

Understanding Object Dynamics for Interactive Image-to-Video-Synthesis

Supplementary Material

Andreas Blattmann Timo Milbich Michael Dorkenwald Björn Ommer
Interdisciplinary Center for Scientific Computing, HCI
Heidelberg University, Germany

Contents

A Additional Visualizations	1
A.1. PokingPlants	2
A.2. iPER	2
A.3. Tai-Chi-HD	2
A.4. Human3.6M	2
A.5. Qualitative Comparison	3
B Additional Experiments	3
B.1. Training Setting	3
B.2. Results	3
B.3. Generalization to Unseen Types of Plants . .	4
C Derivation of Equation (3)	4
C.1. Correspondence between RNNs and the 2- stage Runge-Kutta-Method	4
C.2. Derivation of RNN-Hierarchy	4
D Ablation Study: Modelled Order of ODE- Derivative	5
E Implementation Details	5
E.1. Network Details	5
E.2. Training Details	5
F. Evaluation Details	6

Preliminaries

For all of our reported datasets in the main paper, we provide additional video material resulting in 24 videos in total. For each video three individual cycles are shown (indicated left bottom) as well as the corresponding frame-per-second (FPS) rate (right bottom). Analogous to the main paper, the direction of the poke is indicated by a red arrow starting at the poke location l and the target location is marked by a red dot, if not stated otherwise. The file structure of the videos is as follows:

```
supplementary_material_244
|
```

```

+--A-Additional_Visualizations
|
+--A1-PokingPlants
|
+--A2-iPER
|
+--A3-Tai-Chi-HD
|
+--A4-Human36M
|
+--A5-Qualitative_Comparison
|
+--B-Additional_Experiments
|
+--B2-Impulse_Model
|
+--B3-Generalization

```

In Section **A**, we discuss our qualitative results and further emphasize the effectiveness of our approach by visually comparing with two competitors from the main paper. In Section **B** we acquire two additional experiments which are based on a slightly different training procedure and a different interpretation of the poke: first, we introduce the training procedure. We then visually evaluate the trained model and subsequently show this model to generalize to previously unseen types of plants which are obtained using web-search. After discussing the results, we derive Equation (3) from the main paper within Section **C**. Moreover, to provide empirical evidence for the effects of selecting the highest order of ODE derivative modeled by our method, we show the benefits of increasing the depth of the RNN hierarchy corresponding to an increase in this order in an additional ablation study in Section **D**. In conclusion we describe implementation and evaluation details in the Sections **E** and **F**.

A. Additional Visualizations

Subsequently, we discuss our obtained results for each dataset individually, before visually comparing our ap-

proach with two competing methods from the quantitative evaluation section. Within the section-specific directory `'...--A-Additional_Visualizations'`, each subsection matches its corresponding folder (e.g. `'A.1.PokingPlants'` corresponds to `'...--A1-PokingPlants'`) containing the discussed video sequences. Each video file compares synthesized sequences of our model for a simulated poke and three pokes initiated by human users with the ground truth sequence starting with the same initial image x_0 , except for those videos containing the comparisons to the related methods.

A.1. PokingPlants

We provide four videos (`poking_plants_[1-4].mp4`) for the PokingPlants dataset showing distinct types of plants of substantially different shapes and appearances. Despite these large variances, our model generates realistic and appealing visualizations which are plausible responses to the poke. Pokes of large magnitudes (indicated by longer arrows) result in larger object motion, affecting not only the object parts in the vicinity of the poke location, but also those parts farther away. This illustrates that our model captures long-term relations between distinct object regions. Modest interactions, in contrast, result in much finer object motions. For instance, the subtle poke initiated in the third column of `poking_plants_2.mp4` only results in fine-grained movements of those parts next to the poke location, whereas the large interaction in the last column causes nearly the entire plant to heavily oscillate. In summary, the examples demonstrate the capabilities of our method to flexibly model dynamics even for object categories with large intra-class variance, ranging from large scale low frequency motion to very subtle movements. Moreover, our model consistently accomplishes to generate sequences showing the object regions around the poke location to approach the target location. We also provide examples for background interactions in the last columns of the examples `poking_plants_1.mp4` and `poking_plants_4.mp4` which indicate our model to separate those pixels comprising the object from background clutter.

