mathematics

MDPI

Article

LoRA Fusion: Enhancing Image Generation

Dooho Choi †, Jeonghyeon Im † and Yunsick Sung *

Department of Computer Science and Artificial Intelligence, Dongguk University-Seoul, Seoul 04620, Republic of Korea; likeb789@dgu.ac.kr (D.C.); jounghyunjet@dgu.ac.kr (J.I.)
* Correspondence: sung@dongguk.edu
† These authors contributed equally to this work.

Abstract: Recent advancements in low-rank adaptation (LoRA) have shown its effectiveness in fine-tuning diffusion models for generating images tailored to new downstream tasks. Research on integrating multiple LoRA modules to accommodate new tasks has also gained traction. One emerging approach constructs several LoRA modules, but more than three typically decrease the generation performance of pre-trained models. The mixture-of-experts model solves the performance issue, but LoRA modules are not combined using text prompts; hence, generating images by combining LoRA modules does not dynamically reflect the user's desired requirements. This paper proposes a LoRA fusion method that applies an attention mechanism to effectively capture the user's text-prompting intent. This method computes the cosine similarity between predefined keys and queries and uses the weighted sum of the corresponding values to generate task-specific LoRA modules without the need for retraining. This method ensures stability when merging multiple LoRA modules and performs comparably to fully retrained LoRA models. The technique offers a more efficient and scalable solution for domain adaptation in large language models, effectively maintaining stability and performance as it adapts to new tasks. In the experiments, the proposed method outperformed existing methods in text–image alignment and image similarity. Specifically, the proposed method achieved a text–image alignment score of 0.744, surpassing an SVDiff score of 0.724, and a normalized linear arithmetic composition score of 0.698. Moreover, the proposed method generates superior semantically accurate and visually coherent images.

Keywords: low-rank adaptation (LoRA); image generation; merging LoRA modules

MSC: 68T01

Citation: Choi, D.; Im, J.; Sung, Y. LoRA Fusion: Enhancing Image Generation. *Mathematics* **2024**, *12*, 3474. https://doi.org/10.3390/math12223474

Academic Editors: Xingyu Li, Sihan Huang, Baicun Wang and Xi Gu

Received: 4 October 2024
Revised: 1 November 2024
Accepted: 5 November 2024
Published: 7 November 2024

1. Introduction

Recent advancements at the intersection of computer vision and natural language processing have advanced the field of controlled image generation [1–6], creating images that precisely interpret and visualize user-specified textual prompts. Despite these advancements, a challenge persists: generated images often fail to fully align with the intentions of users, missing critical details and subtleties of the prompts.

Low-rank adaptation (LoRA) [7] has emerged as a pivotal enhancement technique for pre-trained models, introducing trainable low-rank matrices to adjust model weights efficiently. This approach is effective in image generation, where it fine-tunes neural networks to represent specific visual elements accurately, from individual character traits to unique style features. Despite its efficacy, employing multiple LoRA modules introduces complexity, necessitating additional training or fine-tuning to harmonize these adaptations. This extra training increases computational expenses and complicates the maintenance of the distinctiveness of each adaptation. The integration of multiple LoRA modules is primarily driven by the need to address new tasks without retraining the entire model. By combining several LoRA modules, each fine-tuned to distinct aspects of a task, the model can more effectively manage complex, multifaceted requirements, using the specific

strengths of each LoRA module to enhance the overall performance and adaptability of the model.

However, when multiple LoRA modules merge, traditional merging techniques [8–10] often fail to capture the precise user intentions from textual prompts, as observed in practices where LoRA modules have been integrated into complex image generation tasks without considering the interactions between various model adaptations. This oversight can lead to images that do not align with user expectations, especially as the number of integrated LoRA modules increases. Advanced approaches, such as the mixture of experts model [11], the mixture of LoRA experts (MoLE) model [8], and SVDiff [12] discussed in previous work [9], have attempted to address these problems by employing strategies that selectively activate various LoRA modules during the image synthesis process. However, these approaches still require training and do not fully consider the detailed prompts from the user in the merging process, failing to accurately reflect the intended outcomes. These challenges highlight the need for further research into sophisticated techniques for composing multiple LoRA modules, aiming to reduce retraining requirements, integrate subtle user inputs more effectively, and improve the quality and relevance of the generated images.

In contrast to traditional multi-module approaches like MoLE and SVDiff, the proposed method offers enhanced flexibility and performance by addressing limitations. While MoLE and SVDiff support the level of LoRA reusability, the proposed method maximizes this reusability, allowing for dynamic reuse of modules without additional retraining. Furthermore, the design of the proposed method enables a more flexible integration of multiple modules, preserving each module's inherent characteristics and adapting seamlessly to complex tasks. Most importantly, the proposed method better aligns with user intent by allowing for precise adjustments based on specific textual prompts, resulting in outputs that more closely match user expectations. These advantages highlight the proposed method's superior adaptability and responsiveness compared to traditional methods.

