Efficient Video Prediction via Sparsely Conditioned Flow Matching – Supplementary Material–

Aram Davtyan*, Sepehr Sameni*, Paolo Favaro
Computer Vision Group, Institute of Computer Science, University of Bern, Switzerland
{aram.davtyan, sepehr.sameni, paolo.favaro}@unibe.ch

A. Introduction

In the main paper we have introduced RIVER- a new model and an efficient training procedure to perform video prediction based on Flow Matching and randomized past frame conditioning. This supplementary material provides details that could not be included in the main paper due to space limitations. In section B we describe in details the architecture of our model and how we trained it on different datasets. In section C we show the training curve of the model and in section D we conduct an analysis on the training time and memory consumption and compare with that of other methods. In section F we provide more samples generated with our model.

B. Architecture and Training Details

Autoencoder. In this section we provide the configurations of the VQGAN [9] for all the datasets used in the main paper (see Table 5). All models were trained using the code from the official `taming transformers` repository.[1]

Vector Field Regressor. In this section we provide implementation details of the network that regresses the conditional time-dependent vector field $v_t(x \mid x^{\tau-1}, x^c, \tau-c)$. As mentioned in the main paper, the network is implemented as a U-ViT [4]. The detailed architecture is provided in Figure 10 and is shared across all datasets. First, the inputs $x, x^{\tau-1}$ and x^c are channel-wise concatenated and linearly projected to the inner dimension of the ViT blocks. Besides in and out projection layers, the network consists of 13 standard ViT blocks with 4 long skip connections between the first 4 and the last 4 blocks. At each skip connection the inputs are channel-wise concatenated and projected to the inner dimension of the ViT blocks. All ViT blocks apply layer normalization [1] before the multihead self-attention [23] (MHSA) layer and the MLP. The inner dimension of all ViT blocks is 768 and 8 heads are used in all MHSA layers.

All models are trained for 300K iterations with the AdamW [17] optimizer with the base learning rate equal to 10^{-4} and weight decay $5 \cdot 10^{-6}$. A learning rate linear warmup for 5K iterations is used along with a square root decay schedule. For the CLEVRER [28] dataset, random color jittering is additionally used to prevent overfitting. We observed that without it, the objects may change colors in the generated sequences (see Figure 12). In all experiments we used $\sigma_{\text{min}} = 10^{-7}$.

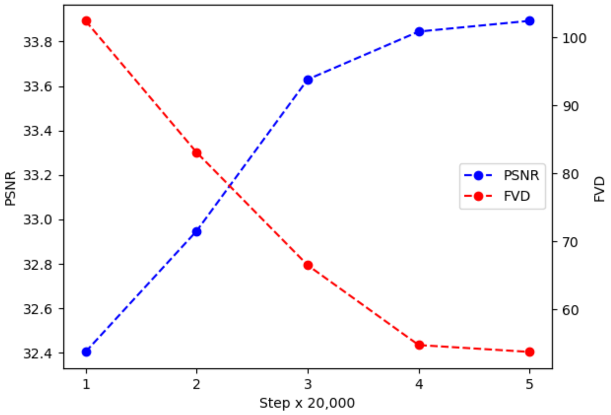

Figure 9. Training curve of RIVER on CLEVRER [28].

Additionally, we would like to highlight once again that the excellent tradeoff of RIVER demonstrated in Figure 1 of the main paper is the motivation to use flow matching instead of diffusion. Flow matching exhibits faster convergence compared to diffusion models. Moreover, on BAIR we observed DDPM fail to converge (see Figure 11). Besides this, the same theoretical arguments used by the authors of flow matching in the case of images can be extended to the case of videos.

C. Training Curve

In Figure 9 we show the FVD [22] and PSNR of RIVER trained on CLEVRER [28] against the iteration time. As we can see, the training is stable and more iterations lead to better results.

*Equal contribution.

[1] https://github.com/CompVis/taming-transformers

	BAIR64×64 [8]	BAIR256×256 [8]	KTH [20]	CLEVRER [28]
embed_dim	4	8	4	4
n_embed	16384	16384	16384	8192
double_z	False	False	False	False
z_channels	4	8	4	4
resolution	64	256	64	128
in_channels	3	3	3	3
out_ch	3	3	3	3
ch	128	128	128	128
ch_mult	[1,2,2,4]	[1,1,2,2,4]	[1,2,2,4]	[1,2,2,4]
num_res_blocks	2	2	2	2
attn_resolutions	[16]	[16]	[16]	[16]
dropout	0.0	0.0	0.0	0.0
disc_conditional	False	False	False	-
disc_in_channels	3	3	3	-
disc_start	20k	20k	20k	-
disc_weight	0.8	0.8	0.8	-
codebook_weight	1.0	1.0	1.0	-

Table 5. Configurations of VQGAN [9] for different datasets. Notice that on the CLEVRER [28] dataset we did not utilize an adversarial training.

