Generating Video with Conditional Control Diffusion Model

Xiaoyang Gao, Zheng Wen and Tao Yang

March 4, 2024

Generating Video with Conditional Control Diffusion Model

XiaoYang Gao [1][0009-0008-0512-4228] Zheng Wen [1] and Tao Yang [1,2,*]

[1] School of Information and Intelligence Engineering, University of Sanya, Sanya 572022, Hainan, China

[2] Academician Workstation of Chunming Rong, University of Sanya, Sanya 572022, Hainan, China

* Correspondence: syauyt@160.com (T.Y.)

Abstract. *We present the Conditional Control Diffusion Model (CCDM), a neural network that converts a text-to-image (T2I) model into a video model by using conditional control while keeping the image quality of the original model. CCDM first trains on real video data, creating a composite model to fuse multiple frames and learn action priors. Then, CCDM adopts the Stable Diffusion architecture and integrates the T2I model, ensuring no changes to the T2I model during video generation. Finally, CCDM feeds back the generated frames to the model as feedback, reducing flickering caused by content changes. We test CCDM on various T2I models from CivitAI with different styles and features. Using prompts from the T2I model's website, we generate videos and show that CCDM can produce dynamic information and handle generation tasks with 8GB VRAM. CCDM has excellent potential for video generation applications.*

Keywords: Diffusion Model, Video Generation, Conditional Control, Computer Vision, Text-to-Image.

1 Introduction

Text-to-image models are a kind of artificial intelligence technology that can generate high-quality images based on natural language descriptions. In recent years, they have received widespread attention and application in academic and non-academic circles. According to the latest survey, text-to-image models have been applied to various domains, such as art, education, entertainment, and social media, providing a low-barrier AI creative pathway for non-researcher users (such as artists and hobbyists), enabling them to create novel and diverse visual content. For example, VQGAN-CLIP [25]has generated realistic paintings of various styles and themes, such as a raccoon queen wearing a red French royal gown. DALL-E[11] has been used to create educational illustrations, such as a transparent sculpture of a duck made of glass. Imagen has been used to produce photorealistic images for social media posts, such as a photo of a corgi dog riding a bike in Times Square.

To further enhance the generative capabilities of text-to-image models, some lightweight personalized methods, such as Dreambooth [4] and Low-Rank Adaptation [5] (LoRA), have emerged. These methods involve personalizing the model through fine-tuning on small datasets and optimizing with consumer-grade devices. Such fine-tuning significantly improves the quality of personalized content generation, allowing users to introduce new concepts or styles into pre-trained text-to-image models at a lower cost. On model-sharing platforms, such as CivitAI[6] and HuggingFace[1], users can easily access various personalized models.

However, training text-to-video models requires abundant high-quality videos and computational resources, hindering further research and application in the field. Recent text-to-video generation methods have attempted to incorporate temporal modelling into the original T2I models and adjust them on video datasets. This challenges personalized T2I models, as non-professional users often need help managing the sensitive hyperparameter tuning, personalized video collection, and intensive computational resource demands.

Given the widespread application of videos, our research aims to explore whether it is possible to transform existing T2I models into video generation models while maintaining the original model's image quality. This exploration is significant for advancing research in text-to-video generation. In this study, we propose a method called CCDM (Conditional Control Diffusion Model), which allows any personalized diffusion model to output high-quality continuous video images without adjustment. Personalized diffusion models typically focus on generating images in different domains and styles, and there is diversity among these models, making it impractical to train them separately on large-scale datasets. Therefore, our method turns to designing a composite network module, which, through this strategy, enables most T2I diffusion models to have excellent video generation capabilities. The composite module consists of two parts: the action modelling module and the multi-frame fusion module. The action modelling module is attached to the user-invoked base T2I model and learns reasonable motion priors by fine-tuning large-scale video clips. Subsequently, the action modelling module adds the first layer of video content for video generation to a group of T2I models.

Our study introduces a keyframe control to enhance generated videos. Within the action modelling module, users can specify keyframes, each corresponding to a specific action or scene envisioned by the user. In the multi-frame fusion module, information from the keyframes guides the video frame generation process, better preserving the user-specified actions and scenes. Smooth transitions are achieved by interpolating and fusing between keyframes, resulting in more natural and coherent videos. This keyframe control method allows users to control the video generation process intuitively. The interactive design improves the user experience and ensures that the generated videos align with users' personalized needs.

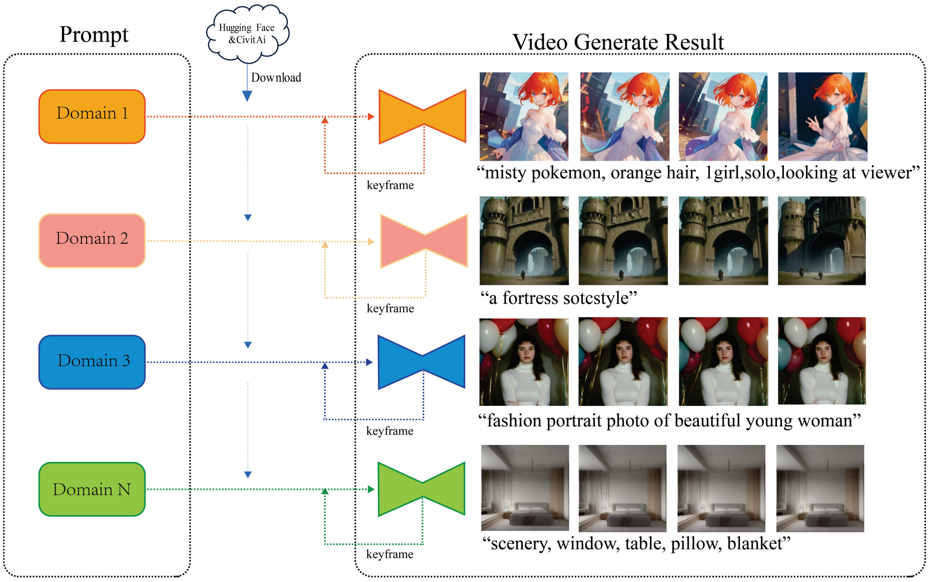

Prompt
Hugging Face &CivitAi
Download
Video Generate Result
Domain 1
Domain 2
Domain 3
Domain N
keyframe
keyframe
keyframe
keyframe
“misty pokemon, orange hair, 1girl,solo,looking at viewer”
“a fortress sotcstyle”
“fashion portrait photo of beautiful young woman”
“scenery, window, table, pillow, blanket”

Fig. 1. The workflow of CCDM. After providing prompts from various domains, they are input into the T2I model derived from either the HuggingFace[1] or CivitAI[6] platform. The CCDM method is incorporated during the generation process to produce high-quality video sequences.

CCDM has been evaluated on multiple representative T2I models, including those in anime style and real-world image domains. Without specific adjustments, most personalized T2I models can achieve video generation by incorporating the trained composite module, thus reducing the flickering commonly seen in T2I video generation methods. CCDM enables users to obtain high-quality personalized videos using standard T2I models, achieving multi-frame fusion to mitigate flickering.

In summary, this paper begins with an introduction (Section 1), and we set the stage for our work. Section 2 provides a succinct review of related work, touching upon fine-tuning neural networks, diffusion models for image synthesis, and personalized text-to-image (T2I) animation. Section 3 delves into the specifics of our proposed method, encompassing the training process (Section 3.1) and logical design with loss computation (Section 3.2). Section 4 details our experiments, including comparisons (Section 4.3) and an ablation study (Section 4.4). The section concludes the paper, summarizing contributions and critical findings. Finally, Section 6 addresses limitations and outlines potential future directions.

2 Related Work

2.1 Finetuning Neural Networks

LoRA

The study emphasizes the significance of LoRA [5] due to its application of models that were fine-tuned using this method. The primary innovation of LoRA [5] is the incorporation of trainable low-rank matrices into each Transformer layer within the pre-trained model. This technique markedly diminishes the parameter count of the pre-trained model without compromising its integrity. LoRA [5] surpasses traditional adaptation methods by delivering enhanced model quality and expedited training speeds, all without additional inference latency. Furthermore, LoRA [5] is compatible with established adaptation strategies, including prefix tuning and layer-wise tuning, which bolsters its overall performance.