This paper introduces a novel method that employs multiple LoRA modules in a single image generation model. This method allows for dynamic and efficient changes tailored to specific tasks. Although using LoRA in image-related tasks is not unprecedented, the proposed method innovatively merges multiple LoRA modules to refine how generative models respond to textual prompts.

By allowing for the dynamic adjustment of attention mechanisms, this integration selectively amplifies the attention to LoRA modules, which are critical to user prompts. The model adapts its focus in real-time and enhances its capability to produce outcomes that closely mirror user expectations. The contributions of the proposed method presented in this paper are as follows:

- Enhanced alignment with user intentions: The alignment of generated images with user intentions is significantly improved by implementing this method, addressing a crucial gap in the current image generation technology.
- Enhanced technical adaptability: This enhancement advances the technical adaptability of generative models and offers new avenues for research and practical applications.
- Improved user satisfaction: This paper contributes toward improving user satisfaction regarding personalized content creation and other areas.

2. Related Work

This section reviews advancements in generative models, focusing on the evolution of diffusion models for image generation. It also examines LoRA for fine-tuning large pre-trained models and the challenges of integrating multiple LoRA modules to enhance performance in complex tasks, such as text-to-image generation [5,6,8–10,13].

2.1. Image and Text-to-Image Generation with Diffusion Models

Diffusion probabilistic models [14] have transformed image generation, producing high-quality synthetic images that mirror the natural distribution of training data. These models operate by parameterizing a Markov chain, gradually denoising random noise into

data-like samples by reversing the diffusion process, which typically adds noise until data become pure Gaussian noise. Learning to reverse this process allows diffusion models to achieve unparalleled fidelity and diversity in image generation, often surpassing traditional generative adversarial networks [15] in terms of stability and mode coverage. Research has expanded on the success of these models by adapting diffusion models for text-to-image generation, where text prompts direct the image creation process. This process involves conditioning the diffusion process on textual data, employing architectures that integrate text with the model denoising phases. These models are adept at producing detailed and contextually accurate images from elaborate descriptions, significantly advancing creative image generation. Their ability to generate such detailed images from text opens new avenues in automated content creation and highlights the significant potential for future enhancements to improve model accuracy and efficiency.

2.2. LoRA Module

The LoRA module significantly reduces the number of trainable parameters and computational overhead in pre-trained deep learning models, such as the generative pre-trained Transformer 3 (GPT-3) [16,17], making it an effective solution for updating large-scale models to new tasks without retraining the entire network. By integrating trainable low-rank matrices into the Transformer architecture [18], LoRA allows for precise, efficient weight adjustments while maintaining the original frozen pre-trained weights. This approach preserves the generalizability of the model and enhances its adaptability for specific tasks or datasets. Notably, LoRA also decreases the memory requirements of the graphics processing unit and increases the training throughput without adding inference latency.

2.3. Composing Multiple LoRA Modules

To optimize the enhancement of pre-trained models without extensive retraining, researchers have developed advanced methodologies for integrating multiple low-rank adaptation (LoRA) modules. These techniques, including the mixture of LoRA experts (MoLE) [8], treat each LoRA module as an independent expert and use a hierarchical weight control mechanism to adjust composition weights dynamically. This approach preserves the unique characteristics of each module while enhancing the overall model performance. Other strategies focus on sophisticated weight management techniques that adjust the influence of each module dynamically during operation, effectively balancing adaptability with performance.

The integration of multiple LoRA modules, while innovative, introduces challenges, such as the dilution of unique attributes and increased computational overhead as the number of modules grows [9]. Traditional linear arithmetic methods used for combining LoRA modules can reduce the specificity and effectiveness of individual adaptations, impairing model performance for tasks requiring distinct capabilities. Additionally, managing multiple LoRA modules in a single framework complicates the model architecture, making it challenging to maintain, especially when adjustments are necessary for evolving data or tasks. Ongoing research continues to seek more efficient integration strategies to lower computational demands and simplify model management, which is essential for advancing automated image generation.

Tables 1 and 2 compare the reusability, flexibility, and user intent adaptability across different traditional approaches, including single LoRA, normalized linear arithmetic (NLA), SVDiff, and the proposed method. As demonstrated, single LoRA lacks reusability and fusion flexibility, limiting its adaptability for diverse applications. While NLA and SVDiff address some issues, particularly in reusability and partial fusion flexibility, they fall short of fully aligning with user intent. In contrast, the proposed method achieves notable improvements across all criteria. It not only supports reusability and flexible fusion but also dynamically adapts to user intent without retraining, making it a robust solution for more complex, specific tasks. This comparative analysis highlights the unique advantages

of the proposed method, positioning it as an effective method for enhanced task-specific adaptability in image generation models.