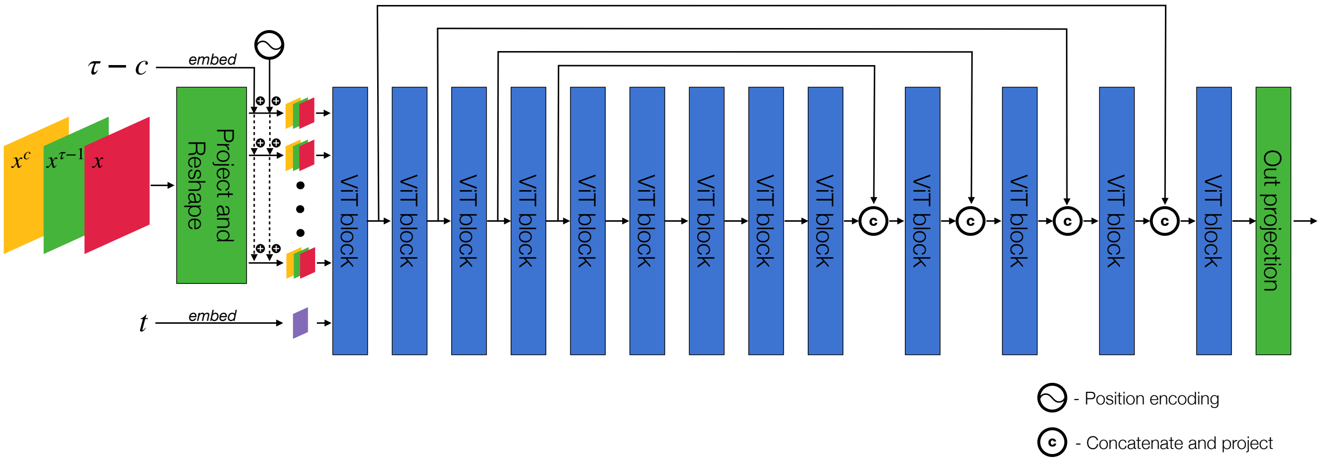

Figure 10. Architecture of the vector field regressor of RIVER. “ViT block” stands for a standard self-attention block used in ViT [7], that is an MHSA layer, followed by a 2-layer wide MLP, with a layer normalization before each block and a skip connection after each block. “Out projection” involves a linear layer, followed by a GELU [12] activation, layer normalization and a 3×3 convolutional layer.

D. Training Time and Memory Consumption

In Table 6, we compare the total training time and GPU (or TPU) memory requirements of different models trained on BAIR64×64 [8]. As we can see, RIVER is extremely efficient and can achieve a reasonable FVD [22] with significantly less compute than the other methods. For example, SAVP [16], which has the same FVD as RIVER, requires 4.6× more compute (measured by Mem×Time) and all the models that take less compute than RIVER have FVDs more than 250.

E. Sampling Speed

In this section we provide more comparisons in terms of the sampling speed with different models. We test the models on the BAIR 64 × 64 dataset, generating 16 frames and measuring the time the generation required. For evaluation we compare to some diffusion-based models with available code (RaMViD [13], MCVD [24]). In addition, we pick one RNN-based model (SRVP [11]) and one Transformer-based (LVT [19]), to cover different model architectures. The results are reported in Figure 13. Due to the sparse past frame conditioning, RIVER is able to generate videos with reasonable sampling time. However, if the focus is on the

Method	Memory (GB)	Time (Hours)	**Mem×Time (GB×Hour)**	FVD [22]
RVD [27]	24	-	-	1272
MoCoGAN [21]	16	23	368	503
SVG-FP [6]	12	24	288	315
CDNA [10]	10	20	200	297
SV2P [2]	16	48	768	263
SRVP [11]	36	168	6048	181
VideoFlow [14]	128	336	43008	131
LVT [19]	128	48	6144	126
SAVP [16]	32	144	4608	116
DVD-GAN-FP [5]	2048	24	49152	110
Video Transformer(S) [25]	256	33	8448	106
TriVD-GAN-FP [18]	1024	280	286720	103
CCVS(Low res) [15]	128	40	5120	99
MCVD(spatin) [24]	86	50	4300	97
Video Transformer(L) [25]	512	336	172032	94
FitVid [3]	1024	288	294912	94
MCVD(concat) [24]	77	78	6006	90
NUWA [26]	2560	336	860160	87
RaMViD [13]	320	72	23040	83
RIVER	25	25	625	106

Table 6. Compute comparisons. We report the memory and training times requirements of different models trained on BAIR64×64 [8]. The overall compute (Mem × Time) shows that RIVER delivers better FVD with much less compute.

