Supplementary Material

A. Experimental Details

A.1. Implementation of Similarity Metrics

This subsection lists approaches for computing mage-to-image similarities and text-to-text similarities. In the image-text-image and text-image-text tasks, the evaluation metrics and distance metrics are closely related, i.e., the evaluation metrics for one task serve as distance metrics for the other task. Hence, we present them jointly in this section.

CLIP: The CLIP ViT-L/14 model[3] is pre-trained on a 400M text-image pair dataset. Specifically, we employ the outputs from the visual projection layer with an embedding size of 768 for the CLIP image encoder. For the CLIP text encoder, the outputs from the textual projection layer with an embedding size of 512 are used.

DINO: The pretrained DINO ViT-B/8 model[4] are used to obtain the image embeddings. DINO has been trained on the images dataset in a self-supervised way and has shown superior performance on representation learning tasks. The input image size is set to 384 and the output embedding size to 768.

FID, IS: Following [46], the torch-fidelity library[5] is used to compute the fidelity scores of the generated images.

SBERT: We use Sentence-BERT[6] with embedding size of 384 and take the [CLS] embeddings from the last transformer layer as the representation for the input text.

WMD: We use the open implementation of NLTK library[7] to compute WMD between a source sentence and a target sentence. Specifically, we use the *glove-wiki-gigaword* vector embeddings with 200 dimensions as the choice for the WMD. Since WMD is a distance metric rather than a score metric, we plot the y-axis in the reversed order to represent that the higher the y, the better the text, as shown in the second subfigure in Figure 3.

A.2. Details of Generation Models and Benchmarks

We use the NoCaps [1] validation set of 4500 images and a subset of 2000 images from the COCO Karpathy test split [27] to support the evaluation. For BLIP[8], we use the ViT-L model finetuned for the image captioning task. For SD[9], we use the weights sd-v1-4.ckpt and DDPM scheduler with sampling steps of 50 and a guidance scale of 7.5. The output image size is 512. The number of minimal words is set to 5 and the maximum number of words is set to 30. We set p=0.9 in the Top-p strategy.

[3]adapted from the public library https://huggingface.co/docs/transformers/modeldoc/clip

[4]adapted from https://github.com/facebookresearch/dino

[5]public implementation available from https://github.com/toshas/torch-fidelity

[6]public implementation from https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[7]https://www.nltk.org/

[8]https://github.com/salesforce/BLIP

[9]https://github.com/huggingface/diffusers

B. Impact of Parameter of Top-p Sampling

p	0.1	0.2	0.3	0.4	0.5
CIDEr	111.8	111.2	110.0	107.2	103.4
p	0.6	0.7	0.8	0.9	0.95
CIDEr	96.2	89.7	83.1	78.7	69.2

Table 6. Impact of p on the Top-p sampling performance of BLIP ViT-L.

Top-p sampling, which is also referred to as nucleus sampling, is a text generation method that samples words from a set of candidates whose cumulative probability exceeds a specified threshold p. By varying p, we achieve a trade-off between the diversity and accuracy of the generated text. Broadly speaking, larger values of p result in more diverse captions, whereas smaller values of p lead to less variable yet more accurate captions for an input image.

Note that in Table 2 of the BLIP paper [35], the Top-p sampling method is used to generate a diverse set of captions which are then utilized for bootstrapping the BLIP model. However, the evaluation result in that row is obtained by the beam search method. To examine the effect of p, we report the performance of BLIP ViT-L on the NoCaps dataset for image captioning, under different choices of p, in Table 6.

C. Influence of Sampling Methods for Image Captioning

Several sampling methods exist for generating text in the image captioning model. Apart from the Top-p sampling approach presented in the main text, we conduct a quantitative evaluation of two different sampling strategies, including Top-k [16] with k=10, and Tempered sampling [9, 39] with T=0.7. Each sampling method serves as a baseline. Table 7 verifies that our conclusion is solid across different sampling methods.

D. Impact of Number of Candidates N

We investigate the effect of the number of candidates N on our findings. We conduct experiments using the BLIP ViT-L model on the NoCaps dataset for image captioning, utilizing the Top-p sampling method. For image generation, we use the SD model and the DDPM scheduler. We sample N candidates for each input image or text in each experiment. We compare our approach to the baseline method, where a candidate is selected randomly.