A.2. iPER

For the iPER [20] dataset, we also provide four videos (`iper_[1-4].mp4`) containing unseen actors in indoor as well as outdoor scenes. When comparing the synthesized motion, which is generated based on simulated pokes (second columns), with the ground truth sequences depicted in the first columns, one can clearly observe, that our proposed method achieves to infer realistic global motion only from those sparse, localized interactions. The model can even

generate complex human dynamics including distinct combinations of the arms and legs resulting from large pokes, as visualized in the third column of `iper_1.mp4`. Furthermore, it generalizes well to out-of-distribution settings as indicated by the example `iper_4.mp4` showing an actor in the wild¹ and is capable to separate background from foreground also in such cases.

A.3. Tai-Chi-HD

We now visualize examples for the Tai-Chi-HD [24] dataset, which does not contain movements as large as those in iPER, but is nonetheless challenging, since it contains many in-the-wild scenes with highly textured background and substantial amounts of camera movements. Thus, we use it to demonstrate our method to be also able to handle such real-world conditions. As for the other datasets discussed so far, we prepared four videos (`taichi_[1-4].mp4`). Also for this dataset, the model infers plausible global motions from the poke as indicated by the examples obtained from the simulated poke, which are similar to the ground truth sequences for all provided visualizations. Moreover, the model generates plausible responses to user interactions, demonstrating our model to be applicable to outdoor conditions. This is further emphasized by the capability of our model to separate the object area from the background in the presence of camera movements and background clutter.

A.4. Human3.6M

Finally, we show exemplary results of our method on the Human3.6m [14] dataset, which is challenging due to the complexity of performed motions and the low number of different unique persons. Consequently during testing, appearance swaps after a small number of predicted frames are frequently observed [28, 22, 7]. As the majority of actions performed by the persons are either based on walking or sitting we visualize examples for both these cases (`h36m_[1, 2].mp4`). In the first example, we use the poke to control the walking direction of the depicted actor as well as the covered distance. Although changing the appearance, our model generates plausible dynamics which are responses to the individual pokes. In the second example, we interact with the depicted person by manipulating its torso, head and arms. In summary, the visualizations demonstrate that our model is able to synthesize complex human motions and to control their temporal progression based on interactions.

¹Nearly the entire dataset is recorded indoor in front of white background, as indicated by the remaining examples. Therefore, we observe the vast majority of competing models trained on iPER to be struggling with such in-the-wild settings.

A.5. Qualitative Comparison

Comparison with SAVP. To visually demonstrate the benefits of our proposed model, we compare it to SAVP [18], the strongest competitor of all video prediction models considered in this paper. We therefore visualize three sequences showing distinct types of plants from the PokingPlants dataset in `comparison_savp_plants.mp4`. Each row contains a ground truth sequence, followed by our results and those from SAVP [18]. Especially on this dataset, we observe large differences in the quality of the generated dynamics. We attribute this to the wide range of covered motions for the distinct types of plants within the dataset. Note, that the spatial video resolution for our model and the baseline was chosen to be 64×64 for these experiments, as this is common practice in video prediction [18, 22, 3, 7]. The visualized sequences indicate that our competitor struggles especially with subtle oscillations corresponding to fine image details. In contrast, our model also successfully synthesizes these fine motion details, thus demonstrating the efficacy of our proposed hierarchical architecture for capturing fine-grained dynamics.

Comparison with Hao et al. Finally, we visually compare our model with the controllable image synthesis approach of Hao et al. [11] which can also be applied to generate video sequences, as stated in the main paper. The comparisons are conducted on the PokingPlants and iPER datasets and can be viewed in the videos `comparison_hao_plants.mp4` and `comparison_hao_iper.mp4`. Each video consists of three individual video sequences. Again, we firstly visualize the ground truth on the left, before showing our results and those of the competitor. Moreover, since we aim at comparing the models in terms of visual quality, we do not visualize the pokes and trajectories but only the generated sequences. For both datasets, we can clearly observe that our model significantly improves upon the competitor in terms of motion consistency and visual quality. Especially on iPER, we observe the approach of Hao et al. to be unable to individually move distinct body parts. Instead, they holistically move the entire body resulting in large errors when compared to the ground truth sequence.

B. Additional Experiments

Our model formulation also allows for different interpretations of the poke p . In this paragraph, we present the alternative interpretation of the poke p as an initial, localized impulse onto the object instead of a local shift of the pixel at location l . We first explain how to train the model to achieve this before showing samples of the resulting synthesized videos for a model trained on the PokingPlants dataset. Lastly, we show that the trained model generalizes also well to unseen types of plants obtained from web

search.