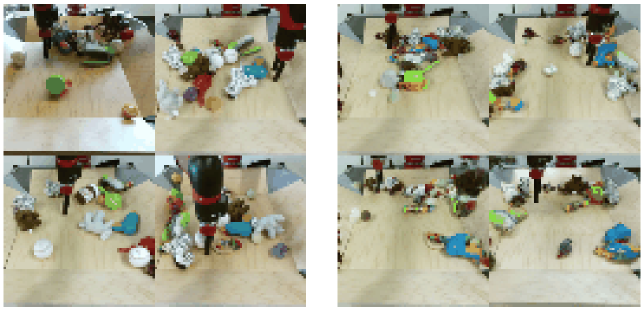

Figure 11. Video generation with different generative models. Use Acrobat Reader to play videos.

inference speed, one might opt for RNN-based models.

F. Qualitative Results

Here we provide more visual examples of the sequences generated with RIVER. See Figures 15 and 17 for results on the BAIR [8] dataset, Figures 14 and 16 for results on the KTH [20] dataset and Figures 18 and 20 for video prediction and planning on the CLEVRER [28] dataset respectively. Besides this, we highlight the stochastic nature of the generation process with RIVER in Figure 19 and the impact of extreme ($s > 0.5$) warm-start sampling strength in Figure 21. For more qualitative results and visual comparisons with the prior work, please, visit our website https://araachie.github.io/river.

References

[1] Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016. 1

[2] Mohammad Babaeizadeh, Chelsea Finn, D. Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. *ArXiv*, abs/1710.11252, 2018. 3

[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. 3

[4] Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. All are worth words: a vit backbone for score-based diffusion models. *arXiv preprint arXiv:2209.12152*, 2022. 1

[5] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv: Computer Vision and Pattern Recognition*, 2019. 3

[6] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International conference on machine learning*, pages 1174–1183. PMLR, 2018. 3

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021. 2

[8] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *CoRL*, 2017. 2, 3

[9] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Pro-*

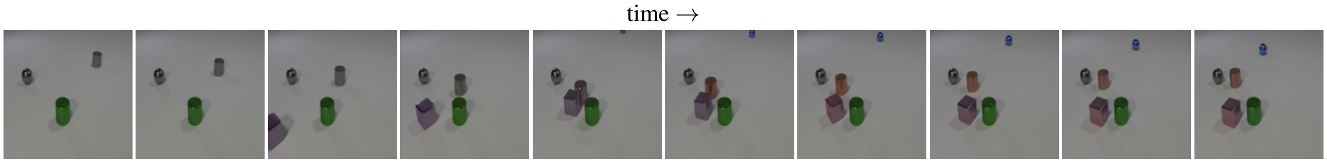

Figure 12. A sequence generated with RIVER trained on the *CLEVRER* dataset without data augmentation. Notice how the color of the grey cylinder changes after its interaction with the cube. In order to prevent such behaviour, both the autoencoder and RIVER are trained with random color jittering as data augmentation. The first frame can be played as a video in Acrobat Reader.

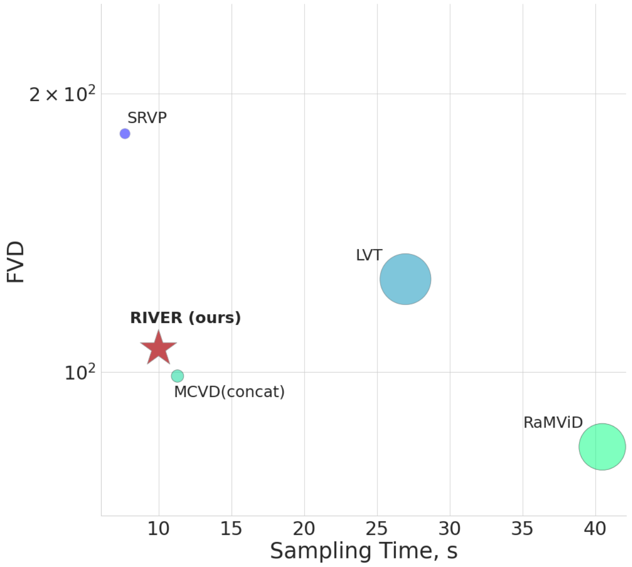

Figure 13. FVD vs. inference speed, the time required to generate a 16 frames long 64×64 resolution video on a single Nvidia GeForce RTX 3090 GPU. The sizes of the markers are proportional to the standard deviation of measured times in 20 independent experiments.

ceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 12873–12883, 2021. 1, 2

[10] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems*, 29, 2016. 3

[11] Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. In *ICML*, 2020. 2, 3

[12] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv: Learning*, 2016. 2

[13] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*, 2022. 2, 3

[14] Manoj Kumar, Mohammad Babaeizadeh, D. Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A conditional flow-based model for stochastic video generation. *arXiv: Computer Vision and Pattern Recognition*, 2020. 3

[15] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Ccvs: Context-aware controllable video synthesis. *Advances in Neural Information Processing Systems*, 34:14042–14055, 2021. 3

[16] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018. 2, 3

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 1

[18] 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. *ArXiv*, abs/2003.04035, 2020. 3

[19] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. *arXiv preprint arXiv:2006.10704*, 2020. 2, 3

[20] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36. IEEE, 2004. 2, 3

[21] S. Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1526–1535, 2018. 3

[22] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *ArXiv*, abs/1812.01717, 2018. 1, 2, 3

