text{DMO}}(d) = -r(\{y_1, \dots, y_T\}) \cdot \nabla_\theta \log p_\theta(\{y_1, \dots, y_T\}|I) \quad (4)$$

$$= -r(d) \nabla_\theta \sum_i^T \log p_\theta(y_i|I, y_{<i}) \quad (5)$$

$$= r(d) \nabla_\theta \sum_i^T \text{CE}(q(y_i), p_\theta(y_i|I, y_{<i})) \quad (6)$$

$$= r(d) \nabla_\theta \mathcal{L}_{\text{LM}}(d), \quad (7)$$

where $q(y_i)$ is a one-hot vector from label y_i. Note that in offline reinforcement learning, importance sampling is commonly employed to bridge the gap between the distribution of the policy model and the offline training data (Precup, 2000). However, especially in the fine-tuning stage after the pre-training, the policy model's distribution is considered to be not significantly different from the distribution of the data; therefore the importance term is ignored in our setting. The resulting objective function (Eq. (7)) can be interpreted as a reward-weighted gradient of self-supervised learning loss. This approach eliminates the bottleneck inherent in RL, specifically the recursive generation process, through the utilization of pre-sampled sequences. Consequently, it enables the model to leverage the parallel computational capabilities of the Transformer architecture, resulting in a substantial enhancement of computational efficiency.

This reward-weighted self-supervised training on augmented datasets is related to noise/similarity-aware supervised training that adaptively assigns different weights to each sample (Atliha and Šešok, 2020; Yang et al., 2023; Kang et al., 2023). However, there are notable differences. First, while those noise-aware methods often focus on large-scale pre-training from noisy datasets and mitigate the effect of noisy samples, our approach features the finetuning stage with relatively small and clean datasets, and deliberately augments datasets to introduce the diversity of samples. Second, while similarity-aware methods often utilize CLIP/BERT scores or custom weights (Ding et al., 2019; Atliha and Šešok, 2020; Yang et al., 2023), we directly employ target metrics for sample weighting. While the CLIP/BERT score is useful for denoising or filtering, training with these measures does not directly lead to the optimization of the final metrics. With these perspectives, our method enables more effective optimization of the target metrics.

4 Experiment Implementations

This section describes the experiment implementations for the evaluation of our DMO training.

4.1 Datasets and Captioning Models

We validate our method with the MS-COCO dataset (Lin et al., 2014) and Flickr8K (Hodosh et al.), which are commonly used in image captioning research. Both datasets have 5 captions per image. For the evaluation, the datasets are split into training, validation, and testing sets according to the Karpathy method (Karpathy and Fei-Fei, 2015) so that the numbers of images in the training, validation and test datasets become 6091/1000/1000 for Flickr8k and 113287/5000/5000 for MS-COCO. For captioning models, we employ GIT-base/large (Wang et al., 2022) and BLIP2-2.7b (Li et al., 2023), which have different sizes of parameters and architectures. GIT has a simplified architecture of one image encoder and one text decoder and the base model has 178M parameters while the large model has 390M parameters for each. BLIP2-2.7B has 2.7B parameters and it employs large pre-trained frozen models for its vision encoder and text decoder. Both models are pre-trained on datasets that include COCO (Lin et al., 2014), Visual Genome (Krishna et al., 2017), CC3M (Sharma et al., 2018), and other large image-text pair datasets. In our experiments, all models are finetuned for 3 epochs with learning rate 1.0×10^{-5} and batch size 960, using a fixed single random seed. Further details are explained in Appendix A.1.

4.2 Text Data Augmentation Strategy

Based on Section 2.3, we adopt the following three augmentation methods accordingly. From each of the following three methods, two augmented captions are randomly sampled for each image and added to the original training dataset.

- **Back-translation**: The En-Fr translation model from MarianNMT (Junczys-Dowmunt et al., 2018) is adopted. Back-translation is applied to each ground truth caption and the same number of back-translated captions as original captions are created.
- **Pre-trained VLM Sampling**: Using the COCO-pre-trained BLIP2-6.7B model, five captions are generated from each image in the training dataset with temperature 1.0.
- **Paraphrasing by LLM**: We employ Llama2-7b-chat (Touvron et al., 2023) to paraphrase captions. The detailed prompt text is presented in Appendix B.

We do not explicitly adopt a noising strategy, as sufficient semantic noise is introduced by each TDA method.