Table 1. Comparison between different methods of merging LoRA models. In this table, '◯' indicates that the method supports the specific feature, while '×' indicates that the feature is not supported by the method. The features compared include LoRA reusability (whether the model can reuse existing LoRA modules), flexible fusion (the ability to combine multiple LoRA modules flexibly), and user intent alignment (how well the method aligns with the user's specific prompt intent).

	Single LoRA	**NLA**	**SVDiff**	**The Proposed Method**
LoRA reusability	×	◯	◯	◯
Flexible fusion	×	×	◯	◯
User intent	×	×	×	◯

Table 2. Comparison of Single LoRA, NLA, SVDiff, and the proposed method.

Method	**Description**	**Advantages**	**Disadvantages**
Single LoRA	Single task-specific LoRA model	Simple, efficient for specific tasks; minimal computational overhead	Lacks flexibility; cannot adapt to multiple tasks or contexts without retraining
NLA (Normalized Linear Arithmetic)	Combines multiple LoRA models using linear arithmetic	Provides partial flexibility in model combination	Performance degradation when combining many LoRA modules; lacks dynamic adaptation to different prompts
SVDiff	Uses singular value decomposition (SVD) to merge LoRA models	Captures key characteristics of multiple models effectively	Complex; may not retain all LoRA-specific features; requires significant computational resources
The proposed method	Dynamically merges multiple LoRA models based on prompt relevance	High adaptability to prompts; uses cosine similarity and softmax for task-specific model blending	Moderate computational cost; slightly more complex than single LoRA; may require fine-tuning parameters for optimal performance

3. The Proposed Method

This section details the proposed method for merging multiple LoRA [7] models using a prompt-driven attention mechanism. The aim is to compute the contribution of each pre-trained LoRA module dynamically based on its alignment with a given prompt, allowing for flexible task-specific adaptation without the need for retraining. The proposed method uses cosine similarity between the prompt and the learned representation of each LoRA module to determine weights, which are applied to merge the models coherently.

3.1. Overview

The core idea of the proposed method is to apply the learned representations of multiple LoRA modules and combine them into a single output model that is dynamically adapted to a specific prompt. The motivation for this method is the observation that each LoRA module can capture distinct semantic properties or task-specific information. Thus, instead of treating all models equally or manually selecting one, the proposed method aims to automatically assign appropriate weights to each model based on the prompt content. The process involves the following steps:

1. Embedding the LoRA key vector and query vector;
2. Cosine similarity and softmax-based LoRA controller between the query vector and LoRA key vector;
3. Weighted value matrix combination for merged model creation.

Figure 1 presents an overview illustration depicting the sequence of the proposed method process described above.

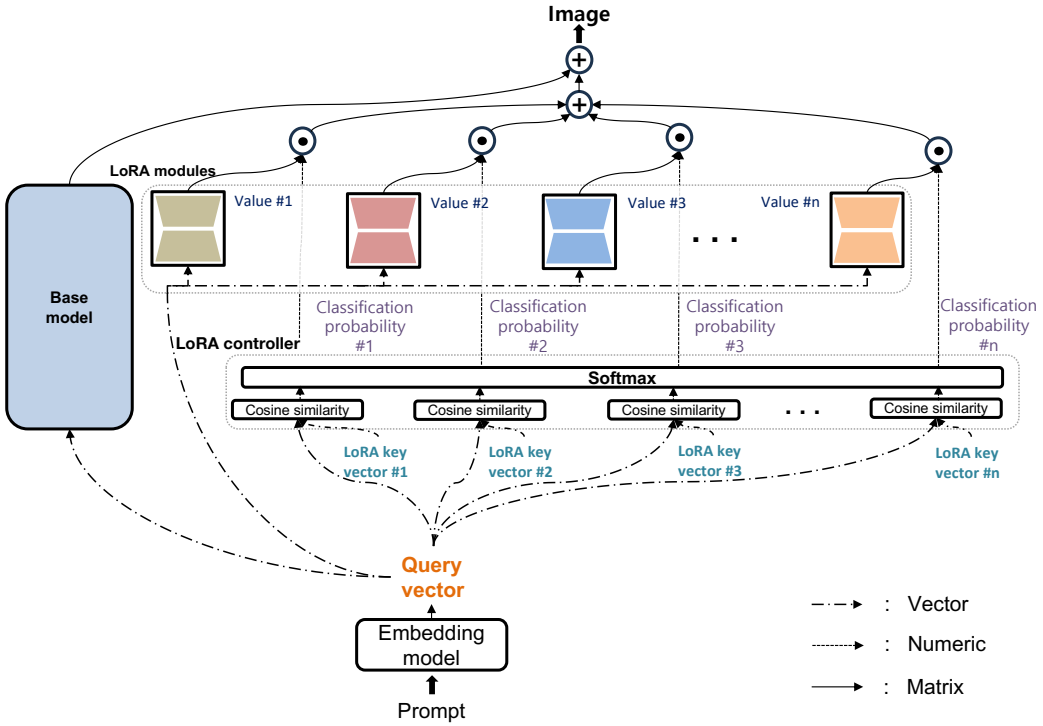

Figure 1. Low-rank adaptation (LoRA) fusion overview of architecture, where a query vector, representing the prompt's semantic content, is compared against multiple LoRA key vectors from each LoRA module. Cosine similarity computes weights via softmax, applying them to value matrices.

