ModuleTeam: Open-Set Multi-Conditional Image Generation with Training-Free Latent Mixture of Any Control Module

Yuwei Zhou
DCST, Tsinghua University
Beijing, China
zhou-yw21@mails.tsinghua.edu.cn

Xin Wang*
DCST, BNRist, Tsinghua University
Beijing, China
xin_wang@tsinghua.edu.cn

Hong Chen
DCST, Tsinghua University
Beijing, China
h-chen20@mails.tsinghua.edu.cn

Yipeng Zhang
DCST, Tsinghua University
Beijing, China
zhang-yp22@mails.tsinghua.edu.cn

Zeyang Zhang
DCST, Tsinghua University
Beijing, China
zy-zhang20@mails.tsinghua.edu.cn

Wenwu Zhu*
DCST, BNRist, Tsinghua University
Beijing, China
wwzhu@tsinghua.edu.cn

Abstract

Multi-conditional image generation aims to create customized images that align with multiple specified conditions. Existing methods, whether through end-to-end training or by fine-tuning adapters to integrate pre-trained control modules of the same category (e.g., LoRA, IP-Adapter, ControlNet, T2I-Adapter), are restricted to a closed set of predefined input conditions. To overcome this limitation, we propose ModuleTeam, a **t**raining-fr**e**e method for l**a**tent **m**ixture of arbitrary control modules, capable of handling open-set conditions by incorporating the corresponding modules. The design of ModuleTeam is rooted in two key findings: (i) modules interfere with each other at the level of model parameters, and (ii) module weights contribute to the generated images by affecting the noise predictions within the diffusion process in an approximately linear manner. The first finding motivates our latent mixture approach, which mixes the control modules by aggregating their latent variables between diffusion model blocks. The second finding enables a multi-inference module reweighting strategy that balances module contributions to generation, requiring no additional training or fine-tuning overhead. Extensive results demonstrate that ModuleTeam not only outperforms existing methods but also provides flexibility in the types of conditions and scalability in their number.

CCS Concepts

• **Computing methodologies** → **Computer vision**.

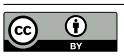
Keywords

Image Generation, Open-Set Multi-Condition, Diffusion Model

ACM Reference Format:

Yuwei Zhou, Xin Wang, Hong Chen, Yipeng Zhang, Zeyang Zhang, and Wenwu Zhu. 2025. ModuleTeam: Open-Set Multi-Conditional Image Generation with Training-Free Latent Mixture of Any Control Module. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3746027.3755686

*Corresponding authors. DCST is the abbreviation of Department of Computer Science and Technology. BNRist is the abbreviation of Beijing National Research Center for Information Science and Technology.



MM '25, Dublin, Ireland

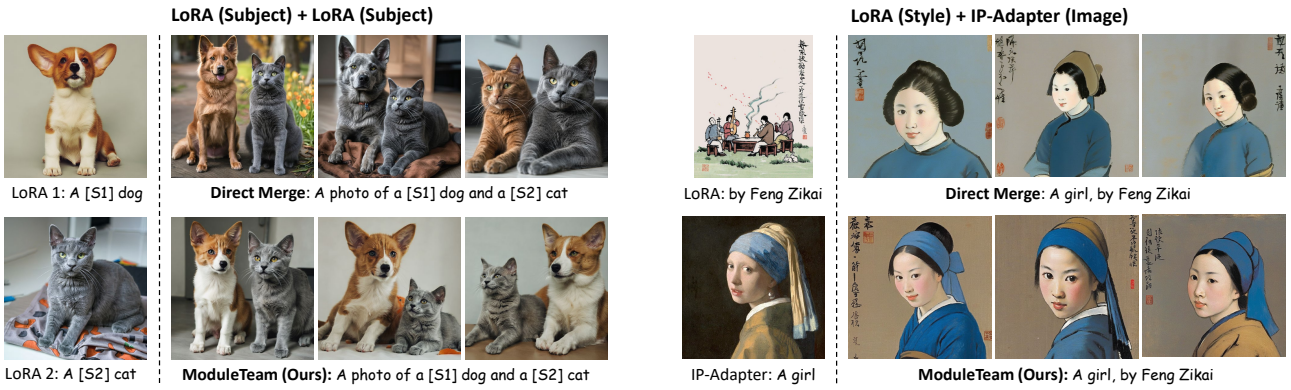
ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755686

1 Introduction

Diffusion models mark a significant milestone in the field of image generation, providing a practical approach to synthesizing high-quality images. While striving for high-quality and high-resolution images, there is also a growing demand for precise control over the generation to enable personalized emotional expression and subtle requirements. The text-to-image paradigm [21, 26, 27, 30] provides an excellent solution but not a perfect one, as text has inherent limitations in its expressive capacity. As a result, multi-conditional image generation has become a widely studied task [1], with an increasing variety of conditions being incorporated into the generation process, including text, subject [29], style [32], spatial location [17], human pose skeletons, edge maps, segmentation maps, and depth maps [20, 38].

Existing methods for multi-conditional generation can be broadly categorized into end-to-end training and adapter-tuning approaches. End-to-end methods [12–14, 24, 35, 40] typically adopt a joint training strategy to train a unified model on a large-scale dataset that accommodates various conditions. While these trained models can handle multiple conditions and enable zero-shot composition, they demand substantial computational resources and lack the flexibility to incorporate unseen conditions. Adapter-tuning methods [8, 18, 23, 31], on the other hand, design strategies to integrate pre-trained control modules, such as LoRA [11], ControlNet [38], IP-Adapter [37], T2I-Adapter [20], or other customized modules, leveraging their respective effective control over the generation model. Compared to end-to-end methods, adapter-tuning methods are more resource-efficient. However, existing adapter-tuning methods are constrained to combining modules of the same type, failing to explore the merging of diverse modules. This is a significant limitation, as no single type of control module can cover all possible conditions. For example, LoRA excels at addressing subject and style conditions, IP-Adapter accepts image inputs, and ControlNet and T2I-Adapter handles structural conditions like edge and depth map. In summary, existing methods are limited to a closed set of predefined conditions, thus failing to meet the requirements of open-set multi-conditional generation.

Based on the aforementioned discussions, we propose to push the boundaries of existing methods by exploring strategies to mix



(a) Module Interference Example **(b) Module Imbalance Example**

Figure 1: Two examples of module interference and imbalance. Between LoRAs, there are incorrect subjects (e.g., dog) and confused attributes (e.g., color). Between the LoRA and IP-Adapter, style condition suppresses image condition.

control modules of different types, thereby enabling the incorporation of open-set multiple conditions. However, achieving this goal presents two non-trivial challenges. (i) The first is *module interference* caused by parameter interactions. For instance, both LoRA and IP-Adapter introduce additional parameters to the Transformer block of U-Net within diffusion models, which can lead to interference when these two modules are used simultaneously. A theoretical analysis is presented in Section 3.2. (ii) The second is *module imbalance* on their contributions to the generated image. This imbalance arises because different types of control modules have distinct target conditions, architectural designs, and parameter scales, which lead to uneven influence on the denoising process. A visual illustration of both challenges is provided in Figure 1.