Method	Nocaps								COCO			
	In-domain		Near-domain		Out-domain		Overall		Karpathy Test			
	CIDEr	SPICE	CIDEr	SPICE	CIDEr	SPICE	CIDEr	SPICE	B@3	B@4	CIDEr	SPICE
Top-k Sampling	79.5	13.2	78.4	12.4	83.4	11.7	79.6	12.4	39.7	28.1	108.4	20.9
Ours	**84.3**	**13.5**	**81.6**	**12.8**	**91.6**	**12.7**	**84.0**	**12.9**	**40.1**	**28.7**	**111.6**	**21.5**
Gain (%)	+6.0	+2.3	+4.1	+3.2	+9.8	+8.5	+5.5	+4.0	+1.1	+2.3	+2.9	+2.9
Tempered Sampling	83.9	13.2	82.7	12.7	89.8	12.0	84.3	12.6	33.0	22.2	92.4	19.5
Ours	**87.8**	**13.4**	**87.0**	**13.1**	**98.1**	**12.8**	**89.3**	**13.1**	**33.6**	**22.8**	**95.4**	**20.4**
Gain (%)	+4.6	+1.5	+5.2	+3.1	+9.2	+6.7	+5.9	+4.0	+1.8	+2.6	+3.2	+4.7

Table 7. Comparison of different sampling methods and our proposed method on Nocaps and COCO datasets. Our method outperforms every sampling method on all metrics. The relative gain of our method compared to each sampling method is given in the last row in each block. B@k: BLEU@k.

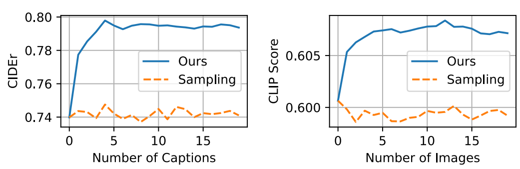

Figure 9. Evaluation of the choice of the number of sample candidates.

Figure 9 depicts the impact of the number of candidates on image-text-image (left) and text-image-text (right) tasks. The figure (left) shows the image captioning score (y-axis) for our approach and the baseline method, with the x-axis representing the number of candidate captions for each input image. As shown in the figure, the performance of the baseline method remains consistent as it randomly samples captions with varying qualities. Conversely, our approach improves significantly after the number of captions has reached around five. When the number of candidates is limited, there may not be enough high-quality captions to choose from, resulting in lower performance. Nevertheless, our approach remains effective even in that stage. After acquiring a reasonable number of candidates, our method consistently outperforms the baseline method by a significant margin. A similar conclusion can be inferred for the text-image-text task, as demonstrated on the right side of the figure.

E. Qualitative Results for Image and Text Generation

To reinforce the findings in Section 3, we provide additional qualitative examples on the NoCaps dataset. The annotations and explanations for each figure are included in their respective captions. Figure 10 shows both positive examples and negative examples for the image-text-image task, whereas Figure 11 presents visualizations for the text-image-text task. All results are obtained from BLIP ViT-L and SD models.

F. Analysis of Different Image-to-Text and Text-to-Image Generative Models

We conduct further experiments with different image captioning and text-to-image models. Table 8 shows the result of BLIP with a VAE-based image generative model LAFITE for the image-text-image task, as well as SD with BLIP-2 for the text-image-text task. In general, our finding, that better reconstruction leads to better generation performance, still holds. We find that SD performs better than LAFITE, very likely due to its larger training data. When coupled with BLIP-2, SD improved the performance upon the baseline, but it performed worse than that in the case of BLIP.

Text Generation					
I2T	T2I	I-C	N-C	O-C	E-C
Baseline		75.1	72.4	78.7	74.1
BLIP [35]	SD [46]	**77.3**	**78.3**	**88.8**	**80.3**
BLIP [35]	LAFITE [64]	75.4	76.2	82.9	77.4
Image Generation					
T2I	I2T	CLIP↓	FID↓	IS↑	
Baseline		40.54	32.37	41.19 $\pm$ 2.91	
SD [46]	BLIP [35]	**33.47**	**29.59**	**45.64** $\pm$ 2.40	
SD [46]	BLIP-2 [34]	39.41	31.34	42.34 $\pm$ 2.23	

Table 8. Comparison of different combinations of generation models. We evaluate two image captioning models and two image generation models on the NoCaps dataset. I2T: Image-to-text model. T2I: Text-to-image model. I-C/N-C/O-C/E-C: In-/Near-/Out-/Entire-domain CIDEr.