Adapter

Adapter methods are extensively used in Natural Language Processing (NLP) to tailor a pre-trained transformer model for various tasks by integrating new modular layers [24,26]. Adapters have been widely applied in computer vision in incremental learning [28] and domain adaptation [27]. CLIP[12] often uses them to adapt pre-trained backbone models to diverse tasks[30,31,32]. Adapters have recently shown promising results in vision transformers, including visual transformers and ViT-Adapter [33]. For instance, the T2I-Adapter [23] adapts Stable Diffusion(SD) [2] to external conditions.

2.2 Diffusion Model for image synthesis

Diffusion models have gained the attention of both researchers and artists for their exceptional capability to create highly detailed images [19]. These models are now being applied in fields beyond image synthesis [16], such as motion and 3D shape generation [37].

Progress in image-space diffusion has been made through various developments, including changes to parameterization, the introduction of advanced sampling techniques, the creation of more robust architectures, and the use of additional information for conditioning.

A particularly impactful method is text-conditioning, which utilizes embeddings from models like CLIP[12] or T5[34]. This technique has become a powerful means for artists to exert precise control over the outputs of diffusion models[9,13].

Latent diffusion model (LDM)

LDM [19]is an efficient variant of diffusion models by applying the diffusion process in the latent space rather than image space. LDM [19]contains two main components.

First, it employs an encoder $\mathcal{E}$ to compress the image $\boldsymbol{x}$ into a latent code $\boldsymbol{z} = \mathcal{E}(\boldsymbol{x})$,and then uses a decoder to reconstruct the image $\boldsymbol{x} \approx \mathcal{D}(\boldsymbol{z})$.Secondly, it learns the distribution of the image latent codes $\boldsymbol{z}_0 \sim p_{data}(\boldsymbol{z}_0)$ within the Denoising Diffusion Probabilistic Models (DDPM) framework. During the forward diffusion process, Gaussian noise is incrementally added at each time step t to obtain $\boldsymbol{z}_t$:

$$q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) = \mathcal{N}\left(\boldsymbol{z}_t; \sqrt{1-\beta_t}\boldsymbol{z}_{t-1}, \beta_t I\right) \tag{1}$$

where $\{\beta_t\}_{t=1}^{T}$ are the scale of noises, and T signifies the number of diffusion timesteps. The backward denoising process reverses the aforementioned diffusion process to predict less noisy $\boldsymbol{z}_{t-1}$:

$$p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t) = \mathcal{N}(\boldsymbol{z}_{t-1}; \mu_\theta(\boldsymbol{z}_t, t), \Sigma_\theta(\boldsymbol{z}_t, t)) \tag{2}$$

The μ_θ and Σ_θ are implemented with a denoising model ϵ_θ with learnable parameters θ, which is trained with a simple objective:

$$\mathcal{L}_{simple} := \mathbb{E}_{\mathcal{E}(\boldsymbol{z}),\epsilon\sim\mathcal{N}(0,1),t}[\| \epsilon - \epsilon_\theta(\boldsymbol{z}_t, t) \|_2^2] \tag{3}$$

When generating new samples, we start from $z_T \sim \mathcal{N}(0,1)$ and employ Denoising Diffusion Implicit Models (DDIM) sampling to predict z_{t-1} of previous timestep:

$$\boldsymbol{z}_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{z_t - \sqrt{1-\alpha_t}\epsilon_\theta(z_t,t)}{\sqrt{\alpha_t}}\right)}_{\text{"predicted } \boldsymbol{z}_0\text{"}} + \underbrace{\sqrt{1-\alpha_{t-1}} \cdot \epsilon_\theta(\boldsymbol{z}_t, t)}_{\text{"direction pointing to } \boldsymbol{z}_t\text{"}}, \tag{4}$$

Where $\alpha_t = \prod_{i=1}^{t} (1-\beta_i)$. We use $z_{t\to0}$ to represent “predicted z0” at timestep t for simplicity.

Note that we use SD[2] (SD) $\epsilon_\theta(\boldsymbol{z}_t, t, \tau)$ as our base model, which is an instantiation of text-guided LDMs pre-trained on billions of image-text pairs. τ denotes the text prompt.

2.3 Personalized T2I animation

Due to the various limitations previously described and the novelty of the composite module setup in this paper, related research work still needs to be completed. While it is expected to add temporal structure to video generation to extend the video generation capabilities of T2I models, existing works have modified and updated the network parameters during the video generation process, compromising the domain knowledge of the original T2I models. In recent years, some works have reported their applications in personalized T2I model videos. For example, Tune-a-Video[20] addresses one-off video generation tasks through minor architectural modifications and sub-network tuning. Text2Video-Zero[22] introduces a training-free method that wraps pre-trained T2I models in videos, given predefined affine matrices. Align-Your-Latents [17], a text-to-video (T2V) model, trains separate temporal layers within the T2I model. The most closely related work to our study’s method is AnimateDiff [10], which adopts a simplified network design structure from Align-Your-Latents [17] and introduces an action module.

3 Method

CCDM is a neural network that transforms the T2I diffusion model into a video model with the help of conditional control. In this section, our study will introduce the model's training methods in 3.1 and CCDM's logical design and loss computation in 3.2.

3.1 Train

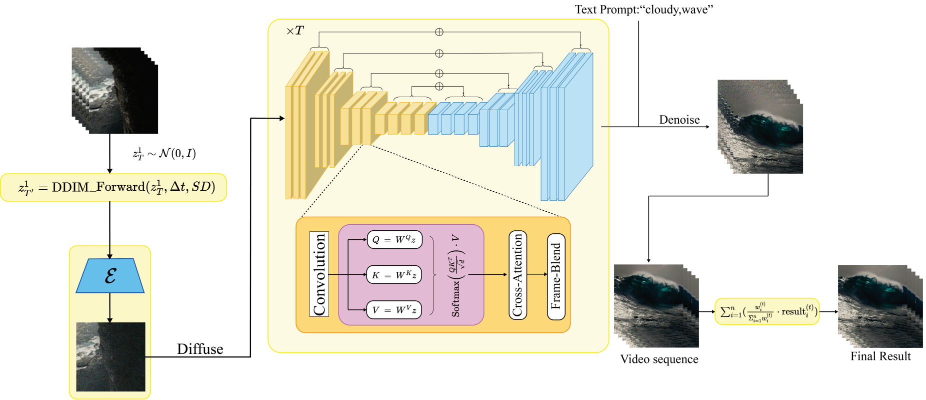

Text Prompt:"cloudy,wave"
$\times T$
$z_T^1 \sim \mathcal{N}(0, I)$
$z_{T'}^1 = \mathrm{DDIM\_Forward}(z_T^1, \Delta t, SD)$
$\mathcal{E}$
Diffuse
Denoise
Convolution
$Q = W^Q z$
$K = W^K z$
$V = W^V z$
$\mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V$
Cross-Attention
Frame-Blend
Video sequence
$\sum_{i=1}^n \left(\frac{w_i^{(t)}}{\sum_{i=1}^n w_i^{(t)}} \cdot \mathrm{result}_i^{(t)}\right)$
Final Result

Fig. 2. Our study imported data, starting with randomly sampled latent encodings z_T^1. Through DDIM's forward noise addition steps, our study obtained z_T^1 using the pre-trained SD model for model training. In the complete video training, the action module and multi-frame fusion module will be trained together. Through training, our study can enable the model to determine the action priors of the all the data information. The yellow and blue parts in the figure represent down-sampling and up-sampling methods, aiming to accelerate computational efficiency and learn higher-level representations in videos, making it easier for the model to capture the global structure of action priors in the video dataset. With the addition of cross-attention, CCDM can significantly enhance the consistency of the foreground subject in video motion. After the module training is completed, specific text prompts and personalized models are given, and according to frame weight enhancement, the video sequence is smoothed to obtain the final generated results.

