SLAMP: Stochastic Latent Appearance and Motion Prediction

Adil Kaan Akan[1] Erkut Erdem[2] Aykut Erdem[1] Fatma Güney[1]
[1] Koç University Is Bank AI Center, Istanbul, Turkey
[2] Hacettepe University Computer Vision Lab, Ankara, Turkey
{kakan20,aerdem,fguney}@ku.edu.tr erkut@cs.hacettepe.edu.tr
https://kuis-ai.github.io/slamp

Abstract

Motion is an important cue for video prediction and often utilized by separating video content into static and dynamic components. Most of the previous work utilizing motion is deterministic but there are stochastic methods that can model the inherent uncertainty of the future. Existing stochastic models either do not reason about motion explicitly or make limiting assumptions about the static part. In this paper, we reason about appearance and motion in the video stochastically by predicting the future based on the motion history. Explicit reasoning about motion without history already reaches the performance of current stochastic models. The motion history further improves the results by allowing to predict consistent dynamics several frames into the future. Our model performs comparably to the state-of-the-art models on the generic video prediction datasets, however, significantly outperforms them on two challenging real-world autonomous driving datasets with complex motion and dynamic background.

1. Introduction

Videos contain visual information enriched by motion. Motion is a useful cue for reasoning about human activities or interactions between objects in a video. Given a few initial frames of a video, our goal is to predict several frames into the future, as realistically as possible. By looking at a few frames, humans can predict what will happen next. Surprisingly, they can even attribute semantic meanings to random dots and recognize motion patterns [15]. This shows the importance of motion to infer the dynamics of the video and to predict the future frames.

Motion cues have been heavily utilized for future frame prediction in computer vision. A common approach is to factorize the video into static and dynamic components [30, 20, 22, 6, 9, 21, 14, 28]. First, most of the previous methods are deterministic and fail to model the uncertainty of the future. Second, motion is typically interpreted as local changes from one frame to the next. However, changes in motion follow certain patterns when observed over some time interval. Consider scenarios where objects move with near-constant velocity, or humans repeating atomic actions in videos. Regularities in motion can be very informative for future frame prediction. In this work, we propose to explicitly model the change in motion, or *the motion history*, for predicting future frames.

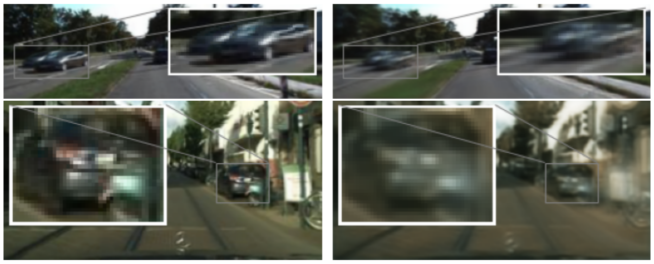

Figure 1: Comparison of the first prediction frames (11th) SLAMP (**left**) vs. state-of-the-art method, SRVP [8] (**right**) on KITTI [10] (**top**) and Cityscapes [3] (**bottom**) datasets. Our method can predict both foreground and background objects better than SRVP. Full sequence predictions can be seen in Supplementary.

Stochastic methods have been proposed to model the inherent uncertainty of the future in videos. Earlier methods encode the dynamics of the video in stochastic latent variables which are decoded to future frames in a deterministic way [4]. We first assume that both appearance and motion are encoded in the stochastic latent variables and decode them separately into appearance and motion predictions in a deterministic way. Inspired by the previous deterministic methods [7, 20, 9], we also estimate a mask relating the two. Both appearance and motion decoders are expected to predict the full frame but they might fail due to occlusions around motion boundaries. Intuitively, we predict a probabilistic mask from the results of the appearance and motion decoders to combine them into a more accurate final prediction. Our model learns to use motion cues in the dynamic parts and relies on appearance in the occluded regions.

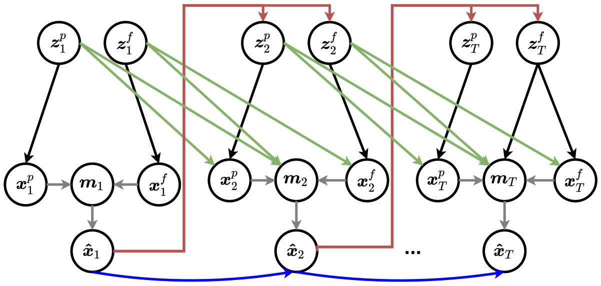

Figure 2: **Generative Model of SLAMP.** The graphical model shows the generation process of SLAMP with motion history. There are two separate latent variables for appearance $\mathbf{z}_t^p$ and motion $\mathbf{z}_t^f$ generating frames $\mathbf{x}_t^p$ and $\mathbf{x}_t^f$ (black). Information is propagated between time-steps through the recurrence between frame predictions (blue), corresponding latent variables (green), and from frame predictions to latent variables (red). The final prediction $\hat{\mathbf{x}}_t$ is a weighted combination of the $\mathbf{x}_t^p$ and $\mathbf{x}_t^f$ according to the mask $\mathbf{m}(\mathbf{x}_t^p, \mathbf{x}_t^f)$. Note that predictions at a time-step depend on all of the previous time-steps recurrently, but only the connections between consecutive ones are shown for clarity.

The proposed stochastic model with deterministic decoders cannot fully utilize the motion history, even when motion is explicitly decoded. In this work, we propose a model to recognize regularities in motion and remember them in the motion history to improve future frame predictions. We factorize stochastic latent variables as static and dynamic components to model the motion history in addition to the appearance history. We learn two separate distributions representing appearance and motion and then decode static and dynamic parts from the respective ones.

Our model outperforms all the previous work and performs comparably to the state-of-the-art method, SRVP, [8] without any limiting assumptions on the changes in the static component on the generic video prediction datasets, MNIST, KTH and BAIR. However, our model outperforms all the previous work, including SRVP, on two challenging real-world autonomous driving datasets with dynamic background and complex object motion.

