ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10888991

Implanting Robust Watermarks in Latent Diffusion Models for Video Generation

Xiaohang Liu[1*], Heng Chang[2*], Jinfu Wei[3], Lei Zhu[1], Li Liu[2], Likun Li[2], Shiji Zhou[3], Chengyuan Li[2], Di Xu[2], Wei Gao[1†]
[1]*School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University*
[2]*Huawei Technologies Co., Ltd., China*
[3]*Tsinghua University, China*

***Abstract*—In the dynamic realm of digital media, latent diffusion models (LDM) have revolutionized the generation of videos, surpassing the capabilities of traditional generative models. This paper presents Stable Video Signature, a pioneering watermarking framework for LDM in video generation. Addressing the pressing need for copyright and model protection, our approach is the first to implant watermarks directly into the generation process of LDM based on video through a novel two-stage process. We first encode watermarks into the video's latent space embedding, ensuring a holistic temporal decoding mechanism of LDM. Then watermark is integrated into the LDM's decoder. In this process, our method can maintain frame consistency, preserving the quality and robustness of generated videos during watermark implantation. We further show that the framework embeds watermarks seamlessly into LDM, maintaining the original functionality of the models and exhibiting resilience against a spectrum of watermark attacks. Our comprehensive experiments on both text-to-video and image-to-video generation tasks substantiate the efficacy of Stable Video Signature. This work not only pioneers watermarking in video generation LDM, but also sets a precedent for safeguarding intellectual property in the age of advanced media synthesis.**

***Index Terms*—digital watermark, diffusion model, video generation.**

I. Introduction

In the burgeoning field of digital media, latent diffusion models (LDM) [1]–[4] have emerged as a transformative force, particularly in the generation of images and videos. Their ability to produce content of remarkable quality and diversity has eclipsed traditional generative models like GANs [5]–[7] and VAEs [7]–[9]. The advent of large-scale LDM, such as Sora [10], has not only marked a significant success in video generation but also broadened the practical applications of LDM that include text-to-video [11]–[14] and image-to-video tasks [15]–[18].

Though digital video content can be rapidly reproduced and disseminated nowadays, ensuring the traceability and ownership of videos generated by LDM is also paramount. The triumph of LDM in video generation brings forth substantial compliance challenges, including copyright and model protection. One user can generate customized videos via the officially released model by the model provider. However, if the model that the user accesses is not protected, the compliance officer is not able to trace the source model. Otherwise, if the model provider can use a private watermark encoder and detector to identify the watermark, where specific information is detected from the video, the model can be safely protected them. Unlike previous works focused on neural network copyright protection in critical tasks, the discussion on the efficacy of watermarks within DMs, particularly for video generation, remains scant.

However, watermarking DMs for video generation presents unique challenges over image generation [19]. On the one hand, post-hoc watermarking solutions usually consist of two disconnected modules: video generation and watermark implantation. However, this disconnected schema fails to offer secure model protection, complicating the defense of intellectual property rights in cases of unauthorized use. For example, one could easily bypass the post-hoc watermark rooter and directly access the intermediate video content with a similar approach as in Yang et al. [20]. Therefore, this challenge calls for seamlessly integrating watermarks into the generative process without compromising the original content's aesthetic and functional integrity.

On the other hand, while video content possesses a rich temporal dimension, it also complicates the task of embedding and maintaining watermarks across frames without impacting the video quality. Unlike static images, videos comprise a sequence of frames, and any watermarking technique must account for the temporal coherence and flow to avoid defects like flickering or inconsistency between frames. We conclude the two challenges as follows:

- Naive disconnected post-hoc methods cannot add watermarks in an end-to-end manner, which leads to easier model theft and vulnerability.
- Adding watermarks to videos without considering the temporal consistency challenges integrating watermarks into the generative process while avoiding flickering or inconsistency.