Learning step-by-step denoising aims to generate samples from the training domain in image diffusion models. This denoising process can occur in pixel or latent space encoded from training data. Specifically, SD[2] uses a preprocessing method similar to VQ-GAN [14] to convert images from 512×512-pixel space into smaller latent images, as operating in latent space has been proven to stabilize the training process. To integrate CCDM into SD[2], we used a preprocessing method similar to VQ-GAN [14], as operating in this space has been proven to stabilize the training process [19].

Similar to VideoLDM, our study trained the action module using the WebVid-10M [8,15]dataset, which contains text-video data, primarily for real-world video data. To minimize flickering, we used a linear setting during training. Unlike SD[2]'s fixed β, our study used a strategy of linearly adjusting β, allowing it to change linearly from the start to the end of the training, with β values varying linearly from 0.0008 to 0.012 to accommodate generation needs outside the dataset (e.g., 2D anime). The training video clips were set to 24 frames to cater to most video formats. In actual testing, our study found that consumer-grade graphics cards could not bear the cost of excessive generation demands (often resolutions greater than 512x512 require 12GB or more of VRAM), so our study's training resolution was set to 256x256. However, this does not mean that artefacts will occur in a high-resolution generation. A 256 resolution can not only be extended to higher resolutions, such as 512, but also maintain a balance between training efficiency and visual quality. The processing approach here is analogous to a method [21] used in intrusion detection.

Due to the nature of videos being a continuous sequence of frames, we observed that even with ample textual input during the generation process, it is challenging to exert precise control over the real-time state of each frame. Here, by real-time state, we refer to the cognitive state of individuals regarding the motion patterns and speeds of objects in the video. For instance, setting the video to 8 or 24 frames per second and providing the cue "ocean waves" may not ensure full control over the real-time states of individual frames. Under different frame rates, relying solely on static images to generate the subsequent frame after the completion of the generation process can result in significant variations in speed. This discrepancy in speed variations becomes apparent in the viewer's perception, causing the waves in the video to appear inconsistently paced.

To address this issue, we modified the pipeline of SD [2]. When setting the overall frame rate to n, the model generates n frames at once. However, it is essential to note that this does not imply the simultaneous generation of the entire video. Instead, the model initially generates a denoised frame, improving the overall coherence of the sequence's motion.

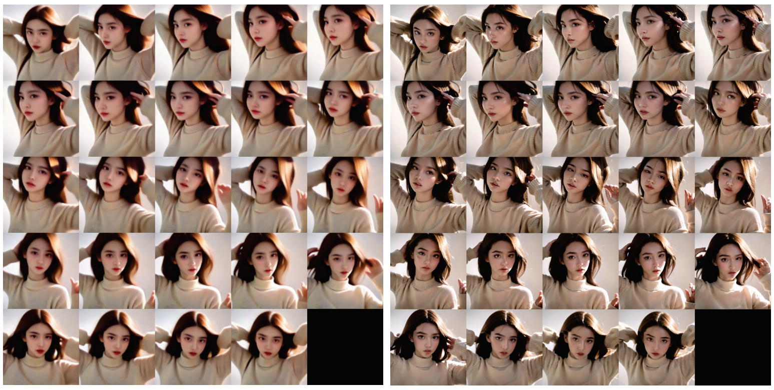


Fig. 3. The left image represents the outcome after 10 steps of generation, while the right image illustrates the final video result after 20 steps of generation.

Our study used four A100 graphics cards and completed training within the past four days. The generated images used an NVIDIA 4060 graphics card, and SD [2] used version v1.6.0. In Cross-Attention, we use the xFormers [3] optimization scheme and employ model hash calculations.

3.2 logical design and loss computation

Considering the text input's embedded representation as $\mathcal{E}(x_0^{1:N})$,the video input as y, and the noise term as ϵ, this study introduces the expectation operator $\mathbb{E}$ to describe the expected nature of the loss function. This expectation operation covers the average of all possible values for text input embeddings, video inputs, and noise terms. The expectation operation is as follows:

$$\mathbb{E}_{\mathcal{E}(x_0^{1:N}),y,\epsilon\sim\mathcal{N}(0,I),t} \tag{5}$$

In this study, a multivariate normal distribution $\mathcal{N}(0,I)$ is adopted, which, due to its characteristics of having a mean of zero and a covariance matrix as the identity matrix, ensures that the computation of the loss function obtains an average value for different text inputs, video inputs, and noise terms, thereby enhancing the robustness and generalizability of the model.

Next, we look at the main loss term:

$$\| \epsilon - \epsilon_\theta(z_t^{1:N}, t, \tau_\theta(y)) \|_2^2 \tag{6}$$

This is sampled from the standard normal distribution $(\mathcal{N}(0,I))$, where $\epsilon_\theta(z_t^{1:N}, t, \tau_\theta(y))$ depends on the model's parameters, the current time step t, the video input y, and the feature representation $z_t^{1:N}$, with $z_t$being the video frame feature representation at time t, N being all frames of the video, $\tau_\theta(y)$ used to convert the video input into the model's temperature parameter, to measure the difference between real noise and generated noise.

The design of the loss term for previous and subsequent frames considers the continuity of the video sequence, aiming to enable the model to capture the motion information and temporal correlation in the video. This term includes noise reconstruction items for 2k+1 frames adjacent to the current time step t, where k is the number of frames chosen before and after. To measure the difference between real noise and generated noise, the formula for previous and subsequent frames is as follows:

$$\sum_{i=-k}^{k} \| \epsilon - \epsilon_\theta(z_{t+i}^{1:N}, t+i, \tau_\theta(y)) \|_2^2 \tag{7}$$

The design of the keyframe term considers user-specified keyframes, aiming to make the generated video content more accurately match user expectations at these keyframes.

To enhance the versatility of video generation, this study provides a way for users to select keyframes and customize their keyframes. Users can use plugins like ControlNet[36] to enhance customization effects during the actual keyframe definition process. The result turns out that the generated video content can more accurately match user expectations through keyframes. This study assumes the specified keyframes as j.

$$\sum_{j\in\mathrm{keyframes}} \| \epsilon - \epsilon_\theta(z_j^{1:N}, j, \tau_\theta(y)) \|_2^2 \tag{8}$$

In the multi-frame fusion method, the main goal is to generate videos with spatiotemporal consistency. To achieve this goal, a strategy based on Pyramid Patch Matching is adopted, introducing additional constraints and losses by considering the relationship between previous and subsequent frames and user-specified keyframes. In the study's pyramid multi-scale Patch Matching style transfer method, an attention mechanism is introduced to more accurately capture the relationship between source image patches and target image patches. The specific expression of Patch Matching is the calculation formula for attention weights:

$$z' = \mathrm{Attention}(Q, K, V) = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V \tag{9}$$

Where $Q = W^Q z, K = W^K z, V = W^V z$ are the projections of the input feature map Z,$\sqrt{d}$ is the scaling factor. Utilizing the idea of pyramid multi-scale, Patch Matching is performed at different scales. For each layer of the pyramid, this study can establish such a formula to represent it:

$$z'_{\mathrm{pyramid}} = \mathrm{Attention}_{\mathrm{pyramid}}(Q_{\mathrm{pyramid}}, K_{\mathrm{pyramid}}, V_{\mathrm{pyramid}}) \tag{10}$$

