Controllable Video Generation with Text-based Instructions

Ali Köksal Γ Kenan E. Ak Ying Sun Deepu Rajan Joo Hwee Lim

***Abstract*—Most of the existing studies on controllable video generation either transfer disentangled motion to an appearance without detailed control over motion or generate videos of simple actions such as the movement of arbitrary objects conditioned on a control signal from users. In this study, we introduce Controllable Video Generation with text-based Instructions (CVGI) framework that allows text-based control over action performed on a video. CVGI generates videos where hands interact with objects to perform the desired action by generating hand motions with detailed control through text-based instruction from users. By incorporating the motion estimation layer, we divide the task into two sub-tasks: (1) control signal estimation and (2) action generation. In control signal estimation, an encoder models actions as a set of simple motions by estimating low-level control signals for text-based instructions with given initial frames. In action generation, generative adversarial networks (GANs) generate realistic hand-based action videos as a combination of hand motions conditioned on the estimated low control level signal. Evaluations on several datasets (EPIC-Kitchens-55, BAIR robot pushing, and Atari Breakout) show the effectiveness of CVGI in generating realistic videos and in the control over actions.**

***Index Terms*—Controllable video generation, video generation with textual instructions, motion generation, conditional generative models**

I. Introduction

Deep architectural models such as convolutional neural networks (CNNs) and generative adversarial networks (GANs) enable the generation of high-dimensional data such as images, [1]–[6] and videos [7]–[12]. These models can manipulate the given high-dimensional data conditioned on the desired manipulation. For example, image manipulation and editing architectures [13]–[16] allow users to transfer the style from another image.

Motion manipulation according to text-based instructions on a video where human interacts with objects in a complex scene is indeed extremely more challenging, as there is no simple way to model the interaction. Besides, building a semantic association between instructions and motion is also challenging because text descriptions are often ambiguous for controllable video generation. In the literature, there exist video manipulation architectures such as [12], [17]–[20] that allow users to manipulate motion of objects on a video. They can be grouped into two groups according to the source of manipulation. Most of the existing approaches in the first group use driving videos as a source of manipulation by extracting actions. They can disentangle motion and transfer it to another appearance but they are limited to detailed control over the motion during the generation [9], [21]–[25]. In the second group, existing approaches use control signals that are received from an agent such as mouse click [26], key stroke [8], [27], joystick [28] but in most of them only video generation of simple actions that can be defined as displacement-based actions such as moving arbitrary objects is possible. On the other hand, the proposed framework allows detailed control over motions of generated videos and it can generate complex actions as a combination of simple motions.

With the motivation of building an association between text-based institutions and motions to manipulate the features of the generated motion such as direction, speed, the target, ..., this paper introduces a novel framework, named CVGI, that allows users to manipulate simple human-object interactions such as hand/s going toward the desired object in videos with complex scenes by conditioning through text-based instructions. CVGI receives a text-based instruction from a user and takes an initial frame as input to generate a video sequence that corresponds well with the user input. For example, Figure 1 shows that CVGI can reconstruct the ground truth video by using the same text-based instruction as the instruction of ground truth. It can also generate novel videos with different text-based instructions. As shown, the generated videos are photo-realistic and correspond well with the text-based instructions. CVGI divides the task into two sub-tasks by incorporating the motion estimation layer as seen in Figure 2. The first sub-task, control signal estimation, encodes high-level text-based instructions to low-level control signals as a form of motion representation. The control signal encoder takes instruction and an initial frame to estimate a set of low-level control signals for motions on the future frames. Low-level signals define the location change of the object of interest between two consecutive frames such as displacement center of mass of hand masks and displacement of the robot arm's gripper. The second sub-task, action generation, generates realistic videos frame-by-frame in a loop conditioned on low-level signals. First, it generates the next frame with the initial frame and the first estimated low-level signal. Then it takes the generated frame and the second estimated low-level signal to generate the third frame and so on until n frames are generated for all estimated low-level signals. EPIC-Kitchens-55 dataset [29] contains egocentric videos shot by a head-mounted camera.

Köksal A., Ak K.E., Sun Y., Lim J.H are with Institute for Infocomm Research (I^2R), A*STAR, Singapore (Email: {koksal_ali, kenan_emir_ak, suny, joohwee}@i2r.a-star.edu.sg)

Köksal A. and Rajan D. are with School of Computer Science and Engineering (SCSE), Nanyang Technological University (NTU), Singapore (Email: ali013@e.ntu.edu.sg, asdrajan@ntu.edu.sg)

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes details of the architectures and additional qualitative results.