2. Related Work

Appearance-Motion Decomposition: The previous work explored motion cues for video generation either explicitly with optical flow [30, 29, 19, 20, 22, 6, 9] or implicitly with temporal differences [21] or pixel-level transformations [14, 28]. There are some common factors among these methods such as using recurrent models [25, 21, 6], specific processing of dynamic parts [14, 19, 6, 9], utilizing a mask [7, 20, 9], and adversarial training [28, 22]. We also use recurrent models, predict a mask, and separately process motion, but in a stochastic way.

The previous work which explored motion for video generation are mostly deterministic, therefore failing to capture uncertainty of the future. There are a couple of attempts to learn multiple future trajectories from a single image with a conditional variational autoencoder [29] or to capture motion uncertainty with a probabilistic motion encoder [19]. The latter work uses separate decoders for flow and frame similar to our approach, however, predicts them only from the latent vector. We incorporate information from previous frames with additional modelling of the motion history.

Stochastic Video Generation: SV2P [1] and SVG [4] are the first to model the stochasticity in video sequences using latent variables. The input from past frames are encoded in a posterior distribution to generate the future frames. In a stochastic framework, learning is performed by maximizing the likelihood of the observed data and minimizing the distance of the posterior distribution to a prior distribution, either fixed [1] or learned from previous frames [4]. Since time-variance in the model is proven crucial by the previous work, we sample a latent variable at every time step [4]. Sampled random variables are fed to a frame predictor, modelled recurrently using an LSTM. We model appearance and motion distributions separately and train two frame predictors for static and dynamic parts.

Typically, each distribution, including the prior and the posterior, is modeled with a recurrent model such as an LSTM. Villegas et al. [27] replace the linear LSTMs with convolutional ones at the cost of increasing the number of parameters. Castrejon et al. [2] introduce a hierarchical representation to model latent variables at different scales, by introducing additional complexity. Lee et al. [17] incorporate an adversarial loss into the stochastic framework to generate sharper images, at the cost of less diverse results. Our model with linear LSTMs can generate diverse and sharp-looking results without any adversarial losses, by incorporating motion information successfully into the stochastic framework. Recent methods model dynamics of the keypoints to avoid errors in pixel space and achieve stable learning [23]. This offers an interesting solution for videos with static background and moving foreground objects that can be represented with keypoints. Our model can generalize to videos with changing background without needing keypoints to represent objects.

Optical flow has been used before in future prediction [18, 22]. Li et al. [18] generate future frames from a still image by using optical flow generated by an off-the-shelf model, whereas we compute flow as part of prediction. Lu et al. [22] use optical flow for video extrapolation and interpolation without modeling stochasticity. Long-term video extrapolation results show the limitation of this work in terms of predicting future due to relatively small motion magnitudes considered in extrapolation. Differently from flow, Xue et al. [31] model the motion as image differences using cross convolutions.

State-Space Models: Stochastic models are typically auto-regressive, i.e. the next frame is predicted based on the frames generated by the model. As opposed to interleaving process of auto-regressive models, state-space models separate the frame generation from the modelling of dynamics [12]. State-of-the-art method SRVP [8] proposes a state-space model for video generation with deterministic state transitions representing residual change between the frames. This way, dynamics are modelled with latent state variables which are independent of previously generated frames. Although independent latent states are computationally appealing, they cannot model the motion history of the video. In addition, content variable designed to model static background cannot handle changes in the background. We can generate long sequences with complex motion patterns by explicitly modelling the motion history without any limiting assumptions about the dynamics of the background.

3. Methodology

3.1. Stochastic Video Prediction

Given the previous frames $\mathbf{x}_{1:t-1}$ until time t, our goal is to predict the target frame $\mathbf{x}_t$. For that purpose, we assume that we have access to the target frame $\mathbf{x}_t$ during training and use it to capture the dynamics of the video in stochastic latent variables $\mathbf{z}_t$. By learning to approximate the distribution over $\mathbf{z}_t$, we can decode the future frame $\mathbf{x}_t$ from $\mathbf{z}_t$ and the previous frames $\mathbf{x}_{1:t-1}$ at test time.

Using all the frames including the target frame, we compute a posterior distribution $q_{\phi}(\mathbf{z}_t|\mathbf{x}_{1:t})$ and sample a latent variable $\mathbf{z}_t$ from this distribution at each time step. The stochastic process of the video is captured by the latent variable $\mathbf{z}_t$. In other words, it should contain information accumulated over the previous frames rather than only condensing the information on the current frame. This is achieved by encouraging $q_{\phi}(\mathbf{z}_t|\mathbf{x}_{1:t})$ to be close to a prior distribution $p(\mathbf{z})$ in terms of KL-divergence. The prior can be sampled from a fixed Gaussian

at each time step or can be learned from previous frames up to the target frame $p_{\psi}(\mathbf{z}_t|\mathbf{x}_{1:t-1})$. We prefer the latter one as it is shown to work better by learning a prior that varies across time [4].

The target frame $\mathbf{x}_t$ is predicted based on the previous frames $\mathbf{x}_{1:t-1}$ and the latent vectors $\mathbf{z}_{1:t}$.

In practice, we only use the latest frame $\mathbf{x}_{t-1}$ and the latent vector $\mathbf{z}_t$ as input and dependencies from further previous frames are propagated with a recurrent model. The output of the frame predictor $\mathbf{g}_t$

contains the information required to decode $\mathbf{x}_t$.

Typically, $\mathbf{g}_t$ is decoded to a fixed-variance Gaussian distribution whose mean is the predicted target frame $\hat{\mathbf{x}}_t$ [4].

