Diffusion Models for Video Prediction and Infilling

Tobias Höppe
KTH Stockholm
tobihoeppe@gmail.com

Arash Mehrjou
MPI for Intelligent Systems

Stefan Bauer
KTH Stockholm

Didrik Nielsen
Norwegian Computing Center

Andrea Dittadi
Technical University of Denmark
adit@dtu.dk

Abstract

Video prediction and infilling require strong, temporally coherent generative capabilities. Diffusion models have shown remarkable success in several generative tasks, but have not been extensively explored in the video domain. We present Random-Mask Video Diffusion (RaMViD), which extends image diffusion models to videos using 3D convolutions, and introduces a new conditioning technique during training. By varying the mask we condition on, the model is able to perform video prediction, infilling, and upsampling. Due to our simple conditioning scheme, we can utilize the same architecture as used for unconditional training, which allows us to train the model in a conditional and unconditional fashion at the same time. We evaluate the model on two benchmark datasets for video prediction, on which we achieve state-of-the-art results, and one for video generation. High-resolution videos are provided at https://sites.google.com/view/video-diffusion-prediction

1 Introduction

Videos contain rich information about the world, and a vast amount of diverse video data is available. Training models on this data for video prediction or video infilling—i.e., observing a part of a video and generating missing frames —can be used in planning, estimating trajectories, and video processing. In addition, video models can be valuable for downstream tasks such as action recognition [17] and pose estimation [26]. Video prediction can be modelled in a deterministic or stochastic form. Deterministic modelling [34, 35, 41, 42] tries to predict the most likely future, but this often leads to averaging the future states [19]. Therefore, most recent methods are based on generative modeling, either using variational methods [2, 3, 8, 30, 44] or GANs [7, 20]. Diffusion models [1, 10, 13, 21, 23, 31, 32] have seen tremendous progress on static visual data, even outperforming GANs in image synthesis [9]. Only a few concurrent works have recently considered diffusion models for video generation. [47] uses diffusion models for autoregressive video prediction, by modeling residuals for a predicted frame, [14] focuses on unconditional video generation, [12] uses diffusion models to predict long videos, and [39], the most closely related to our work, also considers video prediction and infilling. The essence of diffusion models are two stochastic (diffusion) processes implemented by Stochastic Differential Equations (SDEs), a forward and a backward one. Let $\mathbf{x}_0 \in \mathbb{R}^d$ be a sample from the empirical data distribution, i.e., $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0)$ and d be the data dimension. The forward diffusion process takes $\mathbf{x}_0$ as the starting point and creates the random trajectory $\mathbf{x}_{[0,T]}$ from $t = 0$ to the final time $t = T$. The forward process is designed such that $p(\mathbf{x}_T \mid \mathbf{x}_0)$ has a simple unstructured distribution. One example of such SDEs is

$$d\mathbf{x}_t = f(\mathbf{x}_t, t)dt + g(t)dw := \sqrt{\frac{d[\sigma^2(t)]}{dt}}dw, \tag{1}$$

NeurIPS 2022 Workshop on Score-Based Methods.

where w is the Brownian motion. A desirable property of this process is the fact that the conditional distribution $p(\mathbf{x}_t \mid \mathbf{x}_0)$ takes a simple analytical form:

$$p(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \mathbf{x}_0, \left(\sigma^2(t) - \sigma^2(0)\right)\mathbf{I}\right). \tag{2}$$

Upon learning the gradient of $p(\mathbf{x}_t)$ for each t, one can reverse the above process and obtain the complex data distribution from pure noise as

$$d\mathbf{x}_t = [f(\mathbf{x}_t, t) - g^2(t)\nabla_{\mathbf{x}} \log p(\mathbf{x}_t)]dt + g(t)dw', \tag{3}$$

where w' is a Brownian motion independent of the one in the forward direction. Hence, generating samples from the data distribution boils down to learning $\nabla_{\mathbf{x}} \log p(\mathbf{x})$.