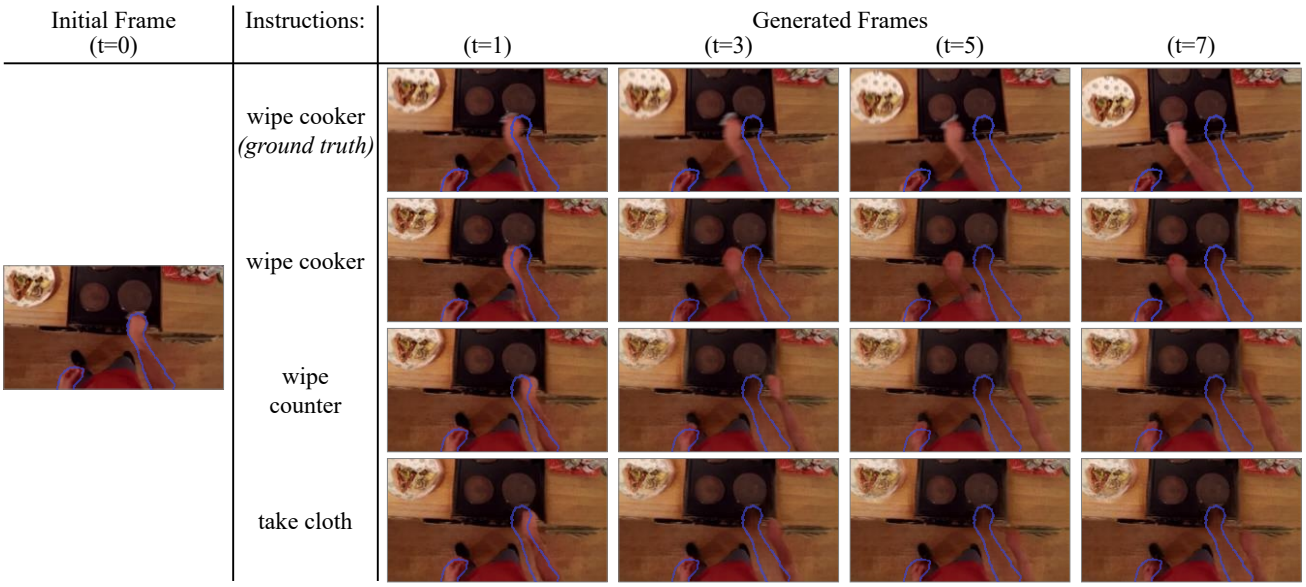


Fig. 1: The proposed framework, named CVGI, generates controllable videos conditioned on text-based instructions received from a user. CVGI generates novel photo-realistic videos from an initial frame and textual instructions. Generated frames are from top: ground truth, duplication of ground truth, and two novel videos. In the duplication of ground truth, CVGI generates hands at similar positions to the ground truth. In the novel videos, CVGI generates videos with different hand movements based on textual instruction. Note that the boundary of the hand masks of the initial frame is indicated as blue in the generated frames to highlight the difference in hand movements.

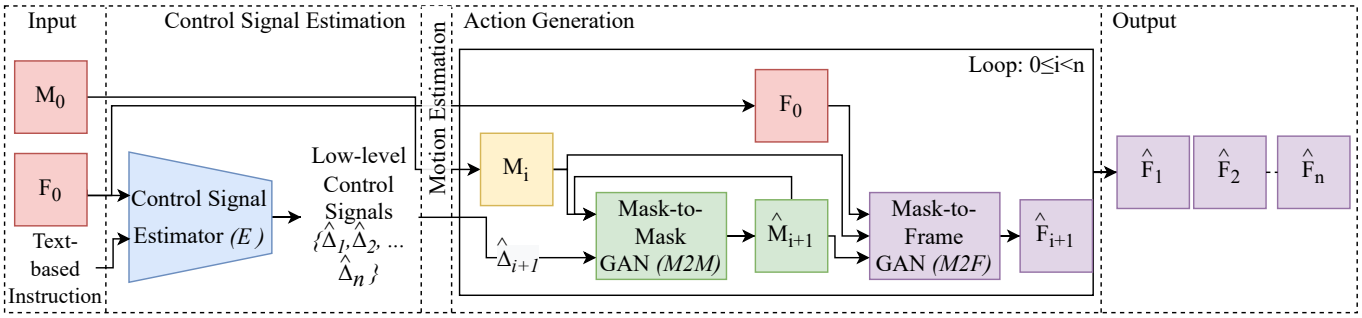


Fig. 2: CVGI divides the task into two sub-tasks: control signal estimation and action generation. In the first sub-task, an encoder estimates low-level control signals with a given initial frame conditioned on text-based instructions. In the second sub-task, two GANs (M2M and M2F) are trained one after another in a loop. M2M is trained to control motion and generates the next mask of the object of interest conditioned on the estimated control signal. M2F is trained to perform motion-aware mask-to-frame translation and generates realistic frames with given masks.

Egocentric videos capture useful visual input for the understanding of the person's activities because scenes of egocentric videos approximate the vision of the person who wears the camera. In spite of useful scenes, wearable cameras lead to a lot of motion and dynamic scenes in egocentric videos. In dynamic scenes, it is hard to focus on the object of interest and represent its meaningful movement and interaction with low-level control signals. Consequently, motion generation in egocentric videos is indeed extremely challenging. Moreover, to the best of our knowledge, CVGI is the first study that can generate controllable egocentric videos where the motion of the object of interest is controlled in detail. Therefore, CVGI incorporates the hand masks to differentiate the object of interest to control and the action generator synthesizes video sequences by employing two GANs: mask-to-mask (M2M) and mask-to-frame (M2F) for egocentric videos. M2M models the association between motions of masks of the object of interest, i.e., hand masks and low-level signals and it is trained to synthesize masks that correspond with low-level signals. It takes masks of input frames and generates masks for the next frames conditioned on the low-level signal. M2F GAN maps masks to frames by being aware of hands' motion to generate realistic frames. For BAIR robot pushing [30] and Atari Breakout datasets [8], we employ a single GAN similar to M2M.

The contributions of this work can be summarized in three aspects: **(1)** We propose CVGI which generates novel human-object interaction videos where hand/s go toward the desired

object by manipulating motions on complex scenes conditioned on text-based instructions from users. **(2)** We overcome the challenge with two innovations. Control signal estimation models human-object interaction in terms of motions and builds the association between text-based instructions and low-level control signals. Action generation models motions with low-level signals and generates realistic videos. **(3)** With the evaluations on three public datasets: EPIC-Kitchens-55 [29], BAIR robot pushing [30], and Atari Breakout [8], we demonstrate that CVGI generates photo-realistic videos that correspond well with instructions.