3.2. SLAMP

We call the predicted target frame, appearance prediction $\mathbf{x}_t^p$ in the pixel space. In addition to $\mathbf{x}_t^p$, we also estimate optical flow $\mathbf{f}_{t-1:t}$ from the previous frame $t-1$ to the target frame t. The flow $\mathbf{f}_{t-1:t}$ represents the motion of the pixels from the previous frame to the target frame. We reconstruct the target frame $\mathbf{x}_t^f$ from the estimated optical flow via differentiable warping [13]. Finally, we estimate a mask $\mathbf{m}(\mathbf{x}_t^p, \mathbf{x}_t^f)$ from the two frame estimations to combine them into the final estimation $\hat{\mathbf{x}}_t$:

$$\hat{\mathbf{x}}_t = \mathbf{m}(\mathbf{x}_t^p, \mathbf{x}_t^f) \odot \mathbf{x}_t^p + (\mathbf{1} - \mathbf{m}(\mathbf{x}_t^p, \mathbf{x}_t^f)) \odot \mathbf{x}_t^f \qquad (1)$$

where $\odot$ denotes element-wise Hadamard product and $\mathbf{x}_t^f$ is the result of warping the source frame to the target frame according to the estimated flow field $\mathbf{f}_{t-1:t}$. Especially in the dynamic parts with moving objects, the target frame can be reconstructed accurately using motion information. In the occluded regions where motion is unreliable, the model learns to rely on the appearance prediction. The mask prediction learns a weighting between the appearance and the motion predictions for combining them.

We call this model SLAMP-Baseline because it is limited in the sense that it only considers the motion with respect to the previous frame while decoding the output. In SLAMP, we extend the stochasticity in the appearance space to the motion space as well. This way, we can model appearance changes and motion patterns in the video explicitly and make better predictions of future. Fig. 3 shows an illustration of SLAMP (see Supplementary for SLAMP-Baseline).

In order to represent appearance and motion, we compute two separate posterior distributions $q_{\phi_p}(\mathbf{z}_t^p|\mathbf{x}_{1:t})$ and $q_{\phi_f}(\mathbf{z}_t^f|\mathbf{x}_{1:t})$, respectively. We sample two latent variables $\mathbf{z}_t^p$ and $\mathbf{z}_t^f$ from these distributions in the pixel space and the flow space. This allows a decomposition of the video into static and dynamic components. Intuitively, we expect the dynamic component to focus on changes and the static to what remains constant from the previous frames to the target frame. If the background is moving according to a camera motion, the static component can model the change in the background assuming that it remains constant throughout the video, e.g. ego-motion of a car.

The Motion History: The latent variable $\mathbf{z}_t^f$ should contain motion information accumulated over the previous frames rather than local temporal changes between the last frame and the target frame. We achieve this by encouraging $q_{\phi_f}(\mathbf{z}_t^f|\mathbf{x}_{1:t})$ to be close to a prior distribution in terms of KL-divergence. Similar to [4], we learn the motion prior conditioned on previous frames up to the target frame: $p_{\psi_f}(\mathbf{z}_t^f|\mathbf{x}_{1:t-1})$. We repeat the same for the static part represented by $\mathbf{z}_t^p$ with posterior $q_{\phi_p}(\mathbf{z}_t^p|\mathbf{x}_{1:t})$ and the learned prior $p_{\psi_p}(\mathbf{z}_t^p|\mathbf{x}_{1:t-1})$.

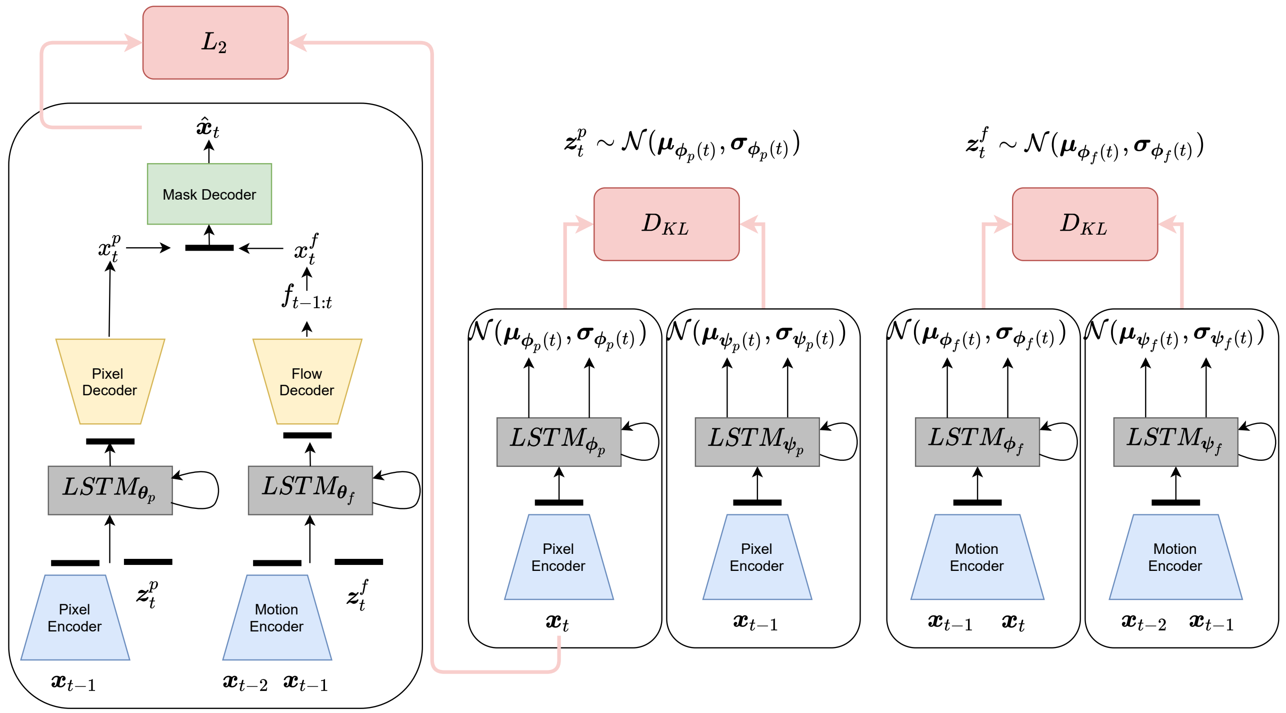
L_2
$\hat{\boldsymbol{x}}_t$
Mask Decoder
x_t^p
x_t^f
$f_{t-1:t}$
Pixel Decoder
Flow Decoder
$LSTM_{\boldsymbol{\theta}_p}$
$LSTM_{\boldsymbol{\theta}_f}$
Pixel Encoder
$\boldsymbol{z}_t^p$
Motion Encoder
$\boldsymbol{z}_t^f$
$\boldsymbol{x}_{t-1}$
$\boldsymbol{x}_{t-2}$ $\boldsymbol{x}_{t-1}$
$\boldsymbol{z}_t^p \sim \mathcal{N}(\boldsymbol{\mu}_{\phi_p(t)}, \boldsymbol{\sigma}_{\phi_p(t)})$
D_{KL}
$\boldsymbol{z}_t^f \sim \mathcal{N}(\boldsymbol{\mu}_{\phi_f(t)}, \boldsymbol{\sigma}_{\phi_f(t)})$
D_{KL}
$\mathcal{N}(\boldsymbol{\mu}_{\phi_p(t)}, \boldsymbol{\sigma}_{\phi_p(t)})$
$\mathcal{N}(\boldsymbol{\mu}_{\psi_p(t)}, \boldsymbol{\sigma}_{\psi_p(t)})$
$\mathcal{N}(\boldsymbol{\mu}_{\phi_f(t)}, \boldsymbol{\sigma}_{\phi_f(t)})$
$\mathcal{N}(\boldsymbol{\mu}_{\psi_f(t)}, \boldsymbol{\sigma}_{\psi_f(t)})$
$LSTM_{\phi_p}$
$LSTM_{\psi_p}$
$LSTM_{\phi_f}$
$LSTM_{\psi_f}$
Pixel Encoder
Pixel Encoder
Motion Encoder
Motion Encoder
$\boldsymbol{x}_t$
$\boldsymbol{x}_{t-1}$
$\boldsymbol{x}_{t-1}$ $\boldsymbol{x}_t$
$\boldsymbol{x}_{t-2}$ $\boldsymbol{x}_{t-1}$

