Video Inpainting Using Diffusion Model

Wondmgezahu Teshome
teshome.w@northeastern.edu

1 Introduction

Video inpainting involves filling in missing parts of a video (masked regions) with content that seamlessly integrates with the rest of the video. The goal of video inpainting is to create a final result that appears natural and realistic as if the missing or damaged parts were never there. This can be used to remove unwanted objects from a video or to restore old or damaged footage [1], and in surveillance video processing.
One simple approach to video inpainting is to apply image inpainting techniques to each frame of the video individually. This means that the missing or occluded parts of each frame are filled in without considering the content of the other frames in the video. However, applying image inpainting techniques to each frame of the video individually, without considering the content of the other frames, can result in inconsistencies between the frames. This can lead to visual artifacts and unnatural transitions in the final result, which can reduce the quality of the video inpainting.
To address this problem, several methods have been proposed for video inpainting such as patch-based, flow-based, and generative approaches. The flow-based video inpainting methods split the synthesis process into three steps: flow completion, pixel propagation, and content hallucination [22]. Patch-based approaches use patches of the video to fill in missing or damaged regions. In order to harmonize the inpainted regions with the rest of the image, people in the past have used generative approaches such as GANs or Autoregressive modeling.
In this work, a conditional DDPM is utilized to address the issue of video inpainting. The model takes the masked frames as conditional input and generates frames that resemble the original. To ensure consistency in the generated frames, multiple resampling steps are employed, where the sampled frame is fed back into the model for further sampling. A prior work using diffusion models has been proposed by Lugmayr et al. [23] for image inpainting, where they use off-the-shelf unconditionally trained DDPM as the generative prior, and alter the reverse diffusion iterations by sampling the unmaksed regions using the given image information. In this work, instead of using the pretrained unconditional DDPM, the model was trained in a conditional input setting.

2 Related Work

Image completion has been an active area of research in recent years. One approach to image inpainting is using denoising diffusion probabilistic models (DDPMs), as proposed by Lugmayr et al. in their paper *Repaint* [23]. Other approaches to image inpainting include score-based methods [19][17], and generative inpainting [7],[27]. He et al. [3] and Xiong et al. [12] have also proposed methods for image completion.

Video inpainting has also been studied extensively. Patch-based synthesis techniques have been proposed by Newson et al. [4] and Wexler et al. [2]. These methods can produce high-quality results but can be computationally expensive and may struggle with large or complex missing regions. CNN-based approaches have been proposed by Wang et al. [6], Chang et al. [9], Li et al. [11], and Zhu et al. [8]. These methods may produce less realistic results. Flow-based approaches have been proposed by Xu et al. [13], Kim et al. [10], and Zhang et al. [14]. These methods can produce temporally coherent results but may struggle with complex motion or occlusions.

In this work, I propose to extend the use of diffusion models, as used in the *Repaint* paper [23] for image inpainting, to video inpainting. By applying diffusion models to video inpainting. The aim is to achieve high-quality and temporally coherent results by applying diffusion models to video inpainting.

3 Preliminaries

Denoising Diffusion Probabilistic Models (DDPMs)

Suppose we are given a dataset $x_1, x_2, \ldots, x_N$, where each point is drawn independently from an underlying data distribution $p(x)$. The aim of generative modeling with this dataset is to find a model that matches the data distribution so that we can create new data points by drawing from the distribution. Diffusion probabilistic model is a parameterized Markov chain trained using variational inference to produce samples matching the data after finite time[15]. Diffusion Models are generative models that work by destroying training data through the successive addition of Gaussian noise, and then learning to recover the data by reversing this noising process. After training, we can use the Diffusion Model to generate data by simply passing randomly sampled noise through the learned denoising process.