II. Related Work

In the literature, many studies for novel video generation exist they can be grouped into two according to the source of manipulation. The first group learns a motion that is extracted from a video and transfers the motion to another object/subject. The second group receives user input to control the motion and generates novel videos that correspond well with user input. Another important difference between the groups other than the source of manipulation is the ability of the detailed control over the synthesized motion that can be defined as explicit control. Although studies in the first group can control the motion, they are limited to explicitly controlling the synthesized motion as they transfer the motion that is learned from a source video as it is. On the other hand, studies in the second group learn to associate motions and user input and can manipulate the motion according to user input.

Most existing studies that transfer motions learn content which is the object and its appearance and motion which is the dynamics of content and they generate videos of moving faces, human body, and arbitrary objects. [25] proposes a framework that generates controllable videos for human faces conditioned on a driving vector that can be extracted from a given video, audio, or pose vector. [24] builds separate latent spaces for content and motion of videos for the generation of novel videos of faces, human body, and artificial objects by controlling content and motion latent vectors. [31] and [32] predict future frames with a given input frame conditioned on estimated future human body poses. [19] introduces a framework that animates an arbitrary object on the given image conditioned on a motion that is derived from a driving video sequence by using sparse keypoint trajectories. In [33], a similar framework addresses the same problem without using any annotation. [23] distinguishes the appearance and pose of humans and generates images with given appearance with different poses. Similar to [23], [22] generates controllable human behavior by transferring motion that is rendered over keypoints of the human body. [21] generates dance videos by transferring motion with a network that is trained to translate pose to appearance and vice versa. [9] introduces a framework that includes a generator and multiple discriminators disentangles actions and objects of a given video. It generates human-object interaction videos based on text description by replacing objects to generate novel videos. The aforementioned studies disentangle content and motion of videos and transfer motions to generate novel videos of contents. On the other hand, our framework controls the motion of objects according to user input. It generates novel videos by manipulating a given frame conditioned on control signals that are estimated from user input.

Most of the studies that receive user input as the source of manipulation generate novel videos by synthesizing simple motions as depicting desired motion on the generated frames. [34] introduces a framework that generates action conditional videos in Atari games by predicting future frames with given previous frames and an action label for player actions. [35] generates action-conditioned videos for robotic arm actions by predicting future frames in long-range from previous frames. [18] introduces a framework that generates variable-length videos for artificial or arbitrary objects conditioned on captions by separately learning short-term and long-term context. [17] generates controllable videos with a given frame conditioned on sparse trajectories specified by users and improves the video quality by hallucinating pixels that cannot be copied based-on flow. [28] extracts a character from a given video, and generates videos of the extracted character performing motions that are controllable with low-level signals received from an agent on any background. The framework has two modules, first generates poses corresponding with signals from an agent such as a joystick, second translates poses to frames. [27] is trained to imitate the game engines and renders the next screen conditioned on keystrokes by users. [8] learns actions in an unsupervised manner to cluster motions and then generates videos of discrete actions with a given initial frame conditioned on keystrokes from users. [26] proposes a framework that generates videos where the motion of specific objects can be controlled through mouse clicks. It receives an input frame, its segmentation map, and mouse click and incorporates a graph convolution network to model the motions of objects. As summarized above, most existing studies generate controllable videos with low-level signals received by an agent such as keyboard, joystick, and mouse. On the other hand, our framework builds a semantic association between text-based instructions and motions. This association allows controlling generated videos according to text-based input that can describe complex actions such as human-object interaction.

In addition to the above-mentioned video generation frameworks, there exist recent studies that generate videos based on text. [12] proposes a generic solution for various visual generation tasks. Its generic model can also generate videos based on text. [36] proposes a framework, CogVideo, for text-to-video generation. CogVideo includes a transformer-based architecture and uses a pretrained text-to-image model. Similarly, [37] employs bi-directional masked transformers-based model. It is trained with a large amount of text-image pairs and a smaller amount of video-text pairs. [38] proposes an efficient video generation model. It is a GAN-based model and can control the generated videos by conditioned on the discrete label of the category. Likewise, [39] is a conditional model, but it is based on diffusion architecture instead of GAN. Similarly, [40] employs a diffusion model. The proposed text-to-video model leverages a text-to-image model. The summarized video generation models can generate videos that are

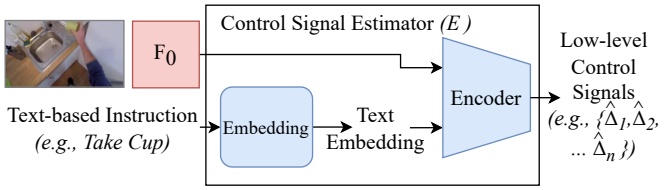


Fig. 3: Control signal estimator includes an embedding layer and a CNN-based encoder. It takes the initial frame and text-based instructions and estimates low-level control signals for the object of interest.

aligned with the given text or condition, but they are limited to explicit control over the motion. For example, [40] can generate a flying dog based on text, but it cannot manipulate the generated video by controlling the motion such as changing direction. Furthermore, they are similar to the second group in terms of the source of manipulation and similar to the first group in terms of the ability to explicitly control the generated motion. On the other hand, our framework can control the motion explicitly with text-based instructions.