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1

[24] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation. *arXiv preprint arXiv:2205.09853*, 2022. 2, 3

[25] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. *arXiv preprint arXiv:1906.02634*, 2019. 3

[26] Chenfei Wu, Jian Liang, Lei Ji, F. Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. In *ECCV*, 2022. 3

[27] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*, 2022. 3

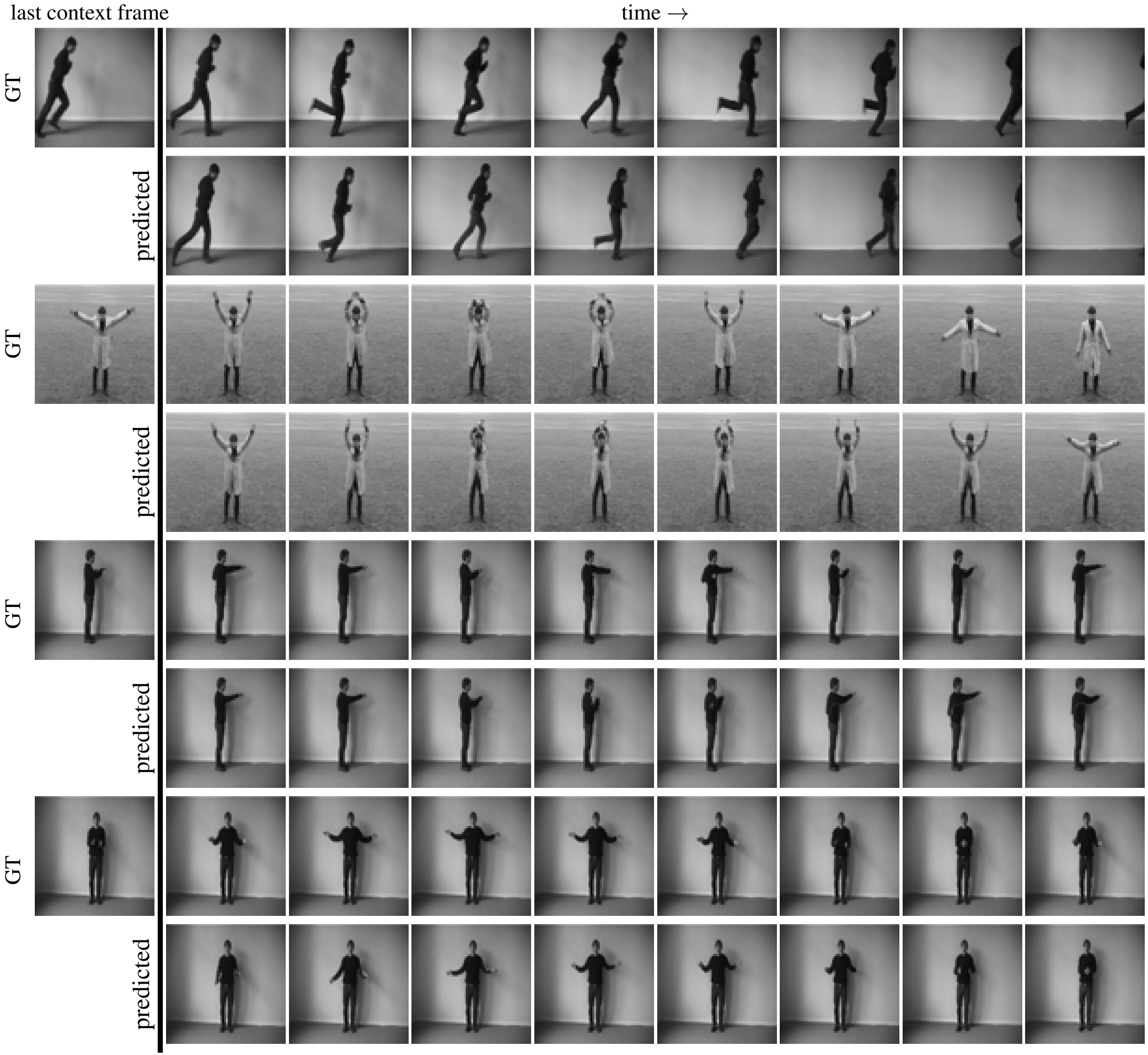

Figure 14. Video prediction on the *KTH* dataset. Odd rows show frames of the original video. Even rows show the video generated by RIVER when fed the context frames of the row above (GT). We observe that RIVER is able to generate sequences with diversity and realism. The images in the first column after the bold vertical line can be played as videos in Acrobat Reader.