Figure 3: **SLAMP.** This figure shows the components of our SLAMP model including the prediction model, inference and learned prior models for pixel and then flow from left to right. Observations $\mathbf{x}_t$ are mapped to the latent space by using a pixel encoder for appearance on each frame and and a motion encoder for motion between consecutive frames. The blue boxes show encoders, yellow and green ones decoders, gray ones recurrent posterior, prior, and predictor models, and lastly red ones show loss functions during training. Note that L_2 loss is applied three times for appearance prediction $\mathbf{x}_t^p$, motion prediction $\mathbf{x}_t^f$, and the combination of the two $\hat{\mathbf{x}}_t$ according to the mask prediction $\mathbf{m}(\mathbf{x}_t^p, \mathbf{x}_t^f)$. We only show L2 loss between the actual frame $\mathbf{x}_t$ and the final predicted frame $\hat{\mathbf{x}}_t$ in the figure. For inference, only the prediction model and learned prior models are used.

3.3. Variational Inference

For our basic formulation (SLAMP-Baseline), the derivation of the loss function is straightforward and provided in Supplementary. For SLAMP, the conditional joint probability corresponding to the graphical model in Fig. 2 is:

$$p(\mathbf{x}_{1:T}) = \prod_{t=1}^{T} \; p(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{z}_t^p, \mathbf{z}_t^f) \qquad (2)$$
$$p(\mathbf{z}_t^p | \mathbf{x}_{1:t-1}, \mathbf{z}_{t-1}^p) \;\; p(\mathbf{z}_t^f | \mathbf{x}_{1:t-1}, \mathbf{z}_{t-1}^f)$$

The true distribution over the latent variables $\mathbf{z}_t^p$ and $\mathbf{z}_t^f$ is intractable. We train time-dependent inference networks $q_{\phi_p}(\mathbf{z}_t^p | \mathbf{x}_{1:T})$ and $q_{\phi_f}(\mathbf{z}_t^f | \mathbf{x}_{1:T})$ to approximate the true distribution with conditional Gaussian distributions. In order to optimize the likelihood of $p(\mathbf{x}_{1:T})$, we need to infer latent variables $\mathbf{z}_t^p$ and $\mathbf{z}_t^f$, which correspond to uncertainty of static and dynamic parts in future frames, respectively. We use a variational inference model to infer the latent variables.

Since $\mathbf{z}_t^p$ and $\mathbf{z}_t^f$ are independent across time, we can decompose Kullback-Leibler terms into individual time steps. We train the model by optimizing the variational lower bound (see Supplementary for the derivation):

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) \geq \mathcal{L}_{\boldsymbol{\theta}, \boldsymbol{\phi}_p, \boldsymbol{\phi}_f, \boldsymbol{\psi}_p, \boldsymbol{\psi}_f}(\mathbf{x}_{1:T}) \qquad (3)$$
$$= \sum_t \mathbb{E}_{\substack{\mathbf{z}_{1:t}^p \sim q_{\phi_p} \\ \mathbf{z}_{1:t}^f \sim q_{\phi_f}}} \log p_{\boldsymbol{\theta}}(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}^p, \mathbf{z}_{1:t}^f)$$
$$- \beta \Big[D_{\mathrm{KL}}(q(\mathbf{z}_t^p | \mathbf{x}_{1:t}) \;||\; p(\mathbf{z}_t^p | \mathbf{x}_{1:t-1}))$$
$$+ D_{\mathrm{KL}}(q(\mathbf{z}_t^f | \mathbf{x}_{1:t}) \;||\; p(\mathbf{z}_t^f | \mathbf{x}_{1:t-1})) \Big]$$

The likelihood $p_{\boldsymbol{\theta}}$, can be interpreted as an L_2 penalty between the actual frame $\mathbf{x}_t$ and the estimation $\hat{\mathbf{x}}_t$ as defined in (1). We apply the L_2 loss to the predictions of appearance and motion components as well.