To tackle the challenges, in this paper, we design a **module**-based **t**raining-fr**e**e l**a**tent **m**ixture method (ModuleTeam) for open-set multi-conditional image generation. To avoid module interference, we design a latent mixture approach to combine control modules in the latent space rather than at the model parameter level. Concretely, we load one control module and perform one forward propagation through each U-Net block at a time, and then aggregate the resulting latent variables between blocks, instead of loading all modules simultaneously and passing the input through the model once. To balance module contributions, we introduce a multi-inference module reweighting strategy that assigns proportional weights to each module according to the change it induces in the noise predictions output by the U-Net. This approach is grounded in our observation that the changes in noise predictions exhibit an approximately linear relationship with the module weights, as illustrated in Figure 2. Since the final image is derived from the cumulative noise predictions across multiple timesteps, balancing the influence on these predictions ensures that each module contributes approximately equally to the final output. Notably, the entire method requires no additional training or fine-tuning, relying solely on multiple inference passes.

To validate the effectiveness of our method, we conduct empirical experiments on a collected dataset that includes diverse conditions, respectively adapted to different types of control modules. Comparative results demonstrate that ModuleTeam outperforms existing baselines, and ablation studies confirm that the proposed latent mixture approach and multi-inference module reweighting strategy contribute to improvements in open-set multi-conditional image generation. Overall, our contributions are listed as follows.

- To the best of our knowledge, this is the first work to incorporate control modules of different types into diffusion models for open-set multi-conditional image generation.
- We propose a latent mixture approach and a multi-inference module reweighting strategy to effectively address the challenges of module interference and imbalance.
- We uncover and theoretically explain an approximately linear relationship between module weights and noise prediction changes, which serves as the basis for determining module weights.
- Empirical results demonstrate that our method outperforms existing approaches in performance while offering flexibility in the types of conditions and scalability in their number.

2 Related Work

2.1 Conditional Image Generation

Conditional image generation, also known as controllable image generation, requires the generated image to comply with specified conditions [1]. Text-to-image diffusion models [21, 26, 27, 30] are the most prominent examples, where the condition is a textual description. With the rapid advancement of diffusion models, a variety of novel conditions and their corresponding control modules have emerged. For example, LoRA (Low-Rank Adapter) [11] has been widely adopted to incorporate conditions including subject [5, 6, 29**?**], identity [7, 33], style [10, 32], and so on [3, 4, 22, 39]. IP-Adapter [37] is specifically designed to handle image conditions. ControlNet [38] is effective in processing structural conditions, such as Canny edges, depth maps, normal maps, M-LSD lines, HED soft edges, ADE20K segmentation, OpenPose, and user sketches. T2I-Adapter [20] targets conditions like segmentation, sketches, Canny edges, color, depth, and keypoints. Notably, a single control module can only handle one condition at a time. As a result, multi-condition scenarios often require either the development of newly designed modules or the specific adaptation of existing modules.

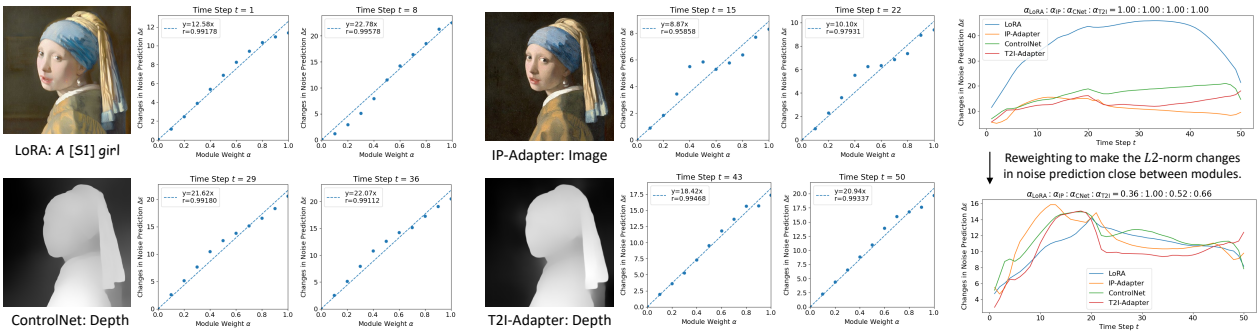


Figure 2: Approximately linear relationship between module weights and $L2$-norm changes in noise predictions. Based on this observation, we can reweight the module weights to achieve close $L2$-norm changes between modules in the denoising process.

2.2 Multi-Conditional Image Generation

Multi-conditional image generation, as the name suggests, extends conditional generation to incorporate multiple conditions. It presents non-trivial challenges, particularly in making generated images consistent with different conditions while maintaining image quality. Pioneering works can be grouped into two branches based on the training strategy: end-to-end methods and adapter-tuning methods.

The end-to-end methods focus primarily on innovating model architectures and training strategies. For example, Cocktail [12] proposes a generalized ControlNet architecture, a controllable normalisation layer, and a spatial guidance sampling method. Composer [13] explores global and localized conditioning mechanisms and adopts a joint training strategy to recompose decomposed images. DiffBlender [14] provides an embedding network and designs local and global self-attention for spatial and non-spatial tokens. UniControl [24] introduces a task-aware HyperNet for condition selection and trains the model across unique tasks. Uni-ControlNet [40] designs local and global control adapters, trains them separately, and integrates them directly. VideoComposer [35] equips the model with a unified Spatio-Temporal Condition encoder and leverages a two-stage training for text-to-video generation and conditions incorporation. Despite their effectiveness, these methods are resource-intensive and restricted to predefined conditions.

Adapter-tuning methods concentrate on the utilization and integration of pre-trained control modules, whether existing or custom-designed. For example, CTRL-Adapter [18] introduces an adapter with spatial/temporal convolution and attention mechanisms to map outputs from ControlNet into the diffusion model. Mix-of-Show [8] tunes embedding-decomposed LoRAs for individual concepts and employs gradient fusion in the center node to combine the LoRAs. Orthogonal Adaptation [23] independently tunes LoRAs for each concept while enforcing orthogonality constraints between them to achieve concept disentanglement. ZipLoRA [31] separately tunes a LoRA for subject and style, and trains a merge vector to combine the LoRAs by minimizing their cosine similarity. Compared to end-to-end strategies that require training from scratch, modular methods significantly reduce the number of trainable parameters and save substantial computational resources. However, existing methods are generally limited to integrating control modules of the same type, such as only ControlNets or only LoRAs.

