iPOKE: Poking a Still Image for Controlled Stochastic Video Synthesis

Andreas Blattmann\textsuperscript{1,2} Timo Milbich\textsuperscript{1,2} Michael Dorkenwald\textsuperscript{2} Björn Ommer\textsuperscript{1,2}
\textsuperscript{1}Ludwig Maximilian University of Munich \textsuperscript{2}IWR, Heidelberg University, Germany

Abstract

How would a static scene react to a local poke? What are the effects on other parts of an object if you could locally push it? There will be distinctive movement, despite evident variations caused by the stochastic nature of our world. These outcomes are governed by the characteristic kinematics of objects that dictate their overall motion caused by a local interaction. Conversely, the movement of an object provides crucial information about its underlying distinctive kinematics and the interdependencies between its parts. This two-way relation motivates learning a bijective mapping between object kinematics and plausible future image sequences. Therefore, we propose iPOKE – invertible Prediction of Object Kinematics – that, conditioned on an initial frame and a local poke, allows to sample object kinematics and establishes a one-to-one correspondence to the corresponding plausible videos, thereby providing a controlled stochastic video synthesis. In contrast to previous works, we do not generate arbitrary realistic videos, but provide efficient control of movements, while still capturing the stochastic nature of our environment and the diversity of plausible outcomes it entails. Moreover, our approach can transfer kinematics onto novel object instances and is not confined to particular object classes. Our project page is available at https://bit.ly/3dJN4Lf.

1. Introduction

Imagine a 3-year-old standing next to a stacked pyramid of glasses in a shop. Can you sense the urge to pull one glass out—just to observe what happens. We have an inborn curiosity to understand how the world around us reacts to our actions, so we can eventually imagine and predict their outcome beforehand. This ability to predict is the prerequisite for targeted, goal-oriented interaction with our world rather than random manipulation of our environment. Once we are older, we have also learned to generalize and predict the dynamics of previously unseen objects when they are pulled or poked; and the less audacious have understood that it is often more effective to have others do daring experiments like the one above (and pay the bill) while they are learning by merely watching the outcome. While such experiments are not just fun to watch, they also help to imagine the many possible outcomes caused by the stochastic nature of the many factors beyond our control.

Given a single static image, how can an artificial vision system imagine, i.e. synthesize, the many possible outcomes when locally manipulating the scene? It needs to learn how a local poke affects different parts of an object and the resulting kinematics [49]. Conditioned on only the start frame and the displacement of a single pixel, we want to synthesize multiple videos, each showing the different plausible future dynamics. To render this generative, stochastic approach widely applicable, training should only require videos of objects in motion, but no ground truth information regarding the forces acting on an object such as a local poke. The representation of the kinematics should then generalize to similar objects not seen during training in contrast to instance specific models [15]. Moreover, the method should work for arbitrary objects, rather than being tuned to just a single class [1, 21]. Therefore, no prior motion model is available, but all kinematics have to be learned from the unannotated
video data. Previous work on video synthesis has mainly explored two opposing research directions: (i) uncontrolled future frame prediction [24, 10, 53, 59] synthesizing videos based on a start frame, but with no control of scene dynamics, and (ii) densely controlled video synthesis [50, 73, 77, 74] demanding tedious, per-pixel guidance how the video will evolve such as by requiring the object motion to be provided per pixel [50, 73, 77] or a future target frame [74]. Our sparsely controlled video synthesis based on few local user interactions constitutes the rarely investigated midground in-between, allowing for specific but still efficient control of kinematics.

In this paper, we present a model for exercising local control over the kinematics of objects observed in an image. Indicating movements of individual object parts with a simple mouse drag provides sufficient input for our model to synthesize plausible, holistic object motion. To capture the ambiguity in the global object articulation, we learn a dedicated latent kinematic representation. The synthesis problem is then formulated as an invertible mapping between object kinematics and video sequences conditioned on the observed object manipulation. Due to its stochastic nature, our latent representation allows to sample and transfer diverse kinematic realizations fitting to the sparse local user input to then infer and synthesize plausible video sequences as shown in Fig. 1.

To evaluate our model on controlled stochastic video synthesis, we conduct quantitative and qualitative experiments on four different datasets exhibiting complex and highly articulated objects, such as humans and plants. Comparisons with the state-of-the-art in stochastic and controlled video prediction demonstrate the capability of our model to predict and synthesize plausible, diverse object articulations inferred from local user control.

2. Related Work

Video Synthesis. Video synthesis denotes the general task of generating novel video sequences. While some works solely focus on transferring a predefined holistic motion between objects [73] or interpolating motion between a starting and end frame [54, 79, 43, 4, 55], the most commonly addressed problem is video prediction. Given an initially observed video sequence, the goal is to infer a likely continuation into the future. To this end, proposed methods either generate a single, deterministic video sequence [76, 70, 69, 77, 6] or model the distribution over likely future sequences [23, 16, 40, 59, 53, 10, 19]. Moreover, the employed model architectures exhibit large divergence with latent RNN-based methods being the dominant modelling choice [53, 24]. However, also more complex models based on transformers [75], pixel-level autoregression [51, 40, 45, 16, 70, 10], factorization of dynamics and content [53, 24] and image warping using optical flow [71, 44, 25] have been proposed. Despite these methods showing promising results, none of them is able to exercise control over the video generation process.

Controllable Video Synthesis. Exercising user control over the video synthesis process requires a detailed understanding of the object kinematics and interplay of the object parts. To circumvent the difficult task of learning object kinematics directly from data, Davis et al. [15] resort to fixed, linear mathematical models. Thus, they can only consider constraint oscillating motion around an object’s rest state. In contrast, our model learns natural, unconstrained object kinematics from video, thus is also applicable to highly complex articulation such as those of humans. Other works rely on a low-dimensional, parametric representation e.g. keypoints to transfer motion between videos [1, 5] or to synthesize videos based on action labels [81]. Given such assumptions, these works cannot be universally applied to arbitrary object categories and allow only for coarse control compared to our fine-grained, local object manipulations. By iteratively warping single images with local sets of estimated optical flow vectors, [27] takes a first step towards sparsely controlled video generation for arbitrary object categories. However, due to the method’s warping based nature, it is still not able to generate temporally coherent motion and requires optical flow guidance for each individual predicted image frame. To overcome such limitations, [6] introduces a hierarchical dynamics model, which can predict complex object dynamics controlled by a single optical flow vector in a given image, but does still not consider the natural motion ambiguity of the remaining, uncontrolled object parts. In contrast, our model learns a dedicated, stochastic kinematics representation modeling this incertitude of the object remainder and, thus, is capable of synthesizing locally controlled but also diverse object motion.

Invertible Neural Networks. Invertible neural networks (INNs) are learnable bijective functions often used to transform between two probability distributions, thus being a natural choice for addressing inverse problems [3], introspecting and explaining neural network representations [22, 32] and domain transfer [62, 63]. Typically, INNs are realized as generative normalizing flows [60, 47, 37] which have recently also found application in image [37, 58] and video synthesis [39, 19]. In this work, we use normalizing flows to learn the missing residual information, i.e. the latent object kinematics, not being determined by the by the sparse local control over part of the object motion.