The original score matching objective [15]:

$$\mathbb{E}_{\mathbf{x}_t}\left[\|s_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\|_2^2\right] \tag{4}$$

is the most intuitive way to learn the score function, but is unfortunately intractable. Denoising Score Matching (DSM) provides a tractable alternative objective function:

$$J_t^{DSM}(\theta) = \mathbb{E}_{\mathbf{x}_0}\mathbb{E}_{\mathbf{x}_t|\mathbf{x}_0}\left[\|s_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid \mathbf{x}_0)\|_2^2\right] \tag{5}$$

whose equivalence with the original score matching objective was shown by [38] and used to train energy models in [29]. Similar to many recent works, we use the DSM formulation of score matching in this work to learn the score function.

In this paper, we extend diffusion models to the video domain via several technical contributions. We use 3D convolutions and a new conditioning procedure incorporating randomness. Our model is not only able to predict future frames of a video but also fill in missing frames at arbitrary positions in the sequence. Therefore, our Random-Mask Video Diffusion (RaMViD) can be used for several video completion tasks. We summarize our technical contributions as follows:

- A novel diffusion-based architecture for video prediction and infilling.
- Competitive performance with recent approaches across multiple datasets.
- Introduce a schedule for the random masking.

2 Random-Mask Video Diffusion

Our method, Random-Mask Video Diffusion (RaMViD), consists of two main features. First, the way we introduce conditional information to the network is different from what has been used so far. Second, by randomizing the mask, we can directly use the same approach for video prediction and video completion (infilling). In the following, we detail each of these aspects of the proposed method.

2.1 Conditional training

Let $\mathbf{x}_0 \in \mathbb{R}^{L,W,H,C}$ be a video with length L. We partition the video $\mathbf{x}_0$ into two parts, the unknown frames $\mathbf{x}_0^{\mathcal{U}} \in \mathbb{R}^{L-k,W,H,C}$ and the conditioning frames $\mathbf{x}_0^{\mathcal{C}} \in \mathbb{R}^{k,W,H,C}$, where $\mathcal{U}$ and $\mathcal{C}$ are sets of indices such that $\mathcal{U} \cap \mathcal{C} = \varnothing$ and $\mathcal{U} \cup \mathcal{C} = \{0, 1, \ldots, L-1\}$. We write $\mathbf{x}_0 = \mathbf{x}_0^{\mathcal{U}} \oplus \mathbf{x}_0^{\mathcal{C}}$ with the following definition for the $\oplus$ operator:

$$(\mathbf{a}^{\mathcal{U}} \oplus \mathbf{b}^{\mathcal{C}})^i := \begin{cases} \mathbf{a}^i \text{ if } i \in \mathcal{U} \\ \mathbf{b}^i \text{ if } i \in \mathcal{C} \end{cases} \tag{6}$$

where the superscript i indicates tensor indexing and in our case corresponds to selecting a frame from a video, and the subscript t indicates the diffusion step, with $t = 0$ corresponding to the data and $t = T$ to the prior Gaussian distribution. If we use an unconditionally trained model and sample via the replacement method [32], we find that the predicted unknown frames $\mathbf{x}_0^{\mathcal{U}}$ do not harmonize well with the conditioning frames $\mathbf{x}_0^{\mathcal{C}}$. To mitigate this issue, we propose to train the model conditionally with *randomized masking*.