To address these challenges, this paper introduces Stable Video Signature (SVS), a novel approach that merges watermarking into the video generation process itself, without any architectural changes. SVS employs a two-stage model watermarking method. In the first stage, we proposed Video-Lock, a video watermark module. Video-Lock's watermark encoder and corresponding extractor are obtained, inspired by HiDDeN [21]. To align with the holistic decoding mechanism of DMs, we incorporate 3DCNN to encode watermarks into the overall latent space embedding of videos. Additionally, we introduce temporal Transformer blocks to capture the temporal consistency during watermarking, thereby maintaining the quality of generated videos. We call this method Video-Lock. In the second stage, we propose a two-branch model distillation architecture. A deflickering decoder is designed in both branches to alleviate the potential artifacts, while the same temporal Transformer block with the first stage is introduced in the upper branch to further ensure consistency among frames. Additionally, watermark integrity is maintained through extraction and comparison with a predefined

*Equal Contribution; †Corresponding authors.

This work was supported by National Science and Technology Major Project (2024ZD01NL00101), Natural Science Foundation of China (62271013, 62031013), Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (2024B1212010006), Guangdong Province Pearl River Talent Program (2021QN020708), Guangdong Basic and Applied Basic Research Foundation (2024A1515010155), Shenzhen Science and Technology Program (JCYJ20240813160202004, JCYJ20230807120808017).

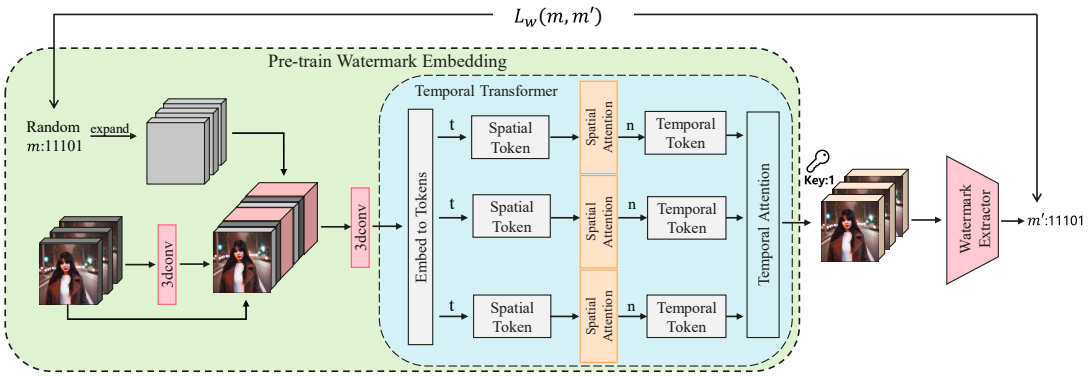

Fig. 1. Architecture of Video-Lock: The module consists of two components: a watermark encoder ($\mathcal{W}_E$), and an extractor($\mathcal{W}_D$). In the watermark embedding process, the original video and a randomly generated watermark m are fed as inputs, yielding a watermarked video as the output. Subsequently, this watermarked video is processed by the watermark extractor to output the watermark m'. The loss is determined by calculating the cross entropy loss between m and m'.

watermark, contributing to the model's robustness. Our extensive experiments across text-to-video and image-to-video generation tasks validate the effectiveness of the SVS in embedding watermarks across various DMs. We further demonstrate that the model's functionality remains unimpaired post-watermarking and showcase the robustness of our watermarks against diverse attacks.

In summary, this work presents the following key contributions:

- To best our knowledge, SVS is the first non-post-hoc approach to add watermarks in DMs for video generation. SVS is a general framework that can root watermarks in DMs through efficient fine-tuning.
- To maintain the quality of generated videos after watermark implantation, we propose to use 3DCNN to accommodate the nature of DMs that decode the video latent as a whole and introduce temporal attention blocks to better capture the consistency among frames.
- Extensive experiments on both text-to-video and image-to-video generation tasks demonstrate that SVS could effectively root watermarks into DMs without downgrading the quality of generated videos while preserving robustness to various watermark attacks.