3. Approach

Controlled video synthesis seeks to generate a plausible future video sequence \( X \in \mathbb{R}^{T \times H \times W \times 3} \) given an initial frame \( x_0 \) and a user-defined control \( c \) that locally specifies part of the video dynamics,

\[
(x_0, c) \mapsto X = [x_1, \ldots, x_T].
\]
Figure 2. Overview of our proposed framework iPOKE for controlled video synthesis: We apply a conditional bijective transformation $\tau_\theta$ to learn a residual kinematics representation $r$ capturing all video information not present in the user control $c$ defining intended local object motion in an image frame $x_0$ (orange path). To retain feasible computational complexity, we pre-train a video autoencoding framework $(E, GRU, D)$ (blue path) yielding a dedicated video representation $z$ as training input for $\tau_\theta$. Controlled video synthesis is achieved by sampling a residual $r$, thus defining plausible motion for the remaining object parts not directly affected by $c$, and generating video sequences $\hat{X}$ from the resulting $z = \tau_\theta(r|x_0, c)$ using GRU and $D$ (black path).

3.1. Invertible Controlled Video Synthesis

Seeking to find the mapping (3) we naturally arrive at a problem of stochastic video prediction. So far, the dominant approach to such problems are conditional variational autoencoder (cVAE) based models [38, 61, 65]. cVAE employs strong regularization to remove the given conditioning from the remaining data variations, thus facing a trade-off between synthesis quality and capturing all these variations [11, 83], in our case the diverse object kinematics $r$.

To avoid this, we use a conditional bijective, i.e. one-to-one, mapping $\tau$ between each residual $r$ and the corresponding video $X = \tau(r|x_0, c)$ so that all plausible $X$ for a given conditioning can be synthesized. Moreover, the inverse mapping $\tau^{-1}$ allows to recover the residual kinematics for any $X$,

$$r = \tau^{-1}(X|x_0, c),$$

where then can be considered as a random variable $r \sim p(r|x_0, c)$, since $\tau^{-1}$ is unique and $X$ is a random variable defined in (2). To solve the conditional video synthesis task, we now show how to learn $\tau$ such that (i) indeed contains all video information not present in $(x_0, c)$ and (ii) follows a distribution which can be easily sampled from.

Learning the invertible mapping $\tau$. We equip $\tau$ with parameters $\theta$ which, by employing Eq. (5), can be learned from training videos $X$. By the change-of-variables theorem for
we have to ensure to learn (6), we can express KL into Eq. (7) we arrive at the simple objective function

\[ \min_{\theta} \mathcal{L}(\tau_0, X, x_0, c) = \mathbb{E}_{X, x_0, c} \left[ \| \tau_0^{-1}(X|x_0, c) \|^2_2 \right] - \log |\det J_{\tau_0^{-1}}(X|x_0, c)| \]

By selecting \( q(r) = \mathcal{N}(r|0, I) \) [38, 80] and inserting this into Eq. (7) we arrive at the simple objective function

\[ \min_{\theta} \mathcal{L}(\tau_0, X, x_0, c) = \mathbb{E}_{X, x_0, c} \left[ \| \tau_0^{-1}(X|x_0, c) \|^2_2 \right] - \log |\det J_{\tau_0^{-1}}(X|x_0, c)| \]
3.3. Automatic Simulation of User Control

Training our model for controlled video synthesis relies on user controls $c$ and corresponding video sequences $X$ depicting natural object responses to be available. Providing sufficient amounts of such training data for every targeted object category is tedious and costly. Instead, we employ an efficient self-supervised strategy to artificially generate such interactions directly from the observed motion of a collection of cheaply available training videos $X$. To this end, we extract dense optical flow maps $F \in \mathbb{R}^{H\times W\times 2}$ between the start and end frames, $x_0$ and $x_T$, of $X$ whose individual shift vectors can be interpreted as sparse pixel displacements $c = \{(F_{l_n,1}, l_n, 2, 1), F_{l_n,1}, l_n, 2), 2\}^{N_c}_{n=1}$. During training, we randomly sample such simulated pokes at positions $l_n$, which exhibit sufficiently large motion that reliably corresponds to the foreground object. Contrasting [6], which use a similar strategy, but restrict the user control to be defined by only a single poke, we allow a user to control the degree of freedom of the object articulation by training our model on up to 5 simultaneous interactions $c$, i.e. on a variable number $N_c \in [1, 5]$ of local pokes. Note that for inferring user controls after training we do not require optical flow estimates, but use simple mouse drags instead.

4. Experiments

Subsequently, we evaluate our model for controlled stochastic video synthesis on four video datasets showing diverse and articulated object categories of humans and plants. Implementation details and video material can be found in the Appendices F and G, and on our project page.

4.1. Datasets

We evaluate our approach to understand and synthesize object dynamics on the following four datasets:

**Poking-Plants (PP)** [6] consists of 27 videos of 13 different types of pot plants. To learn a single kinematics model for all plants is notably challenging given the large variance in shape and texture of the plants. Overall, PP contains of 43k frames, from which a fifth is used as a test set and the...
180k frames and a test set of 49k frames. We extract the corresponding residual kinematics, as well as three synthesized examples generated from a fixed dataset. Transfer of Kinematics. Besides sampling plausible object kinematics, we can also apply our model to transfer kinematics inferred from a source sequence \( X_s = [x_{s,0}, \ldots, x_{s,T}] \) to a novel object instance. To this end, we extract the corresponding residual kinematics \( r_s = \tau^{-1}_0(X_s|x_{s,0}, c) \) for a user control \( c \) simulated based on \( X_s \) and use Eq. (4) to animate a target image \( x_t \) showing another object instance than \( x_{s,0} \) with similar articulation. The resulting successfully transferred motion sequence \( \hat{X}_t = \tau_0(r_s|x_t, c) \) is shown in Fig. 6 (second row) and compared to \( X_s \) (top row). It can be seen, that the motion contained in \( X_s \) is transferred to \( \hat{X}_t \) but not the object appearance shown in \( x_{0,s} \), indicating that our model indeed learns a residual representation \( r \) solely containing kinematics. Additionally, we visualize the synthesized video sequence based on a random sample \( r \sim q(r) \) of residual kinematics for the same conditioning \( (x_t, c) \) (bottom row), showing substantially different object motion except for the controlled body part and thereby providing evidence that \( r \) is also independent of the user-control \( c \). More results of kinematics transfer can be found in Appendix A.2.

### Understanding Object Kinematics

To demonstrate how well our model captures holistic object kinematics, we analyze its understanding of the interplay of integral object parts. Therefore, we measure the pixel-wise correlations when applying 1000 randomly sampled user controls \( c \) at a fixed location \( l \) of a fixed image \( x_0 \), i.e. varying only direction and magnitude of the shift vector. To measure the correlation in motion of all pixels with respect to the fixed control location (and thus the remaining object parts with the controlled part), we first compute optical flow maps between the start frame \( x_0 \) and the end frame \( x_T \) of all resulting synthesized video sequences. Next, we compute the shift of the tracked pixel locations in \( x_T \) with respect to the interaction location \( l \), thus obtaining 1000 [magnitude, angle] representations of the individual shifts. To measure the correlation of a pixel with \( l \), we now compute the variance over these shifts. Fig. 7 illustrates the resulting correlation maps given different locations \( l \) for both humans and plants. For humans, we obtain high correlations for pixels constituting a certain body parts and to parts which are naturally connected to \( l \), showing our model correctly understands the body structure. For the plant, we see pulling at locations close to the trunk (top mid and right) intuitively affects large parts of the object. Interacting with individual small leaves mostly has only little effect on the remaining object, in contrast to the pixels.

**iPER** [42] contains of 30 humans with diverse styles performing various simple and complex movements. We follow the official train/test split which results in a training set of 180k frames and a test set of 49k frames. **Tai-Chi-HD** [64] is a collection of 280 in-the-wild Tai-Chi videos from Youtube. We follow previous work [64] and use 252 videos for training and 28 videos for testing. Given the large variance in background and camera movement, this dataset tests the real-world applicability of our model. Since the motion between subsequent frames is often small, we skip every other frame. **Human3.6m** [30] is a large scale human motion dataset with video sequences of 7 human actors performing 17 different actions. We follow previous work [76, 53, 24] by center-cropping and downsampling the videos to 6.25 Hz and by using actors S1, S5, S6, S7 and S8 (600 videos) for training and actors S9 and S11 (239 videos) for testing.