Formal definition: Given a prompt, the proposed method begins by encoding this prompt into a vector, referred to as the query vector Q. The prompt encapsulates the semantic meaning and contextual information necessary for the task. For each LoRA module i, a corresponding LoRA key vector K_i is defined, which represents the learned features or characteristics of that particular model. Additionally, each LoRA module has a value matrix V_i, which stores the trained model weights that will contribute to the final merged model. The number of LoRA modules to be merged is denoted by N. The overall objective is to compute a merged value matrix V', a combination of the individual V_i matrices, weighted according to their similarity to the given prompt.

3.2. Embedding the LoRA Key Vector and Query Vector

Before starting the inference process to generate the LoRA key vector K for each pre-trained LoRA module, the process begins by manually creating characteristic text that captures the characteristics of the module. As shown in Figure 2, these characteristic texts are embedded into a high-dimensional vector representation, which serves as the LoRA key vector K for the LoRA module. This LoRA key vector encapsulates the core properties of the LoRA module in a way that allows it to be compared with the query vector Q, which represents the prompt. The LoRA key vectors are precomputed and stored, ensuring that the proposed method can efficiently assess the relevance of each module to different prompts without recalculating the LoRA key vectors. This approach enables rapid and effective similarity computation, allowing the model to dynamically adapt to the prompt by selecting the most relevant LoRA modules based on their LoRA key vectors.

The first step in the inference of the proposed method involves converting the input prompt into a query vector Q, as shown in Figure 3, capturing the semantic content of the prompt in a form suitable for comparison with the LoRA modules. The prompt is processed using an embedding model, mapping the textual prompt to a high-dimensional embedding space to achieve this. The resulting vector Q represents the contextual and semantic information in the prompt, allowing it to be compared with the LoRA key vectors, Ki, of the LoRA modules. This transformation ensures that the prompt is encoded to retain the meaning and intent necessary for selecting the most relevant LoRA modules. A robust embedding model for this conversion step ensures that even complex or subtle prompts are accurately represented, providing a reliable basis for the similarity calculations.

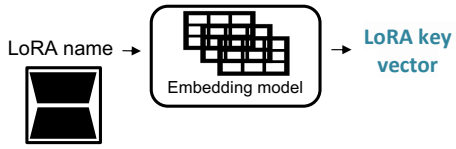

Figure 2. Illustration of the process in which the distinct characteristics of LoRA are captured in a name and then transformed into a LoRA key vector through an embedding model.

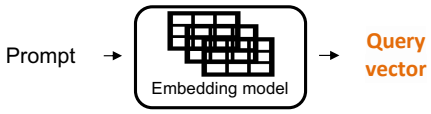

Figure 3. Prompt transformed into a query vector using an embedding model.

3.3. Cosine Similarity and Softmax-Based LoRA Controller Between the Query Vector and LoRA Key Vector

With the LoRA key vector K and query vector Q established, we proceed to the step of comparing these vectors to determine the most relevant LoRA modules for the prompt. The cosine similarity between query vector Q and each LoRA key vector K_i measures how relevant each LoRA module is to the prompt. Cosine similarity allows the proposed method to prioritize LoRA modules whose semantic space aligns more closely with the task described by the prompt by focusing on the angular relationship between vectors, as shown in Figure 4. Cosine similarity is computed as the dot product of two vectors, normalized by their Euclidean norms; this normalization of vector magnitudes allows the measure to focus exclusively on the angle between vectors. Furthermore, cosine similarity is computationally efficient [19]. This focus on the angle is essential for capturing semantic similarity in high-dimensional spaces [20,21]. In this paper, cosine similarity prevents models with larger magnitudes but less relevance from dominating the merged model, ensuring that the final composition more accurately reflects the task.

After computing the cosine similarities S_i for each LoRA module, the next step is to convert these similarity scores into normalized weights that determine the contribution of each model to the final merged output. To achieve this, the softmax function is applied to the similarity scores.

The softmax function is defined as follows: $w_i = \frac{\exp(S_i)}{\sum_{j=1}^{N} \exp(S_j)}$, where w_i represents the weight assigned to the i-th LoRA module. The exponential function $\exp(S_i)$ ensures that all weights are positive, and the normalization term $\sum_{j=1}^{N} \exp(S_j)$ ensures that the weights sum to 1. The softmax function has several important properties that make it ideal for the proposed method:

- Amplification of differences: The exponential nature of the softmax function magnifies differences between similarity scores. Even slight differences in cosine similarity can result in significantly different weights, ensuring that models with slightly higher relevance to the prompt have a much more considerable influence on the final model.
- Smooth normalization: Softmax provides a smooth, continuous normalization of the similarity scores, preventing abrupt changes in the weights when different prompts

are given, leading to a smooth transition between LoRA module configurations as the task changes.