In this paper, we take a step further by exploring an effective way to mix control modules of different types for open-set multi-conditional image generation.

3 Method

In this section, we propose ModuleTeam, a training-free latent mixture method for control modules. We begin with preliminaries about Stable Diffusion (Section 3.1). Then we analyze the challenge of module interference in the parameter space, taking LoRA and IP-Adapter as examples (Section 3.2). To address this issue, we design a latent mixture approach (Section 3.3). After that, we present a multi-inference module reweighting strategy to balance the contributions of different modules to generation (Section 3.4).

3.1 Preliminaries

Stable Diffusion is a pre-trained latent diffusion model [27] with key components of an autoencoder $(\mathcal{E}, \mathcal{D})$ to transform image pixels into latent variables and back, a CLIP [25] text encoder E_T to project texts into embeddings, and a U-Net [28] ϵ_θ to perform the diffusion denoising process by predicting the noise ϵ. Delving into the U-Net architecture, it consists of multiple downsampling, middle, and upsampling blocks. These blocks are further constructed from ResNet blocks [9], Transformer blocks [34], and convolutional layers [15] for the downsampling and upsampling operations.

Currently, mainstream control modules are applied to the U-Net, with occasional ones for the text encoder. Consequently, our research focuses on the mixture of control modules within the U-Net architecture.

3.2 Modules Interference

Between LoRAs. To begin with, we introduce the interference issue between the modules of the same type. Generally, a LoRA adapts the parameters of query, key, value, and output projections $W = \{W_Q, W_K, W_V, W_O)\}$ in the attention layers with $\Delta W = \{\Delta W_Q, \Delta W_K, \Delta W_V, \Delta W_O)\}$, so the attention with LoRA becomes:

$$\begin{aligned} &\text{Attn}\,(W, \Delta W) \\ &= \text{Softmax}\left(\frac{(Q+\Delta Q)(K+\Delta K)^T}{\sqrt{d}}\right)(V+\Delta V)(W_O+\Delta W_O). \end{aligned} \tag{1}$$

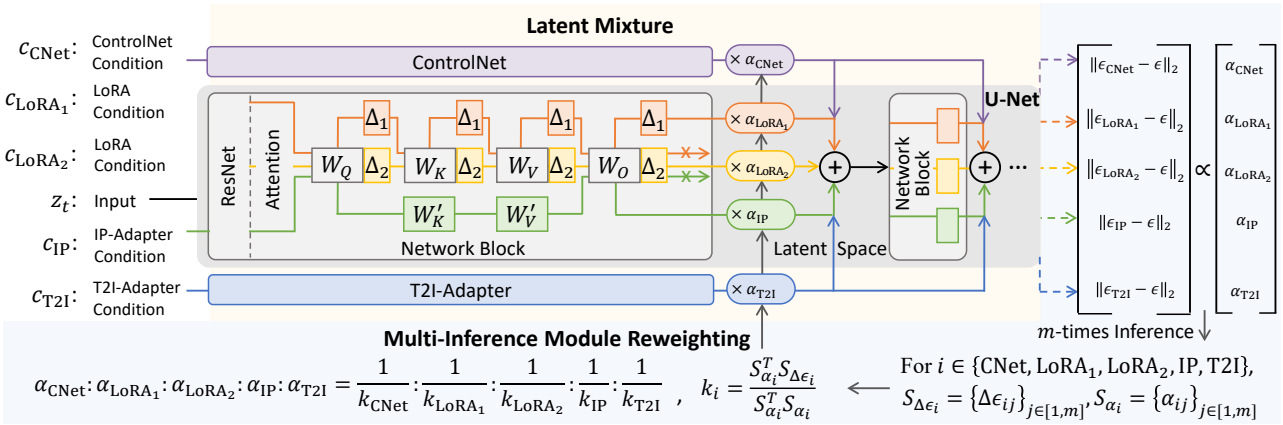


Figure 3: ModuleTeam framework. Yellow (top): Latent Mixture approach mixes modules by summing their latent variables in the latent space between network blocks of the U-Net. Blue (bottom): Multi-Inference Module Reweighting strategy determines module weight proportions based on approximately linear relationship between module weights and noise prediction changes.

Without loss of generality, we consider the simplest case of two LoRAs and observe the dot product score between query and key:

$$\begin{aligned}
&\text{Score}\,(W, \Delta W_1, \Delta W_2) \\
&= (Q + \Delta Q_1 + \Delta Q_2)(K + \Delta K_1 + \Delta K_2)^T \\
&= (QK^T + Q\Delta K_1^T + \Delta Q_1 K^T + \Delta Q_1 \Delta K_1^T) \\
&+ (QK^T + Q\Delta K_2^T + \Delta Q_2 K^T + \Delta Q_2 \Delta K_2^T) \\
&+ (-QK^T + \Delta Q_1 \Delta K_2^T + \Delta Q_2 \Delta K_1^T) \\
&= \text{Score}(W, \Delta W_1) + \text{Score}(W, \Delta W_2) + f(W, \Delta W_1, \Delta W_2).
\end{aligned} \tag{2}$$

The entangled item $f\,(W, \Delta W_1, \Delta W_2)$ reflects the mutual interference between the two LoRAs, which is further amplified by the subsequent Softmax and the multiplication with value and output projection matrices in the attention mechanism.

Between LoRA and IP-Adapter. To elucidate the interference issue between modules of different types, we take LoRA and IP-Adapter as an example. IP-Adapter enables diffusion models to process image input by introducing additional cross-attention with new parameters $W' = \{W'_K, W'_V\}$ alongside the original cross-attention, so the attention with an IP-Adapter becomes:

$$\begin{aligned}
\text{Attn}\,(W, W') &= \left(\text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V + \text{Softmax}\left(\frac{Q(K')^T}{\sqrt{d}}\right)V'\right)W_O \\
&= \text{Attn}\,(W) + \text{Attn}\,(W').
\end{aligned} \tag{3}$$

Consider the attention layer with a LoRA and an IP-Adapter:

$$\begin{aligned}
&\text{Attn}(W, \Delta W, W') \\
&= \left(\text{Softmax}\left(\frac{(Q+\Delta Q)(K+\Delta K)^T}{\sqrt{d}}\right)(V + \Delta V)\right. \\
&\left. + \text{Softmax}\left(\frac{Q(K')^T}{\sqrt{d}}\right)V'\right)(W_O + \Delta W_O) \\
&= \text{Softmax}\left(\frac{(Q+\Delta Q)(K+\Delta K)^T}{\sqrt{d}}\right)(V + \Delta V)(W_O + \Delta W_O) \\
&+ \text{Softmax}\left(\frac{Q(K')^T}{\sqrt{d}}\right)V'W_O + \text{Softmax}\left(\frac{Q(K')^T}{\sqrt{d}}\right)V'\Delta W_O \\
&= \text{Attn}(W, \Delta W) + \text{Attn}(W') + g(W, \Delta W, W'),
\end{aligned} \tag{4}$$

it is observed that this scenario also introduces an entangled term $g(W, \Delta W, W')$ between the LoRA and the IP-Adapter, which interferes with both modules.