4.3 Metrics and Rewards

We evaluate the performance of models by CIDEr, BLEU-4, METEOR (Banerjee and Lavie, 2005), ROUGE-L (Lin, 2004) and SPICE (Anderson et al., 2016). Since our method requires each sample to be scored by a reward function, we directly use the metrics above as the rewards in the training. As the scoring of each metric demands a set of ground truth captions as references, we employ the original training dataset as the reference dataset.

5 Results

We evaluate our proposed method in terms of metric optimization performance, learning stability and computational efficiency. We further investigate how reward-weighting architecture facilitates robust metric optimization by comparing DMO with standard LM training under noisy data and limited data settings.

Training	Evaluation Metrics			
Metric	CIDEr	B4	MET.	ROU.
CIDEr	**97.0**	33.3	**27.7**	**57.5**
BLEU-4	96.8	**33.5**	**27.7**	**57.5**
METEOR	96.1	32.8	27.5	57.0
ROUGE-L	96.0	33.0	27.6	**57.5**
standard LM	95.1	33.4	27.0	57.1

Table 1: Performance evaluation of GIT-base model optimized for each metric by DMO on the textually augmented Flickr8k dataset. ’B4’, ’MET.’ and ’ROU.’ refer to BLEU-4, METEOR and ROUGE-L, respectively.

5.1 Evaluating Metric Optimization Performance

5.1.1 Does DMO Enhance Final Metrics?

First, we examine whether our method effectively improves the targeted metrics for image captioning. We use a textually augmented Flickr8k dataset (TDA-Flickr8k) and apply DMO to the GIT-base so that each CIDEr, BLEU-4, METEOR, and ROUGE-L is optimized respectively. We then evaluate whether DMO improves these metrics compared to training with the standard Language Model (LM) loss. The result is presented in table 1. We find that when optimized for each metric, there is an improvement in each metric compared to training with the standard LM loss. This result implies that our method can effectively enhance the target metrics. Interestingly, optimizing for CIDEr or BLEU-4 leads to improved scores in other metrics as well. This can be attributed to the similarities in the way each evaluation metric is measured. In the following experiments, we use CIDEr as the target metric because CIDEr-optimizing DMO leads to general improvements in scores across other metrics.

5.1.2 Comparison of DMO and LM Training

We compare the performance of DMO training with LM training for different models and dataset settings. We use three models, GIT-base, GIT-large, and BLIP2-2.7b and two datasets, the Flickr8k and the COCO dataset. We apply CIDEr-optimizing DMO to each model and compare CIDEr, BLEU-4, METEOR, ROUGE-L, and SPICE scores with the models trained by standard LM loss. The results are presented in Table 2. We observe that DMO results in significant performance improvements in almost all models and datasets compared with models trained with LM loss without TDA. On

Captioning Model	Optimization Method	Flickr8k					MS-COCO				
		CIDEr	B4	MET.	ROU.	SPICE	CIDEr	B4	MET.	ROU.	SPICE
GIT-base	LM	95.1	33.4	27.0	57.1	21.4	135.6	41.0	30.5	60.4	23.6
GIT-base	LM w/ TDA	93.6	33.2	26.9	57.0	21.0	132.1	39.4	29.8	59.9	23.5
GIT-base	DMO	**99.6**	**35.4**	**27.9**	**58.1**	**22.3**	**137.4**	**41.5**	**30.5**	**60.9**	**24.0**
GIT-large	LM	96.3	33.3	26.9	56.9	21.2	**140.9**	**42.5**	**31.3**	61.3	**24.3**
GIT-large	LM w/ TDA	101.1	34.9	27.9	58.0	22.2	134.7	39.8	30.3	60.4	23.9
GIT-large	DMO	**110.7**	**37.6**	**29.1**	**60.2**	**23.2**	140.6	42.0	31.1	**61.5**	**24.3**
BLIP2-2.7b	LM	101.3	33.8	28.6	**58.5**	23.4	132.2	39.1	29.6	59.5	23.3
BLIP2-2.7b	LM w/ TDA	100.2	32.7	28.3	58.2	22.8	132.9	38.7	29.9	59.7	23.6
BLIP2-2.7b	DMO	**103.7**	**33.8**	**28.7**	58.4	**23.7**	**138.3**	**41.1**	**30.4**	**60.7**	**24.0**

Table 2: Evaluation of three models trained by standard LM training (with and without TDA) and CIDEr-optimizing DMO on Flickr8k and COCO datasets. The performance metrics include CIDEr, BLEU-4 (B4), METEOR (MET.), ROUGE-L (ROU.) and SPICE.

the Flickr8k dataset, all models exhibit score improvements across all metrics with DMO. A similar trend is observed when fine-tuning GIT-base on the COCO dataset. We observe that DMO remains effective even for larger models with up to 7 billion parameters (Appendix E). This suggests that DMO consistently enhances the scores beyond standard LM training across various models and datasets.