- Scalability: Softmax is computationally efficient and can be applied to any number of LoRA modules without significant overhead. The computation only involves exponentials and summations; hence, it scales linearly with the number of models.

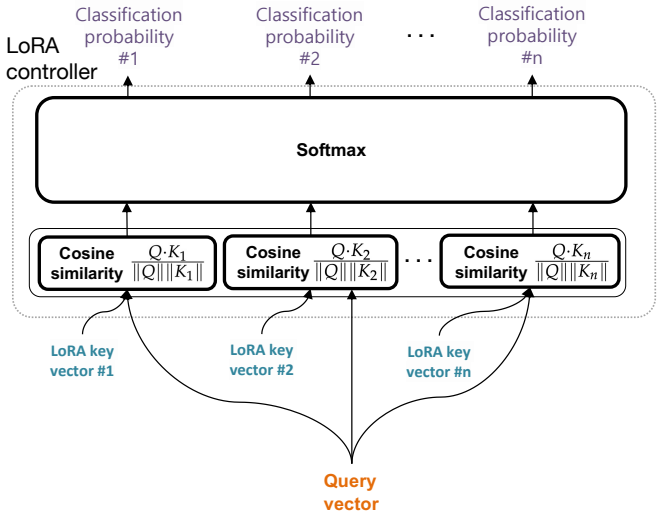

Figure 4. Setting the value for how low-rank adaptation (LoRA) is determined using softmax with query vectors, predefined LoRA key vectors, and cosine similarities.

3.4. Weighted Value Matrix Combination for Merged Model Creation

Once the weights are computed for each LoRA module, the final step is to merge the value matrices V_i of the LoRA modules. The merged value matrix V' is obtained by taking a weighted sum of the individual Value matrices. Specifically, each value matrix V_i is multiplied by its corresponding weight w_i, and then all the weighted matrices are summed together to form the final merged model.

Interpretation of the merged model: The merged model is an ensemble of LoRA modules, where each model contributes according to how well it aligns with the input prompt. This results in a model dynamically adapted to the task and capable of using the strengths of various pre-trained LoRA modules. Because the weights are computed based on prompt-specific similarity, the method ensures that the final model is closely aligned with the task requirements.

Additionally, this approach allows for flexibility in handling multiple LoRA modules. It avoids the limitations of static combinations or manual model selection, providing a fully automated and data-driven method for determining how best to merge the available models. Once the weights w_i are computed for each LoRA module, the final step is to merge the value matrices V_i of the LoRA modules. The merged value matrix V' is a weighted combination of the individual Value matrices, computed as shown in Equation (1):

$$V' = \sum_{i=1}^{N} \left(\frac{\exp\left(\frac{Q \cdot K_i}{\|Q\|\|K_i\|}\right)}{\sum_{j=1}^{N} \exp\left(\frac{Q \cdot K_j}{\|Q\|\|K_j\|}\right)} \right) V_i \tag{1}$$

where V' is the final output matrix and w_i denotes the weight computed via the softmax function. This equation reveals that the final merged model is a linear combination of the original LoRA modules, with the weights determining how much each model contributes, as illustrated in Figure 5. The entire process of computing weights and merging value matrices for the final model is outlined in Algorithm 1.

For a prompt such as “a photo of a cat blowing bubble gum”, the query vector might show high similarity with key vectors from modules like “Cat” and “Bubble Gum”. Using

softmax, these modules receive higher weights, leading the model to prioritize their value matrices in the final merged output, which aligns closely with the prompt's theme.

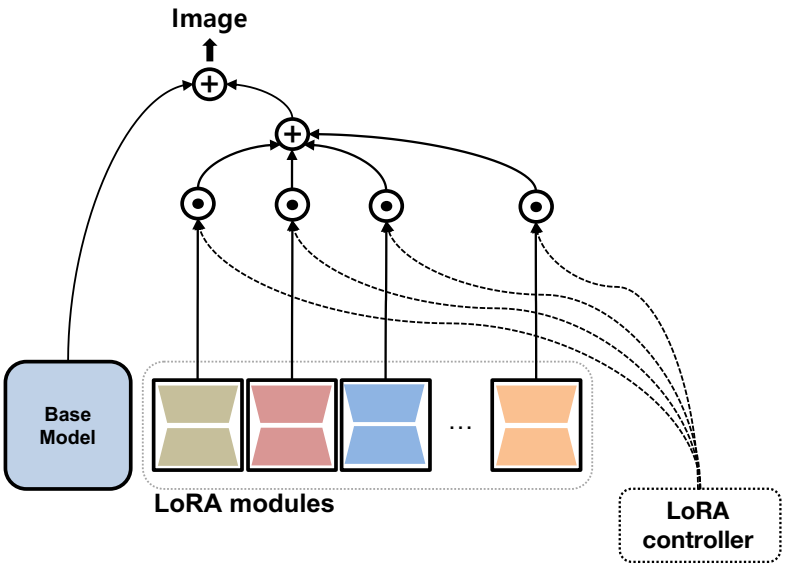

Figure 5. Low-rank adaptation (LoRA) fusion overview of the architecture, where a query vector is compared to multiple LoRA key vectors. Cosine similarity computes the weights via softmax, applying them to the value matrices.