Conditional diffusion models usually optimize

$$\mathbb{E}_{\mathbf{x}_0}\left\{\mathbb{E}_{\mathbf{x}_t|\mathbf{x}_0}\left[\left\|s_\theta(\mathbf{x}_t, \mathbf{x}_0^{\mathcal{C}}, t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid \mathbf{x}_0)\right\|_2^2\right]\right\} \tag{7}$$

where $\mathbf{x}_0^{\mathcal{C}}$ is typically given as a separate input through an additional layer [6] or it is expanded to the dimension of $\mathbf{x}_t$ (e.g., via padding) and concatenated with the input [4, 24, 25]. We, on the other hand, feed the entire sequence to the network s_θ but only add noise to the unmasked frames: $\mathbf{x}_t^{\mathcal{U}} \sim \mathcal{N}\left(\mathbf{x}_0^{\mathcal{U}}, \left(\sigma^2(t) - \sigma^2(0)\right)\mathbf{I}\right)$. The input to the network is then a video where some frames are noisy and some are clean: $\mathbf{x}_t = \mathbf{x}_t^{\mathcal{U}} \oplus \mathbf{x}_0^{\mathcal{C}}$. The loss is computed only with respect to $\mathbf{x}_t^{\mathcal{U}}$:

$$J_t^{\text{RaMViD}}(\theta) = \mathbb{E}_{\mathbf{x}_0}\left\{\mathbb{E}_{\mathbf{x}_t^{\mathcal{U}}|\mathbf{x}_0}\left[\left\|s_\theta(\mathbf{x}_t, t)^{\mathcal{U}} - \nabla_{\mathbf{x}_t^{\mathcal{U}}} \log p(\mathbf{x}_t^{\mathcal{U}} \mid \mathbf{x}_0)\right\|_2^2\right]\right\}. \tag{8}$$

Note that the score function $\nabla_{\mathbf{x}_t^{\mathcal{U}}} \log p(\mathbf{x}_t^{\mathcal{U}} \mid \mathbf{x}_0)$ has the same dimension as $\mathbf{x}_t^{\mathcal{U}}$, whereas in Eq. (7) it had the dimension of the entire video $\mathbf{x}_t$. The reversed diffusion process then becomes:

$$d\mathbf{x}_t^{\mathcal{U}} = [f(\mathbf{x}_t^{\mathcal{U}}, t) - g^2(t)\nabla_{\mathbf{x}_t^{\mathcal{U}}} \log p(\mathbf{x}_t^{\mathcal{U}}|\mathbf{x}_0^{\mathcal{C}})]dt + g(t)dw' \tag{9}$$

2.2 Randomization

As previously mentioned, the proposed model is able to perform several tasks. We achieve this by sampling $\mathcal{C}$ at random. At each training step, we first choose the number of conditioning frames $|\mathcal{C}| = k \in \{1, \dots, K\}$, where K is a chosen hyperparameter. Then we define $\mathcal{C}$ by selecting k random indices from $\{0, \dots, L-1\}$, and we refrain from applying the diffusion process to the corresponding frames. After training, we can use RaMViD by fixing $\mathcal{C}$ to the set of indices of the known frames ($\mathcal{C}$ can be any arbitrary subset of $\{0, \dots, L-1\}$) and generating the unknown frames (those with indices in $\mathcal{U}$). Our approach allows us to use the exact same architecture of unconditionally trained models, thus enabling *mixed training*, where we train the model conditionally and unconditionally at the same time. We set $\mathcal{C} = \varnothing$ (i.e., the model does not have any conditional information $\mathbf{x}_t^{\mathcal{C}}$) with probability p_U, which is a fixed hyperparameter. If $\mathcal{C} = \varnothing$, our objective in Eq. (8) becomes the same as the objective in Eq. (5) used for unconditional training.

3 Experiments

To compare our model to prior work, we train it on three datasets: BAIR robot pushing [11] and Kinetics-600 [5] for video prediction and completion and UCF-101 for unconditional generation [33]. We train all datasets on 64×64 resolution and choose $K = 4$. To quantitatively evaluate prediction, we use the Fréchet Video Distance (FVD) [37],[1] and to evaluate unconditional generation the Inception Score (IS) [28] with the implementation from [27].[2].