Let $\mathbf{x}_t$ for $t = 1, \ldots, T$ be a sequence of latent variables in the same sample space as $\mathbf{x}_0$. The forward process is defined by the Markov chain:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{1}$$

where

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

, and β_t is a small constant that represents a noise level. A sampling of $\mathbf{x}_t$ has the closed-form $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0, (1-\alpha_t)\mathbf{I})$, where $\hat{\alpha} := 1 - \beta_t$ and $\alpha_t := \prod_{i=1}^{t} \hat{\alpha}_i$. Then, using reparametrization trick,$\mathbf{x}_t$ can be expressed as $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + (1-\alpha_t)\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$.

The reverse process denoises $\mathbf{x}_t$ to recover $\mathbf{x}_0$, and is defined by the following Markon chain:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \mathbf{x}_T \sim \mathcal{N}(0, I), p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_\theta(\mathbf{x}_t, t)I). \tag{2}$$

Ho et al. [15], parameterize $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ as:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\alpha_t}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(\mathbf{x}_t, t)), \sigma_\theta(\mathbf{x}_t, t) = \hat{\beta}_t^{\frac{1}{2}} \ where \ \hat{\beta}_t = \begin{cases} \frac{1-\alpha_{t-1}}{1-\alpha_t}\beta_t & t > 1 \\ \beta_1 & t = 1 \end{cases} \tag{3}$$

Under this parametrization, Ho et al. [15] have shown that the reverse process can be trained by solving the following optimization problem:

$$\min_\theta \mathcal{L}(\theta) := \min_\theta \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(0,I), t} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2, \tag{4}$$

$$where \ \mathbf{x}_t = \sqrt{\alpha_t}x_0 + (1-\alpha_t)\epsilon.$$

Conditional Denoising Diffusion Model

Conditional DDPMs are a type of generative model that can be used for tasks such as image inpainting [23], image super-resolution [18], text-to-image generation [26], [28], image-to-image translation [27] etc. In this type of diffusion model, we are interested in learning

conditional distribution $p(\mathbf{x}|y)$, which would enable us to explicitly control the data we generate through conditioning information y. Generally, there are three ways of applying conditions to the learnable model. The first one is conditioning as part of the input data. In this method, the condition input is concatenated with the input data [20], [23]. The second conditioning method is through the attention module, where the input data is used as Query (Q), and the condition as key (K) and value (V) to build a connection between a data and the condition [26]. The third one is Adaptive Instance Normalization (AdaIN), which was proposed by Huang et al. [5]. In this method, AdaIN receives a content input x and a style input y and simply aligns the channel-wise mean and variance of x to match those of y. In my case, I use the first conditioning technique, which is to add conditioning information simply alongside the timestep information, at each iteration. The reverse process in the conditional diffusion model can be defined as:

$$p(x_{0:T}|y) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, y) \tag{5}$$

In the above equation, y could be a text encoding in text-to-image generation [25], or a low-resolution image to perform super-resolution on[18]. We can parameterize the reverse transition with a neural network like the unconditional one, $\hat{\epsilon}_\theta(x_t, t, y)$ where we want to predict the added noise in each iteration. Similarly, as shown in [24], we can predict the original input $x_0 \approx \hat{x}_\theta(x_t, t, y)$ or the score $s_\theta(x_t, t, y) \approx \nabla \log p(x_t|y)$.

Latent Diffusion Models

Latent diffusion models (LDMs) are a type of diffusion model where the diffusion is applied in the latent space instead of pixel space. In the work by Rombach et al. [26], they showed that LDMs achieve new state-of-the-art scores for image inpainting and class-conditional image synthesis and highly competitive performance on various tasks, including text-to-image synthesis, unconditional image generation, and super-resolution, while significantly reducing computational requirements compared to pixel-based DMs. The corresponding objective function used in Rombatch et al[26] is:

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2], , \tag{6}$$

where both τ_θ and ϵ_θ are jointly optimized. τ_θ is a domain-specific encoder that projects a condition input to an intermediate representation. This intermediate representation is then mapped to the intermediate layers of the UNet via a cross-attention layer.