In the analysis comparing with LM training on TDA datasets, we observe that LM training on TDA datasets causes a decline in the performance in certain scenarios, such as training GIT-base/BLIP2 on Flickr8k and GIT-base/large on the COCO. This implies that the TDA datasets possess excessive noise and this noise leads to a deterioration in the performance of the models trained with standard LM loss. In contrast, DMO training, which utilizes the TDA dataset, exhibits rather enhanced performance. BLIP2 trained with DMO on Flickr8k and GIT-base DMO-trained on the COCO show improved scores for almost all metrics while those trained with LM loss show worse performance by introducing TDA. These findings indicate that our DMO can effectively leverage even noisy datasets that would deteriorate the performance of regular LM training. Moreover, when GIT-large is trained on TDA-COCO, a reduction in performance is observed for both LM and DMO training. However, the decline in performance is significantly different: 6.1 points for LM training compared with 0.3 points for DMO training, highlighting DMO's robustness under noisy dataset conditions.

Dataset Setting	Evaluation Metrics				
	CIDEr	B4	MET.	ROU.	SPICE
D_{bktrs}	93.6	32.8	27.3	56.9	22.1
D_{blip2}	98.2	33.0	**28.0**	57.9	22.1
D_{llama}	99.5	34.3	**28.0**	58.0	21.9
D_{all}	**99.6**	**35.4**	27.9	**58.1**	**22.3**
baseline	95.1	33.4	27.0	57.1	21.4

Table 3: Scores of GIT-base model trained with CIDEr-optimizing DMO on each dataset setting. Baseline is the score of LM training without any TDA. B4: BLEU-4, MET: METEOR, ROU: ROUGE-L.

5.1.3 Does TDA-Diversity Matter?

We hypothesize that diversifying the augmentation techniques serves as a replacement for exploration, enhancing the performance of DMO. To validate this hypothesis, we conduct an ablation study and evaluate the performance of models trained with DMO on datasets augmented by a single method and on datasets augmented by multiple methods, respectively. We denote the datasets augmented solely by the back-translation, pre-trained BLIP2 sampling, and Llama2 paraphrasing as D_{bktrs}, D_{blip2}, and D_{llama} respectively. For a fair comparison, each dataset is adjusted to have approximately the same number of image-caption pairs. For datasets $D_{\mathrm{bktrs}}, D_{\mathrm{blip2}}, D_{\mathrm{llama}}$, we increase 5 captions per image by augmentation. Note that the dataset D_{all} is constructed by sampling two augmented captions from each augmentation method and adding them to the original dataset. We use the Flickr8k dataset and train GIT-base by CIDEr-optimizing DMO. The results are presented in Table 3. While all data augmentation methods except for back-translation improve performance over the baseline, which is the score of LM training without any TDA, the highest performance is achieved with the dataset that combines all augmentation methods, suggesting that exposing the model to a variety of expressions from diverse aug-

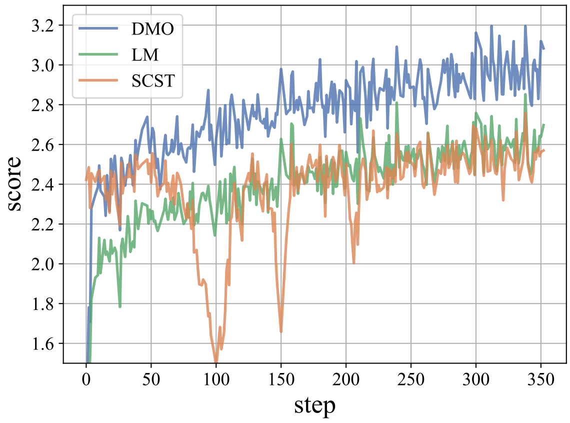


Figure 2: Transition of CIDEr scores for the GIT-base model trained using three different methods: DMO, standard LM training, and SCST. The scores reflect the CIDEr values of sequences greedily generated by each model for images in the mini-batches of the Flickr8k training dataset.

mentation techniques yields the most significant performance improvement. Further analysis of the advantages of combining multiple TDA methods is discussed in Appendix C.