3.1 BAIR

We train four models on the BAIR dataset on 20 frames with $p_U \in \{0, 0.25, 0.5, 0.75\}$ respectively. The models are trained for 250,000 iterations with a batch size of 32 on 8 GPUs. First, we test our method with the typical evaluation protocol for BAIR (predicting 15 frames, given one conditional frame). With all values of p_U, we can achieve state-of-the-art performance, as shown in Table 1. By using $p_U > 0$, we can even increase the performance of our method. However, it seems that there is a tipping point after which the increasing unconditional rate hurts the prediction performance of the model. Thanks to the randomized masking, RaMViD is also able to perform video infilling. Quantitative results are shown in Appendix B. The method works very well for prediction and infilling. However, since the BAIR dataset is arguably rather simple and not very diverse, we will now evaluate RaMViD on the significantly more complex Kinetics-600 dataset.

3.2 Kinetics-600

For the Kinetics-600 dataset, we increase the batch size to 64 and train for 500,000 iterations on 8 GPUs, but train only on 16 frames. First, we evaluate the model on prediction (predict 11 frames given 5 frames). When comparing our models to concurrent work, we find that RaMViD achieves state-of-the-art results by a significant margin (see Table 2). Nevertheless, it struggles with fast

[1] `https://github.com/google-research/google-research/tree/master/frechet_video_distance`

[2] `https://github.com/pfnet-research/tgan2`

Table 1: Prediction performance on BAIR. The values are taken from [3] after inquiring about the evaluation procedure. We have obtained the parameter counts either directly from the papers or by contacting the authors.

Method	FVD (↓)	# parameters
SAVP [18]	116.4	
DVD-GAN-FP [7]	109.8	
TrIVD-GAN-FP [20]	103.3	
VideoGPT [46]	103.3	40M
Video Transfomer [43]	94.0	373M
FitVid [3]	93.6	302M
MCVD [39]	89.5	251.2M
NÜWA [45]	86.9	
RaMViD ($p_U = 0, K = 4$)	86.41	235M
RaMViD ($p_U = 0.25, K = 4$)	84.20	235M
RaMViD ($p_U = 0.5, K = 4$)	85.03	235M
RaMViD ($p_U = 0.75, K = 4$)	86.05	235M

Table 2: Prediction performance on Kinetics-600. Values are taken from [22] after inquiring about the evaluation procedure. We have obtained the parameter counts either directly from the papers or by contacting the authors.

Method	FVD (↓)	# parameters
Video Transfomer [43]	170 ± 5	373M
DVD-GAN-FP [7]	69 ± 1	
CCVS [22]	55 ± 1	366M
TrIVD-GAN-FP [20]	26 ± 1	
RaMViD ($p_U = 0$)	18.69	308M
RaMViD ($p_U = 0.25$)	**16.46**	308M
RaMViD ($p_U = 0.5$)	17.61	308M
RaMViD ($p_U = 0.75$)	27.64	308M

movements: objects moving quickly often get deformed. Similar to what we have seen in Table 1, having an unconditional rate $p_U > 0$ increases the performance up to a tipping point. However, differently from the model trained on BAIR, the FVD score now drops significantly with $p_U = 0.75$. We conjecture that this drop in performance is due to the complexity of the data distribution. In BAIR, the conditional and unconditional distributions are rather similar, while this is not true for Kinetics-600. Also on Kinetics-600 we can perform several video completion tasks. For further experiments, we refer to Appendix C. Since we were able to generate videos unconditionally with the models RaMViD ($p_U = 0.5$) and RaMViD ($p_U = 0.75$), we show quantitative comparision on UCF-101 in Appendix D.

4 Conclusion

We have shown that diffusion models, which have been demonstrated to be remarkably powerful for image generation, can be extended to videos and used for several video completion tasks. The way we introduce conditioning information is novel, simple, and does not require any major modification to the architecture of existing diffusion models, but it is nonetheless surprisingly effective. Although the proposed method targets conditional video generation, we also introduce an alternative masking schedule in an attempt to improve the unconditional generation performance without sacrificing performance on conditional generation tasks. Finally, the focus of this work has been on the diffusion-based algorithm for videos rather than on optimizing the quality of each frame. It has been shown in concurrent works that including super-resolution modules helps create high-resolution videos. Adding a super-resolution module to RaMViD would be a relevant direction for future work.