4 Method

In my approach to video inpainting, I first convert the input video into a latent representation using a pre-trained autoencoder [26]. This allows to represent the video in a more compact and efficient manner. The autoencoder is trained to encode the input video into a lower-dimensional representation while preserving its essential features.

Next, I apply forward diffusion on the latent features to generate new frames that resemble the original video. Forward diffusion involves adding Gaussian noise to the latent representation at each time step. This gradually destroys the information contained in the original video, leaving only noise at the end of the process. However, by training a denoising model to reverse this process, we can recover the original video from the noisy representation.

To achieve this, I make use of both conditional and latent diffusion models. Working in latent space allows for computationally efficient diffusion and sampling. Since the masked video is used as input during training, I use a conditional diffusion model. This allows for the model to take into account the specific characteristics of the masked video and generate frames that seamlessly integrate with the rest of the video. By combining conditional and latent diffusion models, it was possible to achieve high-quality and temporally coherent results in the video inpainting task.

4.1 Problem Formulation

Suppose $x_0 \in \mathbb{R}^{F,W,H,C}$ is the ground truth video, $x_m \in \mathbb{R}^{F,W,H,C}$ the masking video, $y \in \mathbb{R}^{F,W,H,C}$ be masked video (condition input), and $x_t \in \mathbb{R}^{F,W,H,C}$ be the noisy input (diffused x_0), then we want to generate $\hat{x_0}$ (reconstructed video) by optimizing the following:

$$\min_{\theta} \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(0,I), t} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{y})\|_2^2, \tag{7}$$

4.2 Implementation

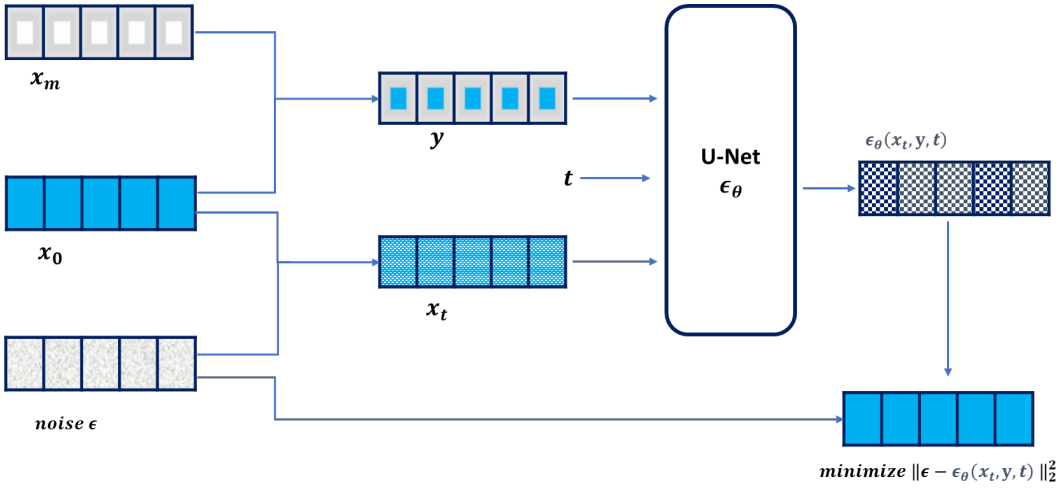

Figure 1: An overview of the implementation

First, the masking image is applied to the ground truth video. I applied a *binary* masking type, where a rectangular mask is applied in each frame of the video. The masked region is filled with a color that matches the background of the original image. This creates a hard mask where the objects in the ground truth are lost and only the background is visible. Next, the masked or corrupted video is converted into a latent space using the pre-trained autoencoder from *stable-diffusion* [26], then the latent input will be fed to the diffusion model. I followed the training in the *RamVid* [21], where the diffusion part of *Ramvid* followed the *DDPM* [15].
For simplicity, the masked frames and the diffused original frames are concatenated through the input channel and fed into the 3D convolutional UNet, and for the output there's only the denoised diffused original frames. I trained the diffusion model on the *CLEVRER* dataset [16]. In order to harmonize the generated frames, I resample multiple times motivated by the work from *RePaint* [23].