### 4.2. Qualitative Evaluation

**Controlled Stochastic Video Synthesis.** In Fig. 3, 4 and 5 we show examples for controlled stochastic video synthesis generated by our proposed model on the PP, iPER and Human3.6m datasets. For each dataset, we show the ground-truth frames following a fixed given image \( x_0 \), as well as three synthesized examples generated from a fixed user control \( c \) (red arrows) and randomly sampled kinematics realizations \( r \sim q(r) \). Examples for the Tai-Chi dataset can be found in the supplemental, where we also show additional synthesized videos based on control inputs from real human users and demonstrate our model to also plausibly react to different poses at the same location. The individual videos are discussed in the Appendices A and B.

**Transfer of Kinematics.** Besides sampling plausible object kinematics, we can also apply our model to transfer the kinematics inferred from a source sequence \( X_s = [x_{s,0}, \ldots, x_{s,T}] \) to a novel object instance. To this end, we obtain high correlations for pixels constituting a certain body parts and to parts which are naturally connected to \( l \), showing our model correctly understands the body structure. For the plant, we see pulling at locations close to the trunk (top mid and right) intuitively affects large parts of the object. Interacting with individual small leaves mostly has only little effect on the remaining object, in contrast to the pixels.
4.3. Quantitative Evaluation

As our proposed task of controlled and stochastic video synthesis has been so far unattempted, we cannot directly compare iPOKE to previous work. To nevertheless quantitatively prove our model to reliably achieve this task, we separately compare against the current state of the art stochastic video prediction models [40, 10, 24] and sparsely controlled video synthesis approaches [27, 6]. For all competitors we used the provided pretrained models, where available, or trained the models using official code.

**Evaluation Metrics.**

**Motion Consistency.** We evaluate the synthesis quality by using the Fréchet Video Distance [68] (FVD, lower-is-better) which is responsive to visual as well as temporal coherence and uses an I3D network [67] trained on the Kinetics [34] dataset as backbone. Unterthiner et al. [68] showed that the metric correlates well with human judgement. The FVD-scores we report are obtained from video of length 10.

**Synthesis Quality.** Since we have no direct means to evaluate how well iPOKE models object kinematics, we compare its synthesized videos against the groundtruth using two commonly used framewise metrics, as producing uncorrect kinematics would lead to large errors between the individual generated and groundtruth frames. We average over time and over 5 samples due to the stochastic nature of our model. As it has been shown to account for high- and low-frequency image differences and also to correlate well with human judgement, LPIPS [82] (lower-is-better) is the metric of choice for this task. Additionally, due to its wide application, we report framewise discrepancy as measured by SSIM [84]. However, as this metric compares image patches based on the L2 distance, it is known to be deceivable by blurry predictions.

**Motion Diversity.** Following previous work [40, 85] we evaluate the diversity by computing mutual distances between the individual frames of different video samples (while fixing the user control) using the LPIPS [82] metric. Moreover, we also directly evaluate the diversity in the pixel space using the MSE, thus measuring low-frequency image differences.

**Controllable Video Synthesis.** We compare our model with the considered methods for sparsely controlled video synthesis [27, 6] on all considered datasets using LPIPS [82], SSIM [84] and FVD [68] on images of resolution of $128 \times 128$. Note that both competing baselines are limited in that they provide no means to stochastically model the inherent ambiguity of the non-controlled object parts. Additionally, [27] lacks a dedicated dynamics model, as this method is based on a warping technique, which we describe in Appendix F, and requires more than one control inputs to reliably generate complex object articulation. Due to these limitations, our model exhibits significantly better temporal and visual consistency as indicated by the large gaps in FVD and LPIPS scores in Tab. 1. To provide a stronger baseline, we also train and evaluate the model of Hao et al. with input trajectories based on groundtruth keypoints (Hao w/KP) which are readily available for the iPER dataset and much more reliable than those based on estimated optical flow. Despite this advantage, we also outperform this baseline in FVD and generate similarly sharp image frames as indicated by comparable LPIPS scores.

Next, we use the displacements between the groundtruth keypoints of the test sequences to construct targeted user controls for each individual part of the human body. By using these manipulations as test-time inputs and estimating keypoints [66] for the resulting generated videos, we assess the targeted control accuracy by measuring the Mean Squared Error (MSE) only between those estimated and groundtruth keypoints which correspond to the poked body parts. Fig. 8 shows the resulting error distributions and means (black dots) showing that we significantly outperform Hao et al. [27] and achieve similar performance to their keypoint-based version. Thus our model allows for accurate control of body parts which are correctly moved to the intended target locations.

**Stochastic Video Synthesis.** To evaluate the visual quality and the diversity of generated videos we compare against recent state of the art methods for stochastic video synthesis (SVS) [40, 53, 24], each of them based on variational autoencoder (VAE). We adopt the SVS evaluation protocol and generate videos of spatial size $64 \times 64$. Tab. 2...
we use the exact architecture of our video-autoencoding. We further compare our model method to a cVAE-baseline (Ours cVAE) for locally controlled video synthesis. Thus, we obtain significantly better motion video quality and diversity scores for all considered datasets. 1: SAVP faced mode collapse due to training instabilities caused by the two involved discriminator networks. As a consequence their model generates entirely equal outputs when sampling. Therefore, we are unable to report diversity scores for this baseline. 2: Reported numbers multiplied with 1e4.

Controlling Future Ambiguity. We now assess the ability of our model to control the degree of freedom in stochastic object articulation by varying the number of local pokes. Intuitively, an increasing number of user controls should result in more accurate predictions and lowered between-sample-variance due to the reduction in future ambiguity. We evaluate the amount of uncertainty in predictions by comparing average reconstruction scores of a fixed number of samples from \( q(r) \) for increasing numbers of user controls. More specifically, we report the mean prediction error and standard deviation of 50 samples (Std-50s) for each of 1000 input images and pokes. On the iPER dataset this is done by measuring MSE between estimated [66] and groundtruth keypoints. For PP dataset we resort to the LPIPS metrics as keypoints are not available. The resulting curves are depicted in Fig. 9. As expected, the decreasing prediction errors and between-sample-variances indicate that our model leverages the additional future information provided by an increased number of pokes. Thus, our model not only generates diverse predictions but also provides means to control their uncertainty by choosing an appropriate number of input pokes.

Ablation Study. As the competing VAE-grounded baselines for SVS are all conditioned on observed motion in form of observed past frames rather than dedicated, local user control, we further compare our model method to a cVAE-baseline (Ours cVAE) for locally controlled video synthesis. Thus, we use the exact architecture of our video-autoencoding framework (cf. Sec. 3.2) except for our latent cINN model.

5. Conclusion
We presented a novel model for controlling and synthesizing object kinematics of arbitrary object categories by locally manipulating object articulation using simple mouse drags. Our model is based on an invertible mapping between the generated video sequences and a dedicated kinematics representation learned from training videos only. To account for the ambiguity in the global object articulation given a local shift determining the motion of only an object part, learning is based on a probabilistic formulation, thus allowing us to sample and synthesize diverse kinematic realizations.