3.3 Latent Mixture

To address module interference, we design a latent mixture method for modules by aggregating their intermediate latent variables in the latent space between network blocks of the U-Net. Following ControlNet, we use *network block* as a unified term for the down-sampling, middle, and upsampling blocks and denote it as $\mathcal{F}(\Theta)$, where Θ represents all parameters in the block $\mathcal{F}$. For modules, we represent LoRA with the parameters $\Delta\Theta$, IP-Adapter with Θ', ControlNet as $\mathcal{M}_{\text{CNet}}(\Phi)$, and T2I-Adapter as $\mathcal{M}_{\text{T2I}}(\Psi)$.

Consider a generalized case where a diffusion model incorporates n_1 LoRAs, n_2 IP-Adapters, n_3 ControlNets, and n_4 T2I-Adapters, with the input to a network block as x and the conditions as c, the latent mixture at this block can be expressed as:

$$\begin{aligned}
&\sum_{i=1}^{n_1} \mathcal{F}(x, c_{\text{LoRA},i}; \Theta, \Delta\Theta_i) + \sum_{j=1}^{n_2} \mathcal{F}(x, c_{\text{IP},j}; \Theta, \Theta'_j) \\
&+ \sum_{k=1}^{n_3} \mathcal{M}_{\text{CNet}}(x, c_{\text{CNet},k}; \Phi_k) + \sum_{l=1}^{n_4} \mathcal{M}_{\text{T2I}}(c_{\text{T2I},l}; \Psi_l).
\end{aligned} \tag{5}$$

Compared to *direct merge* where additional conditions and parameters are loaded into the model without any dedicated design:

$$\begin{aligned}
&\mathcal{F}\big(x, \{c_{\text{LoRA}}\}^{n_1}, \{c_{\text{IP}}\}^{n_2}, \{c_{\text{CNet}}\}^{n_3}, \{c_{\text{T2I}}\}^{n_4}; \\
&\quad \Theta, \{\Delta\Theta\}^{n_1}, \{\Theta'\}^{n_2}, \{\Phi\}^{n_3}, \{\Psi\}^{n_4}\big),
\end{aligned} \tag{6}$$

latent mixture offers the following advantages: (i) Each condition processes only the parameters of its corresponding module, avoiding interference from other modules. (ii) The unified mixture in latent space is scalable to accommodate future and unseen modules. (iii) Aggregation by summing latent variables allows for convenient reweighting of different modules.

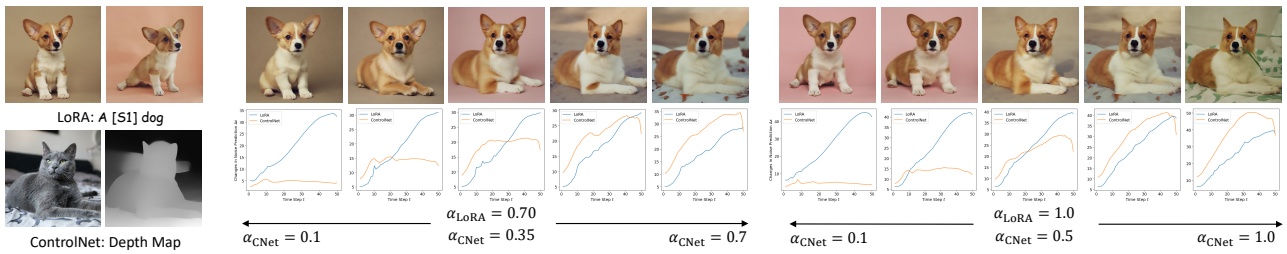


Figure 4: We visualize the generated images alongside the corresponding $L2$-norm changes in noise prediction. When the $L2$-norm changes are close to each other, the contribution of the given conditions to the images achieves a relative balance.

3.4 Multi-Inference Module Reweighting

Apart from the mixture mechanism, the proportion in the mixture also plays a crucial role. Notably, the optimal weight distribution across control modules can vary depending on individual preferences or specific generation goals. In light of this, our work investigates a default option: balancing the influence of different conditions on the generated image in the absence of user-specified condition weights.

Objective 1. Find a set of weights $\{\alpha\}^n$ for the modules $\{\mathcal{M}\}^n$ such that, for any $i, j \in [1, n]$, the weighted modules $\alpha_i\mathcal{M}_i$ and $\alpha_j\mathcal{M}_j$ have the balanced contribution to the generated image.

Statement 1. *The contribution of conditions to the images generated by a diffusion model correlates with the changes induced by the conditions on the noise predictions across all denoising steps. If the magnitude of the changes in noise predictions caused by the conditions is similar, the contributions of the conditions to the final images are approximately equal.*

Statement 1 is more of a hypothesis than a definitive claim, but it is reasonable since the final image results from the cumulative effect of noise predictions across all denoising steps. While theoretical proof is unavailable, the empirical experiments visualized in Figure 4 have provided partial validation for this statement. Based on Statement 1, Objective 1 can be transformed into Objective 2.

Objective 2. Find a set of weights $\{\alpha\}^n$ for the modules $\{\mathcal{M}\}^n$ such that for any $i, j \in [1, n]$, the weighted modules $\alpha_i\mathcal{M}_i$ and $\alpha_j\mathcal{M}_j$ leads to the equal $L2$-norm changes in noise predictions:

$$\begin{aligned} &\forall i, j \in [1, n], t \in [1, T], \\ &||\epsilon_\theta(x_t, t, c_i, \alpha_i\mathcal{M}_i) - \epsilon_\theta(x_t, t)||_2 \\ &= ||\epsilon_\theta(x_t, t, c_j, \alpha_j\mathcal{M}_j) - \epsilon_\theta(x_t, t)||_2, \end{aligned} \tag{7}$$

where T is the denoising steps and $\epsilon_\theta(\cdot)$ is the predicted noise with the parameters θ of the U-Net at the timestep t.

Statement 2. *For each denoising step, the changes in noise predictions caused by conditions exhibit an approximately linear relationship with the control module weights, provided the weights are sufficiently small.*

To prove Statement 2, we first divide the U-Net into multiple network blocks, then further into ResNet blocks and Transformer blocks, and ultimately break down into attention layers, convolutional layers, linear layers, activation layers, normalization layers, and matrix transformation layers. Each layer is analyzed individually for the changes in results after incorporating weighted modules. Herein, we take the attention layer as an example.