During sampling, the generated samples are not consistent across each frames, and there was a need for resampling. It involves reintroducing the forward process, where a less noisy video is encoded back to a noisier video and then denoised again. This preserves the generated information in the video and results in a more semantically harmonized output. The generated latent later is converted back to the pixel space using the decoder from the pre-trained autoencoder.

5 Experiments

I tested the performance of the model on the CLEVRER dataset. I evaluated the model under different settings, including stationary and variable maskings in each frame. I also varied the number of resampling steps to assess the impact on the final generated results.

- Applying stationary masking on each frames

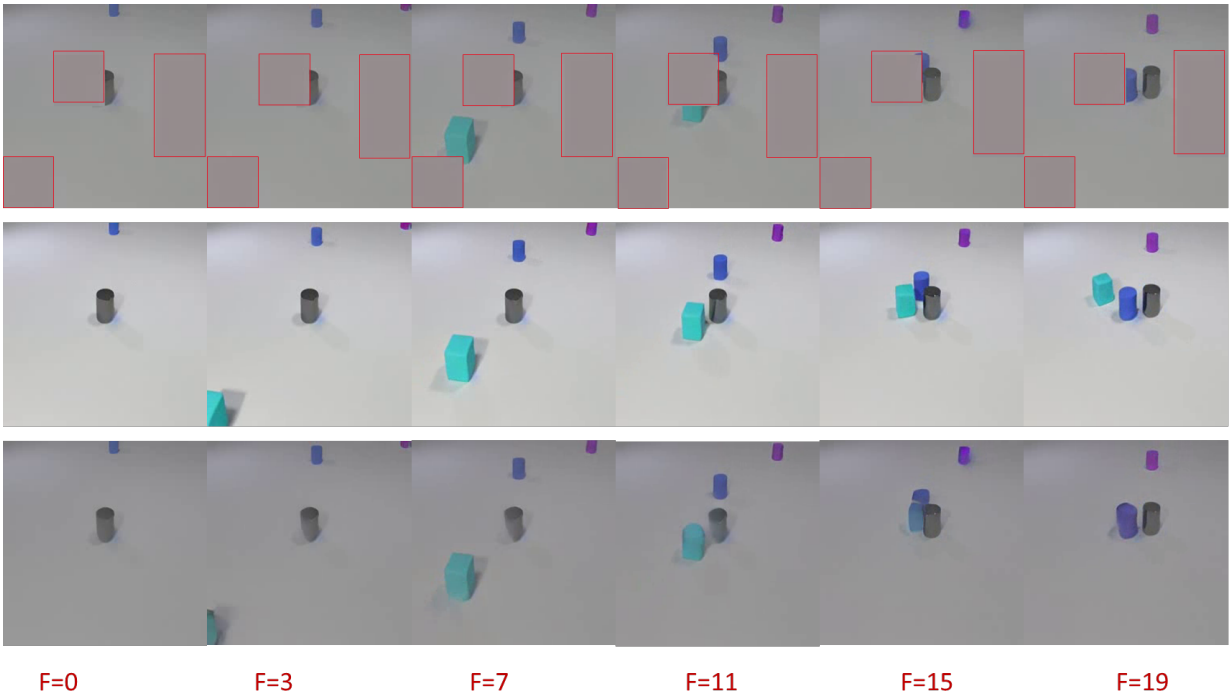

Figure 2: The top row shows the masked frames, the middle row shows the ground truth frames, and the bottom row shows the generated frames. The red box shows the masking region. Note that, the red boundary is added manually to highlight the masking areas and is not part of the original frames. While the total frame length was 20, only selected frames are shown in the figure for the sake of illustration.

- Applying variable masking on each frames In this setting, different masking regions are applied to each frame of the video. The figure below shows a sample of the generated frames, along with the corresponding ground truth and masked frames.