The posterior terms for uncertainty are estimated as an expectation over $q_{\phi_p}(\mathbf{z}_t^p | \mathbf{x}_{1:t})$, $q_{\phi_f}(\mathbf{z}_t^f | \mathbf{x}_{1:t})$. As in [4], we also learn the prior distributions from the previous frames up to the target frame as $p_{\psi_p}(\mathbf{z}_t^p | \mathbf{x}_{1:t-1})$, $p_{\psi_f}(\mathbf{z}_t^f | \mathbf{x}_{1:t-1})$. We train the model using the re-parameterization trick [16]. We classically choose the posteriors to be factorized Gaussian so that all the KL divergences can be computed analytically.

3.4. Architecture

We encode the frames with a feed-forward convolutional architecture to obtain appearance features at each time-step. In SLAMP, we also encode consecutive frame pairs into a feature vector representing the motion between them. We then train linear LSTMs to infer posterior and prior distributions at each time-step from encoded appearance and motion features.

Stochastic video prediction model with a learned prior [4] is a special case of our baseline model with a single pixel decoder, we also add motion and mask decoders. Next, we describe the steps of the generation process for the dynamic part.

At each time step, we encode $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ into $\mathbf{h}_t^f$, representing the motion from the previous frame to the target frame. The posterior LSTM is updated based on the $\mathbf{h}_t^f$:

$$\begin{aligned} \mathbf{h}_t^f &= \mathrm{MotionEnc}(\mathbf{x}_{t-1}, \mathbf{x}_t) \\ \boldsymbol{\mu}_{\boldsymbol{\phi}_f(t)}, \boldsymbol{\sigma}_{\boldsymbol{\phi}_f(t)} &= \mathrm{LSTM}_{\boldsymbol{\phi}_f}(\mathbf{h}_t^f) \end{aligned} \tag{4}$$

For the prior, we use the motion representation $\mathbf{h}_{t-1}^f$ from the previous time step, i.e. the motion from the frame $t-2$ to the frame $t-1$, to update the prior LSTM:

$$\begin{aligned} \mathbf{h}_{t-1}^f &= \mathrm{MotionEnc}(\mathbf{x}_{t-2}, \mathbf{x}_{t-1}) \\ \boldsymbol{\mu}_{\boldsymbol{\psi}_f(t)}, \boldsymbol{\sigma}_{\boldsymbol{\psi}_f(t)} &= \mathrm{LSTM}_{\boldsymbol{\psi}_f}(\mathbf{h}_{t-1}^f) \end{aligned} \tag{5}$$

At the first time-step where there is no previous motion, we assume zero-motion by estimating the motion from the previous frame to itself.

The predictor LSTMs are updated according to encoded features and sampled latent variables:

$$\begin{aligned} \mathbf{g}_t^f &= \mathrm{LSTM}_{\boldsymbol{\theta}_f}(\mathbf{h}_{t-1}^f, \mathbf{z}_t^f) \\ \boldsymbol{\mu}_{\boldsymbol{\theta}_f} &= \mathrm{FlowDec}(\mathbf{g}_t^f) \end{aligned} \tag{6}$$

There is a difference between the train time and inference time in terms of the distribution the latent variables are sampled from. At train time, latent variables are sampled from the posterior distribution. At test time, they are sampled from the posterior for the conditioning frames and from the prior for the following frames. The output of the predictor LSTMs are decoded into appearance and motion predictions separately and combined into the final prediction using the mask prediction (Eq. (1)).

4. Experiments

We evaluate the performance of the proposed approach and compare it to the previous methods on three standard video prediction datasets including Stochastic Moving MNIST, KTH Actions [24] and BAIR Robot Hand [5]. We specifically compare our baseline model (SLAMP-Baseline) and our model (SLAMP) to SVG [4] which is a special case of our baseline with a single pixel decoder, SAVP [17], SV2P [1], and lastly to SRVP [8]. We also compare our model to SVG [4] and SRVP [8] on two different challenging real world datasets, KITTI [11, 10] and Cityscapes [3], with moving background and complex object motion. We follow the evaluation setting introduced in [4] by generating 100 samples for each test sequence and report the results according to the best one in terms of average performance over the frames. Our experimental setup including training details and parameter settings can be found in Supplementary. We also share the code for reproducibility.

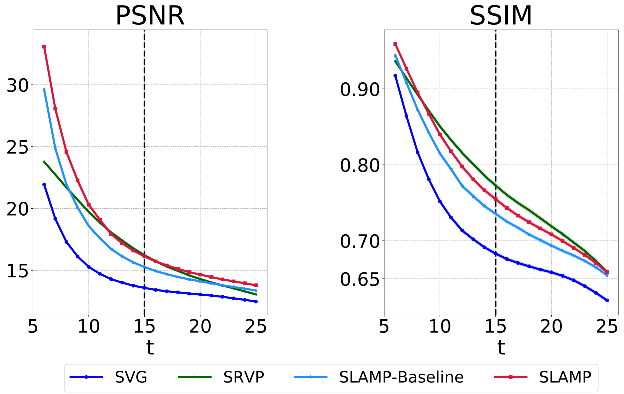

Figure 4: **Quantitative Results on MNIST.** This figure compares SLAMP to SLAMP-Baseline, SVG [4], and SRVP [8] on MNIST in terms of PSNR (**left**) and SSIM (**right**). SLAMP clearly outperforms our baseline model and SVG, and performs comparably to SRVP. Vertical bars mark the length of the training sequences.

Table 1: **FVD Scores on KTH and BAIR.** This table compares all the methods in terms of FVD scores with their 95%-confidence intervals over five different samples from the models. Our model is the second best on KTH and among top three methods on BAIR.