5.2 Stability and Computational Efficiency

DMO replaces the exploration with TDA and resolves the learning instability and computational bottleneck of RL. We examine DMO's stability and efficiency by comparing them with those of RL and standard LM training. We use the GIT-base model trained on the Flickr8k dataset. As an RL method, we employ SCST (Rennie et al., 2017), which is one of the most prominent reinforcement learning methods for image captioning, utilized in training many SOTA models (Wang et al., 2022; Xu et al., 2023). We assess the stability of training by conducting experiments with five seeds and calculating the average and variance of the final scores at the 3rd and 20th epoch. To equalize the number of parameter updates, the number of explorations in RL is set to equal the number of image-text pairs in the training dataset. SCST requires intricate hyperparameter tuning and we present the detailed configuration for SCST training in Appendix A.2. The result is presented in Table 4. With 3-epoch training, our DMO produces the highest score. SCST is still unstable at 3 epochs, with a standard deviation (SD) of 4.1, which is significantly larger than LM's SD of 0.74 or DMO's SD of 1.26. Figure 2 shows the CIDEr score progression on training data up to 5 epochs for DMO, LM, and SCST, respectively. While DMO and LM show steady score improvements, SCST exhibits large fluctuations. This instability in SCST is attributed to its poor capability of sampling high-reward sequences in early training. On the other hand, DMO utilizes samples from TDA throughout the entire training, which makes training more stable.

Method	3 epochs	20 epochs
LM	94.4 ± 0.74	94.2 ± 1.65
SCST	93.6 ± 4.10	**99.8** ± 1.50
DMO	**97.7** ± 1.26	98.0 ± 1.36

Table 4: The average and standard deviation of CIDEr scores when the model was trained for 3 epochs and 20 epochs with five different seeds by each method.

Optimization Method	Forwarding iterations	Execution time
LM	1.68×10^3	13 sec
SCST	1.50×10^5	9971 sec
DMO	1.68×10^3	14 sec

Table 5: The number of model forwarding iterations and execution time for the GIT-base to complete 3 epochs on Flickr8k. The time is measured during the loss computation and the number of iterations is measured by counting the number of batch model forwarding.

By the end of the 20th epoch, SCST achieves the highest score, owing to SCST's ability to continually explore and obtain new samples. DMO displays minimal score improvements from epoch 3 to epoch 20, suggesting that training for only 3 epochs may suffice for model optimization in DMO training while SCST requires at least 20 epochs. This indicates that DMO optimizes the performance of the model more rapidly than SCST.

Additionally, we measure the number of model forwarding iterations and the training time required for each method to complete 3 epochs. To eliminate the impact of differences in implementation and hardware differences on timing, we specifically measure the duration between feeding the data to the model and obtaining the loss values. The result is presented in Table 5. With respect to computational efficiency, we find that LM and DMO have the same number of batch forwardings, while SCST requires approximately 100 times more (the rationale behind the results is explained in Ap-

Noise ratio	LM		DMO	
0%	95.1	(−0%)	**98.6**	(−0%)
20%	82.1	(−14%)	**97.9**	(−1%)
40%	69.6	(−27%)	**98.0**	(−1%)
60%	58.6	(−38%)	**90.9**	(−8%)
80%	5.1	(−95%)	**81.9**	(−17%)

Table 6: CIDEr scores of GIT-base model trained with noisy dataset. Noise ratio is the ratio of original ground-truth captions replaced by irrelevant random captions. The numbers in parentheses represent the percentage decrease from the score at the 0% noise ratio.

Dataset Size	LM		DMO	
100%	95.1	(−0%)	**98.6**	(−0%)
80%	92.1	(−3%)	**97.4**	(−1%)
60%	93.8	(−1%)	**97.6**	(−1%)
40%	90.4	(−5%)	**95.4**	(−3%)
20%	86.5	(−9%)	**95.0**	(−4%)

Table 7: CIDEr scores of GIT-base model trained with the limited number of data. Dataset size is the volume of the available training data. The numbers in parentheses represent the percentage decrease from the score at the 100% dataset volume.

pendix A.1). In terms of execution time, while DMO training takes approximately the same duration as LM training, SCST requires about 1000 times longer. The slight increase in time for DMO training compared to LM training is due to the necessity of reward-weighting operations. On the other hand, SCST requires the recursive generation process, resulting in a number of forwardings approximately 100 times greater and a duration approximately 1000 times longer, compared to LM and DMO training. These results emphasize the DMO's substantially greater computational efficiency compared to that of SCST.