In our research, we primarily utilize SD[2] as our main tool, which includes the capability of image-to-image functionality. Our study specifically employs this image-to-image approach with a preference for style transfer. To this end, we have also established a corresponding formula for balanced mode style transfer to target multi-frame weight balancing. We first define the weight of each frame as w_i,which is used to balance the contribution of each frame in the style transfer. For each frame i,with t beingthe current iteration, we update the weights as follows:

$$w_i^{(t+1)} = \frac{w_i^{(t)}}{w_i^{(t)}+1} + \frac{1}{w_i^{(t)}+1} \tag{11}$$

After incorporating style generation, the final result can be expressed as:

$$\mathrm{frame}_{\mathrm{target}}^{(t)} = \sum_{i=1}^{n} \left(\frac{w_i^{(t)}}{\sum_{i=1}^{n} w_i^{(t)}} \cdot \mathrm{result}_i^{(t)}\right) \tag{12}$$

we can deduce that the overall loss function is as follows:

$$\begin{aligned}\mathcal{L} = {} & \mathbb{E}_{\mathcal{E}(x_0^{1:N}),y,\epsilon\sim\mathcal{N}(0,I),t}[\| \epsilon - \epsilon_\theta(z_t^{1:N}, t, \tau_\theta(y)) \|_2^2 \\ & +\lambda_1 \sum_{i=-k}^{k} \| \epsilon - \epsilon_\theta\left(z_{t+i}^{1:N}, t+i, \tau_\theta(y)\right) \|_2^2 \\ & +\lambda_2 \textstyle\sum_{j\in \mathrm{keyframes}} \| \epsilon - \epsilon_\theta(z_j^{1:N}, j, \tau_\theta(y)) \|_2^2]\end{aligned} \tag{13}$$

λ_1 and λ_2 are balancing factors used to control the importance of the previous and subsequent frame terms and the keyframe term relative to the main loss term. Excessive λ values cause the model to lean towards controlling the consistency of previous and subsequent frames, while smaller λ values focus more on noise reconstruction.

Theoretically, for models with an anime style, smaller λ values yield better results, while larger λ values present better outcomes for realistic styles. However, influenced by LoRA and Dreambooth [4], different video models with the same parameters may exhibit different effects. Specific adjustments should be made for different models to achieve better results. Although we have set λ_2, it is actually an optional choice; when video generation through keyframes is not required, the value of λ_2 is 0.

We define the primary loss term using the square of the L2 norm, a decision made after careful consideration of model optimization and the training process. We aim to minimize the difference between generated and real noise, making the model-generated video more accurate at the current time step. By optimizing the primary loss term, we encourage the generated noise to match the real noise more consistently, thereby improving the accuracy of the generated video at the current time step. This design choice considers mathematical optimizability and the stability of model training to ensure that the generated video meets task requirements.

We define the primary loss term using the square of the L2 norm, a decision made after careful consideration of model optimization and the training process. We aim to minimize the difference between generated and real noise, making the model-generated video more accurate at the current time step. By optimizing the primary loss term, we encourage the generated noise to match the real noise more consistently, thereby improving the accuracy of the generated video at the current time step. This design choice considers mathematical optimizability and the stability of model training to ensure that the generated video meets task requirements.

Although the L1 loss function has advantages in robustness and is often used for feature selection because it leads to more parameters being zero, resulting in sparse solutions, in our model, this could reduce fitting ability, increase the number of iterations required for convergence, lead to different minima, and cause significant impacts such as information loss. Therefore, we weighed these factors and chose the loss function suitable for our task.

Our use of the L2 norm and attempt to avoid sparse solutions is explained as follows: Suppose X and Y are both n-dimensional vectors, $i.e., X = (x_1, x_2, x_3, \dots x_n), Y = (y_1, y_2, y_3, \dots y_n)$, The L1 norm is known as

$$\mathrm{D}(\mathrm{X},\mathrm{Y}) = \sum_{\mathrm{i}=1}^{\mathrm{n}} |\mathrm{x}_\mathrm{i} - \mathrm{y}_\mathrm{i}| \tag{14}$$

and the L2 norm is

$$||\mathrm{X}||_2 = \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x}_\mathrm{i}^2} \tag{15}$$

Correspondingly, the L1 loss function, known as the Least Absolute Deviations (LAD) or Least Absolute Errors (LAE), aims to minimize the total absolute difference between the target value y_i and the estimated value $f(x_i)$, so the corresponding loss function is

$$\mathrm{L} = \sum_{\mathrm{i}=1}^{\mathrm{n}} |\mathrm{y}_\mathrm{i} - \mathrm{f}(\mathrm{x}_\mathrm{i})| \tag{16}$$

The L2 norm loss function, also known as the Least Squares Error (LSE), minimizes the squared sum of the differences between the target value y_i and the estimated value $f(x_i)$, so the L2 loss function is

$$\mathrm{L} = \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{y_i} - \mathrm{f}(\mathrm{x_i}))^2 \tag{17}$$

When we assume $L(W)$ represents the loss without regularization, which is a hyperparameter controlling the size of regularization, the corresponding L1 loss function is modified to

$$\mathrm{L} = \mathrm{L(W)} + \lambda \sum_{\mathrm{i}=1}^{\mathrm{n}} |\mathrm{w_i}| \tag{18}$$

and the L2 loss function is modified to

$$\mathrm{L} = \mathrm{L(W)} + \lambda \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{w_i^2} \tag{19}$$

Now, calculating the gradient for one of the parameters w_i (the same applies to other parameters), where η is the step size and $sign(w_i)$ is, the sign function

$$\mathrm{w_i} > 0, \mathrm{sign}(\mathrm{w_i}) = 1; \mathrm{w_i} < 0, \mathrm{sign}(\mathrm{w_i}) = -1 \tag{20}$$

The gradient for L1 is derived as

$$\frac{\partial \mathrm{L}}{\partial \mathrm{w_i}} = \frac{\partial \mathrm{L(W)}}{\partial \mathrm{w_i}} + \lambda \mathrm{sign}(\mathrm{w_i}), \mathrm{w_i} = \mathrm{w_i} - \eta \frac{\partial \mathrm{L(W)}}{\partial \mathrm{w_i}} - \eta\lambda \mathrm{sign}(\mathrm{w_i}) \tag{21}$$

Similarly, for the L2 gradient, we have

$$\frac{\partial \mathrm{L}}{\partial \mathrm{w_i}} = \frac{\partial \mathrm{L(W)}}{\partial \mathrm{w_i}} + 2\lambda \mathrm{w_i}, \mathrm{w_i} = \mathrm{w_i} - \eta \frac{\partial \mathrm{L(W)}}{\partial \mathrm{w_i}} - \eta 2\lambda \mathrm{w_i} \tag{22}$$

When w_i is less than 1, the penalty term for L2 gradually decreases, while the L1 penalty remains large, so L1 causes the parameter to become zero, while L2 is unlikely to do so.

For our video generation task, we strive to achieve continuity and smoothness between generated video frames to avoid unnatural jumps or mutations, commonly called flickering, in this paper. The squared term of L2 regularization aligns more with our task requirements because it penalizes sparsity less and focuses more on the smoothness of parameters. During optimization, the squared term of L2 regularization helps ensure temporal coherence in the generated video sequence. Additionally, considering that some frames may cause larger errors due to significant motion or changes, the squared term of L2 regularization is more sensitive to significant errors, helping the model better adapt to these changes without being overly affected by outliers. Such design decisions enhance the robustness and adaptability of the model, thereby improving the overall quality of the videos generated.

4 Experiments

We implement CCDM with SD[2] to test various models, including those with different domain styles and types, such as SD[2] models with real-world styles or LoRA models with anime character styles. Please refer to the appendices for more information on the results under different parameters.

4.1 Evaluate

To validate the effectiveness and generalizability of our method, we downloaded some representative SD[2] models from CivitAI[6] in different domains. CivitAI[6] is a public platform that allows artists to train and share their personalized model results. On each model file's recommendation page, we used the example provided by the model's homepage as our prompt input for our study.

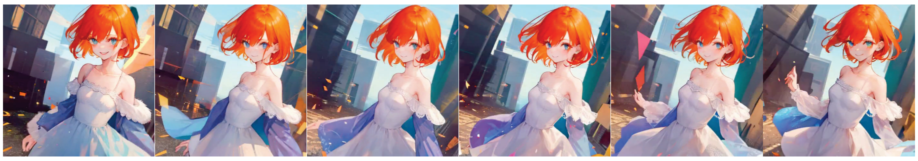
Our model selection covers various aspects, from cartoon images to real-world domains. However, we need to create a new evaluation standard to evaluate our method: we do not use GPT-generated or generic text prompts during generation because personalized models can only generate expected content with a specific text distribution based on fine-tuning effects. In other words, our prompts must contain text with a fixed format, i.e., trigger vocabulary. Otherwise, even with the addition of LoRA[5] or Dreambooth [4], the model's generated effects can only reach the base model level and cannot play a reasonable testing effect in our generality test. The following results section will mainly showcase the results generated using these models.