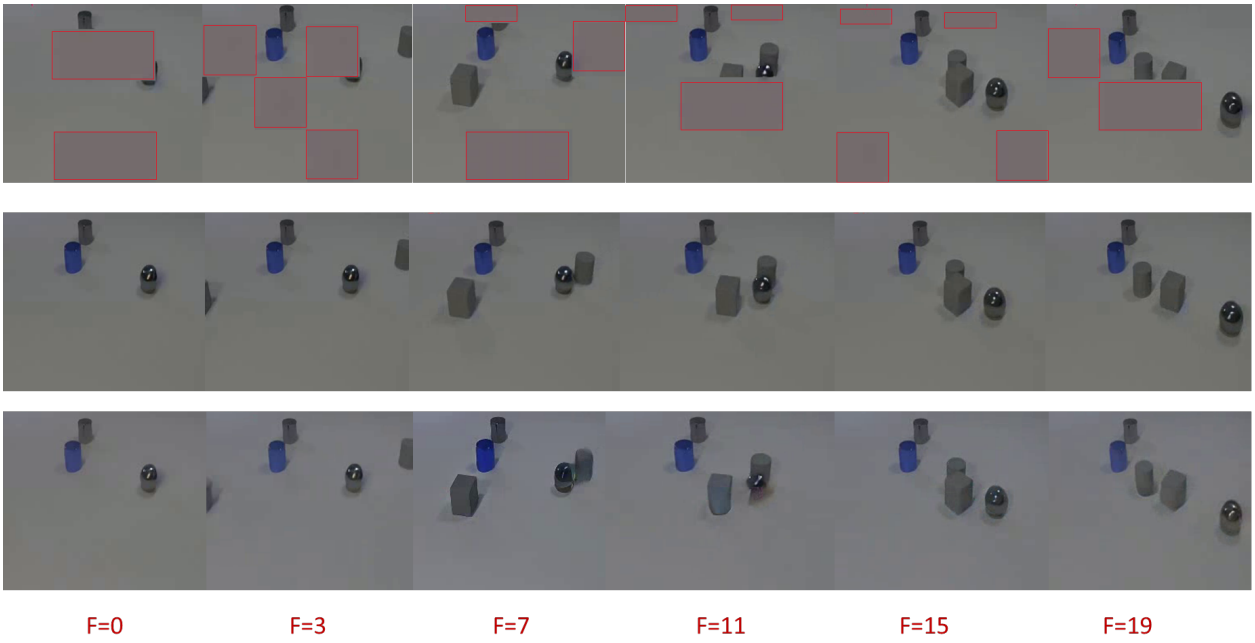

Figure 3: From top to bottom rows: masked frames, ground truth frames and generated frames

- Results with different Resampling steps *(n)*

  The figure below shows the effect of applying different numbers of resampling steps. As the number of resampling steps increases, the generated samples become more harmonized across the frames, resulting in improved consistency and realism. Resampling involves reintroducing the forward process, where a less noisy video is encoded back to a noisier video and then denoised again. This preserves the generated information in the video and results in a more semantically harmonized output.