[28] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. *ArXiv*, abs/1910.01442, 2020. 1, 2, 3

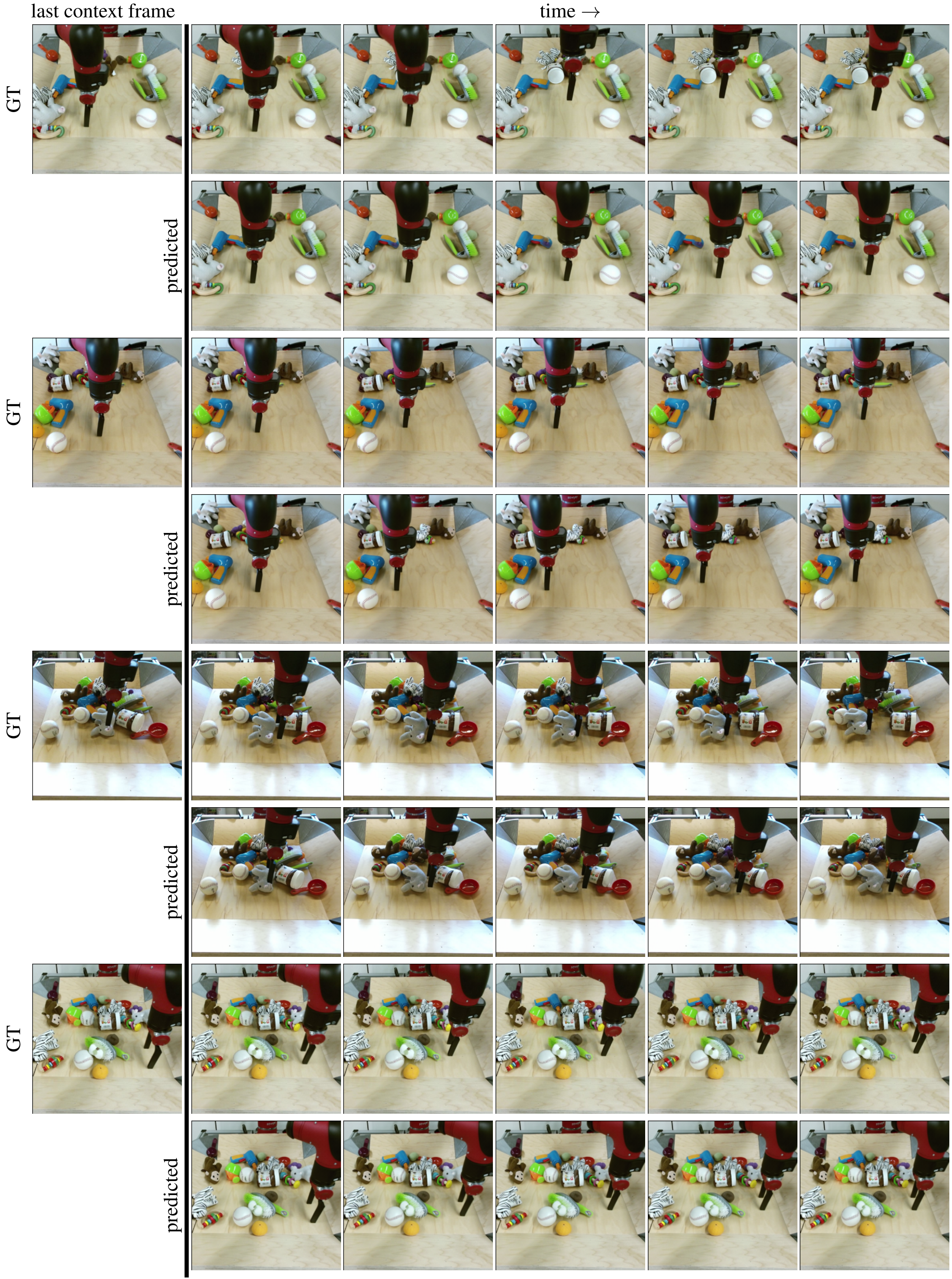

Figure 15. Video prediction on the *BAIR* dataset at 256 × 256 resolution. The model predicts the future frames conditioned on a single initial frame. The frames in the first column after the bold vertical line can be played as videos in Acrobat Reader.