Acknowledgements
This research is funded in part by the German Federal Ministry for Economic Affairs and Energy within the project “KI-Absicherung – Safe AI for automated driving” and by the German Research Foundation (DFG) within project 421703927.
Appendix

Preliminaries

For all of our reported datasets in the main paper, we provide additional video material resulting in 14 videos in total. Analogous to the main paper, the direction of the control input 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. For some examples were either the arrow depicting the user input or the target location are small, we additionally overlay them with larger arrows and dots. We loop each video three times and replay it with three fps. The file structure of the videos is as follows:

```
04284-supp
  |--A-Additional_Visualizations
    |--A1-Controlled Stochastic Video Synthesis
      |--A1.1-Samples_PP.mp4
      |--A1.2-Samples_Iper.mp4
      |--A1.3-Samples_Human36m.mp4
      |--A1.4-Samples_TaiChi.mp4
    |--A2-Kinematics_Transfer
    |--A3-Control_Sensitivity
  |--B-Control Inputs of Human Users
    |--B1-Human_Control_PP.mp4
    |--B2-Human_Control_Iper.mp4
    |--B3-GUI-Demo_1.mp4
    |--B3-GUI-Demo_2.mp4
  |--C-Qualitative Comparison with Baselines
    |--C1-Controllable Video Synthesis
      |--C1.1-Cmp_Hao_PP.mp4
      |--C1.2-Cmp_Hao_Iper.mp4
    |--C2-Stochastic Video Synthesis
      |--C2.1-Cmp_IVRNN_PP.mp4
```
In Section A, we show and discuss qualitative video results of iPOKE on the four considered datasets for the tasks of controllable stochastic video synthesis (Section A.1) and transfer of kinematics between two unrelated object instances (Section A.2), which are the two main tasks considered in our paper. Our model iPOKE, however, also allows human users to animate still images by poking, as demonstrated by examples shown in Section B, which further contains examples of interactions of our model with human users via a graphical user interface. For the quantitative evaluations conducted in the main paper, we additionally provide qualitative comparisons in Section C. A derivation of our objective function is given in Section D. Moreover, we also provide the implementation details of our model in Section E and these of all considered baselines in Section F. Finally, describe the metrics used for evaluation in more detailed in Section G.

A. Additional Visualizations

Here we show and discuss additional video material showing the results of iPOKE on all considered datasets for our main task of controllable stochastic video synthesis. Moreover we also provide additional examples of successfully transferred kinematics from a source sequence to a given still image frame on the iPER [42] dataset.

A.1. Controllable Stochastic Video Synthesis

For each individual dataset presented in the main paper we show a video containing results of iPOKE for multiple object instances. Within each video, the first column depicts the ground truth video sequences starting with the same image frame $x_0$ which is used as part of the conditioning for our model. The remaining columns contain five generated video sequences corresponding to different samples drawn from the residual kinematics prior $q(r)$ for the same conditioning $(x_0, c)$. The user control $c$ is visualized as a red arrow starting at the controlled location $l = c_{3:4}$ and pointing to the intended target location, which is depicted by a red dot and plotted after the sequence has ended. As the arrows are sometimes tiny, we additionally visualize larger arrows indicating the direction of the user control input in the first column as an overlay over the ground truth sequences. For generating the user control, we used the procedure explained in Section 3.3 in the main paper. Subsequently, we discuss each video in detail.

A1.1-PokingPlants. For the PokingPlants dataset we show the results of iPOKE for five plants with substantially varying shapes and appearances in the video sequence A1.1-Samples_Plants.mp4. Despite these large variances, iPOKE manages to generate appealing video sequences showing plausible reactions to the user control $c$. In all cases, the intended end position of the controlled object part is reached. The remainder of the object, however, shows unique, plausible motion for each individual sample from $q(r)$ for the same conditioning. In summary, for the PokingPlants dataset, iPOKE is capable of rendering realistic video sequences by locally controlling the motion of a given part of the shown object instances. Moreover, diverse, plausible motion is generated for the remaining object parts.

A1.2-iPER. In the video A1.2-Samples_Iper.mp4 we visualize results for the test set of iPER [42], thus showing persons previously unseen by iPOKE. We clearly see that our model generalizes to unseen appearances. Moreover, iPOKE generates realistic video sequences showing the poked human body parts approaching their intended target location, while retaining significant motion variance for these parts not directly affected by the user control $c$.

A1.3-Human3.6m. To also evaluate the capabilities of iPOKE to generate human motion not contained in iPER, i.e. behavior such as walking and sitting, we additionally use the Human3.6m [30] dataset. The resulting synthesized output of our model can be seen in the video samples_human36m.mp4. Also in this case, iPOKE achieves to generate realistic looking video sequences showing such complex human motion for control inputs ranging from very small (e.g. third row) to large (e.g. second and last rows) control inputs. Unfortunately, iPOKE changes the appearance of the depicted person if the generated motion is large. However, this is a common issue, when using the Human3.6m test set [53, 70, 76, 24], as the train set only contains five distinct appearances and, thus, generalization of appearance is very hard to achieve.

A1.4-Tai-Chi. To further evaluate the capabilities of iPOKE to synthesize human motion for in-the-wild settings we use the Tai-Chi dataset [64]. This is a very challenging dataset due to diverse video backgrounds, which also contains motion. The human motion contained in the Tai-Chi, however, is not as large and diverse as for the iPER and Human3.6m datasets. Nevertheless, our model is able to synthesize realistic looking sequences as visualized in A1.4-Samples_Tai-Chi.mp4.
A.2. Kinematics Transfer

In the video A2-Kinematics_Transfer_Iper.mp4 we show the ability of iPOKE to transfer kinematics from a source sequence (top row) to an image showing a person of arbitrary appearance exhibiting a similar starting posture (mid row). For comparison, we synthesized sequences in which we induce random object kinematics by sampling from our residual kinematics prior \( q(r) \) (bottom row). For a further description on transfer of kinematics, cf. the respective paragraph in Section 4.2 in the main paper. The implications arising from observing these video examples are twofold: Firstly we can see that our model, due to its conditionally invertible nature, is capable of transferring kinematics from a given sequence to a still image in a fine-grained manner, as subtle details of the motion shown in the source sequence are preserved. Secondly, these experiments show that our residual kinematics representation \( r \) is indeed independent of the conditioning factors \((x_0, c)\) as i) our model only transfers kinematics from the source sequence to the target image and not the appearance of the depicted person and ii) the sequences obtained from drawing random samples from \( q(r) \) exhibit significantly different motion compared to the corresponding transferred sequences except for the controlled location, which is correctly approaching its target location in both cases. Thus, these two factors, are not influenced by the residual kinematics representation but only the kinematics of the body parts not directly influenced by the controlled location, due to this independence. This empirical evidence of the independence of \( r \) from \((x_0, c)\) is further backed by theoretical arguments which are discussed in Section D.

A.3. Control Sensitivity

In this paragraph, we directly verify that iPOKE also generates plausible object responses to control inputs \( c \) at the same pixel location \( c_{3:4} \) with different shift vectors \( c_{1:2} \) for a given initial image frame \( x_0 \). To this end, we sample the initial frame \( x_0 \) of a sequence \( X \) from the iPER [42] test set and simulate a vector \( c \) based on the procedure introduced in Sec. 3.3 from which we only take the respective interaction location \( c_{3:4} \). To obtain different shift vectors at this location we randomly sample \( L \) directions from a uniform distribution \( U(0, 2\pi) \) and magnitudes from \( U(\mu_{|F|}, \max_{|F|}) \), where \( \mu_{|F|} \) and \( \max_{|F|} \) denote the mean and maximum optical flow magnitudes in the optical flow map \( F \) between \( x_0 \) and the end frame \( x_T \) of \( X \). Thus, we obtain \( L \) different control inputs \( \{(c^i, x_0) : c_{3:4} = c_{3:4}^i\}_{i=1}^L \) at the same location for a given \( x_0 \). Drawing different residual samples \( \{r_i\}_{i=1}^L \) and applying the forward transformation results in video representations \( z_i = \tau(r_i|x_0, c_i) \) which can afterwards be decoded to object responses by our autoencoding framework (cf. Sec. 3.2 and E.1). Each row of the video A2-Control_Sensitivity_Iper.mp4 shows such generated object responses of iPOKE for \( L = 4 \) different shift vectors for the same static input image \( x_0 \) and compares these to the ground truth video starting with \( x_0 \). From observing these results, we see that the poked body part always approaches its intended target location which is defined by the respective shift vector \( c_{1:2} \), thus indicating the sensitivity of our model with respect to the control input.