5.3 Noise Robustness and Data Efficiency

We experimentally show that DMO can train models robustly even in data-noisy or low-resource settings, by effectively leveraging the reward. To evaluate the pure effect of reward utilization, we compare LM training and DMO training without TDA.

5.3.1 Evaluation on Extremely Noisy Dataset

In this experiment, we examine how robustly our method can train models on the noisy dataset. To simulate the case where the training dataset is extremely noisy, we construct datasets in which a certain percentage of the ground truth captions in the Flickr8k dataset are replaced with entirely irrelevant captions (we randomly sampled from the COCO dataset). With these datasets, we train GIT-base both with DMO and LM loss and observe how training is affected by those toxic samples. In DMO training, we use the original clean dataset for the reference dataset to ensure that the quality of each sample is accurately scored. We increase the noise ratio from 0% to 80% and evaluate the CIDEr scores of the model trained by LM and DMO. The results are presented in Table 6. Compared to the baseline, both LM and DMO training exhibit a decline in performance; however, while LM training experiences a significant performance drop, DMO manages to minimize this reduction. This indicates that, by utilizing scores as cues, our proposed method can effectively discern samples that should be learned from samples that should be ignored, enabling robust learning even from noisy datasets.

5.3.2 Evaluation under Low-Resource Setting

We simulate a scenario where training is constrained by limited data samples due to low computational resources, as is typical in edge device training that is aimed at minimizing time and battery consumption. We construct small datasets to evaluate how effectively data can be utilized under conditions of low resource availability. We reduce the amount of training data progressively from 20% to 80%. For the scoring in DMO training, we use the full original dataset as the reference dataset. The results are presented in Table 7. While LM training exhibits a 9% drop in scores as the data size decreases, DMO demonstrates robust learning even with limited data, exhibiting a smaller decrease of 4%. This result demonstrates that our method can efficiently learn even from a small number of data samples by leveraging the importance score of each sample.

6 Conclusion

In this paper, we present *Direct Metric Optimization* (DMO), which is a lightweight final-metric-optimizing training method. We hypothesize that diverse text augmentation can substitute the exploration in RL, and show that self-supervised training on reward-weighted augmented data leads to direct and stable metric optimization. Our experiments demonstrate that DMO can directly optimize evaluation metrics across models of various architectures and parameter sizes, and stably achieves performance comparable to the SOTA RL method while saving hundreds of times more model forwarding iterations and greater amounts of computation time. With these practical advantages of stable and lightweight cost of tuning, DMO emerges as a new promising choice for metric optimization in the era of large-scale VLMs.

Limitations

Although our experiments yield promising results, it is important to acknowledge the limitations of our method. The first limitation is the quality suboptimality of TDA. Our approach substitutes the exploration phase in RL with diverse data augmentation. However, in theory, data augmentation is distinct from exploration because it does not actively pursue higher rewards. Consequently, over extended training periods, RL methods, which consistently seek new and higher-quality samples, can outperform our DMO which relies on a fixed dataset. However, considering that RL methods for VLM are often very sensitive to hyperparameters and challenging to optimize, our DMO offers distinct practical advantages such as learning stability and the straightforward training process without the need for intricate hyperparameter tuning—benefits that are absent in most of RL approaches.

Another limitation is the data augmentation overhead. While DMO avoids the computationally expensive exploration process in RL, data augmentation still necessitates a certain computational cost. Therefore, considering the data preparation phase in addition to the training phase, the computational costs required for DMO increase and DMO's superiority over RL methods in terms of computational costs is diminished. However, a key distinction from exploration is that TDA can be conducted on separate machines (e.g., cloud servers) from the one the target VLM is deployed on. This aspect becomes particularly beneficial for model tuning in scenarios where resources such as time, memory, and battery of devices are constrained, as is typical in edge device training. Collecting augmented data on different servers enables models on the resource-constrained device to bypass the data augmentation overhead, making DMO a genuinely lightweight metric optimization method.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *ECCV*.

Viktar Atliha and Dmitrij Šešok. 2020. Text augmentation using bert for image captioning. *Applied Sciences*, 10(17):5978.