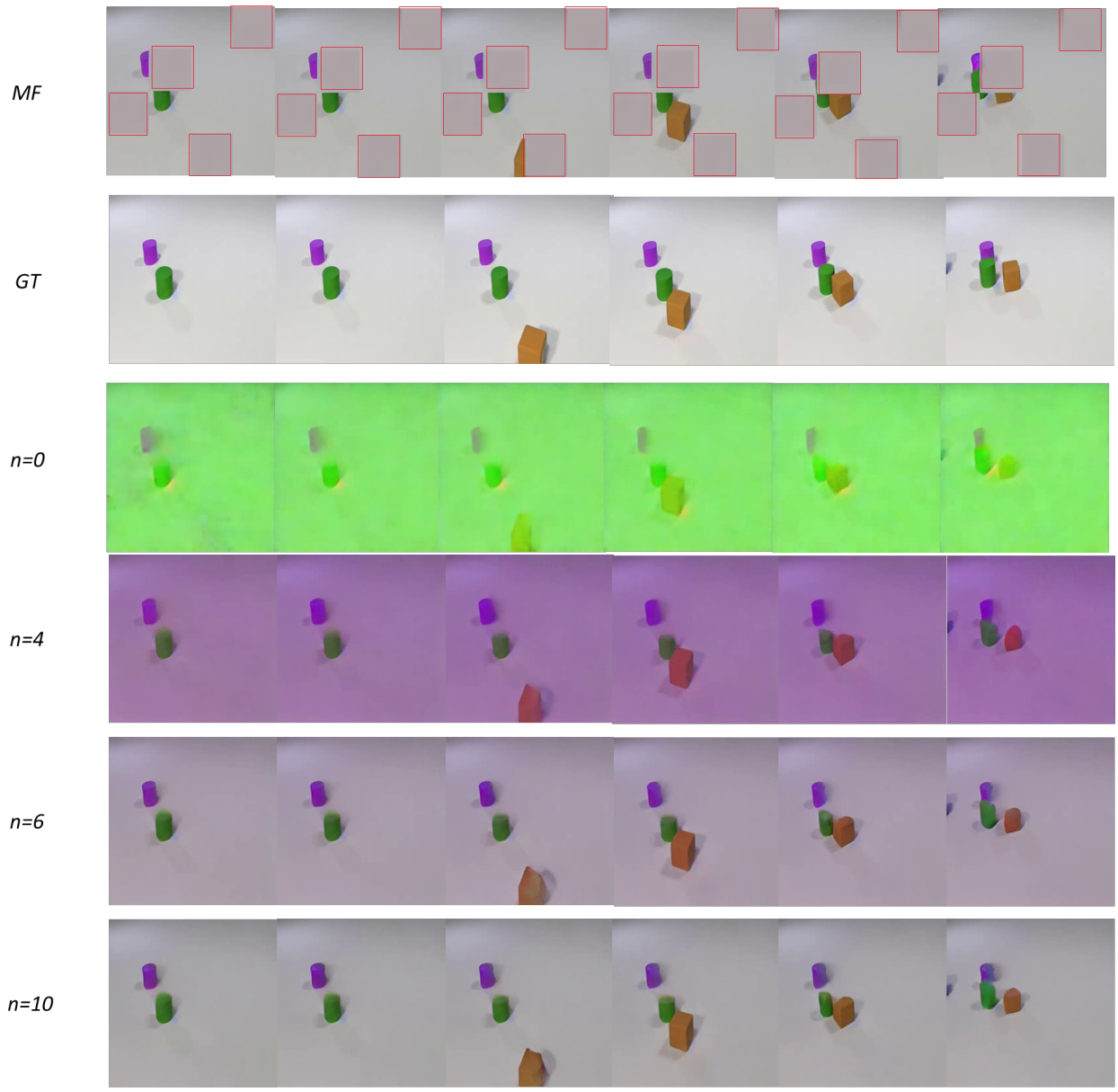

Figure 4: Rows arranged from top to bottom: MF - Masked frames; GT - Ground truth frames; Generated frames with default sampling (n=0); Generated frames with 4 resampling steps (n=4); Generated frames with 6 resampling steps (n=6); Generated frames with 10 resampling steps (n=10)

6 Conclusion

In this work, a conditional DDPM was applied to the video inpainting problem, and tested on the *CLEVRER* dataset. Two types of masking techniques were used: stationary and variable. The variable masking technique produced better generative results in comparison. To improve the consistency and realism of the generated frames, a resampling technique was employed. As the number of resampling steps increased, the quality of the generated samples improved. However, the resampling technique has its limitations. As the number of resampling steps increases, so does the inference time. To address this limitation, future work could explore a trade-off between the number of resampling steps and diffusion steps. Additionally, other variants of *DDPM* that offer faster sampling could be investigated. Further improvements could also be made by testing the model on more challenging datasets.