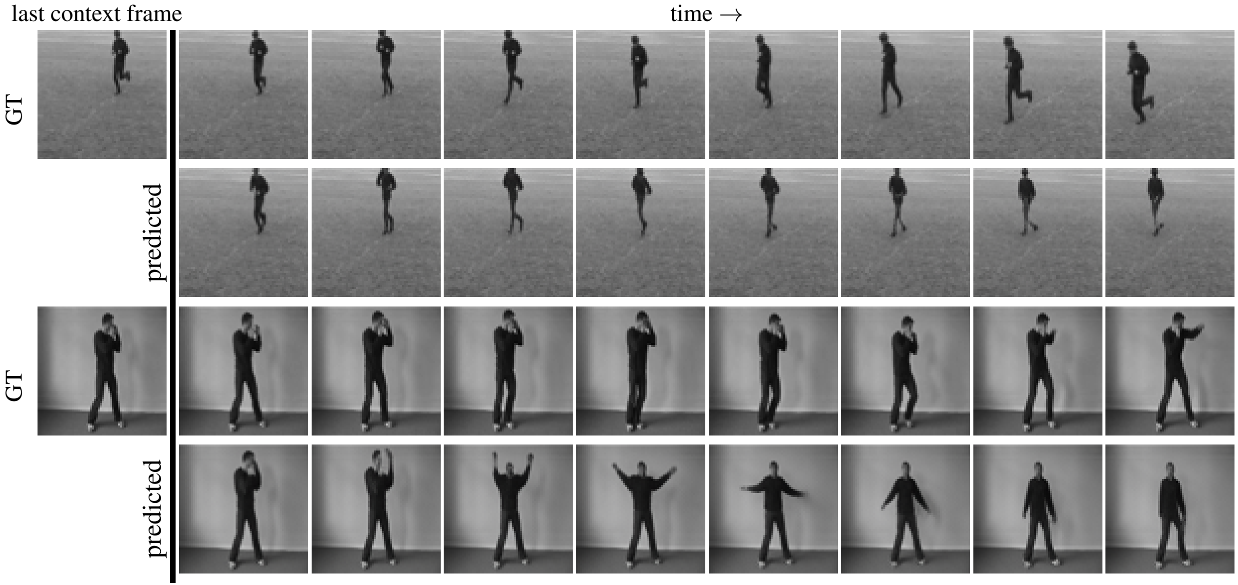

Figure 16. Failure cases on the *KTH* dataset. A common failure mode is when a certain action gets confused with another one, which results in a motion that morphs into a different one. In all examples the model is asked to predict 25 future frames given the first 5. The images in the first column after the bold vertical line can be played as videos in Acrobat Reader.

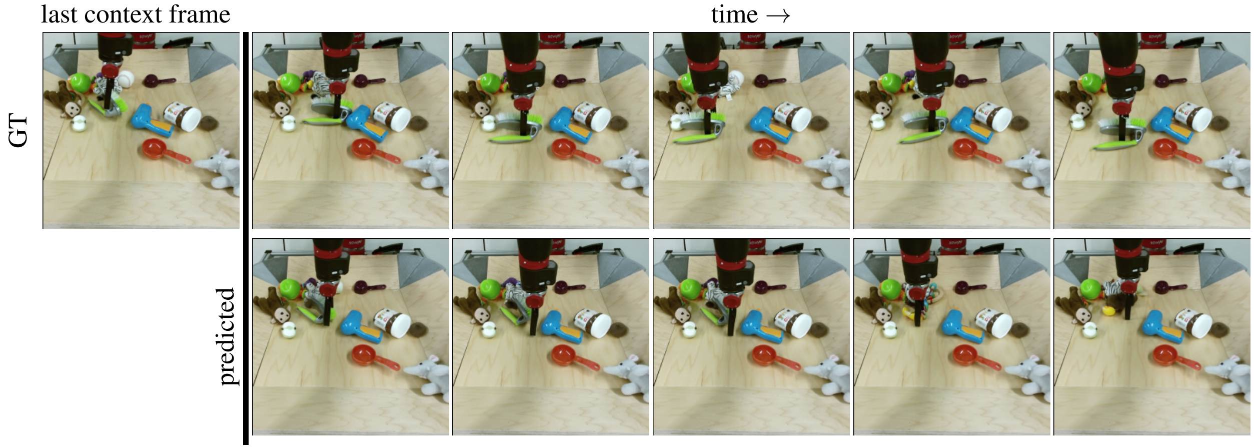

Figure 17. Failure case on the *BAIR* dataset. A common failure mode emerges when generating longer sequences and is when the interaction causes objects to change their class, shape or even to disappear. The images in the first column after the bold vertical line can be played as videos in Acrobat Reader.