Ashutosh Baheti, Ximing Lu, Faeze Brahman, Ronan Le Bras, Maarten Sap, and Mark Riedl. 2023. Improving language models with advantage-based offline policy gradients. *arXiv preprint arXiv:2305.14718*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Guiguang Ding, Mingkui Chen, Sicheng Zhao, et al. 2019. Neural image caption generation with weighted training and reference. *Cognitive Computation*, 11:763–777.

Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. 2023. Improving clip training with language rewrites. In *NeurIPS*.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.

Junlong Gao, Shiqi Wang, Shanshe Wang, Siwei Ma, and Wen Gao. 2019. Self-critical n-step training for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6300–6308.

Micah Hodosh, Peter Young, and Julia Hockenmaier. Flickr8k dataset.

Youngsoo Jang, Jongmin Lee, and Kee-Eung Kim. 2022. Gpt-critic: Offline reinforcement learning for end-to-end task-oriented dialogue systems. In *10th International Conference on Learning Representations, ICLR 2022*. International Conference on Learning Representations, ICLR.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Wooyoung Kang, Jonghwan Mun, Sungjun Lee, and Byungseok Roh. 2023. Noise-aware learning from web-crawled image-text data for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2942–2952.

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

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Bohan Li, Yutai Hou, and Wanxiang Che. 2022. Data augmentation approaches in natural language processing: A survey. *AI Open*, 3:71–90.

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

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. 2017. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision*, pages 873–881.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Doina Precup. 2000. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80.

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.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024.

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

Ruizhe Shi, Yuyao Liu, Yanjie Ze, Simon S Du, and Huazhe Xu. 2023. Unleashing the power of pre-trained language models for offline reinforcement learning. *arXiv preprint arXiv:2310.20587*.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, et al. 2023. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint arXiv:2309.14525*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

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

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, et al. 2023. mplug-2: A modularized multi-modal foundation model across text, image and video. *arXiv preprint arXiv:2302.00402*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.

Kaicheng Yang, Jiankang Deng, Xiang An, Jiawei Li, Ziyong Feng, Jia Guo, Jing Yang, and Tongliang Liu. 2023. Alip: Adaptive language-image pre-training with synthetic caption. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2922–2931.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017a. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Le Zhang, Yanshuo Zhang, Xin Zhao, and Zexiao Zou. 2021. Image captioning via proximal policy optimization. *Image and Vision Computing*, 108:104126.

Li Zhang, Flood Sung, Feng Liu, Tao Xiang, Shaogang Gong, Yongxin Yang, and Timothy M Hospedales. 2017b. Actor-critic sequence training for image captioning. *arXiv preprint arXiv:1706.09601*.

Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. 2023. Beyond hallucinations: Enhancing lvlms through hallucination-aware direct preference optimization. *arXiv preprint arXiv:2311.16839*.

A Experiments Setting in Detail

A.1 Hardware Environment

We use two RTX A6000 Ada GPUs for GIT-base/large and four H100 GPUs for BLIP2-2.7B. Because each GPU has a different size of VRAM, we adopt the gradient accumulation method so that the batch size becomes 960 regardless of the size of the VRAM of the GPU that is used in each experiment. We present the hyperparameters for GIT-base/large and BLIP2-2.7B in Table 8. This configuration explains the number of model iterations in LM/DMO training presented in Table 5. The training uses 6091 images and 11 captions per image (including augmented captions) across 3 epochs. Dividing this by a mini-batch size of 60 and 2 GPUs results in approximately 1.68×10^3 iterations. For SCST, the maximum token length is set to 128 and the model recursively generates tokens up to this length, resulting in nearly 100 times more model forwarding iterations compared to LM/DMO training.

A.2 Configuration for SCST Training

Due to the instability of training with SCST, it necessitates pre-fine-tuning through standard self-supervised training (Rennie et al., 2017). Therefore,

Model	Learning Rate	Batch Size	Mini-Batch Size	Grad. Acc. Step	GPU
GIT-base	1.0×10^{-5}	960	60	8	RTX A6000 Ada × 2
GIT-large	1.0×10^{-5}	960	30	16	RTX A6000 Ada × 2
BLIP2-2.7B	1.0×10^{-5}	960	30	8	H100 × 4

Table 8: Hyperparameters for GIT-base/large and BLIP2-2.7B. The number of batch sizes is equal to the product of the mini-batch size, the number of gradient accumulation steps (Grad. Acc. Step) and the number of GPUs.