Suppose the input is x and module weight is α, the change in a self-attention $\text{Attn}(W)$ with **weighted LoRA** can be derived by starting from the dot product score:

$$\begin{aligned} \Delta\text{Score} &= \text{Score}(W, \Delta W) - \text{Score}(W) \\ &= (Q + \alpha\Delta Q)(K + \alpha\Delta K)^T - QK^T \\ &= \alpha(Q\Delta K^T + \Delta QK^T) + \alpha^2\Delta Q\Delta K^T. \end{aligned} \tag{8}$$

For Softmax, it can be analyzed with a first-order approximation:

$$\begin{aligned} &\text{Softmax}(\text{Score}(W, \Delta W))_{ij} \\ &= \frac{\exp(\text{Score}(W, \Delta W)_{ij})}{\sum_k \exp(\text{Score}(W, \Delta W)_{ik})} \\ &= \frac{\exp(\text{Score}(W)_{ij})\exp(\Delta\text{Score}_{ij})}{\sum_k \exp(\text{Score}(W)_{ik})\exp(\Delta\text{Score}_{ik})}. \end{aligned} \tag{9}$$

According to Taylor's Formula, there are $\exp(a) \approx 1 + a$ and $\frac{1}{a+b} \approx \frac{1}{a} - \frac{a}{b^2}$ when $b \ll a$. If we abbreviate $\sum_k \exp(\text{Score}(W)_{ik})\Delta\text{Score}_{ik}$ to ΔZ and $\sum_k \exp(\text{Score}(W)_{ik})$ to Z, Eq. (9) becomes:

$$\begin{aligned} &\text{Softmax}(\text{Score}(W, \Delta W))_{ij} \\ &\approx \frac{\exp(\text{Score}(W)_{ij})(1 + \Delta\text{Score}_{ij})}{Z + \Delta Z} \\ &\approx \frac{\exp(\text{Score}(W)_{ij})}{Z} + \frac{\exp(\text{Score}(W)_{ij})}{Z}\Delta\text{Score} \\ &\quad - \frac{\Delta Z}{Z^2}\exp(\text{Score}(W)_{ij})(1 + \Delta\text{Score}_{ij}). \end{aligned} \tag{10}$$

For a small α, the change in Softmax function becomes:

$$\begin{aligned} \Delta\text{Softmax} &= \text{Softmax}(\text{Score}(W, \Delta W)) - \text{Softmax}(\text{Score}(W)) \\ &\approx \left(\text{Softmax}(\text{Score}(W)(Q\Delta K^T + \Delta QK^T)\right)\alpha. \end{aligned} \tag{11}$$

If we ignore all second-order terms such as $\alpha^2\Delta Q\Delta K^T$ and negligible terms such as $\frac{\Delta Z}{Z^2}$, we can derive an approximately linear relationship between the module weight and the attention change:

$$\begin{aligned} &\text{Attn}(W, \Delta W) \\ &= \text{Softmax}(\text{Score}(W, \Delta W))(V + \alpha\Delta V)(W_O + \alpha W_O) \\ &= \text{Attn}(W) + \Delta\text{Attn}_1 + \Delta\text{Attn}_2 + ..., \end{aligned} \tag{12}$$

where ΔAttn_d is the d^{th}-order term in the form of $\alpha^d h_d(x, W, \Delta W)$. When the input x and the parameters W and ΔW are fixed, the result of the function h_1 is a constant. Ignoring high-order terms, the attention change is ΔAttn_1, exhibiting an approximately linear relationship with the weight α.

Table 1: Quantitative comparison between baselines and ModuleTeam method. "ModuleTeam (LoRAs)" represents the version of ModuleTeam that includes only LoRAs. The symbol "↑" indicates that higher values correspond to better performance, and vice versa. The best results are in bold. Since the baselines cannot handle all conditions, unavailable results are marked as "-".

Condition	Method	Subject (DINO ↑)	Style (CLIP-I ↑)	Image (CLIP-I ↑)	Canny (SSIM ↑)	Sketch (SSIM ↑)	Depth (MSE ↓)
Subject-Subject	Multiple LoRAs	0.4389	-	-	-	-	-
	Cones 2	0.2382	-	-	-	-	-
	FastComposer	0.2933	-	-	-	-	-
	Custom Diffusion	0.3935	-	-	-	-	-
	ModuleTeam (LoRAs)	**0.4784**	-	-	-	-	-
Subject-Style	Multiple LoRAs	0.5842	0.5813	-	-	-	-
	ZipLoRA	0.6530	0.5321	-	-	-	-
	ModuleTeam (LoRAs)	**0.6656**	**0.6022**	-	-	-	-
Image-Image	Multiple IP-Adapters	-	-	0.6950	-	-	-
	ModuleTeam (IP-Adapters)	-	-	**0.7436**	-	-	-
Canny-Sketch Canny-Depth Sketch-Depth	Multiple T2I-Adapters	-	-	-	0.5906	0.8392	90.17
	Multiple ControlNets	-	-	-	0.8515	0.8035	90.17
	DiffBlender	-	-	-	-	0.6619	96.94
	Uni-ControlNet	-	-	-	0.5168	0.7002	92.51
	ModuleTeam (T2I-Adapters)	-	-	-	0.6091	**0.8634**	87.35
	ModuleTeam (ControlNets)	-	-	-	**0.8776**	0.8421	**82.41**
All Conditions	Direct Merge	0.3660	0.5040	0.6218	0.4848	0.7113	108.3
	ModuleTeam	**0.4104**	**0.5582**	**0.7027**	**0.6466**	**0.7571**	**98.82**

With **weighted IP-Adapter**, the attention change is:

$$\begin{aligned}\Delta\text{Attn} &= \text{Attn}(W, W') - \text{Attn}(W) \\ &= \left(\text{Softmax}\left(\frac{Q(K')^T}{\sqrt{d}}\right)V'W_O\right)\alpha.\end{aligned} \tag{13}$$

It is clear that ΔAttn is linearly related to α.

With **weighted ControlNet or T2I-Adapter**, the input x becomes $x+\alpha\Delta x$, where Δx is $\mathcal{M}_{\text{CNet}}(x, c_{\text{CNet}}; \Phi)$ or $\mathcal{M}_{\text{T2I}}(x, c_{\text{T2I}}; \Psi)$. In this case, by rewriting ΔQ from $x(W_Q + \Delta W_Q)$ to $(x + \alpha\Delta x)W_Q$, Eq. (8) to (12) still hold. Therefore, the conclusion of the approximately linear relationship remains valid.