II. Method

A. Video watermark module: Video-Lock

The framework of the Video-Lock comprises two components: Watermark Embedding ($\mathcal{W}_E$) and Watermark Extractor ($\mathcal{W}_D$). For specific details, please refer to Figure 1.

1) Embedding: The input to $\mathcal{W}_E$ consists of x_0 and a k-bit message $m \in \{0,1\}^k$, where x_0 represents the input video and m represents a randomly generated bitstream that represents the watermark. We believe that watermarking videos does not necessitate complex network architectures. Hence, we employ a simple 3D convolution to extract features from x_0. To embed watermarks into videos more effectively, the temporal Transformer is introduced in the encoding process. Inspired by Arnab et al. [22], this block consists of two separate Transformer encoders. The first part, the spatial encoder, only interacts with tokens extracted from the same frame, which are then passed on to the second part, the temporal encoder, to interact with tokens in the same position across different frames. Temporal Transformer is capable of endowing our model with global consistency, which ensures the stability and reliability of our watermark information.

$$x_w = \mathcal{W}_E(x_0). \tag{1}$$

2) Extractor: After undergoing the embedding module, the original video is transformed into a watermarked video. We then train a decoding network to extract the watermark from the processed video. The decoding network needs to maintain symmetry with the encoding network, as the network encodes the watermark's representation in the video, which the decoding network can better learn from. This approach enables the decode network to extract the watermark from the video more effectively.

$$m' = \mathcal{W}_D(\mathcal{A}(x_w)). \tag{2}$$

3) Training: During the training phase, to ensure the robustness of the watermark, we introduce several watermark attack methods. We enumerate commonly used attack methods, such as compression, cropping, and rotation. These attack methods are combined during the training phase, with the input being the encoded video with the watermark and the output being the video after being attacked by various methods. It should be noted that the resolution of the video may change after being attacked. Some of these attacks are non-differentiable, so this paper uses two losses for learning. One loss is for the normal output without attacks, which is used to fine-tune the encode and decode processes. The other loss is used to extract the watermark from the decoded image after being attacked, which is used to fine-tune the decoding process.

The loss function proposed in our study comprises three components: message loss $\mathcal{L}_w$, similarity loss $\mathcal{L}_i$, and deviation loss $\mathcal{L}_v$. The message loss is employed to ensure the decoding accuracy of the watermark. It is calculated by computing the cross-entropy between the input and the decoded messages.

$$\mathcal{L}_w = CrossEntroy(m', m). \tag{3}$$

The similarity loss is designed to ensure that the introduction of a watermark does not result in artifacts or other issues in the video.

$$\mathcal{L}_i = MSE(x_w, x_0). \tag{4}$$

To ensure that the video watermark is evenly distributed in every video frame, we calculate the variance of the loss sequence composed of the image loss of each frame of the original video and the video with the watermark added.

$$\mathcal{L}_v = \text{Var}(x_w - x_0) \tag{5}$$

The final combined loss function can be summarized as:

$$\mathcal{L} = \lambda_w(\mathcal{L}_w + \mathcal{L}'_w) + \lambda_i\mathcal{L}_i + \lambda_v\mathcal{L}_v. \tag{6}$$

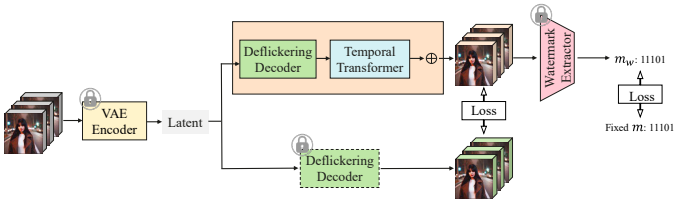

Fig. 2. Model Distillation: A video is first encoded via a VAE encoder. Concurrently, the lower branch's Deflickering Decoder parameters remain frozen. The upper branch's output of the same Deflickering Decoder undergoes a temporal transformation to ensure consistency. Subsequently, the outputs of both branches are merged using weighted concatenation to yield the final output, evaluated against a loss function for fidelity. Additionally, the upper branch's output is processed by a watermark extractor to extract a watermark m_w.