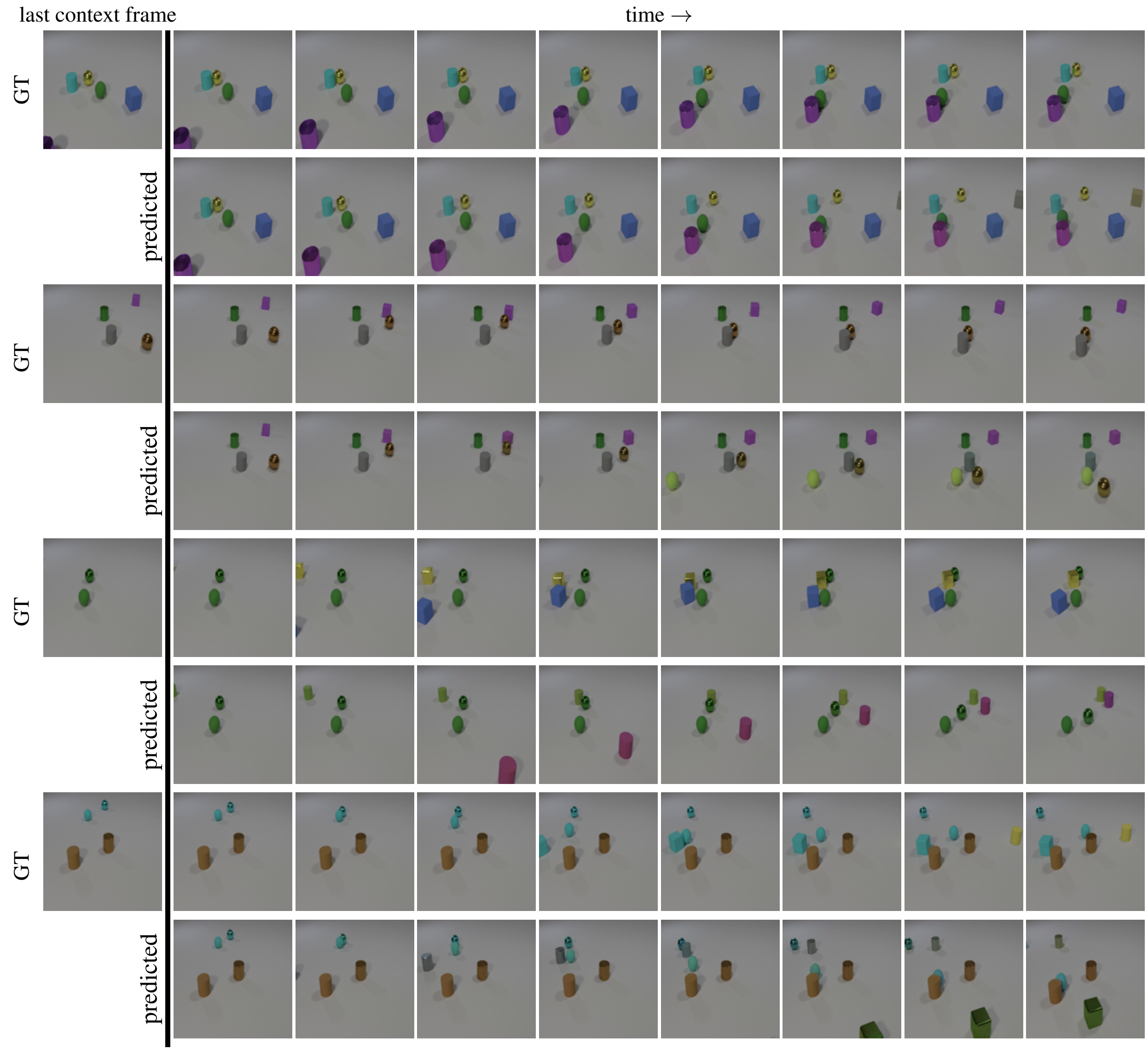

Figure 18. Video prediction on the *CLEVRER* dataset. In order to predict the future frames, the model conditions on the first 2 frames. Only the last context frame is shown. The model succeeds to predict the motion that was observed in the context frames. However, it cannot predict new objects as in the ground truth and introduces random new objects due to the stochasticity of the generation process. The images in the first column after the bold vertical line can be played as videos in Acrobat Reader.

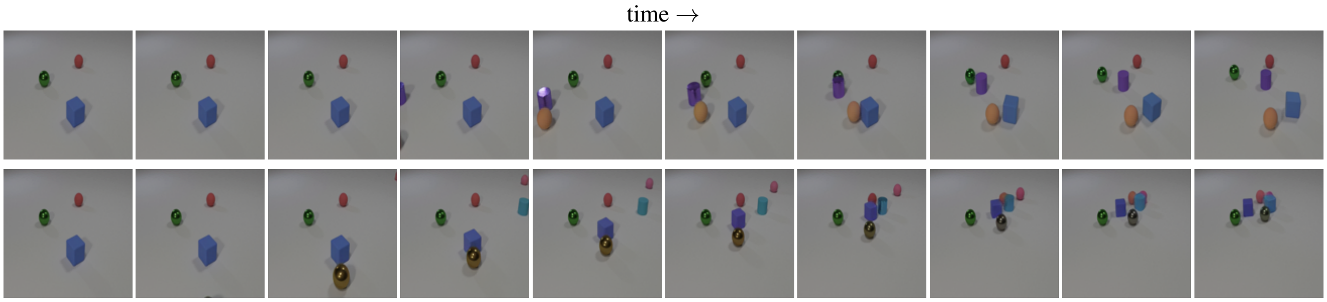

Figure 19. Two sequences generated with RIVER trained on the *CLEVRER* dataset. The model was asked to predict 19 frames given 1. Note the very different fates of the blue cube in these two sequences. The images in the first column can be played as videos in Acrobat Reader.