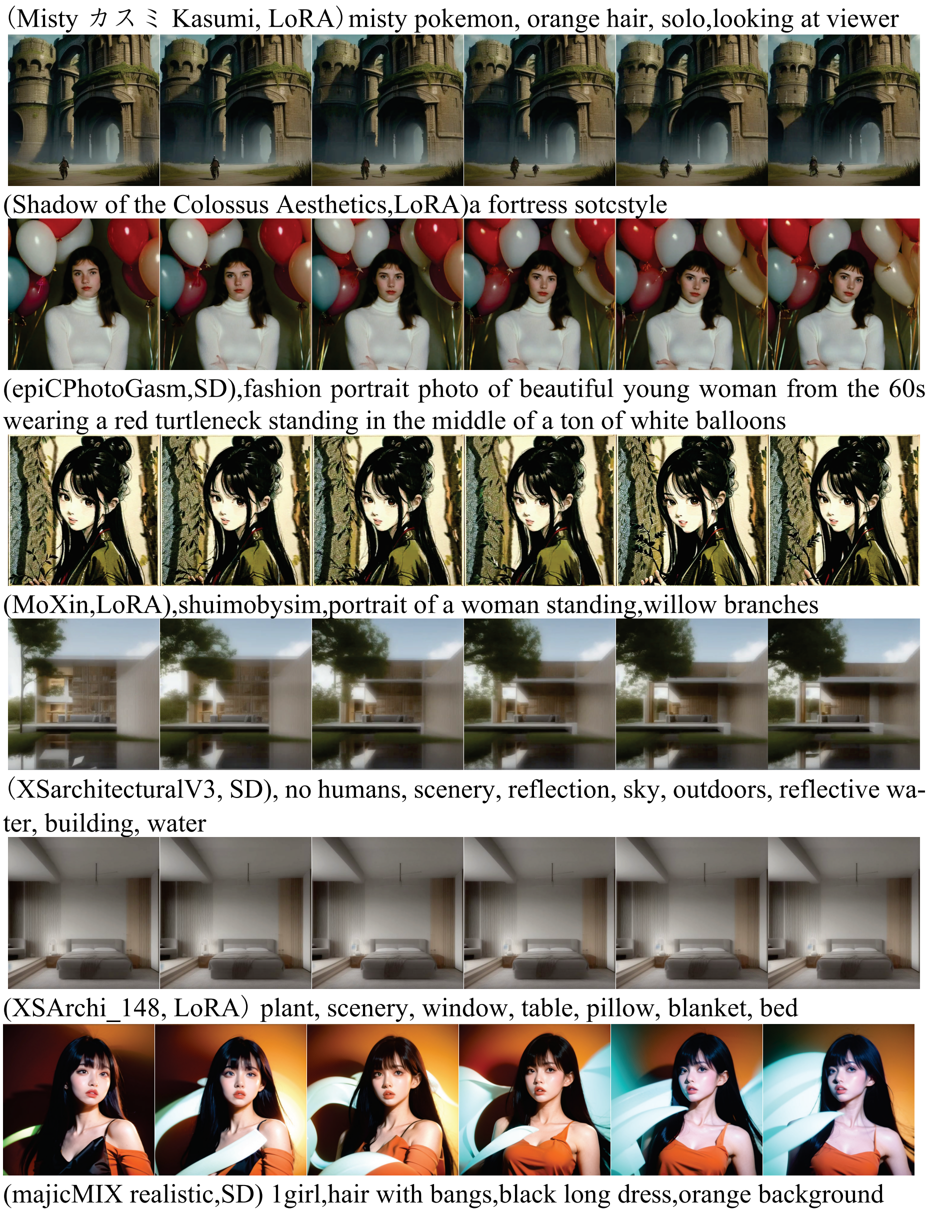
4.2 Result

Due to space constraints, only six frames were chosen from each video-generated content. All models download from the CivitAI [6] platform. To reduce the impact of specificity, all parameter settings (except for image resolution) use the same data as the model's homepage. The prompts are below the corresponding images.

To verify that our method applies to various personalized models derived from finetuning the SD [2] model, we selected models from multiple domains from CivitAI [6], including anime, still life, landscape, real-world, and other types. Through our method combined with personalized models, our approach demonstrates the ability to generate high-quality videos that are also highly engaging.

Furthermore, our tests are not limited to lightweight personalized models. We also used some larger models based on modifications of SD v1.5. These models are uniformly labelled as SD [2], while lightweight methods are labelled according to their respective training methods.





（Misty カスミ Kasumi, LoRA）misty pokemon, orange hair, solo,looking at viewer

(Shadow of the Colossus Aesthetics,LoRA)a fortress sotcstyle

(epiCPhotoGasm,SD),fashion portrait photo of beautiful young woman from the 60s wearing a red turtleneck standing in the middle of a ton of white balloons

(MoXin,LoRA),shuimobysim,portrait of a woman standing,willow branches

（XSarchitecturalV3, SD), no humans, scenery, reflection, sky, outdoors, reflective water, building, water

(XSArchi_148, LoRA）plant, scenery, window, table, pillow, blanket, bed

(majicMIX realistic,SD) 1girl,hair with bangs,black long dress,orange background

Fig. 4. The model names and types are indicated in parentheses, with the prompts at the end. For better readability, we only display part of the text prompts.

4.3 Comparation

In Figure 4, we will compare our method and AnimateDiff[10]. Although various methods such as Tune-a-Video[20] and CoDeF[18] can also used for personalized T2I videos, these methods require input videos to complete processing, so we do not consider them in this study. Both methods use default hyperparameters, with a resolution 512x512 and 16 frames per second. We also used the same prompts ("A forbidden, castle high up in the mountains, pixel art, intricate details2, hdr, intricate details") and parameters for comparison, as shown below:

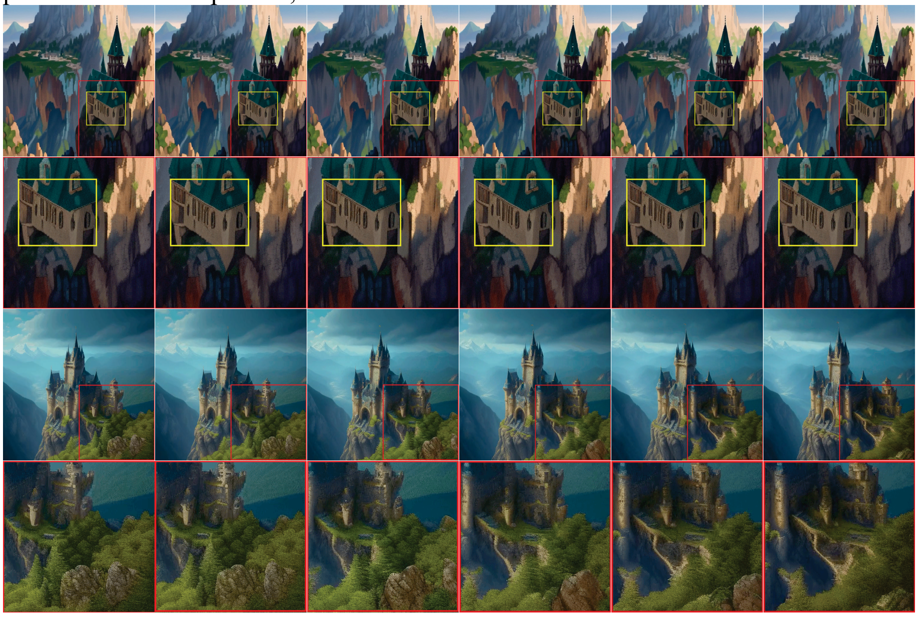


Fig. 5. The first row shows results generated by the CCDM method, the second row shows zoomed-in portions of the images generated by the CCDM method, the third row shows results generated by the Baseline method, and the fourth row shows zoomed-in portions of the Baseline method's results. The original image is divided into four square images, with the selected zoomed images taken from the original image's bottom right area.