References

[1] Korbinian Abstreiter, Stefan Bauer, Bernhard Schölkopf, and Arash Mehrjou. Diffusion-based representation learning. *arXiv preprint arXiv:2105.14257*, 2021.

[2] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *International Conference on Learning Representations*, 2018.

[3] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021.

[4] Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models, 2021.

[5] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600, 2018.

[6] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021.

[7] Aidan Clark, Jeff Donahue, and Karen Simonyan. Efficient video generation on complex datasets. *CoRR*, abs/1907.06571, 2019.

[8] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1174–1183. PMLR, 7 2018.

[9] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[10] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. *arXiv preprint arXiv:2112.07068*, 2021.

[11] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 344–356. PMLR, 11 2017.

[12] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos, 2022.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[14] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022.

[15] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[16] Emmanuel Kahembwe and Subramanian Ramamoorthy. Lower dimensional kernels for video discriminators. *Neural Networks*, 132:506–520, 2020.

[17] Yu Kong and Yun Fu. Human action recognition and prediction: A survey, 2018.

[18] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction, 2018.

[19] Maomao Li, Chun Yuan, Zhihui Lin, Zhuobin Zheng, and Yangyang Cheng. Stochastic video generation with disentangled representations. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 224–229, 2019.

[20] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data, 2020.

[21] Sarthak Mittal, Guillaume Lajoie, Stefan Bauer, and Arash Mehrjou. From points to functions: Infinite-dimensional representations in diffusion models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

[22] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. CCVS: Context-aware controllable video synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[23] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 7 2021.

[24] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models, 2021.

[25] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement, 2021.

[26] Caner Sahin, Guillermo Garcia-Hernando, Juil Sock, and Tae-Kyun Kim. A review on object pose recovery: from 3d bounding box detectors to full 6d pose estimators, 2020.

[27] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal GAN. *International Journal of Computer Vision*, 128(10-11):2586–2606, 5 2020.

[28] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved Techniques for Training GANs. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett,

editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.
[29] Saeed Saremi, Arash Mehrjou, Bernhard Schölkopf, and Aapo Hyvärinen. Deep energy estimator networks. *arXiv preprint arXiv:1805.08306*, 2018.
[30] Vaibhav Saxena, Jimmy Ba, and Danijar Hafner. Clockwork variational autoencoders. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.
[31] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 7 2015. PMLR.
[32] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.
[33] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012.
[34] Jiangxin Sun, Jiafeng Xie, Jian-Fang Hu, Zihang Lin, Jianhuang Lai, Wenjun Zeng, and Wei-shi Zheng. Predicting future instance segmentation with contextual pyramid convlstms. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 2043–2051, New York, NY, USA, 2019. Association for Computing Machinery.
[35] Adam Terwilliger, Garrick Brazil, and Xiaoming Liu. Recurrent flow-guided semantic forecasting. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1703–1712, 2019.
[36] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2018.
[37] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *CoRR*, abs/1812.01717, 2018.
[38] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.
[39] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation, 2022.
[40] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.
[41] Carl Vondrick and Antonio Torralba. Generating the future with adversarial transformers. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2992–3000, 2017.
[42] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2443–2451, 2015.
[43] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *International Conference on Learning Representations*, 2020.
[44] Bohan Wu, Suraj Nair, Roberto Martín-Martín, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2318–2328, 2021.
[45] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. NÜwa: Visual synthesis pre-training for neural visual world creation, 2021.
[46] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers, 2021.
[47] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation, 2022.

