The **Appendix** is structured as follows:

- Appendix A discusses our choice of evaluation metrics, specifically explaining why we did not use CLIPScore.
- Appendix B presents additional ablation studies, discussing the optimal self-attention layer for fine-tuning and the sensitivity of our method to the sampling guidance scale w.
- Appendix C provides the comparison between our method and an adapted training-free-based approach, which confirms the superiority of fine-tuning-based approaches to enhance text-image alignment.
- Appendix D offers the pseudocode of our method.
- Appendix E provides a deeper exploration, including comparison experiments with a better backbone model.
- Appendix F discusses the potential limitations of our work.

A. The Discussion upon Evaluation Metrics

CLIPScore [27] is widely used to evaluate text-image alignment by processing generated images and their corresponding guided texts through CLIP's vision and text encoders, respectively, and then computing the cosine similarity between their latent representations. However, we do not use this metric because the inherent bias in the CLIP model can distort the evaluation. As discussed in Section 4, CLIP's text module produces biased representations when handling correlated tokens, which contributes to the pre-trained model's failure in generating accurate images. Consequently, using CLIPScore can introduce bias and lead to inaccurate judgments.

(a) a sample generated by the pre-trained model with poor text-image alignment

(b) a sample generated by the pre-trained model with good text-image alignment

(c) a sample generated by our method with good text-image alignment

Figure 9. The comparison of several generated images aligning the provided text differently. Figure 9a shows a clear object omission problem, yet it achieves the highest CLIPScore as reported in Table 6, highlighting the discrepancy between CLIPScore and human-perceived text-image alignment.

Image	CLIPScore	BERTScore	Human Evaluation
Figure 9a	0.2891	0.7823	Worst
Figure 9b	0.2495	0.8035	Middle
Figure 9c	0.2157	0.8076	Best

Table 6. Text-image alignment scores using different evaluation criteria for images in Figure 9. A higher score indicates better alignment for both CLIPScore and BERTScore.

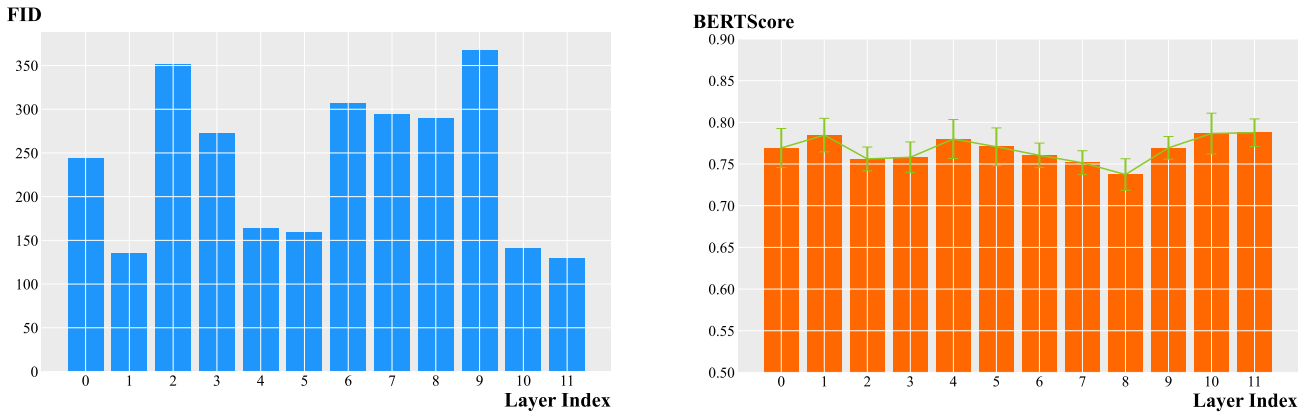
To illustrate this issue, Figure 9 and Table 6 present typical examples. The conditional prompt is *"rusty tracks with an old train"*. We selected two images generated by the pre-trained model (Figures 9a and 9b) and one image generated by our method (Figure 9c). Notably, Figure 9a fails to depict *"train"*, demonstrating an explicit object omission problem, while Figure 9b—generated with a different seed—does show *"train"* in the upper right corner. Figure 9c exhibits the best text-image alignment by successfully presenting both *"train"* and *"tracks"*. Yet, Table 6 shows that the worst image (Figure 9a) achieves the highest CLIPScore. This is because CLIP, when processing correlated tokens, favors biased representations and, as a result, assigns higher similarity scores to biased images that align well with its internal representations but not with human judgment.