Without given motion prompts, our method presents some dynamic information but almost maintains the result of the first frame image. In comparison, it is evident that although the Baseline provides some video angle deflection, the trees themselves not only undergo changes in state and quantity after the angle change, but from the sixth image (fourth row, sixth column), we also discovered the addition of a road. As the image continues to move, the changes in this route become more apparent. Correspondingly, in the video sequence generated by CCDM, although the windows on the castle may occasionally shift positions as time progresses, they return to their original position in the next frame due to the role played by multi-frame fusion when referencing previous and subsequent frames, which is more stable than the Baseline, further verifying the video generation stability of CCDM.

In order to provide a more detailed comparison, we selected a set of prompt information (photo of a ginger woman, in space, futuristic space suit, (freckles: 0.8) cute face, sci-fi, dystopian, detailed eyes, blue eyes) and utilized an SD model specifically trained for real-world scenarios. Similar to the Baseline we selected, we observed that many comparable methods also utilize the WebVid [8,15] dataset for training, such as Make-A-Video[35] and Nvidia Video LDM. Therefore, opting for an SD model trained on real-world scenarios allows a more direct demonstration of the comparative effects.

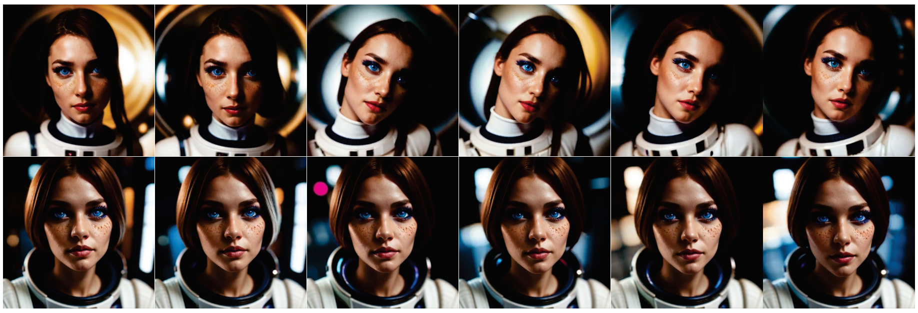


Fig. 6. Compare the generated results. The first row displays the baseline generation outcomes, while the second row shows results from the CCDM method. The generated videos are all 32 frames in length. Due to space limitations, only 6 frames of video results are presented in this study.

Based on the captured images, this study observes significant variations in movement and noticeable color flow in the background in the fourth frame of the Baseline. In contrast, under the control of the CCDM method, the background transitions more smoothly, the character's movement is less pronounced, and there are more apparent distinctions in clothing changes. Specifically, in the Baseline, the red button on the left side of the spacesuit disappears from the fourth to the seventh frame, reappearing only in the final frame. In contrast, in the CCDM method, aside from minor effects from background lighting, there is minimal variation in attire. This study attempts to quantify the differences between the two videos by using optical flow to calculate the inter-frame motion magnitude. The table below presents the calculated results by accumulating the total magnitude, computing the average motion magnitude, and subsequently normalizing the values:

Table 1. Computational results for the castle (top row in Figure 4) and woman in space suit (bottom row in Figure 4), where smaller values indicate less motion variation in the generated videos.

Method	Rating in castle ↓	Rating in woman in space suit↓
Baseline	0.662%	9.421%
Ours	**0.064%**	**4.544 %**

In the practical testing of this study, it was found that the impact of flickering on motion magnitude is more significant than the effect of lens shifts. Moreover, it is easier to generate lens movement or deflection with prompts containing motion information when setting prompts. These prompts more often result in background changes and flickering effects. There is yet to be a well-established method for evaluating purely generated videos, especially in expressing the impact of flickering in videos generated by diffusion models, which is challenging to quantify directly. However, the purpose of employing optical flow in this study is to minimize normal video motion while enhancing the calculation of the flickering impact factor.

Simultaneously, to validate image quality, this study conducted tests for text and image consistency. CLIP values were calculated for each frame in the video and averaged. Additionally, the study used the CLIP values obtained solely from images generated by SD[2] as a quality parameter before incorporating the action module. This was done to demonstrate that the image quality of the model generated under the CCDM method only slightly impacts the original model and is less than the influence of the Baseline method on the model.

Table 2. Evaluate performance through CLIP scores, SSIM, and PSNR, considering higher scores as indicative of better performance.

Method	CLIP Score(avg)↑	SSIM Score(avg)↑	PSNR dB(avg)↑
SD	**0.992**	**0.585**	**11.30**
Ours	0.931	0.443	9.179
Baseline	0.921	0.426	7.813

The table above shows that using SD[2] for static image generation consistently achieves the highest scores across all three evaluation methods. However, based on our selected baseline, we obtained even higher scores. As mentioned earlier, a standardized method for testing video generation is needed, primarily due to the requirement for a comparison image source in most tests. Therefore, we initially opted to generate a static image as a reference and conduct testing using image quality assessment methods such as SSIM and PSNR.

During testing, we observed score fluctuations due to image content variations, leading to an offset of 0.15 for SSIM and 0.8 for PSNR. To address this, we conducted tests on multiple images and obtained the average scores to mitigate the impact of content-related fluctuations.

4.4 Ablation study

This study conducted an ablation experiment to assess the Conditional Control Diffusion Model (CCDM) training methodology and outcomes. Here, we will explain the selection of β values corresponding to the effects of the action prior module. As previously mentioned, to better adapt the model to various styles, this study modified the training values of β. In the figure, options A and B adopted the scaled linear training approach, while option C utilized the linear training approach. Although both A and B follow the training structure of SD[2], the difference lies in A enabling differential random seeds, while B does not. This allows the study to verify that the SD [2]framework can achieve partial control over the output even without any auxiliary plugins.

Based on the observed results, this study hypothesizes that slightly modifying the β values during the forward diffusion training in the training phase contributes to the pre-trained model's adaptability to new tasks and domains. This adjustment enables the model to make ideal predictions and generate actions for the next frame without considering the temporal structure of the video sequence. Using the same diffusion schedule might mislead the model, causing it to believe it is still optimizing for image reconstruction, thereby reducing the training efficiency of the action prior module in this study. This may also decrease the quality of the generated videos when the module is in use, leading to more flickering and color bleeding in the videos. The "Limitations and Future Works" section will further illustrate similar issues.

Table 3. Three diffusion schedule configurations in our ablative experiments.

Configuration	Schedule	β_{start}	β_{end}
A（SD）	scaled linear	0.00085	0.012
B（SD）	scaled linear	0.00085	0.012
C（CCDM）	linear	0.00085	0.012

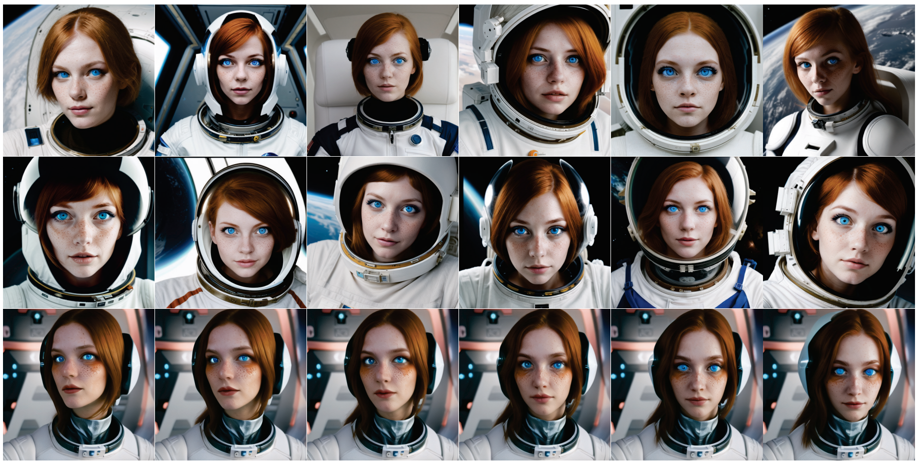


Fig. 7. Ablation Work. This study experimented with Diffusion Schedules under three different environmental conditions and qualitatively compared the results.