Algorithm 1: Optimized method for merging LoRA models

Input: Prompt P, LoRA models $\{LoRA_1, LoRA_2, \ldots, LoRA_N\}$
Output: Merged value matrix V'

$Q \leftarrow f_{\mathrm{E}}(P)$
$V' \leftarrow 0$;
$Z \leftarrow 0$;
for $i \leftarrow 1$ **to** N **do**
  $K_i \leftarrow LoRA_i$.key ;
  $V_i \leftarrow LoRA_i$.value ;
  $S_i \leftarrow \frac{Q \cdot K_i}{\|Q\| \|K_i\|}$;
  $w_i \leftarrow \exp(S_i)$;
  $Z \leftarrow Z + w_i$;
  $V' \leftarrow V' + (w_i \times V_i)$;
$V' \leftarrow \frac{V'}{Z}$;
return V' ;

In Algorithm 1, $f_{\mathrm{E}}(P)$ is the function about converting input prompt to a query vector. Time complexity is $O(N)$, where N is the number of LoRA modules. Space complexity is $O(N)$, as it requires storing similarity scores and weights for each of the N modules.

4. Experiments

This section presents experiments conducted to validate the proposed method's effectiveness in dynamically merging LoRA modules, focusing on text–image alignment and computational efficiency.

4.1. Experimental Environment

Experimental Setup: The proposed method dynamically adjusted the weights of multiple LoRA modules based on the user's prompt, leveraging cosine similarity for weight calculation and softmax for fusion. To validate this method, this paper applied the proposed

approach to a text-to-image generation task using the Stable Diffusion architecture, specifically the DreamBooth model [22]. DreamBooth was well-suited for generating images from textual prompts due to its fine-tuning capability on personalized visual concepts.

In this experiment, the proposed method processed user prompts by calculating the cosine similarity between the prompt and pre-defined LoRA modules, dynamically adjusting the contribution of each LoRA to the final image. For the evaluation, this paper used three different LoRA modules per experiment, each pre-trained on distinct visual concepts such as space, landscapes, and architecture, to combine these into a coherent output based on the user's text input. Table 3 provides a comparison of various LoRA models across categories such as character, clothing, style, background, and object. Each model is characterized by features optimized for specific applications and contexts, enhancing adaptability and performance in targeted scenarios.

Table 3. Comparison of LoRA models across various categories.

Category	LoRA Model	Characteristic
Character	Asian female	Realistic appearance
Character	White female	Iconic look, limited flexibility
Character	African American male	Strong build, suited for action
Character	Cat	hairy
Clothing	Thai University Uniform	Authentic, limited to formal use
Clothing	School Dress	Classic look, culturally specific
Style	Japanese Film Color Style	Nostalgic, limited to vintage themes
Style	Bright Style	Bright scenes, unsuitable for dark themes
Background	Library Bookshelf Background	Academic setting, indoor only
Background	Forest Background	Natural look, outdoor-only use
Object	Umbrella	Realistic, limited to rainy scenes
Object	Bubble Gum	Playful, limited to casual scenes

The generated images were set to a resolution of 1024 × 768; for each experiment, this paper used 5 different text prompts to generate a total of 200 images per prompt. These images were then used to calculate the average performance across various metrics. Table 4 summarizes the detailed experimental environment.

Table 4. Key parameter settings of the experimental environment.

Parameter	Value
Models Used	
Base model	Stable diffusion 1.5
LoRA Modules	
Number of LoRA modules	11
Image generation settings	
Height of the generated images	1024 pixels
Width of the generated images	768 pixels
Images generated per Prompt	200
Total images generated	1000
Evaluation metrics	
Text–image alignment	Measured using CLIP
Method details	
Similarity metric used	Cosine similarity
Weight normalization function	Softmax

4.2. Metrics

This paper evaluates the proposed method using the primary metric text–image alignment, which evaluates how well the generated image aligns with the textual input from the user. This paper measures this using the CLIP model [23], which computes the similarity between the generated image and text prompt in a shared feature space. This paper compares the performance of the proposed method against the following two baseline methods:

- Normalized linear arithmetic (NLA) composition: A simple method that assigns equal weights to each LoRA module, averaging their outputs.
- SVDiff [12]: A state-of-the-art method that preserves LoRA characteristics while combining them into a single output.