Rewrites the following image descriptions:

A young child splashes in a green and yellow wading pool. => A little boy is splashing around in the water in a kiddie pool.

A piece of banana, some strawberries, shish-kabobs, and a muffin are on a tray. => A tray of fruit and a muffin on a table.

Three people are sitting on a bus stop bench in between two tropical palm trees. => Three individuals are seated on a bench at a bus stop, flanked by two tropical palm trees.

<target captions> =>

Figure 3: The prompt that is used in Llama2-paraphrasing. The part ’<target caption>’ in the prompt text is replaced by the caption to be augmented.

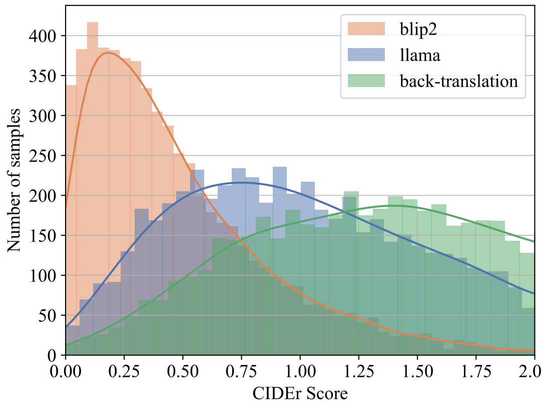


Figure 4: CIDEr score distributions of captions augmented by back-translation, BLIP2-sampling, and Llama2-paraphrasing.

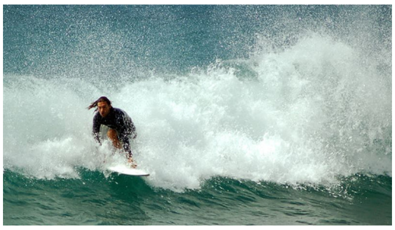
we initially fine-tune the model for three epochs with a learning rate of 1.0×10^{-5} before applying SCST. Given that fine-tuning has already been completed, we reduce the learning rate to 5.0×10^{-6} and only update the parameters of the text decoder to stabilize training. Moreover, during the sampling process in SCST, we opt for a temperature of 0.1. This is because we observe that higher temperatures, such as 0.5 or 1.0, often lead the model to generate random, meaningless sequences of words, which ultimately results in model collapse.

B Prompt for Llama2 Paraphrasing

For the Llama2-paraphrasing method, we employ the same prompting method proposed in La-CLIP (Fan et al., 2023). We present the prompt that is used in our experiments in Figure 3. The part ’<target captions>’ in the prompt text is replaced by the caption to be paraphrased. In the prompt, three examples of paraphrasing are provided. The first and second examples are constructed by regarding the two captions for the same image in each Flickr8K and COCO dataset as the captions before and after paraphrasing. The third paraphrasing example is made by feeding ChatGPT4 a caption from Flickr8k with the prompt "rewrite this image caption".

C Distributions Difference by TDA Method

In this section, we explore how the quality of samples varies across different TDA methods. We present the distribution of scores of samples generated by each TDA method in Figure 4. Scores are CIDEr scores based on the training dataset of Flickr8k. The back-translation tends to yield higher scores while BLIP2-sampling tends to produce samples of lower scores. Examples of generated captions by each TDA method are shown in Figure 5. We find that captions generated by the back-translation show little change compared to the ground truth (GT) captions. On the other hand, BLIP2 sampling generates captions that are significantly different from GT in terms of style and level of detail. Back-translation receives the GT captions to augment captions. Thus captions generated by



Ground truth
- a man surfing in the ocean.

Back-translation
- a man surfing the ocean.

BLIP2-sampling
- a woman wearing a wet suit riding a single white surfboard on a large wave in fresh ocean water.

Llama2-paraphrasing
- a man rides a wave in the ocean.

Figure 5: Examples of captions generated by each augmentation method, back-translation, BLIP2-sampling, and Llama2-paraphrasing.

Captioning	Optimization	Flickr8k				
Model	Method	CIDEr	BLEU4	METEOR	ROUGE_L	SPICE
LLaVA-v1.5-7b	LM	101.5	34.7	**28.9**	58.9	22.4
LLaVA-v1.5-7b	LM w/ TDA	98.0	32.8	27.6	57.9	20.7
LLaVA-v1.5-7b	DMO	**103.4**	**35.0**	28.7	**59.3**	**23.1**

Table 9: Performance evaluation of LLaVA-v1.5-7b model trained on Flickr8k dataset by LM training with and without TDA, and DMO.