The analysis of other layers is in the Supplementary Material. Overall, the U-Net ϵ_θ with weighted modules $\alpha\mathcal{M}$ can be abstracted into the following expression:

$$\begin{aligned}\epsilon_\theta(x_t, t, c, \alpha\mathcal{M}) &= \epsilon_\theta(x_t, t) + \Delta\epsilon_1 + \Delta\epsilon_2 + \dots \\ &\approx \epsilon_\theta(x_t, t) + \alpha \cdot h_1(x, t, c, \theta, \mathcal{M}),\end{aligned} \tag{14}$$

where $\Delta\epsilon_1$ is the first-order change linearly related to α with the fixed x_t, t, θ, and $\mathcal{M}$. Therefore, Statement 2 is proved. From Eq. (14), we derive $||\epsilon_\theta(x_t, t, c, \alpha\mathcal{M}) - \epsilon_\theta(x_t, t)||_2 \propto \alpha$, so Objective 2 can be further transformed into Objective 3.

Objective 3. Estimate the slope set $\{k\}^n$ of the approximately linear relationship between module weights and the changes in noise prediction. Find a set of weights $\{\alpha\}^n$ such that for any $i, j \in [1, n]$, there is $\alpha_i k_i = \alpha_j k_j$ to ensure equal $L2$-norm changes, which means:

$$\alpha_1 : \alpha_2 : \dots : \alpha_n = \frac{1}{k_1} : \frac{1}{k_2} : \dots : \frac{1}{k_n}. \tag{15}$$

It is worth noting that the linear relationship is approximate primarily because timestep t is in a range $[1, T]$ and input x_t are not fixed but depends on random noise input and additional conditions besides t. Therefore, we need to sample a set of module weights $\mathcal{S}_\alpha = [\alpha_{ij}]_{j\in[1,m]}^{i\in[1,\,n]}$, conduct multiple inference processes, and calculate the corresponding set of changes in noise prediction $\mathcal{S}_{\Delta\epsilon} = [\Delta\epsilon_{ij}]_{j\in[1,m]}^{i\in[1,\,n]}$, where m is the inference number. Thus, we can estimate k_i that satisfies $\mathcal{S}_{\alpha_i} k_i = \mathcal{S}_{\Delta\epsilon_i}$:

$$\begin{aligned}&S_{\alpha_i} = S_\alpha[i, :],\ \mathcal{S}_{\Delta\epsilon_i} = \mathcal{S}_{\Delta\epsilon}[i, :], \\ &k_i = (S_{\alpha_i}^T S_{\alpha_i})^{-1} S_{\alpha_i}^T \mathcal{S}_{\Delta\epsilon_i},\ i \in [1, n],\end{aligned} \tag{16}$$

and further determine the weight proportions between modules.

This reweighting strategy significantly reduces the effort and time required for users to carefully perform grid searches and evaluate different weight combinations for multiple modules. Suppose the time complexity of a single search is $O(1)$, the search step size is s, and the weight range is $[0, 1]$, the time complexity of a full grid search over n modules is $O((1/s)^n)$. In contrast, our method requires only $O((1/s) + m)$, where $O(m)$ accounts for sampling by multiple inferences and $O(1/s)$ corresponds to determining the specific weights. In practice, we fix the maximum weight to 1 and determine the remaining weights accordingly, reducing the overall complexity to $O(m)$.

4 Experiments

In this section, we describe the experimental setup (Section4.1), present the performance of ModuleTeam through a comparative study (Section 4.2), and provide further analysis via ablation studies (Section 4.3). Finally, we detail the empirical way to select the hyperparameter, namely the inference number m (Section 4.4).

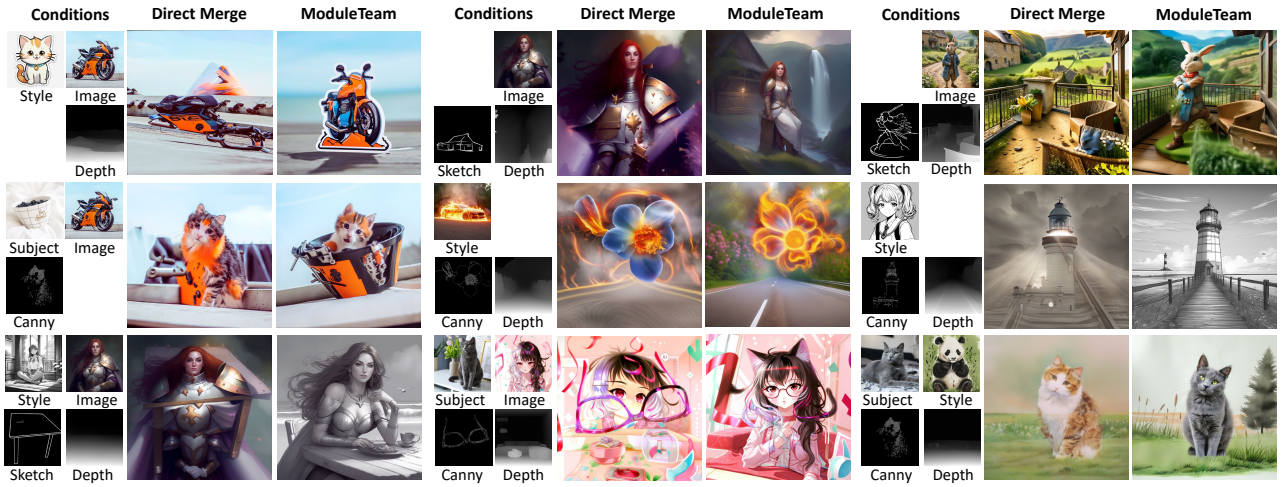


Figure 5: Qualitative comparison on multi-condition generation.

4.1 Experimental Setup

Datasets. We construct a dataset for open-set multi-conditional image generation, consisting of 10 subjects, 10 styles, 20 images, 10 Canny edge maps, 10 sketch maps, and 20 depth maps. The dataset is sourced from DreamBench [29], CustomConcept101 [16], IP-Adapter dataset [37], HuggingFaces, Civitai, and Unsplash. For a fair and robust evaluation, we generate 1000 images for each condition combination, such as subject-image or style-sketch-depth. Prompts are constructed by following the format of "an image of [style], [image], [canny], [depth]". Dataset samples and implementation details are provided in the Supplementary Material.

Comparable Methods. We compare our method with existing multi-conditional image generation approaches based on the types of conditions they support. Cones 2 [19], FastComposer [36], and Custom Diffusion [16] are designed for generating images conditioned on multiple subjects or concepts. ZipLoRA [31] focuses on the combination of subject and style. Cocktail [12], DiffBlender [14], and Uni-ControlNet [40] integrate multiple structural conditions, such as Canny edge, sketch, and depth maps.