G. Implementations of Tokenizer Transformation

We elaborate on the gradient backpropagation process between the output of BLIP and the input of SD, shown on the left side of Figure 6. Our framework includes three types of text tokenizers: BLIP utilizes the word-piece tokenizing method; BLIP-2 uses the byte-pair-encoding tokenizing method; SD employs the CLIP tokenizer, which uses the word-level tokenizing strategy. One of the challenges we faced is to align the output token distributions from BLIP with those from SD, allowing SD to interpret the output of BLIP. To address this issue, we use these tokenization strategies to tokenize each sentence in the COCO training set and learn a one-to-one hard-coded mapping from a source token to a target token. Despite using different strategies, we found that these tokenizing methods have a high ratio of overlapping between tokens. Specifically, when applied to the COCO training set, more than 60% captions can be tokenized into the same set of tokens for BLIP and SD. For unmatched tokens, we map them to the most similar ones or to the [UNK] token. We visually examined this method by generating images conditioned on token distributions and found it to be practical. Additionally, we remove the prefix tokens of BLIP, *a photo of*, and add [BOS] and [EOS] tokens for SD.

H. Discussion on the Loss Function

Parameter Update While optimizing $\mathcal{L}_{IR}$ for image captioning, both SD and BLIP are trained. $\mathcal{L}_{TG}$ only updates the parameters of BLIP. Likewise, both SD and BLIP are trained when optimizing $\mathcal{L}_{TR}$, and only SD is updated by $\mathcal{L}_{IG}$. The reasons for training SD in $\mathcal{L}_{IR}$ are twofold. First, SD needs to adapt to new distributions coming from BLIP. As in the standard training, the input of BLIP is discrete tokens. Whereas in our approach, the input is token distributions. Second, SD could be improved because of additional training data sampled from BLIP. In addition, since they are refreshed at each iteration, one model is able to provide better samples to train the other model throughout the training.

Connection to CycleGAN CycleGAN is a generative model that learns bidirectional mappings from domain X to domain Y, where X and Y are images. Its cyclic loss ensures that the mapping between the input and output domains is consistent, i.e., if we take an image from domain X, pass it through the generator network to obtain an image in domain Y, and then pass that image through the generator network again to obtain an image in domain X, we should obtain an image that is similar to the original image in domain X. Likewise, we aim to enforce consistent mapping between the input and output domains, but we deal with two distinct domains, i.e., image and language, which may require much more complex mapping. In addition, we also optimize a single objective, similar to CycleGAN which is trained on the weighted cyclic loss and adversarial loss. CycleGAN employs a hyperparameter λ to control the relative importance of the cyclic loss to the adversarial loss, we found that a similar weighting did not yield substantial differences in performance in our work. Therefore we do not adjust this hyperparameter in our approach.

Pseudo-Code The pseudo-code of our train framework is presented in Algorithm 1.

Algorithm 1 Training Framework

Model UNet $\boldsymbol{\epsilon}_\psi$, SD's Text Encoder π, BLIP b_θ
Input an image-text pair $(\mathbf{x}_0, \mathbf{y})$

repeat
# Image-Text-Image (BLIP $\rightarrow$ SD)
$\mathbf{x}_0, \mathbf{y} \sim q(\mathbf{x}_0, \mathbf{y})$ ▷ Sample an image-text pair from the dataset
$\hat{\mathbf{y}} = b_\theta(\mathbf{x}_0, \tilde{\mathbf{y}})$ ▷ $\hat{\mathbf{y}} \in \mathbb{R}^{L \times V}$: Output token distribution. $\tilde{\mathbf{y}}$: (causal) masked text input
$\mathcal{L}_1 = \mathrm{CE}(\mathbf{y}, \hat{\mathbf{y}})$ ▷ CE: cross entropy loss
$\mathbf{c} = \pi(\hat{\mathbf{y}})$ ▷ Text encoder encodes BLIP's output into embeddings
$t \sim \mathrm{Uniform}(\{1, \cdots, T\})$ ▷ Sample a timestep for the diffusion process
$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample noise for timestep t
$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}$ ▷ Add noise to image
$\hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t, \mathbf{c})$ ▷ UNet predicts noise $\hat{\boldsymbol{\epsilon}}$ from the noisy image $\mathbf{x}_t$
$\mathcal{L}_2 = \|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}\|^2$
Text-Image-Text (SD $\rightarrow$ BLIP)
$\mathbf{x}_0, \mathbf{y} \sim q(\mathbf{x}_0, \mathbf{y})$ ▷ Sample an image-text pair from the dataset
$\mathbf{c} = \pi(\mathbf{y})$ ▷ Text encoder encodes input text into embeddings
$t \sim \mathrm{Uniform}(\{1, \cdots, T\})$ ▷ Sample a timestep for the diffusion process
$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample noise for timestep t
$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}$ ▷ Add noise to image
$\hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t, \mathbf{c})$ ▷ UNet predicts noise $\hat{\boldsymbol{\epsilon}}$ from the noisy image $\mathbf{x}_t$
$\mathcal{L}_3 = \|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}\|^2$
$\hat{\mathbf{x}}_0 = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \sqrt{1-\alpha_t}\hat{\boldsymbol{\epsilon}})$ ▷ 1-step approximation of $\mathbf{x}_0$
$\mathcal{L}_4 = \mathrm{CE}(\mathbf{y}, b_\theta(\hat{\mathbf{x}}_0, \tilde{\mathbf{y}}))$ ▷ $\tilde{\mathbf{y}}$: (causal) masked text input
Take gradient descent step on
BLIP: $\nabla_\theta(\mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 + \mathcal{L}_4) = \nabla_\theta(\mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_4)$
SD: $\nabla_\psi(\mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 + \mathcal{L}_4) = \nabla_\psi(\mathcal{L}_2 + \mathcal{L}_3 + \mathcal{L}_4)$
until converged
return ψ, θ