B.1. Training Setting

Recall that in our main experiments, the poke p was defined as the shift between the poke location and its desired target location in the last frame \hat{x}_T of the predicted video sequence $\hat{X}_{1:T}$ (cf. Section 3 in the main paper). Instead, we now normalize the magnitudes within all estimated flow maps to be in $[0, 1]$, i.e we remove the information about the exact target location and only retain the direction and a magnitude which does not define a pixel shift anymore. By parameterizing the latent interaction $\phi(t)$ as

$$\phi(t) = \begin{cases} \phi, & t = 0 \\ 0, & t > 0 \end{cases}, \quad (1)$$

with $\phi = E_\phi(p, l)$, $\phi(t)$ can be seen as an initial force to the initial object state. When using this parameterization of $\phi(t)$, we can in fact train our model to produce sequences which do not longer show plausible object reactions to the shift of a pixel. Instead, they now visualize an object response to an initial impulse acting at location l . However, as the magnitude of such an initial impulse should influence the amount motion in the reaction of the depicted object, we have to ensure the model to observe sequences with small amounts of motion for pokes with small magnitude and videos showing much motion for these pokes with large magnitude during training. To this end, we estimate the average of motion within each training sequence $X_{0:T}$ as

$$M(X_{0:T}) = \frac{1}{T} \sum_{i=1}^T \text{mag}(D(x_i, x_{i-1})), \quad (2)$$

with $D(x_i, x_{i-1}) \in \mathbb{R}^{H \times W \times 2}$ the estimated flow map between two consecutive images x_i and x_{i-1} and $\text{mag}(D(x_i, x_{i-1}))$ the spatial average of flow magnitudes. Using these estimates, we can sample smaller pokes for sequence with smaller amounts of motion and larger ones for those with more motion, resulting in a learned model which indeed synthesizes videos showing object responses to an initial impulse.

B.2. Results

We show results of our model trained on the PokingPlants dataset using the procedure explained above. We train the model to reconstruct sequences of length 10 and predict 25 frames during inference. We here provide 3 unique video examples, which are denoted as `impulse_model_[1-3].mp4` and are located in the directory `--B-Additional_Experiments` within `--B1-Impulse_Model` subfolder. For illustrating the poke within the videos, we now plot a single arrow

starting at the poke location. The length of this arrow is proportional to the magnitude of the poke and, thus, defines the amount of motion to be expected within a generated sequence. Within the leftmost examples in the videos `impulse_model_1.mp4` and `impulse_model_2.mp4` only a very subtle poke is induced. As our model has learned to couple the poke magnitude to the global amount of motion visualized in the predicted sequence, it generates similarly subtle movements in these cases. Thus, these plants immediately return to their initial states after performing small oscillations. Pokes with larger magnitudes, however, cause larger motions to elapse. Hence, in these cases, the poked plants do not approach their rest state again within the 25 frames of the generated video sequence. In summary, the provided videos demonstrate that also for this training setting, the model understands the dynamics of a given object class and, thus, can be used to synthesize video sequences illustrating the response to an initial impulse onto the object body.

B.3. Generalization to Unseen Types of Plants

Finally, we illustrate the generalization capabilities of our approach by applying it to four unseen types of plants, which were obtained by using image search in the internet. The resulting generations can be viewed in `generalization_[1-4].mp4` within the subfolder `--B2-Generalization`. The applied model was trained on the PokingPlants dataset as well as the vegetation samples from the Dynamic Texture Database [9]. For each example, we provide video sequences based on four different pokes and also visualize the nearest neighbor from the train set (last column). The nearest-neighbor is computed in the feature space of our object encoder \mathcal{E}_ϕ . For the trees shown in `generalization_1.mp4` and `generalization_3.mp4` as well as for the pot plants in the remaining two examples, we observe the model to predict plausible dynamics. Remarkably, the model also captures the relations between distinct parts for the two pot plants, as indicated by the columns two and three of `generalization_2.mp4`. In the second column, where a subtle interaction is applied, only the directly affected lead and some related parts are slightly moving. However, when initiating a poke with larger magnitude, the entire plant is heavily shaking. Finally our model also achieves to separate foreground from background in example `generalization_3.mp4`, despite its substantially textured background area.

C. Derivation of Equation (3)

In the following section, we will derive our RNN-hierarchy introduced in Equation (3) in the main paper. Prior to the derivation, we will shortly introduce the correspondence between common RNN architectures [13, 4]

and the 2-stage Runge-Kutta-Method [17], an approximation method for solving ODEs. For more details and the exact proof, see [23].