III. Approach

CVGI learns to generate realistic and temporally consistent videos by manipulating motion on complex scenes of egocentric videos according to the given text-based instructions. Figure 2 shows the overall flow of the framework. CVGI takes an initial frame F_0 as context image, mask of the object of interest M_0 such as the mask of hands, instruction d and it generates n next frames $\hat{F}_1, \hat{F}_2, ..., \hat{F}_n$ that corresponds to the given instruction. We divide the task into two sub-tasks: control signal estimation and action generation. Control signal estimator builds association between text-based instructions and low-level control signals $\hat{\Delta}_1, \hat{\Delta}_2, ..., \hat{\Delta}_n$. For egocentric videos such as videos on EPIC-Kitchens-55, the action generator which consists of two GANs (M2M and M2F) generates frames according to the control signal in a loop. First, it generates future masks and then translates them to frames one by one. Note that, for BAIR Robot pushing and Atari Breakout datasets, the action generator consists of one GAN similar to M2M GAN with two differences. First, it directly takes the initial frame and generates the next frame conditioned on low-level signals. Second, its generator's image loss is changed to L2 norm instead of L1 norm to improve the visual quality.

A. Control Signal Estimator

As illustrated in Figure 3, control signal estimator E converts high-level text-based instructions that describe the actions to a set of low-level control signals. It takes initial frames as context images along with instructions to predict motion for the next frames. It contains an embedding layer and a CNN-based encoder. The embedding layer takes a textual instruction d and computes text embedding. The encoder conditioned on text embedding predicts a set of low-level control signal (displacement) $\hat{\Delta}_1, \hat{\Delta}_2, ..., \hat{\Delta}_n$ for the object that is desired to perform the motion with the given initial frame F_0. They are trained to minimize mean square error (MSE) that is computed between the ground truth control signals $\Delta_1, \Delta_2, ..., \Delta_n$ and estimated control signals $\hat{\Delta}_1, \hat{\Delta}_2, ..., \hat{\Delta}_n$ as follows.

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \left\| \Delta_i - \hat{\Delta}_i \right\|_2 , \text{ where } \hat{\Delta}_{1,2,...,n} = E(F_0, d). \quad (1)$$

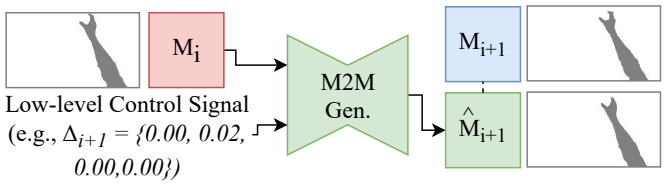
B. Action Generator

The action generator aims to manipulate the motion of the object according to low-level signals. It employs two GANs: mask-to-mask (M2M) and mask-to-frame (M2F) which are trained in a loop. M2M GAN synthesises the motion over masks and M2F GAN is responsible for motion-aware photo-realistic frame synthesis.

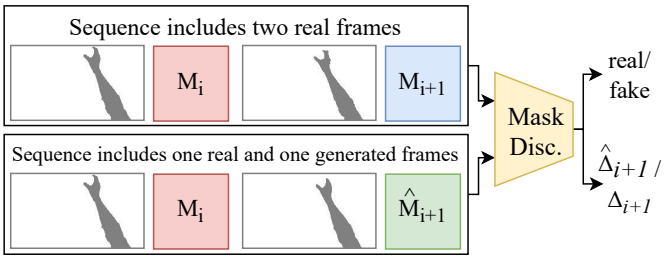
Mask-to-mask GAN employs a conditional generator G_{mask} and a sequence discriminator D_{mask}. M2M GAN uses a continuous signal as a condition in G_{mask} and ground truth in D_{mask}, unlike most existing conditional GANs that usually use discrete signals such as the label of categories. Using continuous signals improves the ability to control motion and it is elaborated in the ablation study. G_{mask} is an encoder-decoder-based generator that takes a single mask and is trained to generate another mask conditioned on the low-level control signal. During training, it is trained in both forward and backward directions by changing the order of the frames to capture more variability. Thus, training with both forward and backward passes increases the variation of training samples and is able to learn a motion together with its reverse. Besides, it improves the understanding of motion's direction. Figure 4 shows only toward forward for sake of visual simplicity. As seen in Figure 4 (a), G_{mask} is trained to generate the next mask from the initial mask conditioned on the corresponding low-level control signal, i.e., $G_{mask}(M_i, \Delta_{i+1}) \rightarrow \hat{M}_{i+1}$. For backward training, the initial frame is generated from the next frame conditioned on the inverse of the control signal (negative displacement), i.e., $G_{mask}(M_{i+1}, (\Delta_{i+1})^{-1}) \rightarrow \tilde{M}_i$. As seen in Figure 4 (b), D_{mask} takes an input of two consecutive frames that are concatenated to train to distinguish real and fake sequences. Over D_{mask}, we introduce an auxiliary regressor to distinguish the motion of objects of interest on given consecutive frames. Thus, D_{mask} is a sequence discriminator with two heads, i.e., $D_{mask}(M_i, M_{i+1}) \rightarrow \{D^{adv}_{mask}(M_i, M_{i+1}), D^{reg}_{mask}(M_i, M_{i+1})\}$. The first is an adversarial head that distinguishes sequences of two frames as real and fake and provides adversarial training. The second is a regressor head and measures the displacement of the objects of interest on the given two frames.