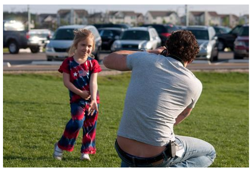
back-translation closely resemble GT captions. On the other hand, BLIP2-sampling generates captions solely from images. Therefore, captions generated by BLIP2-sampling often deviate from the GT captions and sometimes include incorrect descriptions (e.g., BLIP2 misidentified the person in the image as "woman" in Figure 5). Moreover, because the paraphrasing by Llama2 takes the GT captions as a prompt, the generated captions by Llama2 are semantically close to the GT captions. However, the changes from GT captions are greater than those of captions augmented by back-translation, owing to the prompt which encourages Llama2 to change expressions (e.g., 'surf' is paraphrased as 'ride a wave' in Figure 5). Interestingly, considering the result that DMO training on a dataset solely augmented by back-translation does not improve scores (shown in Table 3), TDA that often produces samples of high scores may not provide an advantage for DMO if it hardly alters the expression of the original GT data. Rather, TDA methods that produce samples of diverse expressions and structures can improve the performance of DMO by introducing various information that is not present in the original dataset. With the analysis above, we emphasize that the diversity of samples is important for DMO training and especially diversifying the augmentation techniques themselves is an effective approach because utilizing multiple augmentation methods generates a diverse set of samples across various distributions.

D Examples of Captions Generated by DMO, LM Training and SCST

Figure 6 shows examples of images and corresponding ground truth captions and captions generated by a GIT-base model trained by DMO, standard LM training and SCST. Models are trained for three epochs on the textually augmented Flicker8k dataset. Expressions which seem to be unique to the image are underlined. Those examples show that the model trained by DMO captures distinctive information and objects within the image and depicts them in the generated captions.

E Experiment with Larger Model

To verify whether DMO remains effective for larger models, we experiment with LLaVA-v1.5-7b (Liu et al., 2024). We train LLaVA by standard LM training with and without TDA, and by DMO on the Flickr8k dataset, for 6 epochs with learning rate 1.0×10^{-5}. We present the results in Table 9. The results show that DMO optimizes metrics of LLaVA better than standard LM training either with or without the TDA dataset, which is consistent with the result presented in Section 5.1.2.

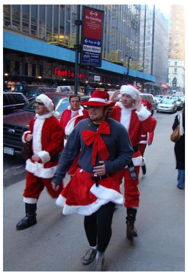


Ground truth: a girl wearing red and blue clothing poses for a man kneeling to take her picture.

DMO: a man is kneeling down to take a picture of a girl in a red and blue dress.

LM: a man and a girl are playing in a field.

SCST: a man and a little girl are playing in the grass.

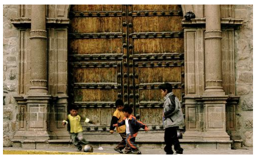


Ground truth: a parade of people is marching in santa costumes on a city street.

DMO: a group of people dressed in santa costumes walk down a city street.

LM: a group of people dressed in costumes walk down a street.

SCST: a group of men in red costumes are walking down the street.

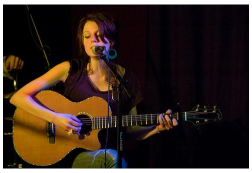


Ground truth: some children are playing soccer in front of a large wooden door.

DMO: three children are playing soccer in front of a large wooden door.

LM: three boys are playing soccer in front of a building.

SCST: three children are playing soccer in front of a building.

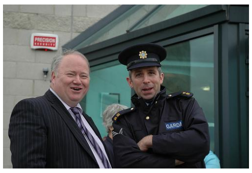


Ground truth: a woman sings into a microphone while playing guitar.

DMO: a woman is singing into a microphone while holding a guitar.

LM: a woman is singing a guitar.

SCST: a woman is playing a guitar.



Ground truth: a man and a police officer are smiling at the photographer.

DMO: a man in a suit and tie is standing next to a police officer.

LM: a man in a suit and tie is standing in front of a building with a man in a suit.

SCST: two men in uniform are smiling at the camera.

Figure 6: Examples of ground truth captions and captions generated by GIT-base model trained by DMO, standard LM training and SCST. Each model is trained on the textually augmented Flickr8k dataset for 3 epochs. Expressions which seem to be unique to the image are underlined.