C.1. Correspondence between RNNs and the 2-stage Runge-Kutta-Method

Runge-Kutta-Methods [17] are a family of multi-stage, discrete approximation methods for solving ODEs of the form

$$\dot{\sigma} = f(\sigma(t)). \quad (3)$$

Niu et al. [23] showed that common RNN architectures as LSTMs [13] and GRUs [4] can be seen as realizations of the 2-stage Runge-Kutta-Method [17] (RK2). Given an initial value $\sigma(0) = \sigma_0$, RK2 approximates $\sigma(t)$ at discrete time-steps $i \in [1, T]$. Let h be the step-size between two consecutive states σ_i and σ_{i+1} , which the ODE is approximated at, then the RK2 method reads

$$\begin{aligned} \sigma_{i+1} &= \sigma_i + h \cdot K_1(\sigma_i) + h \cdot K_2(\sigma_i + h) \\ &= \sigma_i + K_1^*(\sigma_i) + K_2^*(\sigma_i), \end{aligned} \quad (4)$$

with $K_1^*(\sigma_i) := h \cdot K_1(\sigma_i)$, $K_2^*(\sigma_i) := h \cdot K_2(\sigma_i + h)$.² Based on an initial hidden state σ_0 , the update rule of the GRU-cell [4] is

$$\begin{aligned} \sigma_{i+1} &= \sigma_i + z(\sigma_i) \odot \sigma_i + z(\sigma_i) \odot h(\sigma_i) \\ &= \sigma_i + G_1(\sigma_i) + G_2(\sigma_i), \end{aligned} \quad (5)$$

with $z(\sigma_i)$ the update gate vector and $h(\sigma_i)$ the candidate activation [4, 5] and \odot the hadamard product³. By comparing the Equations (4) and (5), the correspondence becomes evident. Note that this is only for clarification, for the exact proof, see Niu et al [23].

C.2. Derivation of RNN-Hierarchy

We will now use RK2 to solve an N -th ODE $f = f(\sigma^{(1)}, \sigma^{(1)}, \dots, \sigma^{(N)})$, with $\sigma^{(n)}$ the n -th time derivative of $\sigma(t)$. By introducing the following hierarchy of variables [10]

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma^0 \\ \sigma^1 \\ \sigma^2 \\ \vdots \\ \sigma^N \end{pmatrix} := \begin{pmatrix} \sigma \\ \sigma^{(1)} \\ \sigma^{(2)} \\ \vdots \\ f(\sigma^0, \dots, \sigma^{N-1}) \end{pmatrix}, \quad (6)$$

we can map this N -th order ODE onto a system of trivial first order ODEs of the form $\sigma^n = \dot{\sigma}^{n-1} = f_n(\sigma^{n-1})$. The n -th element of $\boldsymbol{\sigma}$ is proportional to the n -th time derivative of the original variable σ . Each of these ODEs has the form

²For the exact schemes to compute K_1 and K_2 , see [17].

³Note that G_1 and G_2 are also functions of the input state of the GRU cell [4, 5]. We will later make use of this.

Dataset	PP			iPER [20]		
Method	LPIPS ↓	PSNR ↑	FVD ↓	LPIPS ↓	PSNR ↑	FVD ↓
$N = 1$	0.08	21.13	170.74	0.07	22.62	233.22
$N = 2$	0.08	21.15	138.72	0.06	23.12	178.52
Ours ($N = 3$)	0.06	21.81	89.67	0.05	23.11	144.92

Table 1. Ablation study on modelled order N of derivative of ODE approximation corresponding to the depth of the RNN hierarchy. Since the approximation gets more accurate by including higher orders of derivatives, i.e. increasing the depth of the hierarchy, the model improves in prediction accuracy (LPIPS, PSNR) and dynamics consistency (FVD) for increasing N .

of Eq. (3) and - given initial values σ_0^n - can be approximated in discrete time by using RK2. Thus, we can apply GRU cells \mathcal{F}_n to solve each ODE individually. However, as the argument of each function f_n is the predecessor σ^{n-1} of the variable σ^n , which f_n shall be solved for, we use the approximation σ_{i+1}^{n-1} as the input state of \mathcal{F}_n , resulting in

$$\sigma_{i+1}^n = \mathcal{F}_n(\sigma_i^n, \sigma_{i+1}^{n-1}) \quad (7)$$

for $n \in [1, N]$, which is Eq. (3) in the main paper.