During training, for each consecutive mask and the low-level control signal (displacement) $\{\{M_i, M_{i+1}\}, \Delta_{i+1}\}$, G_{mask} and D_{mask} are trained in an adversarial manner where generated masks towards forward $\hat{M}_{i+1}$ and backward $\tilde{M}_i$ are denoted as follows:

$$\begin{aligned} \hat{M}_{i+1} &= \mathop{\mathbb{E}}_{M_i, \Delta_{i+1}} G_{mask}(M_i, \Delta_{i+1}), \\ \tilde{M}_i &= \mathop{\mathbb{E}}_{M_{i+1}, \Delta_{i+1}} G_{mask}(M_{i+1}, (\Delta_{i+1})^{-1}). \end{aligned} \quad (2)$$



(a) Forward training of G_{mask}



(b) Forward training of D_{mask}

Fig. 4: Forward training of M2M GAN that includes (a) a conditional generator G_{mask} and (b) a sequence discriminator D_{mask}. G_{mask} is trained to generate masks conditioned on low-level control signals which contain four floating-point numbers. The first two control the right hand and the next two control the left hand. D_{mask} is a sequence discriminator with two heads. The first head is trained to distinguish real and fake sequences. The second head is trained to estimate the displacement of objects of interest.

G_{mask} learns to construct the next mask conditioned on ground truth control signals and given mask by minimizing the following image loss function:

$$\mathcal{L}_{img} = \|M_{i+1} - \hat{M}_{i+1}\|_1 + \|M_i - \tilde{M}_i\|_1. \tag{3}$$

In addition, least squared adversarial loss [41] that is computed by D_{mask}'s adversarial head is employed as follows to make generated masks indistinguishable from real masks:

$$\mathcal{L}^G_{adv} = (D^{adv}_{mask}(M_i, \hat{M}_{i+1}))^2 + (D^{adv}_{mask}(\tilde{M}_i, M_{i+1}))^2. \tag{4}$$

Besides, the following regression loss (MSE) is computed between ground truth control signals and estimated control signals by the auxiliary regressor. This is so as to enforce generated masks correspond well with low-level control signals

$$\begin{aligned}\mathcal{L}^G_{cs} =& \|D^{reg}_{mask}(M_i, \hat{M}_{i+1}) - \Delta_{i+1})\|_2 \\ &+ \|D^{reg}_{mask}(\tilde{M}_i, M_{i+1}) - \Delta_{i+1})\|_2.\end{aligned} \tag{5}$$

The full objective function of G_{mask} is formulated as follows:

$$\mathcal{L}^G_{mask} = \lambda_{img}\mathcal{L}_{img} + \lambda_{adv}\mathcal{L}^G_{adv} + \lambda_{cs}\mathcal{L}^G_{cs}\ , \tag{6}$$

where λ_{img}, λ_{adv}, and λ_{cs} are positive weights to balance loss functions and their default values are 1.0, 1.0, and 10.0, respectively.

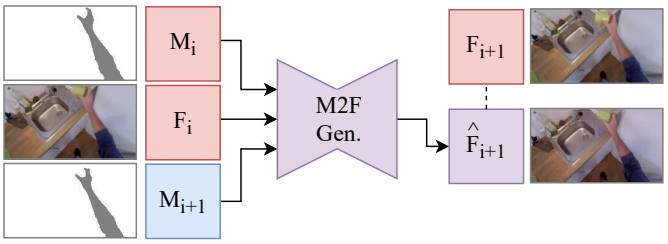


Fig. 5: Forward training of G_{frame}. G_{frame} is trained to generate the next frame by taking the initial frame, its mask, and the mask of the next frame.

D_{mask} is trained to minimize the following least squared adversarial loss to distinguish real and fake sequences:

$$\begin{aligned}\mathcal{L}^D_{adv} =& (D^{adv}_{mask}(M_i, M_{i+1}))^2 \\ &+ \frac{1}{2}\left[(D^{adv}_{mask}(M_i, \hat{M}_{i+1}) - 1)^2\right] \\ &+ \frac{1}{2}\left[(D^{adv}_{mask}(\tilde{M}_i, M_{i+1}) - 1)^2\right].\end{aligned} \tag{7}$$

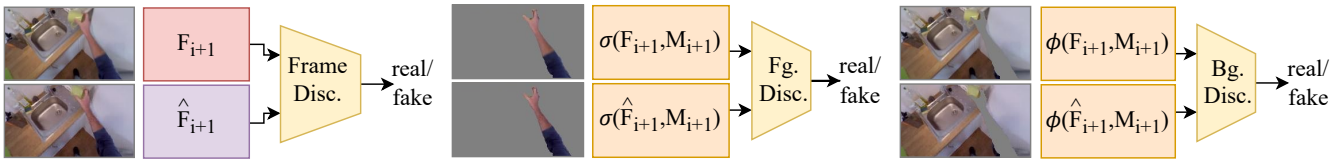
In addition, D_{mask} learns to predict the displacement of the object by minimizing the following loss:

$$\mathcal{L}^D_{cs} = \|D^{reg}_{mask}(M_i, M_{i+1}) - \Delta_{i+1}\|_2. \tag{8}$$