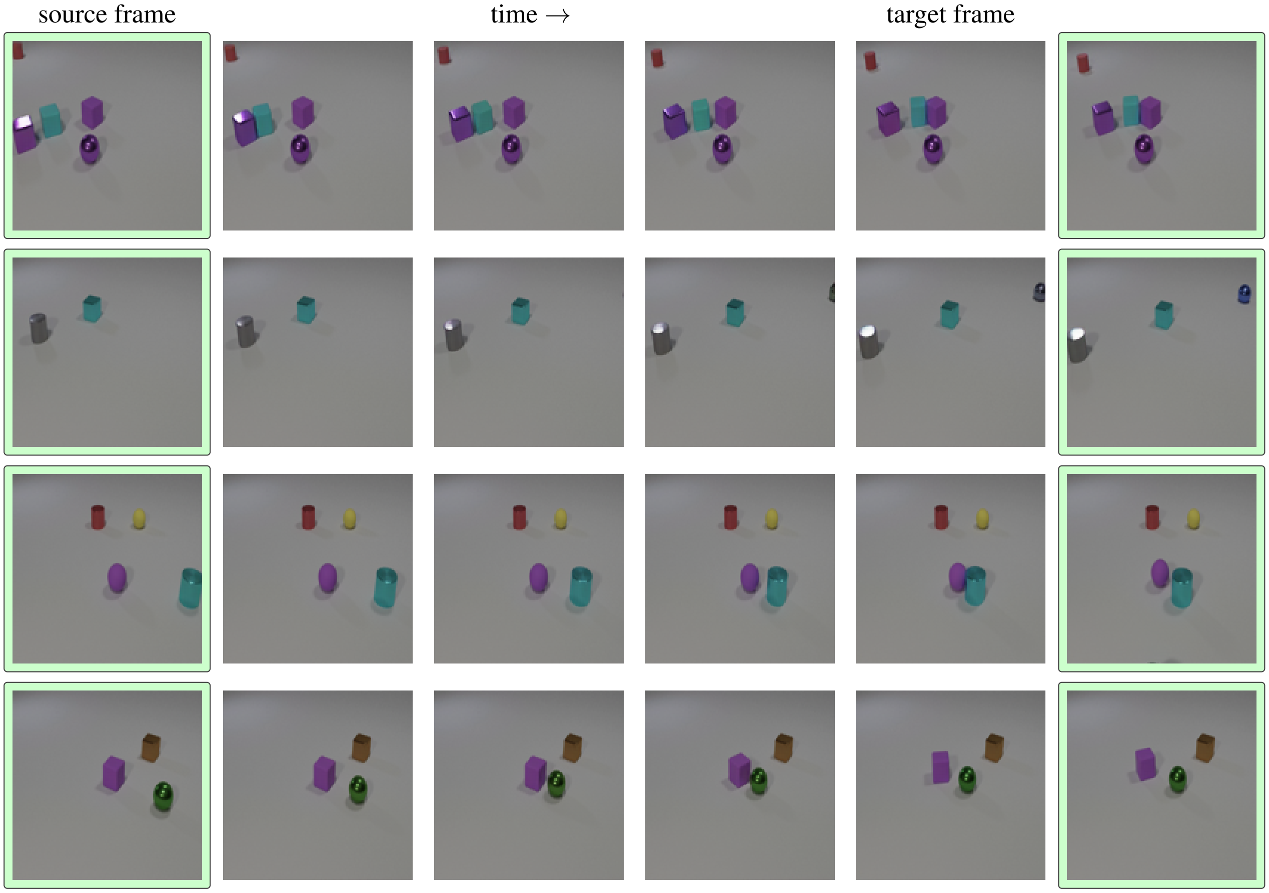

Figure 20. Visual planning with RIVER on the *CLEVRER* dataset. Given the source and the target frames, RIVER generates intermediate frames, so that they form a plausible realistic sequence. The images in the first column can be played as videos in Acrobat Reader.

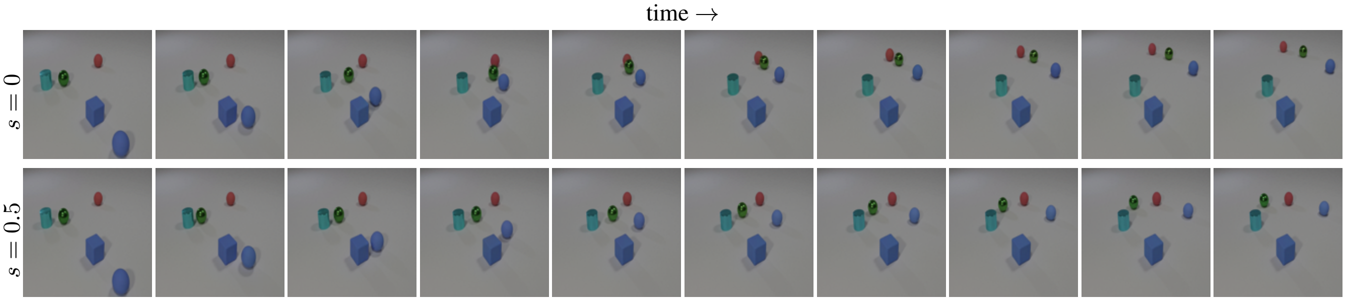

Figure 21. The effect of extreme ($s = 0.5$) warm-start sampling strength. The first frame in each row can be played as a video in Acrobat Reader.