D. Ablation Study: Modelled Order of ODE-Derivative

To further provide empirical evidence on the effects of selecting of the highest order of derivative which is modelled by our proposed method corresponding to the depth N of the RNN-hierarchy, we analyze the effects of varying N in this section. Thus, $N = 1$ corresponds to \mathcal{F} consisting of a single RNN cell in the latent bottleneck between \mathcal{E}_σ and \mathcal{G} and using no skip connections between these sub-networks while learning Ω . Starting from this baseline, we increase the hierarchy until $N = 3$ is reached, which constitutes our proposed model. As each additional RNN cell operates on a higher spatial size than its predecessor, we add a skip connection on the respective spatial level between \mathcal{E}_σ and \mathcal{G} , on which this additional RNN operates while learning \mathcal{F} . The resulting three models, which are compared, are all training on videos with a spatial size of 64×64 on the object categories of humans (on the iPER [20] dataset) and plants (on the PP dataset) investigated in the main paper.

Tab. 1 shows that our proposed model, whose RNN hierarchy is capable of modelling a higher number of derivative than the baselines, obtains lower FVD scores, indicating that complex object dynamics can be more accurately modelled by using a deeper RNN hierarchy. This further arises from comparing the performance of the two baselines, where $N = 1$ obtains worse results than $N = 2$. Additionally, the increasing N yields an enhanced image quality as highlighted by lowered LPIPS and PSNR scores for raised N , which is a further benefit of our proposed hierarchy of RNNs.

E. Implementation Details

Here we give a detailed explanation of the network architecture as well as the training procedure for our model and the baselines which are used for comparison.

E.1. Network Details

Encoders Within the encoders \mathcal{E}_σ and \mathcal{E}_ϕ we subsequently apply ELU-activated [6] 2D-convolutional-layers with kernel size 3 and stride 2, until a spatial resolution of 8×8 is reached, resulting in $N = 3$ and $N = 4$ layers for video sequences of spatial resolutions 64×64 and 128×128 . After each conv-layer, we employ instance normalization [25]. The last conv-layer is followed by a final ResNet-Block [12]. The initial number of channels is 32 after the first conv-layers and is increased by a factor of two after each subsequent layer.

Decoder The decoder consists N subsequently applied ResNet-Blocks [12] where $N = 3$ for a spatial video resolution of 64×64 and $N = 4$ for videos of spatial size 128×128 . Similar to the encoders, the conv-layers within each ResNet-Block are ELU-activated and followed by instance-normalization layers. Upsampling is achieved by using transposed convolutions instead of common convolutions as first layers of those ResNet-Blocks.

Hierarchical Image-to-Sequence-Model Our hierarchical image-to-sequence model consists of a hierarchy of N Conv-GRU cells, where $N = 3$ for spatial video resolutions of 64×64 and $N = 4$ for videos of size 128×128 . The upsampling layers \mathcal{U}_n are implemented as transposed convolutions. The hidden dimension of each individual GRU-cell is equal to the number of channels of the encoder \mathcal{E}_σ at the same spatial level.

Discriminators The static discriminator \mathcal{D}_S is implemented as a patch discriminator [15]. For the temporal discriminator \mathcal{D}_T we use a 3D Resnet-18 [12]. Within both discriminators instance-normalization [25] is used after each layer.

E.2. Training Details

Our model

Foreground-Background-Separation We assume parts of the background of the videos within the train set to be static and the foreground to obtain a sufficient amount of motion, indicated by a specific magnitude of optical flow. Thus, during training, we only consider locations with an optical flow magnitude larger than the mean of magnitudes of the flow map D for sampling the interaction location l . However, as we also want our model to separate the pixels on the object surface from those in the background, we sample a tenth of all poke locations in each epoch out of background pixels and construct artificial pokes by sampling the poke magnitudes and angles at these locations from the locations within

the foreground. If our model gets such an artificial poke as input, it is trained to reconstruct a still sequence obtained by repeating the source image x_0 T times. Thus, the model indeed learns to separate the pixels comprising the object from those in the background, as indicated by our video examples, where we show examples for such artificial pokes for each dataset.

Discriminators. The discriminators are optimized using the hinge formulation [19, 1]. For stabilizing the GAN training gradient penalty [21, 8] is used for the temporal discriminator. Additionally, we add a feature matching loss [27] to the overall objective \mathcal{L} for \mathcal{D}_T , which we weigh with a factor of 2. For training the spatial discriminator, we sample 16 individual images from the predicted and ground truth sequences.