Evaluation Metrics. To evaluate controllability, we adopt specific metrics tailored to different conditions, following previous works. DINO [2] is the average pairwise cosine similarity between ViT-S/16 DINO embeddings, which is recommended to measure subject fidelity [29]. CLIP-I [25] is the average pairwise cosine similarity between ViT-L/14 CLIP embeddings and is commonly used to assess style [31, 32] and image fidelity [37]. SSIM (Structural Similarity Index Measure) evaluates the structural similarity between Canny edge or sketch maps. MSE (Mean Squared Error) is employed to quantify pixel-wise differences in depth maps.

4.2 Main Results

Quantitative Results. Table 1 presents the main results of our quantitative experiments. Following prior works, we select the multi-conditional generation tasks that allow for fair comparisons with state-of-the-art methods. In addition, we include the direct merging of modules in the same type as competitive baselines. Finally, we evaluate the performance of ModuleTeam under the most general setting where all condition types and their corresponding

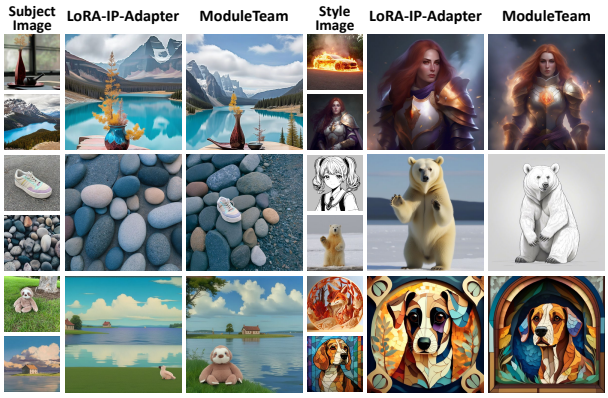


Figure 6: Subject-image and style-image generation.

Figure 7: Subject-depth and style-canny generation.

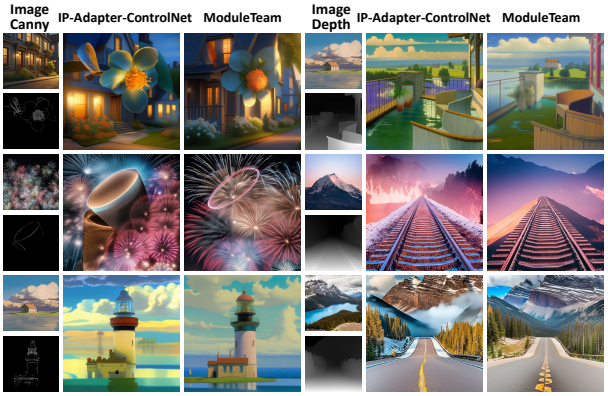


Figure 8: Image-canny and image-depth generation.

modules are integrated. The results demonstrate that ModuleTeam consistently outperforms all baselines while supporting flexible and scalable integration of diverse conditions.

Qualitative Results. Figures 5 to 8 showcase the qualitative results, focusing on combinations of conditions corresponding to modules of different types. It can be observed that the direct merge baseline often overemphasizes certain conditions while overlooking others. In contrast, ModuleTeam maintains subject consistency while effectively preserving the intended style and image in both subject-style and subject-image generation. Similarly, in image-Canny and image-depth generation, ModuleTeam produces more natural and coherent outputs that better integrate conditions. Additional qualitative results are in the Supplementary Material.

4.3 Ablation Studies

To evaluate the effectiveness of the two key components of ModuleTeam, latent mixture and module reweighting, we conduct an ablation study on subject-subject and style-image generation tasks. As shown in Table 2 and Figure 9, both components play an indispensable role in our method. Module reweighting prevents certain conditions from being suppressed by others, ensuring that all specified conditions are preserved. Meanwhile, latent mixture mitigates mutual interference between modules, thereby improving fidelity to each individual condition.

Table 2: Ablation on latent mixture and module reweighting.

	Subject-Subject (DINO ↑)	Style-Image (CLIP-I ↑)	(CLIP-I ↑)
Direct Merge	0.4389	0.5333	0.8881
w/o Reweighting	0.4724	0.5384	0.8811
w/o Latent Mixture	0.4413	0.5198	0.9004
ModuleTeam Full	**0.4784**	**0.5419**	**0.9100**

4.4 Hyperparameter Selection

ModuleTeam introduces only one hyperparameter, the inference number m used in the multi-inference module reweighting. This

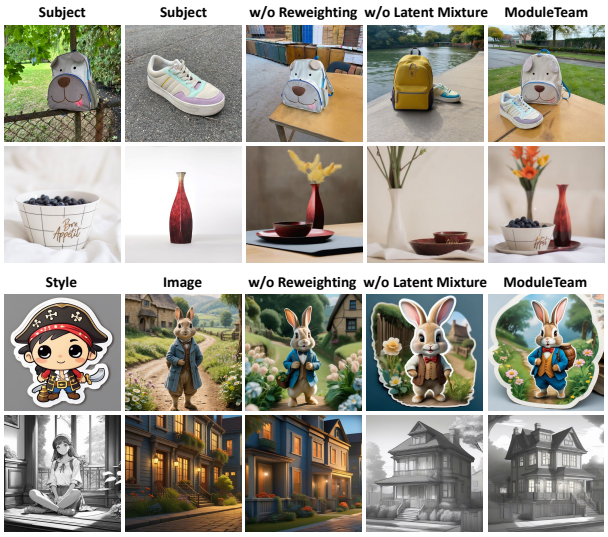


Figure 9: Ablation on latent mixture and module reweighting.

hyperparameter influences both the accuracy of estimating k and the computational cost, thus requiring a trade-off in selection. To determine a suitable value for m, we perform experiments by sampling different conditions and weight combinations 1000 times to compute a reference k value for each condition, which we treat as ground truth. We then plot the relative error against varying values of m to identify a small m that keeps the relative error of k below 5%. As shown in Figure 10, we set $m = 20$ for our method.

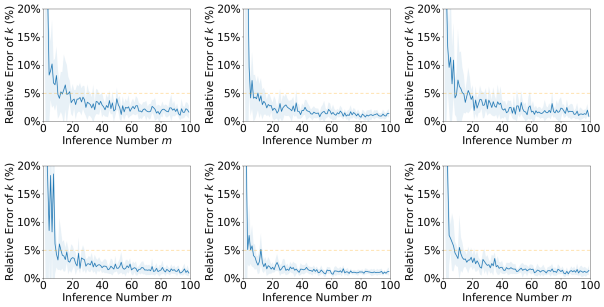


Figure 10: Hyperparameter selection of inference number m. To ensure the relative error of k within 5%, we select m as 20.