In summary, due to its limited capacity, CLIP is more biased than some large language or vision-language models, making CLIPScore unsuitable for measuring text-image alignment in scenarios with potential token-level correlations. Instead, our two evaluation measures—incorporating either a large vision-language model or human feedback—provide more reliable assessments.

B. Further Ablation Study

B.1. Layer Selection

The ablation studies in Section 6.3 demonstrate how our sampling guidance and the hyper-parameter τ affect the final generated images. In addition to these functional components, we conducted an experiment to determine which self-attention layer in the text module is most effective for our fine-tuning process. Using the prompt *"a peaceful trail stops at a trash bin"* (the same as in Section 6.3), the results are presented in Figure 10. We observed that fine-tuning the last (11th) layer of the text module yields the best performance across both evaluation metrics, whereas fine-tuning middle layers (layers 6 to 9) does not significantly enhance text-image alignment.



(a) Variation of FID by fine-tuning different self-attention layers.

(b) Variation of BERTScore by fine-tuning different self-attention layers.

Figure 10. Ablation study on which layer to apply our method.

B.2. The Effect of Guidance Scale

We still use the conditional prompt *"a peaceful trail stops at a trash bin"* to investigate our method's sensitivity to the sampling guidance scale w. We set the default value at 7.5, as recommended for pre-trained Stable Diffusion. Table 7 presents the results. Our findings indicate that the guidance scale affects alignment within a certain range. In particular, when w is set too low (*e.g.*, 3.5), Eq. 6 becomes ineffective because the subtraction operation does not sufficiently steer samples toward an improved direction. In addition, although setting w at 7.5 may not be optimal for both evaluation metrics simultaneously, our method consistently outperforms baseline approaches in terms of both BERTScore and FID, provided that w is not set too low. This further confirms the robustness of our approach.

	Ours					Baselines		
w	3.5	5.5	7.5 (Default)	9.5	11.5	Pre-trained Model	PAG	DisenDiff
BERTScore ↑	77.30	77.89	78.76	78.94	78.73	77.44	76.61	77.06
FID ↓	162.37	146.26	129.59	137.01	132.97	244.43	228.81	288.84

Table 7. The ablation study on guidance scale w. We omit % for BERTScore and attach the corresponding performances of baseline methods in the last three columns.

C. Comparison with Modifying Attention Maps by Training-Free Approaches

In Section 5.1, we propose an approach that performs token-level decorrelation by fine-tuning the attention maps—specifically, by lowering the attention probabilities of selected entries. One might ask why we do not simply apply training-free techniques to modify the attention maps. To address this, we conducted an experiment using ACT [40], a training-free method originally designed to avoid attention sink by reducing the attention score of specified entries. As shown in Table 8, our method outperforms the training-free modification in almost every layer of the text module. We attribute the superiority of our approach to its ability to preserve interactions among unfocused tokens, as discussed in Section 5.2. Our method uses the pre-trained attention map as a reference and only applies decorrelation to the focused entries, thus minimally disturbing the original dependencies among irrelevant tokens. In contrast, training-free methods face the critical challenge of reducing extra attention probabilities caused by token-level correlations and reallocating them appropriately. Techniques like ACT, which assign constant values to attention scores, cannot effectively capture the complex dependencies across inputs. Therefore, we adopt a fine-tuning-based paradigm to achieve better text-image alignment.

Modified Layer		0	1	2	3	4	5	6	7	8	9	10	11
BERTScore ↑	ACT	73.24	74.51	74.28	74.86	75.38	75.53	75.05	75.47	76.96	76.44	77.89	75.59
	Ours	76.92	78.47	75.61	75.81	78.00	77.07	76.04	75.15	73.74	76.94	78.66	78.76
FID ↓	ACT	364.37	397.24	451.01	292.19	261.51	263.01	241.81	247.61	228.66	242.46	178.35	291.11
	Ours	243.81	135.85	351.79	272.55	163.56	159.66	307.10	293.99	290.16	368.10	141.21	129.59

Table 8. The comparison between our method and the training-free-based modification approach. We omit % for BERTScore.

D. Pseudocode

Algorithm 1 The pseudocode of our fine-tuning based method.