Spatial Video of Predicted Videos. To conduct the comparison experiments with recent state-of-the-art video prediction models as well as for ablating our method, we trained models on videos of spatial resolution 64×64 . Our models which are compared with the controllable video synthesis method of Hao et al. [11] are trained to predict sequences of spatial size 128×128 . All provided videos are outcomes of those models, except for the ones which are visually compared with SAVP [18] on the PokingPlants-Dataset. In this case, we predicted videos of spatial size 64×64 to obtain a fair comparison.

Video Prediction Models. For comparison, we implemented the video prediction baselines [18, 3, 7] based on the official provided code from Github⁴⁵⁶. As no pre-trained models are available for the utilized datasets, we trained models from scratch for all competitors except for SRVP [7], which provide a pretrained model for the Human3.6m [14] dataset. All models are trained to predict sequences of length 10 and spatial size 64×64 based on two context frames. For models trained from scratch, we used the hyperparameters proposed in the respective publications.

Controlled Video Synthesis Model. The method of Hao et al. [11] is implemented based on the official code⁷ and the provided hyperparameters for all used datasets. We used their proposed procedure to construct the motion trajectories based on the same optical flow which was we used to train our own model. We trained their model to predict images of spatial size 128×128 .

F. Evaluation Details

FVD-Scores. To compute the FVD-score [26] for a given model, we generated 1000 video sequences and sampled 1000 random videos of the same length from the ground

truth data. Both the real and the generated examples are the input to an I3D [2] model pretrained on the Kinetics [16] dataset. Subsequently their distributions in the I3D feature space are compared resulting in the reported FVD-scores.

Accuracy Metrics. All reported accuracy metrics are based on 8000 predicted video sequences and the corresponding ground truth videos. As these metrics are calculated based on individual image frames, we compare each frame of a generated sequence with its corresponding frame ground in the ground truth sequence, resulting in T scores for a predicted video of length T , which are subsequently averaged to obtain a scalar value per sequence.

⁴<https://github.com/edouardelasalles/srvp>

⁵https://github.com/facebookresearch/improved_vrn

⁶https://github.com/alexlee-gk/video_prediction

⁷<https://github.com/zekunhao1995/ControllableVideoGen>

References

- [1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2018. 6
- [2] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6
- [3] L. Castrejon, N. Ballas, and A. Courville. Improved conditional vrns for video prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3, 6
- [4] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014. 4
- [5] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014. 4
- [6] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. 5
- [7] Jean-Yves Franceschi, Edouard Delasalles, Mickael Chen, Sylvain Lamprier, and P. Gallinari. Stochastic latent residual video prediction. *ArXiv*, abs/2002.09219, 2020. 2, 3, 6
- [8] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017. 6
- [9] Isma Hadji and Richard P. Wildes. A new large scale dynamic texture dataset with application to convnet understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 4
- [10] J.K. Hale. *Ordinary Differential Equations*. Dover Books on Mathematics Series. Dover Publications, 2009. 4
- [11] Zekun Hao, Xun Huang, and Serge Belongie. Controllable video generation with sparse trajectories. In *CVPR*, 2018. 3, 6
- [12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5
- [13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997. 4
- [14] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. 2, 6
- [15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. 5
- [16] W. Kay, J. Carreira, K. Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, F. Viola, T. Green, T. Back, A. Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *ArXiv*, abs/1705.06950, 2017. 6
- [17] W. Kutta. Beitrag zur näherungsweise Integration totaler Differentialgleichungen. *Zeit. Math. Phys.*, 46:435–53, 1901. 4
- [18] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *CoRR*, 2018. 3, 6
- [19] Jae Hyun Lim and Jong Chul Ye. Geometric gan, 2017. 6
- [20] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2, 5
- [21] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine learning (ICML)*, 2018. 6
- [22] Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin P Murphy, and Honglak Lee. Unsupervised learning of object structure and dynamics from videos. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 2, 3
- [23] Murphy Yuezhen Niu, Lior Horesh, and Isaac Chuang. Recurrent neural networks in the eye of differential equations, 2019. 4
- [24] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Conference on Neural Information Processing Systems (NeurIPS)*, December 2019. 2
- [25] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *ArXiv*, abs/1607.08022, 2016. 5
- [26] 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*, 2018. 6
- [27] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *CoRR*, 2017. 6
- [28] Nevan Wichers, Ruben Villegas, Dumitru Erhan, and Honglak Lee. Hierarchical long-term video prediction without supervision. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 2