5 Conclusion

In this paper, we propose a training-free latent mixture method to incorporate arbitrary control modules for open-set multi-conditional image generation. Specifically, we design a latent mixture approach to effectively mitigate module interference and a multi-inference module reweighting strategy to balance module contributions during generation. Extensive experiments demonstrate that ModuleTeam not only outperforms existing approaches but also exhibits strong flexibility and scalability in handling diverse types and varying numbers of target conditions.

Acknowledgments

This work is supported by National Natural Science Foundation of China No. 62222209, the National Key Research and Development Program of China No. 2023YFF1205001, Beijing National Research Center for Information Science and Technology under Grant No. BNR2023TD03006, and Beijing Key Lab of Networked Multimedia.

References

[1] Pu Cao, Feng Zhou, Qing Song, and Lu Yang. 2024. Controllable Generation with Text-to-Image Diffusion Models: A Survey. *arXiv preprint arXiv:2403.04279* (2024).

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9650–9660.

[3] Hong Chen, Xin Wang, Guanning Zeng, Yipeng Zhang, Yuwei Zhou, Feilin Han, Yaofei Wu, and Wenwu Zhu. 2025. VideoDreamer: Customized Multi-Subject Text-to-Video Generation with Disen-Mix Finetuning on Language-Video Foundation Models. *IEEE Transactions on Multimedia* (2025).

[4] Hong Chen, Xin Wang, Yipeng Zhang, Yuwei Zhou, Zeyang Zhang, Siao Tang, and Wenwu Zhu. 2024. Disenstudio: Customized multi-subject text-to-video generation with disentangled spatial control. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 3637–3646.

[5] Hong Chen, Yipeng Zhang, Xin Wang, Xuguang Duan, Yuwei Zhou, and Wenwu Zhu. 2024. DisenDreamer: Subject-driven text-to-image generation with sample-aware disentangled tuning. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 8 (2024), 6860–6873.

[6] Hong Chen, Yipeng Zhang, Simin Wu, Xin Wang, Xuguang Duan, Yuwei Zhou, and Wenwu Zhu. 2023. Disenbooth: Identity-preserving disentangled tuning for subject-driven text-to-image generation. In *The Twelfth International Conference on Learning Representations*.

[7] Zhuowei Chen, Shancheng Fang, Wei Liu, Qian He, Mengqi Huang, Yongdong Zhang, and Zhendong Mao. 2023. Dreamidentity: Improved editability for efficient face-identity preserved image generation. *arXiv preprint arXiv:2307.00300* (2023).

[8] Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao, Rui Zhao, Shuning Chang, Weijia Wu, et al. 2024. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. *Advances in Neural Information Processing Systems* 36 (2024).

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[10] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. 2024. Style aligned image generation via shared attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4775–4785.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[12] Minghui Hu, Jianbin Zheng, Daqing Liu, Chuanxia Zheng, Chaoyue Wang, Dacheng Tao, and Tat-Jen Cham. 2023. Cocktail: Mixing multi-modality control for text-conditional image generation. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[13] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. 2023. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778* (2023).

[14] Sungnyun Kim, Junsoo Lee, Kibeom Hong, Daesik Kim, and Namhyuk Ahn. 2023. Diffblender: Scalable and composable multimodal text-to-image diffusion models. *arXiv preprint arXiv:2305.15194* (2023).

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[16] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. 2023. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1931–1941.

[17] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. 2023. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22511–22521.

[18] Han Lin, Jaemin Cho, Abhay Zala, and Mohit Bansal. 2024. Ctrl-Adapter: An Efficient and Versatile Framework for Adapting Diverse Controls to Any Diffusion Model. *arXiv preprint arXiv:2404.09967* (2024).

[19] Zhiheng Liu, Yifei Zhang, Yujun Shen, Kecheng Zheng, Kai Zhu, Ruili Feng, Yu Liu, Deli Zhao, Jingren Zhou, and Yang Cao. 2023. Cones 2: Customizable image synthesis with multiple subjects. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*. 57500–57519.

[20] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. 2024. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 4296–4304.

[21] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).

[22] Zirui Pan, Xin Wang, Yipeng Zhang, Hong Chen, Kwan Man Cheng, Yaofei Wu, and Wenwu Zhu. 2025. Modular-Cam: Modular Dynamic Camera-view Video Generation with LLM. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 6363–6371.

[23] Ryan Po, Guandao Yang, Kfir Aberman, and Gordon Wetzstein. 2024. Orthogonal adaptation for modular customization of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7964–7973.

[24] Can Qin, Shu Zhang, Ning Yu, Yihao Feng, Xinyi Yang, Yingbo Zhou, Huan Wang, Juan Carlos Niebles, Caiming Xiong, Silvio Savarese, et al. 2023. Unicontrol: A unified diffusion model for controllable visual generation in the wild. *arXiv preprint arXiv:2305.11147* (2023).

[25] 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*. PMLR, 8748–8763.

[26] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* 1, 2 (2022), 3.

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 234–241.

[29] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. 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*. 22500–22510.

[30] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems* 35 (2022), 36479–36494.

[31] Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. 2025. Ziplora: Any subject in any style by effectively merging loras. In *European Conference on Computer Vision*. Springer, 422–438.

[32] Kihyuk Sohn, Nataniel Ruiz, Kimin Lee, Daniel Castro Chin, Irina Blok, Huiwen Chang, Jarred Barber, Lu Jiang, Glenn Entis, Yuanzhen Li, et al. 2023. Styledrop: Text-to-image generation in any style. *arXiv preprint arXiv:2306.00983* (2023).

[33] Dani Valevski, Danny Lumen, Yossi Matias, and Yaniv Leviathan. 2023. Face0: Instantaneously conditioning a text-to-image model on a face. In *SIGGRAPH Asia 2023 Conference Papers*. 1–10.

[34] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[35] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. 2024. Videocomposer: Compositional video synthesis with motion controllability. *Advances in Neural Information Processing Systems* 36 (2024).

[36] Guangxuan Xiao, Tianwei Yin, William T Freeman, Frédo Durand, and Song Han. 2024. Fastcomposer: Tuning-free multi-subject image generation with localized attention. *International Journal of Computer Vision* (2024), 1–20.

[37] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. 2023. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721* (2023).

[38] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3836–3847.

[39] Yipeng Zhang, Xin Wang, Hong Chen, Chenyang Qin, Yibo Hao, Hong Mei, and Wenwu Zhu. 2025. ScenarioDiff: text-to-video generation with dynamic transformations of scene conditions. *International Journal of Computer Vision* 133, 7 (2025), 4909–4922.

[40] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. 2024. Uni-controlnet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems* 36 (2024).