B. Control Inputs of Human Users

So far, we have only considered simulated control inputs as described in Section 3.3 in the main paper. However, as our goal is not only to provide a model for controlled stochastic video synthesis, but also to enable human users to directly animate a still image frame, we now additionally show the results of our model for inputs of human users for the object categories of plants on the PP dataset in Sec. B.1 and humans on iPER in Sec. B.2. Each video contains three unique object instances shown in the rows of the respective video B1-Human_Control_PP.mp4 B2-Human_Control_Iper.mp4. In each row, we show the ground truth sequence on the left, followed by an example using a simulated control input as described in Sec. 3.3 in the main paper. In the remaining three rows, we show examples of control inputs by human users. The results were obtained via our graphical user interface (GUI) which is presented in the videos GUI_Demo_[1,2].mp4 in Section B.3. The GUI will be open sourced together with the project code upon publication.

B.1. PokingPlants

The three plants in the video B1-Human_Control_PP.mp4 again show that iPOKE is capable of generating plausible object kinematics as a reaction to the control. In particular, these examples highlight the accurate object reactions generated by our model even for fine-grained user control as, for instance, shown in the third column of the first row, where a control input with a small magnitude at an outer trunk results very subtle motion, leaving most of the controlled plant mainly unaffected. Large pokes, on the other hand, which are depicted in the remaining columns of the first row, result in large amounts of motion, thus corresponding to what we humans would expect from such a manipulation. Moreover, the results visualize the generalization abilities of iPOKE, as not all of these pokes from humans are within the data distribution.
B.2. iPER

The video B2-Human_Control_Iper.mp4 contains similar results to those discussed in the last section for the iPER dataset. The synthesized videos are realistic looking for inputs defined by human users and the controlled body parts are accurately moved to the intended target location. Moreover it is shown that our model learns to separate the pixels comprising the object from those in the background, as emerging from observing the fourth column in the top row, where a part of the background is controlled. In this case, our model does not generate human kinematics, but renders a still video sequence, where the entire person remains unaffected. This is caused by our training procedure where we explicitly generate control signals in the background to teach the iPOKE to ignore such inputs. For a detailed description, refer to Section E.3.

B.3. Graphical User Interface

Finally we recorded two videos (B3-GUI_Demo_1,2.mp4) showing iPOKE interactively generating kinematics as defined by humans using our graphical user interface. The videos were recorded without cuts and, thus, clearly indicate the robustness of iPOKE. Moreover it can be seen that our model is capable of processing multiple control inputs one after another without generating heavy artifacts or implausible kinematics, thus enabling an interactive experience for human users. We plan to make this GUI publicly available and will release it together with our code.

C. Qualitative Comparison with Competing Methods

To provide visual evidence for the quantitative evaluation presented in the main paper, we now show visual comparisons to Hao et al. [27] for the task of controllable video synthesis and to IVRNN [10] for the task of stochastic video synthesis. The comparisons with these methods are conducted in the iPER and PokingPlants (PP) datasets.

C.1. Controllable Video Synthesis

Similar to Sec. A.1, in C1.1-Comparison_Hao_PokingPlants.mp4 and C1.2-Comparison_Hao_Iper.mp4 we show video generations for iPOKE and compare them to the approach of Hao et al [27] trained on sparse sets of random optical flow vectors and tracked keypoint representations (cf. Sec. 4.3 in main paper for details). We observe that iPOKE produces fluent and realistic object kinematics on both datasets. In contrast, Hao et al. fails to capture and generate the complex, structured motion depicted in the iPER dataset due to only warping individual frames. We attribute the lower FVD scores obtained by this method, compared to ours, to this. On the PP dataset, the competing approach produces blurry outputs, as successive warping cannot represent coherent, fine-grained motion due to noisy optical flow estimates.

C.2. Stochastic Video Synthesis

In C2.1-Comparison_IVRNN_Plants.mp4 and C2.2-Comparison_IVRNN_Iper.mp4 we compare video sequences generated by our model against sequences generated by IVRNN [10], a state-of-the-art approach for unconditional stochastic video synthesis. We chose this particular competitor as it was the best performing of the competing methods on these two datasets when considering video quality and diversity together (cf. Tab. 2 in the main paper). In our case, we generate diverse video sequences depicting different object kinematic realizations inferred by sampling from the residual representation \( r \sim q(r) \) for different fixed user interactions \( c \) on the iPER and PP dataset, thus following a similar setup as in Sec. A.1. For IVRNN [10] we show different video generations for two conditioning frames preceding the sequence to be generated. Our videos clearly show that our model iPOKE successfully synthesizes plausible video sequences which closely follow the user interaction \( c \) while, on the other hand, the remaining object parts exhibit a diverse range of motion when varying \( r \). In contrast, the videos samples of IVRNN hardly show visible differences across different video samples. Moreover, IVRNN fails to capture fine-grained motion, e.g. of individual leaves, on the PP dataset, thus generating blurry video sequences. This can be explained by the trade-off between quality and diversity, which IVRNN, as a VAE-based method, suffers from.

D. Derivation and Implications of Learning Objective

D.1. Independence of \( r \) and \((x_0, c)\)

We intend to learn \( r \) defined in Eq. (4) such that the residual \( r \) contains all information about \( X \) not present in \((x_0, c)\). A way to achieve this is to force independence between \( r \) and the conditioning \((x_0, c)\). We now show that a minimizer of \( \text{KL}(p(r|x_0, c)||q(r)) \), where \( q(r) \) is some prior distribution independent of \((x_0, c)\), forces this independence. By the definition
of the Kullback-Leibler-Divergence, we have

$$\text{KL}[p(r|x_0, c)||q(r)] = \int_{r,x_0,c} p(r, x_0, c) \log \left( \frac{p(r|x_0, c)}{q(r)} \right) = \int_{r,x_0,c} p(r, x_0, c) \log (p(r|x_0, c)) - \int_r p(r) \log (q(r)) .$$

(10)

Following the argument of [63, 2] and using the fact that the Kullback-Leibler-Divergence is always positive, we have

$$\text{KL}[p(r)||q(r)] \geq 0 \Rightarrow \int_r p(r) \log (q(r)) \leq \int_r p(r) \log (p(r)) ,$$

(11)

where $p(r)$ denotes the (unknown) true distribution of $r$. By inserting (11) into (10), we obtain

$$\text{KL}[p(r|x_0, c)||q(r)] \geq \int_{r,x_0,c} p(r, x_0, c) \log (p(r|x_0, c)) - \int_r p(r) \log (p(r))$$

$$= \int_{r,x_0,c} p(r, x_0, c) \log \left( \frac{p(r|x_0, c)}{p(r)} \right)$$

$$= \int_{r,x_0,c} p(r, x_0, c) \log \left( \frac{p(r|x_0, c)}{p(r)p(x_0, c)} \right) = \text{MI}[r, (x_0, c)] .$$

As (12) implies, $\text{KL}[p(r|x_0, c)||q(r)] \geq \text{MI}[r, (x_0, c)]$ is an upper bound on the mutual information $\text{MI}[r, (x_0, c)]$ and, thus, its minimizer also forces independence of $r$ and $(x_0, c)$.