References

[1] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 463–476, 2007. DOI: 10.1109/TPAMI.2007.60.

[2] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 29, no. 3, pp. 463–476, 2007.

[3] K. He and J. Sun, "Image completion approaches using the statistics of similar patches," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 12, pp. 2423–2435, 2014.

[4] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, "Video inpainting of complex scenes," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 1993–2019, Jan. 2014. DOI: 10.1137/140954933. [Online]. Available: https://doi.org/10.1137%2F140954933.

[5] X. Huang and S. Belongie, *Arbitrary style transfer in real-time with adaptive instance normalization*, 2017. arXiv: 1703.06868 [cs.CV].

[6] C. Wang, H. Huang, X. Han, and J. Wang, *Video inpainting by jointly learning temporal structure and spatial details*, 2018. arXiv: 1806.08482 [cs.CV].

[7] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5505–5514.

[8] X. Zhu, Y. Qian, X. Zhao, B. Sun, and Y. Sun, "A deep learning approach to patch-based image inpainting forensics," *Signal Processing: Image Communication*, vol. 67, pp. 90–99, 2018.

[9] Y.-L. Chang, Z. Y. Liu, K.-Y. Lee, and W. Hsu, *Free-form video inpainting with 3d gated convolution and temporal patchgan*, 2019. arXiv: 1904.10247 [cs.CV].

[10] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, *Deep video inpainting*, 2019. arXiv: 1905.01639 [cs.CV].

[11] H. Li and J. Huang, "Localization of deep inpainting using high-pass fully convolutional network," in *proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8301–8310.

[12] W. Xiong, J. Yu, Z. Lin, *et al.*, "Foreground-aware image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5840–5848.

[13] R. Xu, X. Li, B. Zhou, and C. C. Loy, *Deep flow-guided video inpainting*, 2019. arXiv: 1905.02884 [cs.CV].

[14] H. Zhang, L. Mai, N. Xu, Z. Wang, J. Collomosse, and H. Jin, "An internal learning approach to video inpainting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2720–2729.

[15] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[16] K. Yi*, C. Gan*, Y. Li, *et al.*, "Clevrer: Collision events for video representation and reasoning," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HkxYzANYDB.

[17] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, and C. Etmann, *Conditional image generation with score-based diffusion models*, 2021. arXiv: 2111.13606 [cs.LG].

[18] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, *Image super-resolution via iterative refinement*, 2021. arXiv: 2104.07636 [eess.IV].

[19] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, *Score-based generative modeling through stochastic differential equations*, 2021. arXiv: 2011.13456 [cs.LG].

[20] Y. Tashiro, J. Song, Y. Song, and S. Ermon, *Csdi: Conditional score-based diffusion models for probabilistic time series imputation*, 2021. arXiv: 2107.03502 [cs.LG].

[21] T. Höppe, A. Mehrjou, S. Bauer, D. Nielsen, and A. Dittadi, *Diffusion models for video prediction and infilling*, 2022. arXiv: 2206.07696 [cs.CV].

[22] J. Kang, S. W. Oh, and S. J. Kim, "Error compensation framework for flow-guided video inpainting," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XV*, Springer, 2022, pp. 375–390.

[23] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 461–11 471.

[24] C. Luo, *Understanding diffusion models: A unified perspective*, 2022. arXiv: 2208.11970 [cs.LG].

[25] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, *Hierarchical text-conditional image generation with clip latents*, 2022. arXiv: 2204.06125 [cs.CV].

[26] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, *High-resolution image synthesis with latent diffusion models*, 2022. arXiv: 2112.10752 [cs.CV].

[27] C. Saharia, W. Chan, H. Chang, *et al.*, *Palette: Image-to-image diffusion models*, 2022. arXiv: 2111.05826 [cs.CV].

[28] C. Saharia, W. Chan, S. Saxena, *et al.*, *Photorealistic text-to-image diffusion models with deep language understanding*, 2022. arXiv: 2205.11487 [cs.CV].