B. Model Distillation

This part of the pipeline is shown in Figure 2. In our decoding process, we opt for the Video Transformer. SVD also employs this method of decoding. The unique aspect of Stable Video Diffusion lies in its decoder. Unlike traditional decoders that focus on reconstructing images from latent representations, this decoder is tailored for video generation. Being "temporally aware" implies that the decoder considers the temporal continuity and coherence between video frames. This is crucial for generating smooth video sequences where successive frames are consistent with each other, avoiding abrupt changes or "flickers." The "deflickering" aspect addresses a common challenge in video generation – flickering effects that can occur due to inconsistencies between frames. This decoder is designed to minimize such flickering, ensuring a more stable and smooth transition across frames.

We also introduce the Temporal Transformer, which features a globally consistent attention mechanism capable of long-range dependency, enabling the generation of smoother videos. However, this approach reduces video clarity. To maintain clarity, this study concatenates the smooth video output from the temporal Transformer with the original video to form the final decoded output.

We fix the watermark m. The original Deflickering Decode $\mathcal{D}_D$ is finetuned into the watermarked Deflickering Decode $\mathcal{D}_w$.

The input video x_0 passes through VAE encoder $\mathcal{E}$, outputs the latent $z = \mathcal{E}(x_0)$. And z is fed into $\mathcal{D}_w$ to obtain the output $x_w = \mathcal{D}_w(z)$. The extractor recovers $m' = \mathcal{W}_D(x_w)$. The message loss is $\mathcal{L}_w = CrossEntroy(m', m)$.

In addition to this, to ensure the identity of the generated video, we freeze the original Deflickering Decode $\mathcal{D}_D$ and introduce $\mathcal{L}_i = MSE(x_w, \mathcal{D}_D(x_0))$.

C. Generation

In this stage, we generate videos using watermarked LDM, as shown in the newly provided Figure 3. The VAE Decoder and Deflickering Decoder are both core components of LDM, which

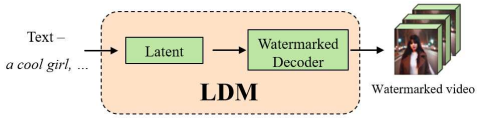

Fig. 3. The pipeline for generating videos using a watermarked LDM. The input text undergoes a diffusion process to obtain the latent representation, which is then decoded by the watermarked decoder to produce the video.

indicates our method has a strong relationship with LDM. We generate videos using the following two methods.

1) AnimateDiff (AD): The AnimateDiff encoder-decoder is a standard VAE encoder-decoder that operates on a per-frame basis and is incapable of training a decoder that generates a holistic watermark. To overcome this limitation, we replaced the AnimateDiff decoder with an SVD decoder and fine-tuned it to generate watermarked videos while ensuring smooth video generation.

2) Stable Video Diffusion (SVD): A high-quality video generation universal model has been constructed, which has shown excellent performance in various downstream tasks after fine-tuning. Initially, videos produced through SVD are embedded with an invisible watermark using a library dedicated to this purpose. Traditionally, this watermarking involved a post-hoc method, where consecutive frames of a video are generated first, followed by the addition of the watermark to each frame in a sequential manner. However, it is observed that this post-hoc approach lacked robustness, as will be elaborated on in the Experiments section.

III. Experiments

In the previous section, we discussed the techniques for incorporating watermarks into videos using large-scale models and detecting video watermarks. In this section, we provide a range of diverse generation tasks and the corresponding results obtained using the video large-scale model, as well as a comparative analysis of the effectiveness of different watermarking algorithms. We aim to answer the following research questions:

(**RQ1**) Can our method effectively implant watermarks in DMs for video generation and withstand common attack methods?