### D.2. Derivation of Objective Function

Again, we start with the definition of the Kullback-Leibler-Divergence and follow [63]

$$\text{KL}[p(r|x_0, c)||q(r)] = \int_r p(r|x_0, c) \log \left( \frac{p(r|x_0, c)}{q(r)} \right)$$

$$= \int_X p(\tau^{-1}_\theta(X|x_0, c)|x_0, c) \cdot |\det J_{\tau^{-1}}(X|x_0, c)| \cdot \log \left( \frac{p(\tau^{-1}_\theta(X|x_0, c)|x_0, c)}{q(\tau^{-1}_\theta(X|x_0, c))} \right),$$

(13)

where we have made a change of integration variable from $r$ to $X$. Further, by rearranging Eq. (6), we have

$$p(\tau^{-1}_\theta(X|x_0, c)|x_0, c) = \frac{p(X|x_0, c)}{|\det J_{\tau^{-1}}(X|x_0, c)|} .$$

(14)

Inserting (14) into (13) yields

$$\text{KL}[p(r|x_0, c)||q(r)] = \int_X p(X|x_0, c) \log \left( \frac{p(X|x_0, c)}{q(\tau^{-1}_\theta(X|x_0, c))|\det J_{\tau^{-1}}(X|x_0, c)|} \right)$$

$$= \mathbb{E}_X \left[ \log p(X|x_0, c) - \log (q(\tau^{-1}_\theta(X|x_0, c))) - \log (|\det J_{\tau^{-1}}(X|x_0, c)|) \right] .$$

(15)

If we now choose $q(r) = \mathcal{N}(0, I)$ and insert this into (15), we obtain our final objective by taking the expectation over $(x_0, c)$, as

$$\text{KL}[p(r|x_0, c)||q(r)] = \mathbb{E}_{X, x_0, c} \left[ \frac{1}{2} ||\tau^{-1}_\theta(X|x_0, c)||^2_2 - \log (|\det J_{\tau^{-1}}(X|x_0, c)|) \right] + H$$

$$\propto \mathbb{E}_{X, x_0, c} \left[ ||\tau^{-1}_\theta(X|x_0, c)||^2_2 - \log (|\det J_{\tau^{-1}}(X|x_0, c)|) \right] ,$$

(16)

where the constant entropy of the data $H = \mathbb{E}_X [\log p(X|x_0, c)]$ can be neglected for optimizing $\tau_\theta$. Summarizing the derived results, we can see, that minimizing $\text{KL}[p(r|x_0, c)||q(r)]$ not only yields a probabilistic generative
model capturing the data distribution, but also forces independence between the latent variable \( r \) and the conditioning \( (x_0, c) \). Interestingly, we can compare our model with cVAE-based models [38, 2, 61, 65] which have to balance between two contradicting loss terms where one ensures good reconstruction properties and the other induces independence between the latent variable and the conditioning. Thus, these models face the trade-off mentioned in the main paper. As derived above, minimizing \( KL[p(r|x_0, c)|q(r)] \) induces both these properties without forcing the model to balance between them (for a broader discussion, see [63]).

E. Implementation Details of iPoke

In this Section, we provide detailed information regarding our employed model architectures and losses as well as specifications of the models’ hyperparameters. Our model is implemented using PyTorch [57]. The corresponding code is available at https://bit.ly/36h8OKr.

E.1. Video Autoencoding Framework

We now provide details regarding the video autoencoding framework which is used to retain tractable computational cost for training our conditionally invertible model \( \tau_\theta \) (cf Section 3.2 in the main paper). The autoencoding framework is trained prior to optimizing \( \tau_\theta \). In the following we will first describe how to train this model, before its implementation details are given.

Model Overview and Losses.

Model Overview. As stated in the main paper, we pre-train a video-autoencoder to obtain a compact, information-preserving video encoding \( z = E(X) \in \mathbb{R}^{h \times w \times d} \) as fixed input for our invertible model \( \tau_\theta \), where \( E \) is a 3D-ResNet-encoder [28]. While training the autoencoding framework, \( z = z_0 \) constitutes the initial hidden state of the latent GRU [12] predicting the (temporally) subsequent states \( \hat{z}_i = \text{GRU}(\hat{z}_{i-1}), \ i = 1, \ldots, T \) which are mapped back into the image domain by using a 2D-decoder, thus yielding the individual images frames \( \hat{x}_i = D(\hat{z}_i) \) of a predicted sequence \( \hat{X} \). The autoencoding framework is visualized together with our invertible model in Fig. 2.

Training the Autoencoder. Here we explain the individual terms of the overall loss function of this model (9) in more detail. The first term, \( L_{rec} \), is an average reconstruction loss between the individual image frames of the ground-truth and predicted sequences, consisting of an \( L1 \)-loss in the pixel space and a perceptual loss \( \ell^p \) [20, 33].

\[
L_{rec} = \frac{1}{T} \left[ \sum_{i=1}^{T} \left\| x_i - \hat{x}_i \right\|_1 + \sum_{i=1}^{T} \ell^p(x_i, \hat{x}_i) \right] \tag{17}
\]

Moreover, we follow previous work [13, 73] and employ a spatial discriminator \( D_S \) and a temporal discriminator \( D_T \). Both discriminators are optimized using the hinge formulation [41, 7]. For stabilizing the GAN training, gradient penalty [52, 26] is used for the temporal discriminator. Additionally, we add a feature matching loss [72] for each individual discriminator, thus yielding the adversarial objectives

\[
L_{D_S} = \frac{1}{T} \sum_{i=1}^{T} \left[ D_S(\hat{x}_i) - \ell_{F_S}(x_i, \hat{x}_i) \right], \tag{18}
\]

and

\[
L_{D_T} = D_T(\hat{X}) + \ell_{F_T}(X, \hat{X}) \tag{19}
\]

for the generator, where \( \ell_{F_T} \) and \( \ell_{F_S} \) denote additional feature matching losses [72]. The counterficting loss objective for the temporal discriminator can be written as

\[
L_{D_T, D} = \rho(1 - D_T(X)) + \lambda_{gp} \left\| \nabla D_T(X) \right\|_2^2 + \rho(1 + D_T(X)) \tag{20}
\]

where \( \left\| \nabla D_T(X) \right\|_2^2 \) denotes the gradient penalty [52, 26] with \( \lambda_{gp} \) as its weighting factor and \( \rho \) the ReLU activation function. For the spatial discriminator, the objective can be formulated as

\[
L_{D_S, D} = \frac{1}{T} \sum_{i=1}^{T} [\rho(1 - D_S(x_i)) + \rho(1 + D_S(\hat{x}_i))]. \tag{21}
\]

Implementation details.

Encoder \( E \). The encoder \( E \) is a 3D-ResNet built up of layers following the structure proposed by [28]. GroupNorm [78]
Training Details. We train our model to reconstruct videos of sequence length $T = 10$ comprising images of spatial size $64 \times 64$ for comparison with the stochastic video synthesis baselines and $128 \times 128$ for all other experiments which are considered within the main paper. When training on sequences of spatial size $64 \times 64$, we use batch size 16 on a single NVIDIA RTX 2080 TI, whereas in the latter case, we use a single NVIDIA A100 with batch size 20. In both cases, all hyperparameters which are subsequently reported were also equally. The models are trained using Adam [35] with a learning rate of $2 \cdot 10^{-4}$, $\beta_1 = 0.5$, $\beta_2 = 0.9$, weight decay of $10^{-5}$, and exponential learning rate decay. The weighting factor of the gradient penalty for $D_T$ is chosen as $\lambda_{gp} = 1.2$.