A Implementation details

Our implementation relies on the official code of [23], and we use the same U-Net architecture but adapted to video data by using 3D convolutions.[3] We do not encode the time dimension and we use two ResNet blocks per resolution for the BAIR dataset, and three blocks for Kinetics-600 and UCF-101. We set the learning rate for all our experiments to 2e-5, use a batch size of 32 for BAIR and 64 for Kinetics-600 and UCF-101, and fix $T = 1000$. We found, especially on the more diverse datasets like Kinetics-600 and UCF-101, that larger batch sizes produce better results. Therefore, to increase the batch size, we use gradient accumulation by computing the gradients for micro-batches of size 2 and accumulate for several steps before doing back-propagation. Even though most previous work uses the cosine noise schedule, we found that the linear noise schedule works better when training the model conditionally. A detailed graphic of our model is shown in Fig. 1.

B BAIR

Since we train with randomized masking, we can also perform video infilling with the same models, without retraining. We condition on the first and last frame (i.e., set $\mathcal{C} = \{0, 15\}$ for sampling) and compute the FVD of the 14 generated frames. Again we find that the performance is very similar for different values of p_U (see Table 3), however, similarly to Table 1, we observe the best results when using $p_U = 0.25$.

Table 3: Infilling performance on BAIR.

Method	FVD (↓)
RaMViD ($p_U = 0$)	85.68
RaMViD ($p_U = 0.25$)	**85.02**
RaMViD ($p_U = 0.5$)	87.04
RaMViD ($p_U = 0.75$)	87.85

C Kinetics-600

We also evaluate RaMViD on two video completion tasks on Kinetics-600. The first task is to fill in a video given the two first and last frames (i.e., $\mathcal{C} = \{0, 1, 14, 15\}$): the challenge here is to harmonize the observed movement at the beginning with the movement observed at the end. In the second task, the conditioning frames are distributed evenly over the sequence (i.e., $\mathcal{C} = \{0, 5, 10, 15\}$), hence the model has to infer the movement from the static frames and harmonize them into one realistic video. RaMViD excels on both tasks, as shown quantitatively in Table 4. Especially when setting $\mathcal{C} = \{0, 5, 10, 15\}$ RaMViD is able to fill the missing frames with very high quality and coherence. This setting can be easily applied to upsampling by training a model on high-FPS videos and then sampling a sequence conditioned on a low-FPS video.

Table 4: Performance of RaMViD on Kinetics-600, when conditioning on different frames.

Method	$\{0, 1, 14, 15\}$	$\{0, 5, 10, 15\}$
RaMViD ($p_U = 0$)	**10.68**	6.28
RaMViD ($p_U = 0.25$)	10.85	**4.91**
RaMViD ($p_U = 0.5$)	10.86	5.90
RaMViD ($p_U = 0.75$)	17.33	7.29

D UCF-101

We have mentioned that we found that RaMViD ($p_U = 0.5$) and RaMViD ($p_U = 0.75$) can generate unconditional videos on Kinetics-600. To quantify RaMViD's unconditional generation, we will

[3] `https://github.com/openai/improved-diffusion`

evaluate these models on the UCF-101 dataset and compare it to other work. We train RaMViD with the same setting as used for Kinetics-600 but for 450,000 iterations. Table 5 shows that our model achieves competitive performance on unconditional video generation, although it does not reach state-of-the-art. The trained models can successfully generate scenes with a static background and a human performing an action in the foreground, consistent with the training dataset. However, the actions are not always coherent and moving objects can deform over time. Note that UCF-101 is a very small dataset given its complexity. Therefore we do observe some overfitting. Since for each action we only have around 25 different settings, our model does not learn to combine those but generates very similar videos to the training set. Due to the characteristics of this dataset we think with more extensive hyperparameter tuning, one can achieve better results with RaMViD in unconditional generation. But our focus does not lie on this.

Table 5: Generative performance of RaMViD on UCF-101. We only compare to models which are also trained on 64×64 resolution. Since the IS score is computed with 112×112 resolution, models trained on higher resolution would have an advantage.