(**RQ2**) Can our proposed method maintain the quality of the generated videos?

(**RQ3**) Does the design of each component of SVS contributes to the effectiveness of watermark implantation?

A. Experimental Setup

1) Dataset: We use WebVID10M [23] to train our model. The WebVID10M dataset is a comprehensive collection of text-video pairs. Downloading these pairs from the original sites may result in the inclusion of visible watermarks on the videos. However, our experiments have demonstrated that these watermarks do not significantly impact the quality of the generated videos, indicating that they are not learned during the fine-tuning process. We also use the Kinetics-400 dataset [24] to evaluate our model.

2) Evaluation metrics: In our study, we have employed a comprehensive set of evaluation metrics to rigorously assess the performance and robustness of our watermarking approach.

Accuracy (Acc): evaluates the accuracy of watermark extraction, reflecting the robustness of the watermark against various attacks. It is defined as the ratio of correctly decoded bits to the total number of embedded bits.

Fréchet Video Distance (FVD): measures the distance between the feature vectors of the original and watermarked videos. It captures not only frame-wise quality but also temporal coherence.

B. Watermark Robustness (*RQ1*)

As far as we know, we are the first to work on video large-model watermarking, and we did not find a direct comparison method. An alternative way to watermark-generated videos is to add watermarks after the video is generated. However, as noted in Section I, there are numerous methods for applying watermarks to images or videos in an end-to-end method. Finally, we had to choose three watermarking

TABLE I
COMPARISON OF ACCURACY BETWEEN OUR METHOD'S AND POST-HOC APPROACHES UNDER VARIOUS ATTACKS.

	AD (Text-to-Video)				SVD (Image-to-Video)			
Attack	SVS	DDS [25]	dwtDct [26]	HiDDeN [21]	SVS	DDS [25]	dwtDct [26]	HiDDeN [21]
None	0.9929	1.0	0.8509	0.9917	0.9946	1.0	0.8834	0.9826
Crop($\sqrt{0.1}$)	0.9839	0.3654	-	0.9919	0.9767	0.3604	–	0.9666
Crop($\sqrt{0.5}$)	0.9928	0.4715	0.5178	0.9953	0.9964	0.4630	0.4900	0.9789
Resize($\sqrt{0.5}$)	0.9928	0.5177	0.5147	0.9658	0.9946	0.5325	0.5114	0.9726
Rot(25°)	0.9929	0.5325	0.5132	-	0.9946	0.5189	0.5048	-
Rot(90°)	0.9929	0.4551	0.6003	0.9455	0.9946	0.4521	0.5983	0.9251
Blur(2.0)	0.9929	1.0	0.4395	0.9205	1.0	1.0	0.4280	0.9058
Jpeg(50)	0.9732	0.4400	0.4009	0.9253	0.9714	0.4000	0.4008	0.8861

TABLE II
VIDEO GENERATION QUALITY

	User study		SSIM ↑	FID ↓	FVD ↓
	Consistency ↑	Input Align ↑			
Original SVD	4.06	4.18	0.95	41.49	478.60
Watermarked SVD	4.07	4.20			
Original AD	3.84	4.39	0.95	23.20	214.86
Watermarked AD	4.10	4.39			

15 users participated in the survey, evaluating the input alignment and consistency of the generated videos with the text prompts using a rating scale of 1 to 5 after using them.

methods commonly used in generative models. They are invisible-watermark[1], blind-watermark[2] and HiDDeN [21]. In these methods, the invisible-watermark is used to watermark videos generated by stable video diffusion. Blind-watermark and invisible-watermark are open-source watermark methods based on DWT-DCT-SVD (DDS) [25] and dwtDct [26]. We compared the accuracy of different attack scenarios under the same video and payload. The comparisons of different networks are shown in table I. We evaluated the robustness of the model using various attack methods, including cropping, scaling and rotation, blur, and compression. Specifically, for the cropping operation, we center crop the frames are cropped in width and height with the proportion p. For scaling and rotation operations, we also scale the video as a whole. For resize, the frames are scaled in width and height with the proportion p. For blur, we apply Gaussian blur across the entire image to simulate the effect of blurring. Finally, for the compression operation, we save the obtained tensor as an image and then read it out again to perform image compression.