E.2. Conditional Invertible Neural Network

In the following, we provide additional implementation details for our conditional invertible model $\tau_0$ and the encoders $\Phi_c$ and $\Phi_{x_0}$ processing the individual components of $(x_0, c)$.

Encoders $\Phi_c$ and $\Phi_{x_0}$. To obtain descriptive representations capturing all details regarding the two conditioning variables $c$ and $x_0$ we pretrain the respective encoders $\Phi_c$ and $\Phi_{x_0}$. We train $\Phi_{x_0}$ and a subsequent decoder to reconstruct image frames by using a combination of a standard $L1$ reconstruction loss and an additional perceptual loss as a training objective. $\Phi_c$ is also combined with a decoder at training time and gets a simulated user control $c$ (cf. Section 3.3) and the related source image frame $x_0$ as inputs. The obtained model is trained to reconstruct the entire flow field between $x_0$ and $x_T$. Again, a combination of $L1$ and an additional perceptual loss is used as loss function. In both cases, the decoders are used only for training and discarded afterwards.

Conditional INN $\tau_0$.

Model Architecture. The cINN is implemented as a convolutional normalizing flow model [60, 47, 37, 56]. As stated in the main paper, it is built up of $K$ subsequently arranged flow blocks, where the $k$-th block comprises $N_k$ sub-blocks, followed by a modified glow block [37], consisting of ActNorm [37], affine coupling [18] and shuffling layers\(^1\). In each sub-block, we employ masked convolutions [48], which subdivide a convolutional kernel into four systematically rotated, efficiently invertible sub-kernels, whose Jacobian is tractable. These properties are crucial when training normalizing flow model [60, 47, 37, 56], as they guarantee tractable training times and efficient sampling from $q(r)$. Moreover, masked convolutions result in the same receptive field than common convolutions and, thus, show improved expressivity compared to widely used architectures such as affine coupling layers [17, 18]. The flow overall architecture and the sub-blocks are visualized in Figure 2 in the main paper. For more details regarding masked convolutions, we refer the reader to [48].

Within all models and datasets, we use $K = 15$ flow blocks. The number $N_k$ of sub-blocks contained in the the $k$-th flow block decreases with increasing $k$. The individual number of sub blocks per flow block can be summarized by the following vector

$$N_{1:K} = [10, 5, 5, 4, 4, 4, 3, 3, 3, 2, 2, 1, 1, 1]$$

where the $k$-th component of $N_{1:K}$ denotes the number of sub-blocks within flow block $k$.

Training Details. We train our cINN model using Adam [36] optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and weight decay of $10^{-5}$. The learning rate is annealed from zero to $10^{-3}$ for the first 500 train steps and linearly decrease afterwards, until training ends. We train the cINN models on a single NVIDIA A100 using batches comprising 40 individual training

\(^1\)We modify these blocks by using shuffling layers instead of invertible $1 \times 1$ convolutions, as originally proposed in glow [37].
E.3. Further Training Details of iPOKE

Foreground-Background-Separation We do not only train our model to infer the implications of a localized poke onto the entire object, we also intend to learn iPOKE such that it only generates object kinematics for control inputs actually acting at the object itself. Hence a poking of a pixel in background areas unrelated to the object should be ignored. We now present an additional training strategy ensuring this property. 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 those locations with an optical flow magnitude larger than the mean of magnitudes of the flow map \( F \) for sampling the controlled location \( l = c_{1:4} \) for a user control with the depicted object. However, as we also want our model to separate the pixels on the object surface from those in the background, we sample the twelfth of all controlled locations in each epoch out of background pixels and construct artificial user controls by sampling the 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. This strategy is applied both to train the autoencoding framework described in Sec. E.1 and the cINN model \( \tau_\theta \). Thus, the model learns to separate the pixels comprising the object from those in the background, as indicated by the video examples in Section B, where we show examples for such artificial user controls.

F. Implementation Details of Baselines

Video Prediction Models. For comparison, we implemented the competing stochastic video prediction baselines [40, 10, 24] based on the officially provided code from Github.\(^2\) As no pretrained models are available for the utilized datasets, we trained models from scratch for all competitors except for SRVP [24], which provide a pretrained model for the Human3.6m [30] dataset. All models are trained to predict sequences of length 10 and spatial size 64 \( \times \) 64 based on two context frames. For models trained from scratch, we used the hyperparameters proposed in the respective publications. For SAVP [40] we further tried all individual hyperparameter-settings which are available in the official code repository\(^5\), but could not manage to obtain any visual diversity.

Controlled Video Synthesis Models. Both controllable video synthesis baselines we compare iPOKE with [27, 6] are implemented based on the official code repositories\(^6\) and the provided hyperparameters for generating videos of spatial size 128 \( \times \) 128 on all used datasets. For [27], due to the lack of pretrained weights, we trained the new models on the considered datasets, contrasting II2V [6], where we entirely used the available pretrained models.

For the method of Hao et al [27] we follow their proposed procedure to predict videos \([x_0, \ldots, x_T]\) of length \( T \): we first sample \( k \) flow vectors between \( x_0 \) and \( x_1 \). We then extend these initial vectors to discrete trajectories of length \( T - 1 \) in the image space by tracking the initial points using consecutive flow vectors between \( x_i \) and \( x_{i+1} \). Videos are finally obtained by successively warping \( x_0 \) using the shift vectors from the trajectories. We follow the official protocol from [27] and set \( k = 5 \).

The motion trajectories are generated from the same optical flow which was used to train our own model. For the model trained on ground truth keypoints on the iPER dataset, we constructed the motion trajectories based on the shifts between the individual keypoints of the source frame and all remaining frames of the considered video sequence.

Variational Counterpart of iPOKE. We here report the implementation details of the cVAE-based counterpart of iPOKE, which we compared our model to in the ablation study in Section 4.3. The model is composed of the individual parts of the autoencoding framework of iPOKE as described in Sec. E.1. Thus, the cVAE-grounded baseline consists of a 3D-Encoder, a convolutional GRU and a 2D-decoder following the respective architectures presented in Section E.1. Additionally, we use two encoders to individually encode the components of the conditioning \((x_0, c)\) with the same architecture than \( \Phi_{x_0} \) and \( \Phi_c \) as described in Section E.2. All these components of the overall model have the same inputs and outputs as the corresponding components of iPOKE. The differences between the two models are i) the regularization of the video encoding \( z \) towards the standard normal prior by using the reparameterization trick [38], as usually done in cVAEs [38, 61], instead of using a cINN operating on the latent level, and ii) the initialization of the hidden state of the latent GRU, which are here the stacked features of the reparametrized video encoding and the representations of the source image \( x_0 \) and the user control \( c \). For iPOKE, in

\(2\)https://github.com/CompVis/interactive-image2video-synthesis
\(3\)https://github.com/facebookresearch/improved_vrnn
\(4\)https://github.com/edouardelasalles/srvp
\(5\)https://github.com/zekunhao1995/ControllableVideoGen
\(6\)https://github.com/CompVis/interactive-image2video-synthesis
contrast, the initial hidden state is only the video representation $z = \tau_\theta(r|x_0, c)$.
The losses for the model are exactly similar to those of our video autoencoding framework as summarized in Eq. (9), except for an additional KL-loss ensuring the mentioned regularization of the video representation towards the standard normal. We tried different options for the weighting factor $\lambda_{KL}$ of this KL-term, as its choice has a major influence on the performance of the trained model and, thus, needs to be selected carefully [11, 83]. The best performance was obtained by choosing $\lambda_{KL} = 0.1$. Additionally, to avoid posterior collapse [46], we use KL-annealing for 5 epochs, before the final value of $\lambda_{KL}$ is reached. Similar to our proposed autoencoding model for iPOKE, the VAE-baseline is trained using Adam [36] with a learning rate of $2 \cdot 10^{-4}$, $\beta_1 = 0.5$, $\beta_2 = 0.9$, weight decay of $10^{-5}$, and exponential learning rate decay. All remaining hyperparameters were chosen to be equal to those of the corresponding iPOKE-model for video sequences of spatial size $64 \times 64$.