Dataset	KTH	BAIR
SV2P	636 ± 1	965 ± 17
SAVP	374 ± 3	**152 ± 9**
SVG	377 ± 6	255 ± 4
SRVP	**222 ± 3**	163± 4
SLAMP-Baseline	236 ± 2	245 ± 5
SLAMP	228 ± 5	—

Evaluation Metrics: We compare the performance using three frame-wise metrics and a video-level one. Peak Signal-to-Noise Ratio (PSNR), *higher better*, based on L_2 distance between the frames penalizes differences in dynamics but also favors blur predictions. Structured Similarity (SSIM),

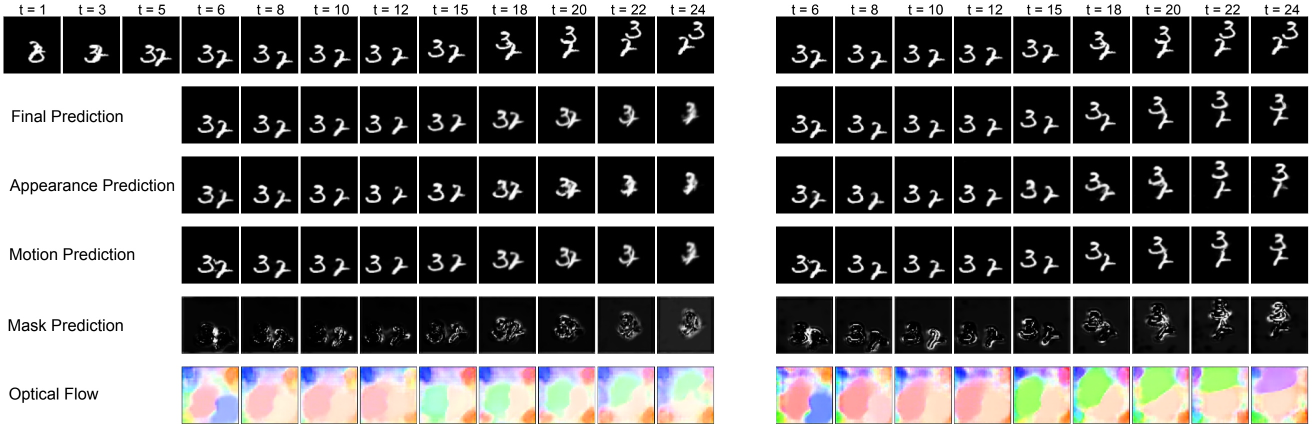

Figure 5: **SLAMP-Baseline (left) vs. SLAMP (right) on MNIST.** The top row shows the ground truth, followed by the frame predictions by the final, the appearance, the motion, and the last two rows show the mask and the optical flow predictions with false coloring. In this challenging case with bouncing and collisions, the baseline confuses the digits and cannot predict last frames correctly whereas SLAMP can generate predictions very close to the ground truth by learning smooth transitions in the motion history, as can be seen from optical flow predictions.

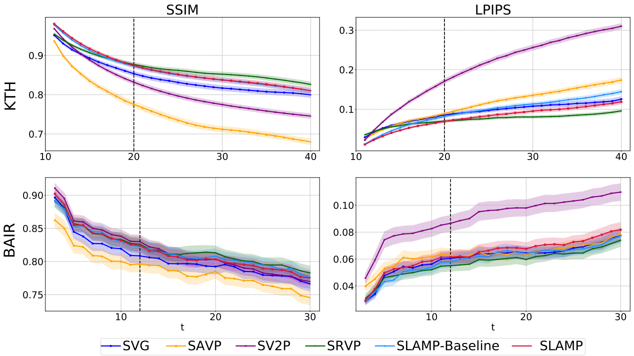

Figure 6: **Quantitative Results on KTH and BAIR.** We compare our results to previous work in terms of PSNR, SSIM, and LPIPS metrics with respect to the time steps on KTH (**top**), and BAIR (**bottom**) datasets, with 95%-confidence intervals. Vertical bars mark the length of training sequences. SLAMP outperforms previous work including SVG [4], SAVP [17], SV2P [1] and performs comparably to the state of the art method SRVP [8] on both datasets.

higher better, compares local patches to measure similarity in structure spatially. Learned Perceptual Image Patch Similarity (LPIPS) [32], *lower better*, measures the distance between learned features extracted by a CNN trained for image classification. Frechet Video Distance (FVD) [26], *lower better*, compares temporal dynamics of generated videos to the ground truth in terms of representations computed for action recognition.

Stochastic Moving MNIST: This dataset contains up to two MNIST digits moving linearly and bouncing from walls with a random velocity as introduced in [4]. Following the same training and evaluation settings as in the previous work, we condition on the first 5 frames during training and learn to predict the next 10 frames. During testing, we again condition on the first 5 frames but predict the next 20 frames.

Fig. 4 shows quantitative results on MNIST in comparison to SVG [4] and SRVP [8] in terms of PSNR and SSIM, omitting LPIPS as in SRVP. Our baseline model with a motion decoder (SLAMP-Baseline) already outperforms SVG on both metrics. SLAMP further improves the results by utilizing the motion history and reaches a comparable performance to the state of the art model SRVP. This shows the benefit of separating the video into static and dynamic parts in both state-space models (SRVP) and auto-regressive models (ours, SLAMP). This way, models can better handle challenging cases such as crossing digits as shown next.

We qualitatively compare SLAMP to SLAMP-Baseline on MNIST in Fig. 5. The figure shows predictions of static and dynamic parts as appearance and motion predictions, as well the final prediction as the combination of the two. According to the mask prediction, the final prediction mostly relies on the dynamic part shown as black on the mask and uses the static component only near the motion boundaries. Moreover, optical flow prediction does not fit the shape of the digits but expands as a region until touching the motion region of the other digit. This is due to the uniform black background. Moving a black pixel in the background randomly is very likely to result in another black pixel in the background, which means zero-loss for the warping result. Both models can predict optical flow correctly for the most part and resort to the appearance result in the occluded regions. However, continuity in motion is better captured by SLAMP with the colliding digits whereas the baseline model cannot recover from it, leading to blur results, far from the ground truth. Note that we pick the best sample for both

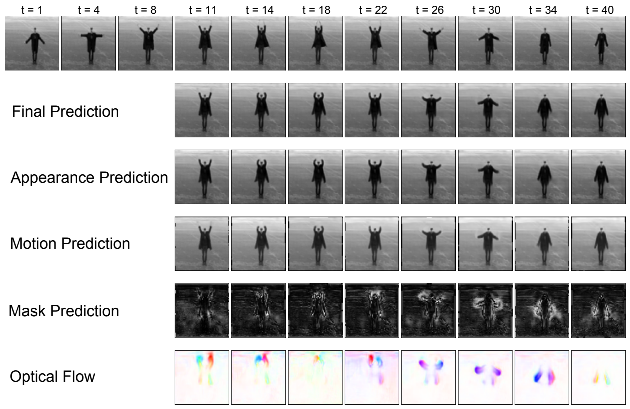

Figure 7: **Qualitative Results on KTH** We visualize the results of SLAMP on KTH dataset. The top row shows the ground truth, followed by the frame predictions by the final, the appearance, the motion, and the last two rows show the mask and the optical flow predictions. The mask prediction combines the appearance prediction (white) and the motion prediction (black) into the final prediction.

models among 100 samples according to LPIPS.