To verify the resistance of our model to various attacks, we generated a large number of videos and subjected them to attacks of varying intensities. The final experimental results are plotted as a line graph, as shown in Figure 4. We also introduce methods to attack video frames. Frame Swap means we randomly rearrange the order of N groups of adjacent frames. Frame Drop refers to randomly deleting a few frames and replacing them with nearby frames. In noise, v represents the variance of the added Gaussian noise.

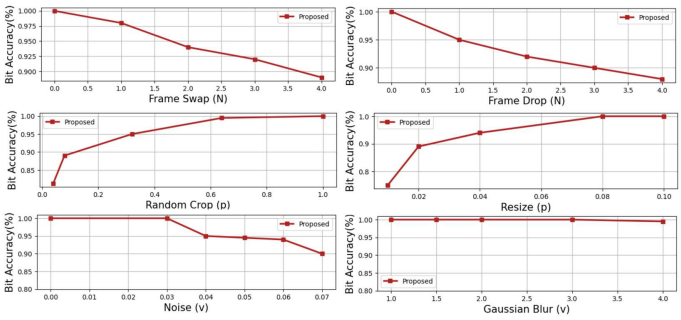

Fig. 4. Decoding accuracy of the proposed method under different attacks and attack intensities.

C. Video Generation Quality (*RQ2*)

1) Metrics: Table II presents the quantitative evaluation of video quality generated by models with and without watermarking. The PSNR between the original and watermarked videos is approximately 25, and the SSIM is around 95. The FVD is about 300. Experimental results indicate that our model watermarking has minimal impact on the quality of video generation.

2) User Study: In this study, we randomly selected five pairs of videos generated by the original model and watermarked model and invited users to evaluate them. Three of the five video sets are generated using stable video diffusion, with the input being randomly selected images of waves, while the other two videos are generated using the AD model, with the input being random text from the test cases in the AD paper. To ensure fairness, users are not informed whether the videos contain watermarks. The data provided in Table II shows that the videos generated by our post-fine-tuned model performed no worse than those generated by the original model. In our user study, respondents are asked to choose the best group of videos. Most users believe that there is little difference between videos generated by the original model and those generated by the watermarked model. Some participants even suggest that our fine-tuned videos are more fluid and more consistent with the input. We analyze that this may be because we use a temporal Transformer to make the videos smoother.

D. Ablation Studies (*RQ3*)

Temporal Transformer block. To verify the effectiveness of the temporal Transformer block $\mathcal{T}$ in Figure 2, we removed this module from the model and re-finetuned it for 100 epochs. The results are shown in Table III. The Pixel-MSE (averaged mean-squared pixel error between aligned consecutive frames) decreases by 0.002, the accuracy is improved by 0.4%, and the accuracy also increases following various attacks. This indicates that our temporal Transformer can enhance the quality of video generation and improve the robustness of the videos.

TABLE III
ABLATION STUDY ON THE TEMPORAL TRANSFORMER BLOCK.

Method	Pixel_MSE↓	Acc↑	Attack acc↑		
			Crop($p=\sqrt{0.5}$)	Jpeg($p=50$)	Rot($\sigma=25°$)
w/o $\mathcal{T}$	0.01749	0.9650	0.9575	0.9095	0.9605
w $\mathcal{T}$	0.01573	0.9665	0.9635	0.9245	0.9675

IV. CONCLUSION

We introduce a novel, effective watermarking approach tailored for video content generated by diffusion models. SVS seamlessly integrates into the video generation process, embedding robust watermarks without compromising video quality. This technique's standout feature is its two-stage process, comprising pre-training and model distillation, which ensures that watermarks are deeply integrated and resistant to various attacks. A key advantage of our approach is its ability to maintain the original quality and integrity of the videos while embedding watermarks. Our method also demonstrates a high degree of resilience against various attacks, proving its efficacy in real-world applications.