G. Evaluation Details

Motion Consistency. To compute the FVD-score [68] 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 the I3D [8] model which was pretrained on the Kinetics [9] dataset. Subsequently their distributions in the I3D feature space are compared resulting in the reported FVD-scores. For all models we concatenate the last conditioning frame (for our model the input frame) with the generated sequence.

Synthesis Quality. All reported accuracy metrics are based on 1000 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 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.

Motion Diversity. To evaluate the diversity for each model, we generated 5 realizations for each video resulting in $5 \times 1000$ videos. The diversity is then computed between the different realizations of the video using LPIPS[82] and MSE in the pixel space. We did not notice any change in numbers when using more realizations.
References

[1] Kfir Aberman, Yijia Weng, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Unpaired motion style transfer from video to animation. *ACM Transactions on Graphics (TOG)*, 39(4):64, 2020. 1, 2

[2] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. *CoRR*, 2016. 4, 13, 14

[3] Lynton Ardizzone, Jakob Kruse, Carsten Rother, and Ulrich Köthe. Analyzing inverse problems with invertible neural networks. In *Int. Conf. Learn. Represent.*, 2019. 2

[4] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3703–3712, 2019. 2

[5] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Bjorn Ommer. Behavior-driven synthesis of human dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12236–12246, June 2021. 2

[6] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Bjorn Ommer. Understanding object dynamics for interactive image-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5171–5181, June 2021. 2, 5, 7, 16

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Int. Conf. Learn. Represent.*, 2019. 14

[8] 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. 17

[9] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4724–4733, 2017. 17

[10] L. Castrejon, N. Ballas, and A. Courville. Improved conditional vrnns for video prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2, 3, 7, 8, 12, 16

[11] Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. In *Int. Conf. Learn. Represent.*, 2017. 3, 17

[12] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014. cite arxiv:1412.3555Comment: Presented in NIPS 2014 Deep Learning and Representation Learning Workshop. 4, 14

[13] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets, 2019. 4, 14

[14] 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. 15

[15] Abe Davis, Justin G. Chen, and Frédéric Durand. Image-space modal bases for plausible manipulation of objects in video. *ACM Trans. Graph.*, 2015. 1, 2

[16] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In Jennifer G. Dy and Andreas Krause, editors, *International Conference on Machine Learning*, pages 1182–1191, 2018. 2, 3

[17] Laurent Dinl, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In Yoshua Bengio and Yann LeCun, editors, *Int. Conf. Learn. Represent.*, 2015. 4, 5, 15

[18] Laurent Dinl, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *Int. Conf. Learn. Represent.*, 2017. 4, 5, 15

[19] Michael Dorkenwald, Timo Milbich, Andreas Blattmann, Robin Rombach, Konstantinos G. Derpanis, and Bjorn Ommer. Stochastic image-to-video synthesis using cinns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3742–3753, June 2021. 2, 4

[20] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Adv. Neural Inform. Process. Syst.*, pages 658–666, 2016. 14

[21] Patrick Esser, Johannes Haux, Timo Milbich, and Björn Ommer. Towards learning a realistic rendering of human behavior. In *ECCV Workshops*, pages 409–425, 2018. 1

[22] Patrick Esser, Robin Rombach, and Björn Ommer. A disentangling invertible interpretation network for explaining latent representations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9220–9229, 2020. 2

[23] Chelsea Finn, Ian J. Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 64–72, 2016. 2

[24] Jean-Yves Franceschi, Edouard Delasalles, Mickael Chen, Sylvain Lamprier, and P. Gallinari. Stochastic latent residual video prediction. 2020. 2, 6, 7, 8, 10, 16

[25] Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, and Trevor Darrell. Disentangling propagation and generation for video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2
[54] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1701–1710, 2018. 2
[55] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5436–5445, 2020. 2
[56] George Papamakarios, Eric T. Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *CoRR*, 2019. 4, 15
[57] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Adv. Neural Inform. Process. Syst.*, pages 8024–8035. 2019. 14
[58] Albert Pumarola, Stefan Popov, Francesc Moreno-Noguer, and Vittorio Ferrari. C-flow: Conditional generative flow models for images and 3d point clouds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7946–7955. 2020. 2
[59] Fitsum Reda, Guilin Liu, Kevin Shih, Robert Kirby, Jon Barker, David Tarjan, Andrew Tao, and Bryan Catanzaro. SDC-Net: Video Prediction Using Spatially-Displaced Convolution. 2018. 2
[60] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis R. Bach and David M. Blei, editors, *International Conference on Machine Learning*, pages 1530–1538. Curran Associates, Inc., 2015. 2, 15
[61] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286. 2014. 3, 14, 16
[62] Robin Rombach, Patrick Esser, and Björn Ommer. Network fusion for content creation with conditional inns. *CoRR*, 2020. 2, 4, 5
[63] Robin Rombach, Patrick Esser, and Björn Ommer. Network-to-network translation with conditional invertible neural networks. In *Adv. Neural Inform. Process. Syst.*, 2020. 2, 5, 13, 14
[64] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Adv. Neural Inform. Process. Syst.*, pages 7135–7145. 2019. 6, 7, 8, 10, 15
[65] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015. 3, 14
[66] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5693–5703. Computer Vision Foundation / IEEE, 2019. 7, 8
[67] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2818–2826. 2016. 7
[68] Thomas Unterrthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Mariner, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *CoRR*, 2018. 7, 17
[69] Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *International Conference on Learning Representations*, 2018. 2
[70] Ruben Villegas, Jimei Yang, Seunghoong Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *Int. Conf. Learn. Represent.*, 2017. 2, 10
[71] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 613–621. 2016. 2
[72] 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. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8798–8807. 2018. 14
[73] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Nikolai Yakovenko, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Adv. Neural Inform. Process. Syst.*, pages 1152–1164. 2018. 2, 4, 14
[74] Tsun-Hsuan Wang, Yen-Chi Cheng, Chieh Hubert Lin, Hwann-Tzong Chen, and Min Sun. Point-to-point video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2
[75] Dirk Weissborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *Int. Conf. Learn. Represent.*, 2020. 2
[76] 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. 2, 6, 10
[77] Yue Wu, Rongrong Gao, Jaesik Park, and Qifeng Chen. Future video synthesis with object motion prediction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5538–5547. 2020. 2, 3
[78] Yuxin Wu and Kaiming He. Group normalization. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Adv. Neural Inform. Process. Syst.*, pages 7135–7145. 2018. 7, 17
[79] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019. 2
[80] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Eur. Conf. Comput. Vis.*, pages 776–791. 2016. 4
[81] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2

[82] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 7, 17

[83] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *CoRR*, 2017. 3, 17

[84] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 7

[85] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Adv. Neural Inform. Process. Syst.*, pages 465–476, 2017. 7