KTH Action Dataset: KTH dataset contains real videos where people perform a single action such as walking, running, boxing, etc. in front of a static camera [24]. We expect our model with motion history to perform very well by exploiting regularity in human actions on KTH. Following the same training and evaluation settings used in the previous work, we condition on the first 10 frames and learn to predict the next 10 frames. During testing, we again condition on the first 10 frames but predict the next 30 frames.

Fig. 6 and Table 1 show quantitative results on KTH in comparison to previous approaches. Both our baseline and SLAMP models outperform previous approaches and perform comparably to SRVP, in all metrics including FVD. A detailed visualization of all three frame predictions as well as flow and mask are shown in Fig. 7. Flow predictions are much more fine-grained than MNIST by capturing fast motion of small objects such as hands or thin objects such as legs (see Supplementary). The mask decoder learns to identify regions around the motion boundaries which cannot be matched with flow due to occlusions and assigns more weight to the appearance prediction in these regions.

On KTH, the subject might appear after the conditioning frames. These challenging cases can be problematic for some previous work as shown in SRVP [8]. Our model can generate samples close to the ground truth despite very little information on the conditioning frames as shown in Fig. 8. The figure shows the best sample in terms of LPIPS, please see Supplementary for a diverse set of samples with subjects of various poses appearing at different time steps.

Table 2: **Results with a Moving Background.** We evaluate our model SLAMP in comparison to SVG and SRVP on KITTI [10] and Cityscapes [3] datasets by conditioning on 10 frames and predicting 20 frames into the future.

Models	PSNR (↑)	SSIM (↑)	LPIPS (↓)
SVG [4]	12.70 ± 0.70	0.329 ± 0.030	0.594 ± 0.034
SRVP [8]	13.41 ± 0.42	0.336 ± 0.034	0.635 ± 0.021
SLAMP	**13.46 ± 0.74**	**0.337 ± 0.034**	**0.537 ± 0.042**

KITTI [11, 10]

Models	PSNR (↑)	SSIM (↑)	LPIPS (↓)
SVG [4]	20.42 ± 0.63	0.606 ± 0.023	0.340 ± 0.022
SRVP [8]	20.97 ± 0.43	0.603 ± 0.016	0.447 ± 0.014
SLAMP	**21.73 ± 0.76**	**0.649 ± 0.025**	**0.2941 ± 0.022**

Cityscapes [3]

BAIR Robot Hand: This dataset contains videos of a robot hand moving and pushing objects on a table [5]. Due to uncertainty in the movements of the robot arm, BAIR is a standard dataset for evaluating stochastic video prediction models. Following the training and evaluation settings used in the previous work, we condition on the first 2 frames and learn to predict the next 10 frames. During testing, we again condition on the first 2 frames but predict the next 28 frames.

We show quantitative results on BAIR in Fig. 6 and Table 1. Our baseline model achieves comparable results to SRVP, outperforming other methods in all metrics except SV2P [1] in PSNR and SAVP [17] in FVD. With 2 conditioning frames only, SLAMP cannot utilize the motion history and performs similarly to the baseline model on BAIR (see Supplementary). This is simply due to the fact that there is only one flow field to condition on, in other words, no motion history. Therefore, we only show the results of the baseline model on this dataset.

Real-World Driving Datasets: We perform experiments on two challenging autonomous driving datasets: KITTI [11, 10] and Cityscapes [3] with various challenges. Both datasets contain everyday real-world scenes with complex dynamics due to both background and foreground motion. KITTI is recorded in one town in Germany while Cityscapes is recorded in 50 European cities, leading to higher diversity.

Cityscapes primarily focuses on semantic understanding of urban street scenes, therefore contains a larger number of dynamic foreground objects compared to KITTI. However, motion lengths are larger on KITTI due to lower frame-rate. On both datasets, we condition on 10 frames and predict 10 frames into the future to train our models. Then at test time, we predict 20 frames conditioned on 10 frames.

As shown in Table 2, SLAMP outperforms both methods on all of the metrics on both datasets, which shows its ability to generalize to the sequences with moving background. Even SVG [4] performs better than the state of

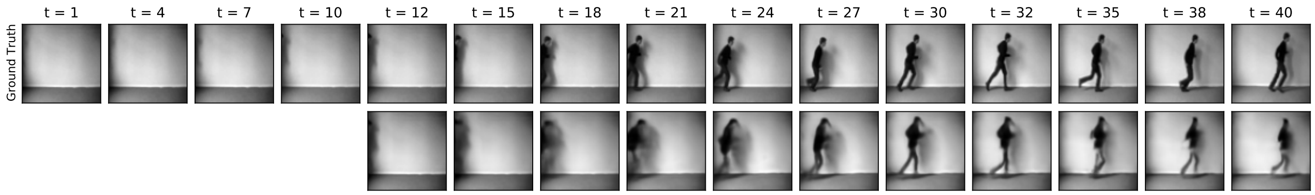

Figure 8: **Subject Appearing after the Conditioning Frames.** This figure shows a case where the subject appears after conditioning frames on KTH with ground truth (**top**) and a generated sample by our model (**bottom**). This shows our model's ability to capture dynamics of the dataset by generating samples close to the ground truth, even conditioned on empty frames.