5 Conclusion

In this study, we successfully introduced the Conditional Control Diffusion Model to enhance the video generation capabilities of personalized Text-to-Image models. The critical aspect of CCDM lies in its composite network module, which integrates action modeling and multi-frame fusion. By interpolating and fusing keyframes, CCDM achieves a more natural and coherent effect in generating videos, allowing users to specify keyframes for fine-tuning during crucial moments.

Comprehensive evaluations confirmed the effectiveness and generalization of CCDM across various personalized T2I models. However, practical applications, particularly involving models of different styles like anime, posed challenges to the learning process of the action module.

Future work will address these challenges, especially in the context of anime models. We plan to adjust the training approach using a DiT+DDPM[38] strategy, testing different ranges of β values to accommodate diverse generation needs. Resolving these issues will further enhance the performance of CCDM, making it more robust and applicable, thereby contributing significantly to advancements in video generation.

6 Limitations and Future Works

Due to using the WebVid[8] dataset, a collection of real-world videos, CCDM outperforms in generating realistic videos. However, in practical applications, this study employs real-world video models and models of different styles like pixel art or anime. These diverse video types challenge the method module to learn motion priors. In specific models, particularly anime models, issues illustrated in the figure below emerge during generation, stemming from a mismatch between the motion module's training data and the characteristics of these models. As a result, when the model is integrated into the U-Net[7] module of SD[2], initial generations may contain erroneous information. Furthermore, this study's multi-frame fusion technique considers both preceding and succeeding frames during generation, hindering the prompt rectification of error information at the video sequence's outset.

The training approach shifted to a DiT+DDPM[38] strategy, employing a linear schedule and β values ranging from 0.0001 to 0.02. The results displayed an anomalous phenomenon resembling overexposure. Although intuitively, using a diffusion schedule consistent with pre-training should help the model retain its learned feature space, deviations in β values during training led to a gradual reduction in image motion range and a subtle saturation trend in color variation. Consequently, a balance between visual quality and video flickering was sought.

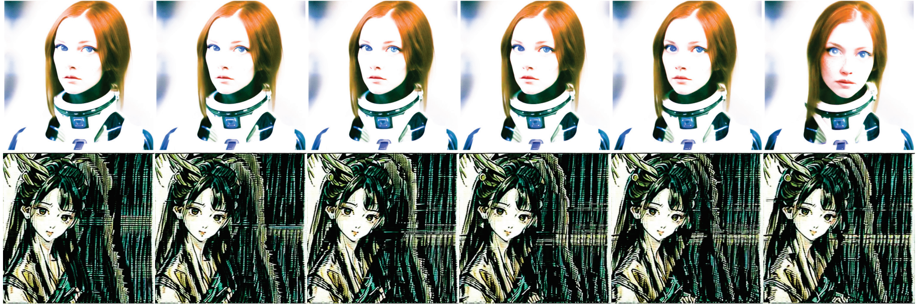


Fig. 8. Some instances of poorly generated results.

In future work, our plan includes expanding the model training dataset to incorporate various video materials, including longer sequences. Additionally, we aim to enhance our algorithm to grant the model greater flexibility in adjustment and control. This improvement is intended to elevate the image quality of generated video sequences, approaching the quality achieved in pure text-based image generation.

References

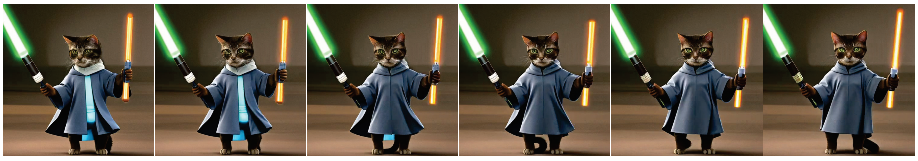
1. Hugging Face. Hugging face. https://huggingface.co/,last accessed 2024/1/20
2. Stability AI. Stable diffusion depth. https://github.com/Stability-AI/stablediffusion,last accessed 2024/1/20
3. Lefaudeux, B., Massa, F., Liskovich, D., Xiong, W., Caggiano, V., Naren, S., Xu, M., Hu, J., Tintore, M., Zhang, S., Labatut, P., Haziza, D.: xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, last accessed 2024/1/20
4. Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 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, pages 22500– 22510, (2023)
5. Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan AllenZhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685, (2021)
6. Civitai. Civitai. https://civitai.com/, last accessed 2024/1/20
7. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention MICCAI International Conference, pages 234–241 (2015)
8. Webvid: https://tensorflow.google.cn/datasets/catalog/webvid, last accessed 2024/1/20
9. Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., Tang, J.: Cogview: Mastering text-to-image generation via transformers. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P.S., Wortman Vaughan, J. (eds.) Advances in Neural Information Processing Systems, volume 34, pages 19822–19835. Curran Associates, Inc., (2021)
10. Guo, Y., Yang, C., Rao, A., Wang, Y., Qiao, Y., Lin, D., Dai, B.: AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning. arXiv:2307.04725, (2023)
11. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical Text-Conditional Image Generation with CLIP Latents. ArXiv, abs/2204.06125
12. 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: International Conference on Machine Learning, pages 8748–8763. PMLR (2021)
13. Nichol, A.Q., 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. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pages 16784–16804. PMLR, 17–23 Jul (2022)

14. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12873–12883 (2021)
15. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1728–1738 (2021)
16. Lu, H., Teng, Y., Li, Y.: Learning Latent Dynamics for Autonomous Shape Control of Deformable Object. IEEE Transactions on Intelligent Transportation Systems, (2022)
17. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 22563–22575, (2023)
18. Ouyang, H., Wang, Q., Xiao, Y., Bai, Q., Zhang, J., Zheng, K., Zhou, X., Chen, Q., Shen, Y.: CoDeF: Content Deformation Fields for Temporally Consistent Video Processing. arXiv:2308.07926, (2023)
19. 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, pages 10684–10695, (2022)
20. Wu, J.Z., Ge, Y., Wang, X., Lei, W., Gu, Y., Hsu, W., Shan, Y., Qie, X., Shou, M.Z.: Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. Proceedings of the IEEE/CVF International Conference on Computer Vision. pp: 7623-7633 ,(2023)
21. Lu, H., Wang, T., Xu, X., et al.: Cognitive memory-guided autoencoder for effective intrusion detection in the Internet of Things. IEEE Transactions on Industrial Informatics, 18(5): 3358-3366, (2021)
22. Khachatryan, L., Movsisyan, A., Tadevosyan, V., Henschel, R., Wang, Z., Navasardyan, S., Shi, H.: Text2video-zero: Text-to-image diffusion models are zero-shot video generators. arXiv preprint arXiv:2303.13439, (2023).
23. Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453, (2023)
24. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De La-roussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for NLP. In International Conference on Machine Learning, pages 2790–2799, (2019)
25. Crowson, Katherine, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. "Vqgan-clip: Open domain image generation and editing with natural language guidance." In European Conference on Computer Vision, pp. 88-105. Cham: Springer Nature Switzerland, (2022)
26. Stickland, A.C., Murray, I.: Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In: International Conference on Machine Learning, pages 5986–5995, (2019)
27. Rebuffi, S.-A., Bilen, H., Vedaldi, A.: Efficient parametrization of multi-domain deep neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8119–8127, (2018)
28. Rosenfeld, A., Tsotsos, J.K.: Incremental learning through deep adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(3):651–663, (2018).
29. Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. In: European Conference on Computer Vision, (2022, October), pages 280-296. Cham: Springer Nature Switzerland.