4.3. No Additional Training Required

One of the critical advantages of the proposed method is that it requires no additional training. Unlike other approaches, such as MoLE [8], which requires hierarchical weight control through training, the proposed method uses pre-trained LoRA modules and combines them based on the prompt in real-time. This method significantly reduces the computational cost and time, making the proposed method more efficient for real-world applications that require dynamic, on-the-fly adaptation.

In addition, the proposed method could scale to multiple LoRA modules without suffering from the performance degradation typical of linear combination methods, such as NLA. Cosine similarity-based weighting ensures that each LoRA contributes proportionally to the final image, making the method stable even when multiple LoRA modules are combined.

4.4. Results

Table 5 and Figure 6 reveal that the proposed method outperformed SVDiff [12] and NLA in terms of text–image alignment and image similarity. Specifically, the proposed method achieved a text–image alignment score of 744, representing a significant improvement over SVDiff at 0.724 and NLA at 0.698. These results demonstrate the ability of the proposed method to effectively combine pre-trained LoRA modules without requiring additional fine-tuning, while dynamically adapting to the user prompt to produce semantically accurate and visually coherent images.

Table 5. Performance comparison of the proposed method against baseline methods in text–image alignment and image similarity.

Method	Image-Alignment
NLA	0.698
SVDiff	0.724
The proposed method	**0.744**

4.5. Detailed Performance Comparison and Analysis

Compared to traditional approaches, the proposed method excels at dynamically adapting to user prompts in real-time. Although traditional methods, such as MoLE, rely on hierarchical weight control through additional training, the proposed method uses cosine similarity to adjust LoRA weights on the fly, enabling it to manage various tasks without retraining. This flexibility allows the proposed method to produce results tailored to specific user inputs, making it highly suitable for applications requiring real-time responses, such as interactive content generation or dynamic image creation. In contrast, some approaches, such as NLA, suffer from performance degradation when multiple LoRA modules are combined because the unique characteristics of each LoRA module become diluted. Moreover, the proposed method overcomes this problem by employing a softmax

function to ensure that the contribution of each LoRA module is proportionally weighted based on its relevance to the prompt.

In terms of computational efficiency, the ability of the proposed method to fuse pre-trained LoRA modules without retraining provides this method with a significant edge in large-scale deployments, where computational resources are often a limiting factor. By eliminating the need for retraining, the proposed method could efficiently adapt to new tasks with minimal computational overhead, making it ideal for real-time applications.

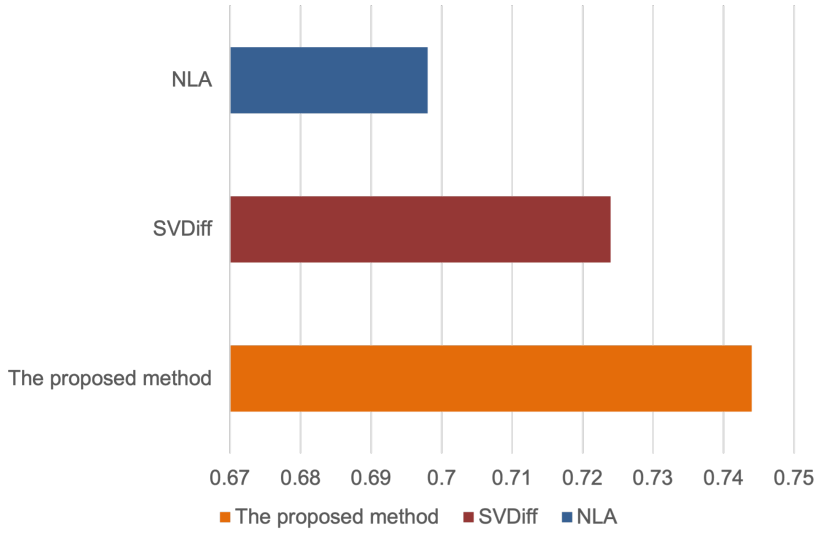

Figure 6. Text–image alignment performance comparison graph.

5. Discussion

The proposed method significantly improves text–image alignment and image similarity over traditional methods, such as NLA and SVDiff, underscoring its effectiveness in dynamically integrating multiple pre-trained modules. The utility of cosine similarity and softmax normalization in the proposed method is critical because these mechanisms ensure that the contribution of each module is precisely adjusted to match the contextual prompts, thereby enhancing the semantic accuracy and visual coherence of the generated images. However, the performance of the proposed method heavily relies on the quality and diversity of the pre-trained modules. If these modules lack a comprehensive representation of diverse visual styles or concepts, the system's performance could notably degrade. This limitation underscores the necessity of developing a more robust set of pre-trained modules that can cover a broader spectrum of visual information to maintain system effectiveness across diverse tasks.

Moreover, the dependency on cosine similarity and softmax normalization for weight adjustments and module fusion introduces vulnerabilities. The ablation study clearly demonstrated that removing these components significantly reduced the effectiveness of the proposed method, confirming their indispensable role in maintaining the stability and functionality of the system. This reliance may limit the adaptability of the proposed method under conditions where these methods are less effective.