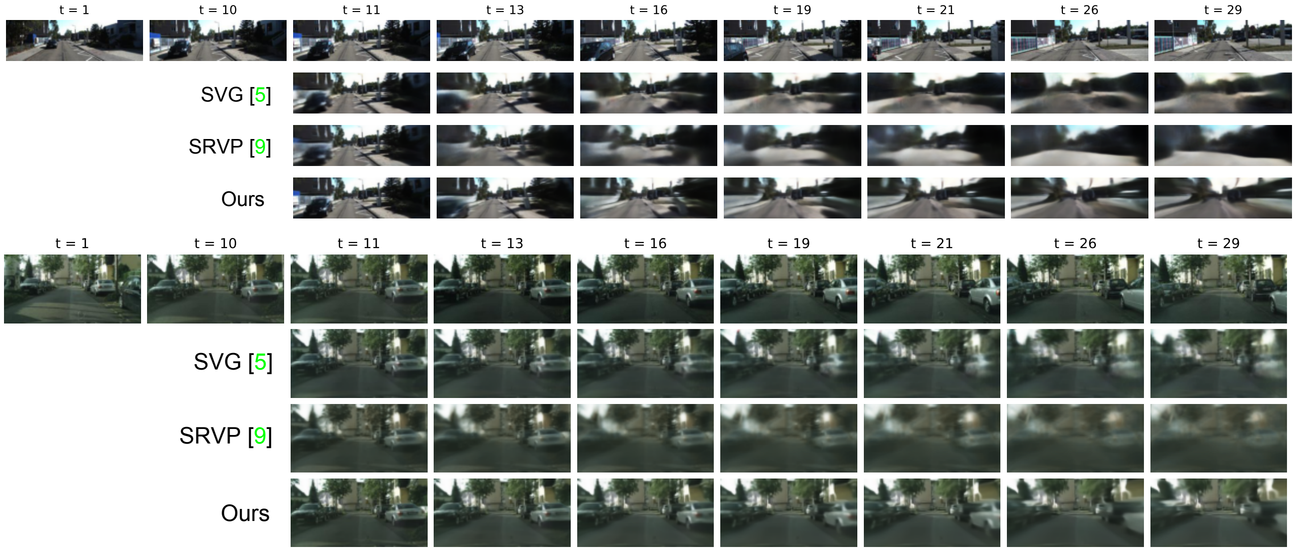

Figure 9: **Qualitative Comparison.** We compare SLAMP to SVG [4] and SRVP [8] on KITTI (**top**) and Cityscapes (**bottom**). Our model can better capture the changes due to ego-motion thanks to explicit modeling of motion history.

the art SRVP [8] in LPIPS metric for KITTI and on both SSIM and LPIPS for Cityscapes, which shows the limitations of SRVP on scenes with dynamic backgrounds. We also perform a qualitative comparison to these methods in Fig. 1 and Fig. 9. SLAMP can better preserve the scene structure thanks to explicit modeling of ego-motion history in the background.

Visualization of Latent Space: We visualize stochastic latent variables of the dynamic component on KTH compared to the static and SVG. (see Supplementary.)

5. Conclusion

We presented a stochastic video prediction framework to decompose video content into appearance and dynamic components. Our baseline model with deterministic motion and mask decoders outperforms SVG, which is a special case of our baseline model. Our model with motion history, SLAMP, further improves the results and reaches the performance of the state of the art method SRVP on the previously used datasets. Moreover, it outperforms both SVG and SRVP on two real-world autonomous driving datasets with dynamic background and complex motion. We show that motion history enriches model's capacity to predict future, leading to better predictions in challenging cases.

Our model with motion history cannot realize its full potential in standard settings of stochastic video prediction datasets. A fair comparison is not possible on BAIR due to the little number of conditioning frames. BAIR holds a great promise with changing background but infrequent, small changes are not reflected in current evaluation metrics.

An interesting direction is stochastic motion decomposition, maybe with hierarchical latent variables, for modelling camera motion and motion of each object in the scene separately.

Acknowledgements. We would like to thank Jean-Yves Franceschi and Edouard Delasalles for providing technical and numerical details for the baseline performances; Deniz Yuret and Salih Karagoz for helpful discussions and comments. K. Akan was supported by KUIS AI Center fellowship, F. Güney by TUBITAK 2232 International Fellowship for Outstanding Researchers Programme, E. Erdem in part by GEBIP 2018 Award of the Turkish Academy of Sciences, A. Erdem by BAGEP 2021 Award of the Science Academy.

References

[1] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018. 2, 5, 6, 7

[2] Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 5, 7

[4] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *Proc. of the International Conf. on Machine learning (ICML)*, 2018. 1, 2, 3, 4, 5, 6, 7, 8

[5] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, 2017. 5, 7

[6] Hehe Fan, Linchao Zhu, and Yi Yang. Cubic lstms for video prediction. In *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 2019. 1, 2

[7] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 1, 2

[8] Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. In *Proc. of the International Conf. on Machine learning (ICML)*, 2020. 1, 2, 3, 5, 6, 7, 8

[9] Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, and Trevor Darrell. Disentangling propagation and generation for video prediction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 1, 2

[10] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 2013. 1, 5, 7

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5, 7

[12] Karol Gregor and Frederic Besse. Temporal difference variational auto-encoder. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018. 3

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 3

[14] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 1, 2

[15] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, jun 1973. 1

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2014. 4

[17] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv.org*, 2018. 2, 5, 6, 7

[18] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 600–615, 2018. 2

[19] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P. Xing. Dual motion gan for future-flow embedded video prediction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2

[20] Ziwei Liu, Raymond A. Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 1, 2

[21] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2017. 1, 2

[22] Chaochao Lu, Michael Hirsch, and Bernhard Scholkopf. Flexible spatio-temporal networks for video prediction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2

[23] 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 (NeurIPS)*, 2019. 2

[24] Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004. 5, 7

[25] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 2

[26] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv.org*, 2019. 6

[27] Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[28] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[29] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 2

[30] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2015. 1, 2

[31] Tianfan Xue, Jiajun Wu, Katherine L Bouman, and William T Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances In Neural Information Processing Systems*, 2016. 2

[32] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 6