I. Ablation Study

Table 9 summarizes the different losses and weight-freezing strategies, highlighting that improvement comes from the proposed reconstruction loss. For simplicity, we only list the effect of different settings on BLIP. Our default setting jointly optimizes both pipelines with both models trainable $p_1 : \mathbf{I} \xrightarrow{\text{BLIP}} \mathbf{T}(L_{TG}) \xrightarrow{\text{SD}} \mathbf{I}(L_{IR})$, and $p_2 : \mathbf{T} \xrightarrow{\text{SD}} \mathbf{I}(L_{IG}) \xrightarrow{\text{BLIP}} \mathbf{T}(L_{TR})$. The third column in the table shows the training signals that BLIP can receive from the two pipelines under the specific setting. "-" means that loss has no effect as the model is frozen.

Weights Update				Loss Function				Effect on BLIP	Experiment	CIDEr
BLIP^{p_1}	SD^{p_1}	SD^{p_2}	BLIP^{p_2}	L_{TG}	L_{IR}	L_{IG}	L_{TR}			
✓	✓	✓	✓	✓	✓	✓	✓	p_1: ground truth + reconstruction p_2: augmentation	Ours, Tab. 4	111.8
✓	×	×	×	✓	×	-	-	p_1: ground truth	Tab. 4	109.7
✓	×	✓	×	✓	✓	✓	✓	p_1: ground truth + reconstruction	Ablation	110.9
×	✓	×	✓	-	✓	-	✓	p_2: augmentation	Ablation	110.2
✓	×	×	×	×	✓	-	-	p_1: reconstruction	Tab. 6	102.3

Table 9. Analysis of different training paradigms of loss terms and model frozen strategies.

J. Qualitative Results for Training Framework

We provide additional qualitative examples of image captioning and image generation by our trained framework in Figure 12 and Figure 13, respectively. Detailed annotations and explanations can be found in the corresponding figure captions.

Figure 10. Examples for the image-text-image task using Top-p sampling. The first column displays the input images, followed by the generated caption and its corresponding generated image. We rank the generated text-image pairs based on the similarity of the images and show the score of the caption below each text. In the first row, *golf car* in the first sample is a more accurate description than *cart* in the fifth sample so that the first generated image is closer to the input image. Additionally, we show a few failed examples in the last two rows.

Figure 11. Examples for the text-image-text task. The first column shows the input text, followed by generated image and its corresponding generated text. We rank the generated image-text pairs by the similarity of the text and show the score of the image in the box. As shown in the first line, the image in the first sample represents the input text better than the image in the fifth sample, and the similarity of the reconstructed text reflected this comparison. Further, we show some failed examples in the last two rows.

Figure 12. Qualitative results for image captioning. We compare the performance of our approach and BLIP ViT-B baseline. A ground truth (GT) caption is given for each image. On average, our method provides more accurate descriptions for input images. The last two rows show examples of our model performing worse than the baseline method.

Figure 13. Qualitative results for image generation. We compare the original SD model with our finetuned model. Each row displays firstly the input text, followed by four images generated by our approach, and four images generated by the SD baseline. We can see that compared with the baseline method, the image generated by our method better reflects the semantics of the input text.