In conclusion, while the proposed method offers substantial improvements in automated image generation, future enhancements should focus on expanding the range and specialization of LoRA modules to better capture diverse image styles and details. Additionally, exploring advanced dynamic weight adjustment methods, such as adaptive weight scaling or context-aware module prioritization, could further reduce the model's reliance on fixed components. This would enhance the system's adaptability and robustness across various tasks and conditions.

6. Conclusions

This research found that the proposed method is a robust approach for dynamically integrating multiple LoRA modules, significantly enhancing the capabilities of text-to-image generation systems. By using real-time data to dynamically adjust the weights of each module, the proposed method successfully generates visually appealing images that closely align with user input. Despite its efficacy, the dependency on high-quality, diverse pre-trained modules and the essential roles of cosine similarity and softmax normalization highlight areas for potential improvement.

Future research should focus on optimizing these weight adjustment mechanisms to enhance the adaptability and efficiency of the system further. Additionally, exploring the application of the proposed method in other domains, such as music or video generation, could uncover new opportunities for creative content creation. Moreover, the proposed method presents a promising avenue for the development of more responsive and adaptable generative models, facilitating advancements in automated content generation that more accurately reflect user intentions and preferences.

Author Contributions: Conceptualization, D.C., J.I., and Y.S.; methodology, D.C., J.I., and Y.S.; software, D.C. and J.I.; validation, D.C. and J.I.; formal analysis, D.C. and J.I.; investigation, D.C. and J.I.; writing—original draft preparation, D.C. and J.I.; writing—review and editing, D.C., J.I., and Y.S.; visualization, D.C. and J.I.; supervision, Y.S.; project administration, D.C. and J.I. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2024-RS-2023-00254592) grant funded by the Korea government(MSIT).

Data Availability Statement: The original contributions presented in the paper are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the paper; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; Lee, H. Generative adversarial text to image synthesis. In Proceedings of the International Conference on Machine Learning, PMLR, New York City, NY, USA, 19–24 June 2016; pp. 1060–1069.
2. Gu, S.; Chen, D.; Bao, J.; Wen, F.; Zhang, B.; Chen, D.; Yuan, L.; Guo, B. Vector quantized diffusion model for text-to-image synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10696–10706.
3. Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv* **2021**, arXiv:2112.10741.
4. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695.
5. Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. Photorealistic text-to-image diffusion models with deep language understanding. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 36479–36494.
6. Reddy, M.D.M.; Basha, M.S.M.; Hari, M.M.C.; Penchalaiah, M.N. Dall-e: Creating images from text. *UGC Care Group I J.* **2021**, *8*, 71–75.
7. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *arXiv* **2021**, arXiv:2106.09685.
8. Wu, X.; Huang, S.; Wei, F. Mixture of lora experts. *arXiv* **2024**, arXiv:2404.13628.
9. Zhong, M.; Shen, Y.; Wang, S.; Lu, Y.; Jiao, Y.; Ouyang, S.; Yu, D.; Han, J.; Chen, W. Multi-lora composition for image generation. *arXiv* **2024**, arXiv:2402.16843.
10. Gu, Y.; Wang, X.; Wu, J.Z.; Shi, Y.; Chen, Y.; Fan, Z.; Xiao, W.; Zhao, R.; Chang, S.; Wu, W.; et al. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. *Adv. Neural Inf. Process. Syst.* **2024**, *36*. [CrossRef]
11. Jordan, M.I.; Jacobs, R.A. Hierarchical mixtures of experts and the EM algorithm. *Neural Comput.* **1994**, *6*, 181–214. [CrossRef]
12. Han, L.; Li, Y.; Zhang, H.; Milanfar, P.; Metaxas, D.; Yang, F. Svdiff: Compact parameter space for diffusion fine-tuning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 7323–7334.

13. Xu, T.; Zhang, P.; Huang, Q.; Zhang, H.; Gan, Z.; Huang, X.; He, X. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1316–1324.
14. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
15. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*.. [CrossRef]
16. Brown, T.B. Language models are few-shot learners. *arXiv* **2020**, arXiv:2005.14165.
17. Radford, A. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://hayate-lab.com/wp-content/uploads/2023/05/43372bfa750340059ad87ac8e538c53b.pdf (accessed on 4 December 2017).
18. Vaswani, A. Attention Is All You Need. *Advances in Neural Information Processing Systems*. 2017. Available online: https://user.phil.hhu.de/cwurm/wp-content/uploads/2020/01/7181-attention-is-all-you-need.pdf (accessed on 11 June 2018).
19. Schütze, H.; Manning, C.D.; Raghavan, P. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008; Volume 39.
20. Singhal, A. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.* **2001**, *24*, 35–43.
21. Mikolov, T. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
22. Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 22500–22510.
23. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 8748–8763.