The full objective function to optimize D_{mask} is defined as follows:

$$\mathcal{L}^D_{mask} = \mathcal{L}^D_{adv} + \mathcal{L}^D_{cs}. \tag{9}$$

Mask-to-frame GAN employs a generator G_{frame} and three frame-based discriminators $D_{frame}, D_{fg}, D_{bg}$ for motion-aware mask-to-frame translation. M2F GAN is only employed for EPIC-Kitchens-55 because videos in this dataset are egocentric shot by a head-mounted camera. In egocentric videos, every object appears moving in most of the frames due to the camera motion. It may cause the generator to confuse which object is the object of interest. Therefore, controllable video generation in egocentric videos requires indicating the object of interest. For this reason, CVGI uses hand masks. After we achieve controlling hand motions with masks, we employ M2F GAN to translate masks to frames. As shown in Figure 5, G_{frame} takes a context frame (the initial frame), the corresponding hand mask, and the mask for the next frame that indicate hands' new location. It is trained to hallucinate pixels at the hands' location on the context images to remove them and create hands at the new location. Similar to G_{mask}, G_{frame} is trained in the forward and backward directions by changing inputs. Figures 5 and 6 show only forward training for simplicity. For forward training, G_{frame} is trained to generate the next frame from initial frame, its mask, and the mask of the next frame, $G_{frame}(M_i, F_i, M_{i+1}) \rightarrow \hat{F}_{i+1}$. For backward training, it is trained to generate the initial frame from the next frame, its mask, and the mask of the initial frame, $G_{frame}(M_{i+1}, F_{i+1}, M_i) \rightarrow \tilde{F}_i$. As seen in Figure 6, three frame-based discriminators are employed. D_{frame} is trained to distinguish real and fake frames, D_{fg} takes frames where the background is masked to distinguish real and fake objects of interest, and D_{bg} takes frames where the foreground is masked to distinguish real and fake background.



(a) Forward training of D_{frame} (b) Forward training of D_{fg} (c) Forward training of G_{bg}

Fig. 6: Forward training of discriminators of M2F GAN. D_{frame} (a) is trained to distinguish real and fake frames, D_{fg} (b) is trained to distinguish real and fake objects of interest, and D_{bg} (c) is trained to distinguish real and fake background. σ and ϕ denote operations to compute hand frames and background frames, respectively.

G_{frame} and D_{frame} are trained with consecutive frames and their corresponding masks $\{\{F_i, F_{i+1}\}, \{M_i, M_{i+1}\}\}$ where generated frames towards forward $\hat{F}_{i+1}$ and backward $\tilde{F}_i$ are denoted as follows:

$$\begin{aligned} \hat{F}_{i+1} &= \mathop{\mathbb{E}}_{M_i, F_i, M_{i+1}} G_{frame}(M_i, F_i, M_{i+1}), \\ \tilde{F}_i &= \mathop{\mathbb{E}}_{M_{i+1}, F_{i+1}, M_i} G_{frame}(M_{i+1}, F_{i+1}, M_i). \end{aligned} \tag{10}$$

G_{frame} is trained with an image loss $\mathcal{L}_{img}$ and least squared adversarial losses $\mathcal{L}^G_{frame}, \mathcal{L}^G_{fg}, \mathcal{L}^G_{bg}$ which are defined in Equation 11. $\mathcal{L}^G_{frame}$ by D_{frame} leads to generating indistinguishable frames from real frames. $\mathcal{L}^G_{fg}$ by D_{fg} is trained with hand frames where the background is masked out and leads to producing realistic hands at the new position. $\mathcal{L}^G_{bg}$ by D_{bg} is trained with background frames where hands are masked out and leads to hallucinating pixels at the hands' previous location.

$$\begin{aligned} \mathcal{L}_{img} &= \|F_{i+1} - \hat{F}_{i+1}\|_2 + \|F_i - \tilde{F}_i\|_2, \\ \mathcal{L}^G_{frame} &= (D_{frame}(\hat{F}_{i+1}))^2 + (D_{frame}(\tilde{F}_i))^2, \\ \mathcal{L}^G_{fg} &= (D_{fg}(\sigma(\hat{F}_{i+1}, M_{i+1})))^2 + (D_{fg}(\sigma(\tilde{F}_i, M_i)))^2, \\ \mathcal{L}^G_{bg} &= (D_{bg}(\phi(\hat{F}_{i+1}, M_{i+1})))^2 + (D_{bg}(\phi(\tilde{F}_i, M_i)))^2, \end{aligned} \tag{11}$$

where σ and ϕ denote operations to compute hand frames and background frames, respectively. G_{frame} is optimized with the following full objective function:

$$\begin{aligned} L^G_{frame} = \lambda_{img}\mathcal{L}_{img} + \lambda_{frame}\mathcal{L}^G_{frame} \\ + \lambda_{fg}\mathcal{L}^G_{fg} + \lambda_{bg}\mathcal{L}^G_{bg}, \end{aligned} \tag{12}$$

where $\lambda_{img}, \lambda_{frame}, \lambda_{fg}$, and λ_{bg} are positive weights to balance loss functions and their default values are 10.0, 1.0, 1.0, and 1.0, respectively. Furthermore, $D_{frame}, D_{fg}, D_{bg}$ are trained to minimize $\mathcal{L}^D_{frame}$, $\mathcal{L}^D_{fg}$, $\mathcal{L}^D_{bg}$, respectively.