[1]https://github.com/ShieldMnt/invisible-watermark
[2]https://github.com/guofei9987/blind_watermark

REFERENCES

[1] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al., "Imagen video: High definition video generation with diffusion models," *arXiv preprint arXiv:2210.02303*, 2022.

[2] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jinren Zhou, and Tieniu Tan, "Decomposed diffusion models for high-quality video generation," *arXiv preprint arXiv:2303.08320*, 2023.

[3] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet, "Video diffusion models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 8633–8646, 2022.

[4] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou, "Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7623–7633.

[5] Changhee Han, Hideaki Hayashi, Leonardo Rundo, Ryosuke Araki, Wataru Shimoda, Shinichi Muramatsu, Yujiro Furukawa, Giancarlo Mauri, and Hideki Nakayama, "Gan-based synthetic brain mr image generation," in *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, 2018, pp. 734–738.

[6] Yongyi Lu, Shangzhe Wu, Yu-Wing Tai, and Chi-Keung Tang, "Image generation from sketch constraint using contextual gan," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 205–220.

[7] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua, "Cvae-gan: fine-grained image generation through asymmetric training," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2745–2754.

[8] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals, "Generating diverse high-fidelity images with vq-vae-2," *Advances in neural information processing systems*, vol. 32, 2019.

[9] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li, "Generating diverse structure for image inpainting with hierarchical vq-vae," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10775–10784.

[10] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh, "Video generation models as world simulators," 2024.

[11] Yitong Li, Martin Min, Dinghan Shen, David Carlson, and Lawrence Carin, "Video generation from text," in *Proceedings of the AAAI conference on artificial intelligence*, 2018, vol. 32.

[12] Yuming Jiang, Shuai Yang, Tong Liang Koh, Wayne Wu, Chen Change Loy, and Ziwei Liu, "Text2performer: Text-driven human video generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22747–22757.

[13] Taegyeong Lee, Soyeong Kwon, and Taehwan Kim, "Grid diffusion models for text-to-video generation," *arXiv preprint arXiv:2404.00234*, 2024.

[14] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al., "Make-a-video: Text-to-video generation without text-video data," *arXiv preprint arXiv:2209.14792*, 2022.

[15] Yaosi Hu, Chong Luo, and Zhenzhong Chen, "Make it move: controllable image-to-video generation with text descriptions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18219–18228.

[16] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris Metaxas, "Learning to forecast and refine residual motion for image-to-video generation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 387–403.

[17] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani, "Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion," *arXiv preprint arXiv:2403.12008*, 2024.

[18] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al., "Stable video diffusion: Scaling latent video diffusion models to large datasets," *arXiv preprint arXiv:2311.15127*, 2023.

[19] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon, "The stable signature: Rooting watermarks in latent diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22466–22477.

[20] Yijun Yang, Ruiyuan Gao, Xiaosen Wang, Nan Xu, and Qiang Xu, "Mma-diffusion: Multimodal attack on diffusion models," *arXiv preprint arXiv:2311.17516*, 2023.

[21] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 657–672.

[22] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6836–6846.

[23] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman, "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*, 2021, pp. 1728–1738.

[24] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al., "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[25] KA Navas, Mathews Cheriyan Ajay, M Lekshmi, Tampy S Archana, and M Sasikumar, "Dwt-dct-svd based watermarking," in *2008 3rd international conference on communication systems software and middleware and workshops (COMSWARE'08)*. IEEE, 2008, pp. 271–274.

[26] Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni, "Robust invisible video watermarking with attention," *arXiv preprint arXiv:1909.01285*, 2019.