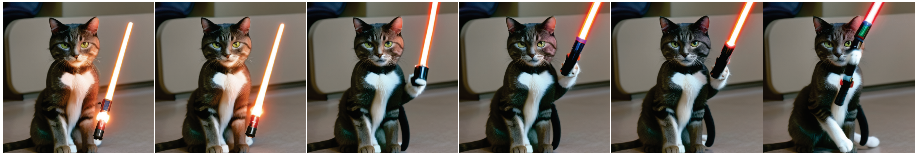
30. Gao, Peng, et al. "Clip-adapter: Better vision-language models with feature adapters." International Journal of Computer Vision (2023): 1-15.
31. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pages 8748–8763. PMLR, (2021)
32. Sung, Y.L., Cho, J., Bansal, M.: Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5227-5237, (2022)
33. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qi-ao, Y.: Vision transformer adapter for dense predictions. International Conference on Learning Representations, (2023).
34. Raffel, C., Shazeer, N., Roberts, A., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 21(1):5485-5551, (2020)
35. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792, (2022)
36. Zhang, L., Rao, A., Agrawala, M."Adding conditional control to text-to-image diffusion models." In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3836-3847,(2023)
37. Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K. Lion: Latent point diffusion models for 3d shape generation. In Advances in Neural Information Processing Systems (NeurIPS), (2022)
38. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, volume 33, pages 6840–6851. Curran Associates, Inc., (2020)

Appendices

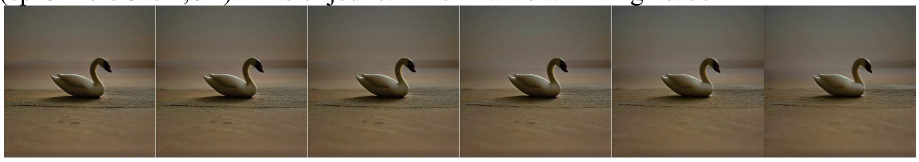
Qualitative Results



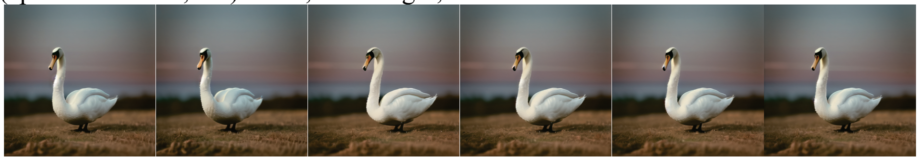
(epiCPhotoGasm,SD）a master jedi cat in star wars with a lightsaber



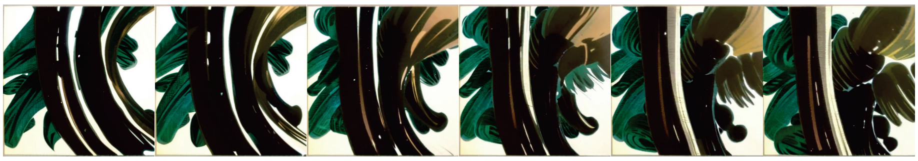
(epiCPhotoGasm,SD)a master jedi cat in star wars with a lightsaber



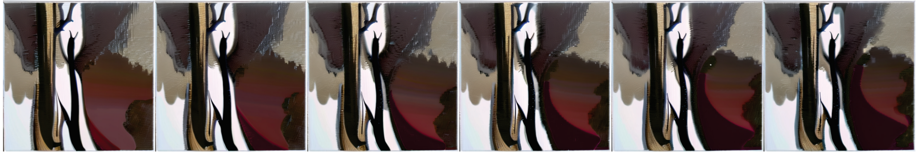
(epiCPhotoGasm, SD) Swan, at Twilight, tilt shift



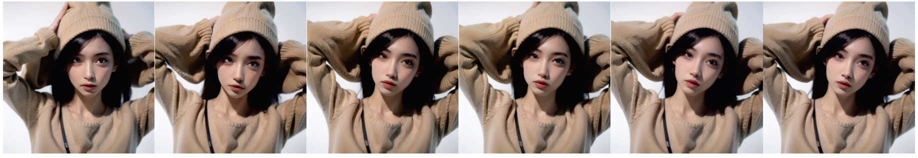
(epiCPhotoGasm,SD)Swan, at Twilight, tilt shift



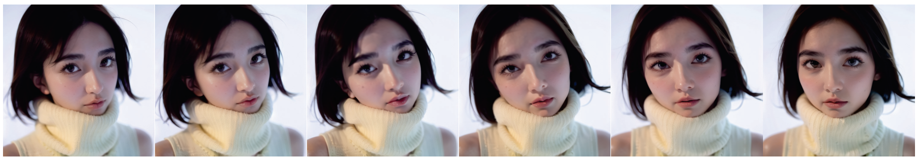
(MoXin,LoRA)a branch of flower,traditional chinese ink painting



(MoXin,LoRA）a branch of flower,traditional chinese ink painting



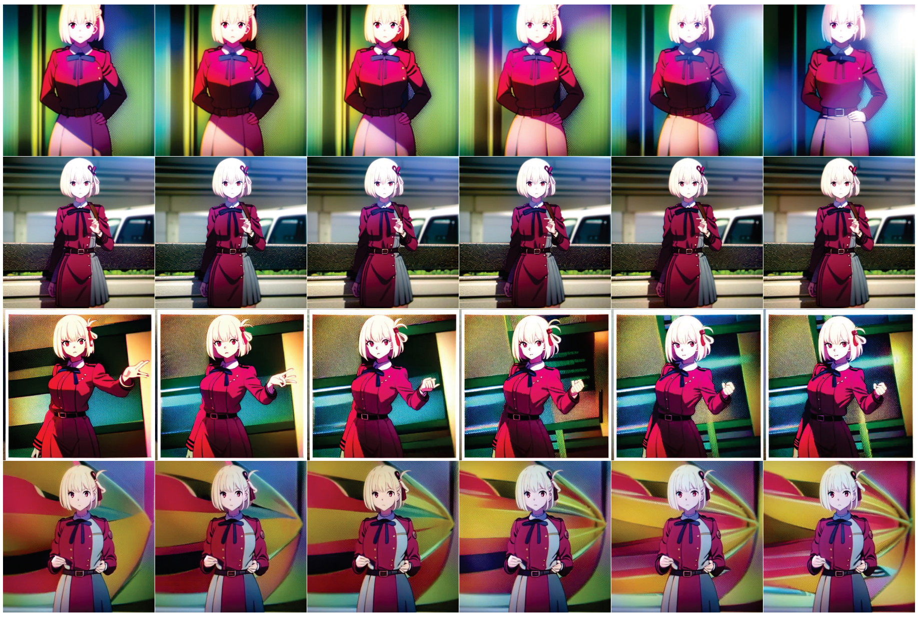
（majicMIX realistic,SD)1girl,sweater,white background



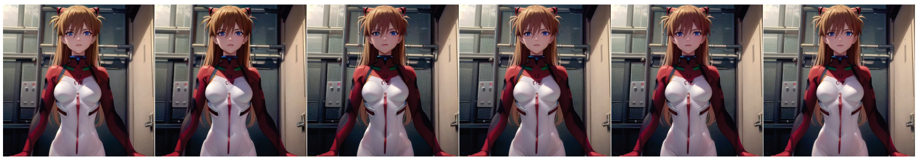
(majicMIX realistic,SD)1girl,sweater,white background

Fig. 9. Qualitative results of our CCDM generation method for different prompts. For better readability, only partial text prompts are shown.

Model Diversity



(Chisato Nishikigi- Lycoris Recoil,LoRA) chisato nishikigi, short hair, blonde hair,



(Asuka Langley Soryu,LoRA) blue eyes, hair ornament, asukalangley

Fig. 10. Model diversity. Four consistent tests were conducted for each of the two models to demonstrate that the CCDM method had no impact on the models. Additionally, two sets of results generated with the same prompts and personalized models are shown, indicating that the personalized models maintain their diversity even after being inflated with CCDM. For better readability, only partial text prompts are displayed.