$$\begin{aligned} \mathcal{L}^D_{frame} &= (D_{frame}(F_i))^2 + (D_{frame}(F_{i+1}))^2 \\ &\quad + (D_{frame}(\hat{F}_{i+1}) - 1)^2 + (D_{frame}(\tilde{F}_i) - 1)^2, \\ \mathcal{L}^D_{fg} &= (D_{fg}(\sigma(F_i, M_i)))^2 + (D_{fg}(\sigma(F_{i+1}, M_{i+1})))^2 \\ &\quad + (D_{fg}(\sigma(\hat{F}_{i+1}, M_{i+1})) - 1)^2 \\ &\quad + (D_{fg}(\sigma(\tilde{F}_i, M_i)) - 1)^2, \\ \mathcal{L}^D_{bg} &= (D_{bg}(\phi(F_i, M_i)))^2 + (D_{bg}(\phi(F_{i+1}, M_{i+1})))^2 \\ &\quad + (D_{bg}(\phi(\hat{F}_{i+1}, M_{i+1})) - 1)^2 \\ &\quad + (D_{bg}(\phi(\tilde{F}_i, M_i)) - 1)^2. \end{aligned} \tag{13}$$

IV. Experiments

We evaluate our approach with three public datasets: EPIC-Kitchens-55 where there are two objects of interest *(hands)* to control the motion in 2D, BAIR robot pushing dataset where there is a single object of interest *(robotic arm)* to control the motion in 2D, and Atari Breakout where there is a single object of interest *(base of breakout game)* to control the motion in 1D.

EPIC-Kitchens-55 dataset [29] contains approximately 40k first-person videos where humans interact with objects during daily activities in the kitchen. Actions on video clips are annotated with a text-based description composed of an action label (verb) and an object label (noun). In the evaluation with EPIC-Kitchens-55, we use the video clips where at least one hand is visible. CVGI is trained with video clips of the first kitchen (P01). Hand masks for M2M and M2F GANs are extracted automatically by the pretrained hand-segmentation model introduced in [42]. The model is trained on Extended GTEA Gaze+ dataset [43] for 100 epochs. The trained model is used to extract hand masks for each frame of EPIC-Kitchens-55. In addition to extending with hand masks, we also extend the annotations with the low-level control signals. As the ground truth control signals, we compute the displacements of the center of mass of the hand masks for every consecutive two frames. Furthermore, we augment the masks of consecutive frames by flipping horizontally (reflection) and warping masks with random translations in x and y directions.

BAIR robot pushing dataset [30] contains roughly 44k video clips of robotic arm pushing objects on a table. Each video clip consists of 30 frames in 256x256 resolution. Besides, the dataset provides the ground truth location of the robotic arm's gripper. In order to evaluate our approach, we extend the annotations of the dataset with the low-level control signals and text-based instructions. Thus, we compute the displacement of the gripper for every two consecutive frames as ground truth low-level signals. We prepare text-based instructions composed of a verb and adjective over the computed displacements. Verbs depict action with nine variations (8 for directions and 1 for stationary) and adjectives depict the speed of the motion with three variations (slowly, -, and quickly). The combination of verbs and adjectives composes 25 unique actions in the space of text-based instructions.

Atari Breakout dataset [8] contains roughly 1400 video clips in resolution 160x210 of the Atari Breakout video game