Input: The token list TOK which is generated from tokenizing the conditional prompt, the index of token t_{target} which represents the missing object, the original pre-trained Stable Diffusion model θ_0
Parameters to be optimized: the 11-th self-attention block in the text module of a Stable Diffusion model θ
Output: A decorrelated model $\widetilde{\theta}$

// Fine-tuning Stage
Initialize $\widetilde{\theta}$, θ with θ_0
Obtain A_{ref}, the self-attention map of pre-trained model θ_0, as reference
Find the token (except BOS token) with the highest attention value on TOK$[t_{\text{target}}]$, record its index as t_{corr}
while the training epoch has not reached its end **do**
  Obtain the attention map of θ, record it as A
  Use Eq. 2 to fine-tune θ, then obtain its new attention map $\widetilde{\mathrm{A}}$
  if Eq. 5 computed with $\widetilde{\mathrm{A}}$ from θ is smaller than that from $\widetilde{\theta}$ **then**
    Update $\widetilde{\theta}$ with current θ
  end if
end while
// Sampling Stage
Estimate the conditional noise with Eq. 6, conduct denoising operations until finishing reconstructing the sample

E. Deeper Exploration

We have also deployed our method on SDXL [25] (with $\sim 3\times$ the size of Stable Diffusion V1.5), which performs better than Stable Diffusion V1.5. The results are listed in Table 9. We could observe that using a newer and larger pretrained model improves text-image alignment to some extent. However, Figure 11's samples reveal that SDXL still suffers from this problem, especially those in the red box. Thus, solely relying on model size is not a definitive solution, as correlation is inherent in pre-training data. Instead, decorrelation operations like ours directly face the core issue and offer a more efficient approach, as confirmed by the performance improvement when applying our method to SDXL in Table 9.

As discussed in the main text, the limitations of the text encoder in Stable Diffusion are a major source of the misalignment issue. This problem may be largely mitigated in models with stronger text understanding, *e.g.*, Stable Diffusion 3 [9] or FLUX [21]. However, we emphasize that while enhancing the text encoder or scaling its size can improve generation quality, it does not address the core issue—namely, the inherent bias in the training data. Without access to higher-quality or more balanced data, model-level improvements alone are insufficient. In contrast, our method provides a more efficient and practical solution to reduce misalignment.

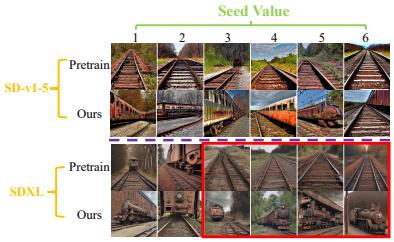


Figure 11. The images generated by different models under the same fixed seeds.

Model	Methods	Object Pair 1		Object Pair 2		Object Pair 3	
		BERTScore ↑	FID ↓	BERTScore ↑	FID ↓	BERTScore ↑	FID ↓
SDv1-5	Pre-train	76.99 ±1.23	200.40	78.67 ±1.61	300.74	75.14 ±2.18	261.28
	Ours	**77.68** ±1.07	**137.29**	**79.68** ±1.49	**232.42**	**75.92** ±2.14	170.79
SDXL	Pre-train	**77.84** ±1.02	164.33	79.24 ±1.02	218.45	76.08 ±3.99	167.63
	Ours	77.57 ±1.91	**143.83**	**80.34** ±0.94	**212.41**	**77.70** ±2.72	**149.41**

Table 9. The experiments conducted on SDXL.

F. Potential Limitations and Further Discussions

In practical scenarios, leveraging more advanced or larger models can partially alleviate the misalignment issue. Such improvements generally fall into two categories: (1) adopting alternative text encoders such as BERT [7] or T5 [28]; and (2) using stronger base models like FLUX [21] or Stable Diffusion 3 [9]. These approaches often lead to a higher proportion of high-quality generations, potentially reducing the necessity for our fine-tuning method.

However, replacing the text encoder is non-trivial due to architectural constraints. For example, the Stable Diffusion series predominantly relies on CLIP, with some recent versions incorporating T5. In contrast, BERT is not widely adopted in this pipeline. Although BERT may offer improved text understanding, integrating it would likely require retraining the entire base model from scratch, which is computationally expensive. Our method, by comparison, is compatible with most mainstream Stable Diffusion variants and requires modifying only 2.66% of the trainable parameters in Stable Diffusion v1.5, offering a more efficient and accessible alternative.

Moreover, while our study focuses on correlation-induced misalignment within the Stable Diffusion framework, we believe that the identified issue—and our proposed mitigation strategy—may extend to other generative architectures. Exploring this broader generalization is left for future work.