Method	**IS** ($\uparrow$)	
VGAN [40]	8.31 ± 0.09	
MoCoGAN [36]	12.42 ± 0.03	3.3M
TGAN-F [16]	13.62	17.5M
TGANv2 [27]	26.60 ± 0.47	200M
Video Diffusion [14]	**57** ± 0.62	
RaMViD ($p_U = 0.5$)	20.84 ± 0.08	308M
RaMViD ($p_U = 0.75$)	21.71 ± 0.21	308M

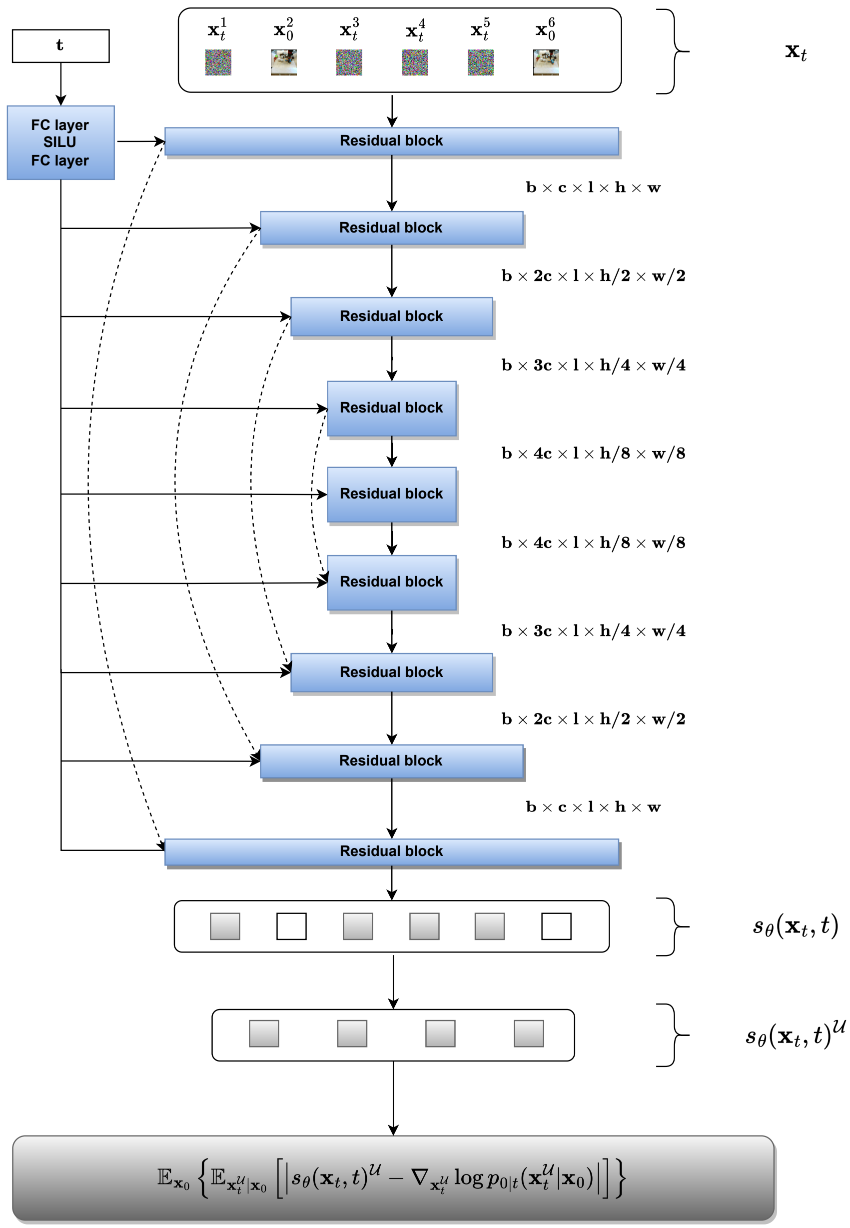

Figure 1: Sketch of our method. In the last step, we only compute the loss with respect to the frames that were corrupted with noise. The number of channels **c** is 128, and **l** is the video length.