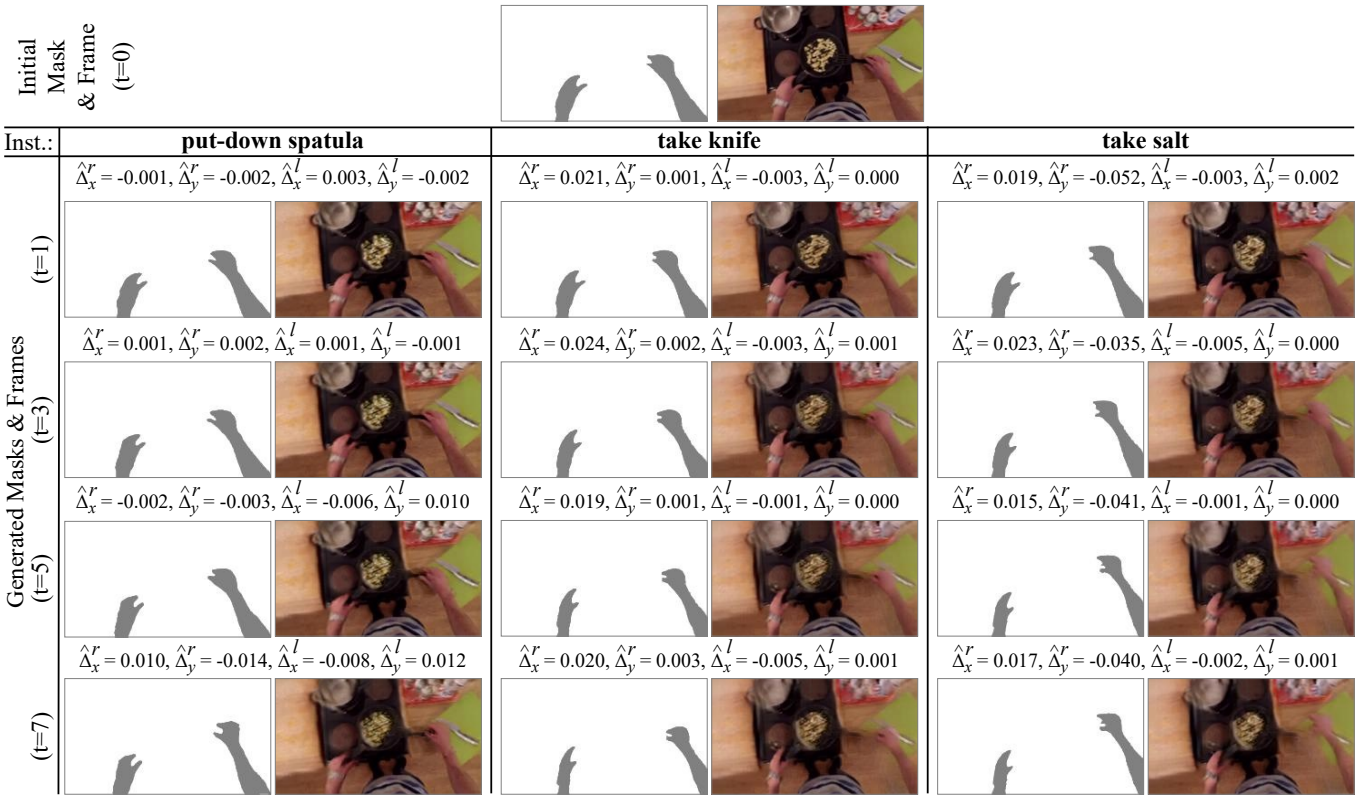


Fig. 7: Qualitative evaluation on EPIC-Kitchens-55 dataset. Given initial frame and mask and different instructions, estimated low-level control signals, estimated masks, and generated frames by CVGI are shown for three different textual instructions. $\hat{\Delta}_x^r$, $\hat{\Delta}_y^r$, $\hat{\Delta}_x^l$, and $\hat{\Delta}_y^l$ denote the estimated low-level control signals in 2D for right and left hand, respectively.

environment. Similar to BAIR robot pushing dataset, we extend the annotations of the dataset with the low-level control signals and text-based instructions. The displacement of the base is computed with respect to the location of the base's most-left pixel. Text-based instructions are prepared by the computed displacements. Although the variation of adjectives is three as in the BAIR robot pushing dataset, the variation of verbs is three due to the one-dimensional motion. So, the action space of text-based instructions has 7 unique actions.

A. Training Details

CVGI's modules are trained separately with video sequences of the training set that contains a set of frames, a set of masks, low-level control signals, and a text-based instruction, $S : \{\{F_0, F_1, ..., F_n\}, \{M_0, M_1, ..., M_n\}, \{\Delta_0, \Delta_1, ..., \Delta_n\}, d\}$. In all experiments, modules are trained from scratch for 500k iterations and we use Adam optimizer [44] with batch size of 16, learning rate of 0.0002, β_1=0.5, β_2 =0.999. For Epic-Kitchens-55, CVGI is trained to produce 7 future frames. i.e., the default value of the hyperparameter n is selected as 7 experimentally. Based on rigorous experimentation, we observe that 7 is an optimal number for generating future frames to avoid excess accumulation of errors in terms of artifacts. On the other hand, for BAIR robot pushing and Atari Breakout datasets, CVGI generates frames by producing the next frame (default value of n is 1) because motions in both datasets are simple motions that start in a frame and end in the next frame typically. To produce longer video sequences, generation can be re-initiated by using the last generated frame, its mask, and the text-based instruction, for EPIC-Kitchens-55 and the last generated frame and the text-based instruction for BAIR robot pushing and Atari Breakout datasets.

B. Qualitative Results

Figures 7, 8, and 9 show the results of qualitative evaluation of CVGI on EPIC-Kitchens-55, BAIR robot pushing, and Atari Breakout datasets, respectively. Figure 7 shows the initial frame, the corresponding mask, and text-based instructions along with estimated low-level control signals, estimated masks, and frames of generated sequences. In Figures 8 and 9, estimated low-level control signals and generated next frames by CVGI conditioned on different instructions are shown for one sample initial frame due to space limitations.

In Figure 7, CVGI is able to generate novel videos depicting different hand motions by using the same initial frame and hand mask according to instructions. Estimated low-level control signals change according to instructions which enables M2M GAN to produce different hand masks, which in turn allow M2F GAN to generate videos with different hand movements. In addition to the difference in generated videos, they are semantically consistent with instructions. As seen in Figure 7, hands in the generated videos move towards desired objects according to the instruction. Thus,

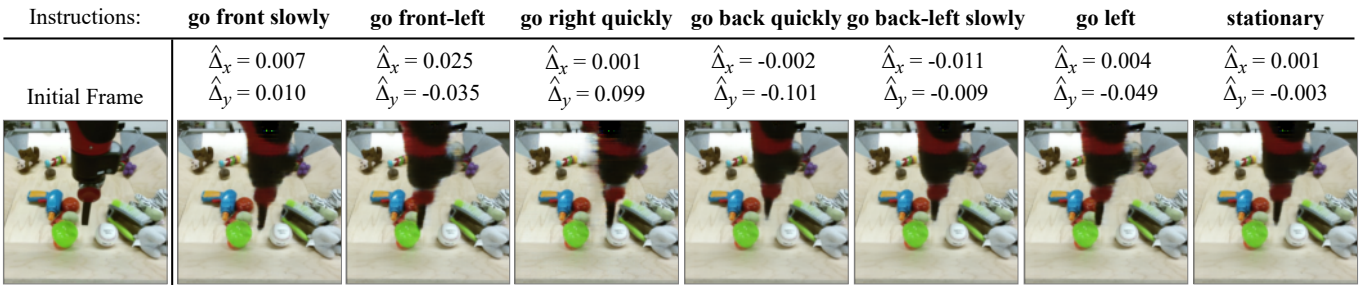


Fig. 8: Qualitative evaluation on BAIR robot pushing dataset. Given initial frame and different instructions, estimated low-level control signals and generated frames by CVGI are shown for different textual instructions. $\hat{\Delta}_x$ and $\hat{\Delta}_y$ denote the estimated low-level control signals in 2D for robotic arm.

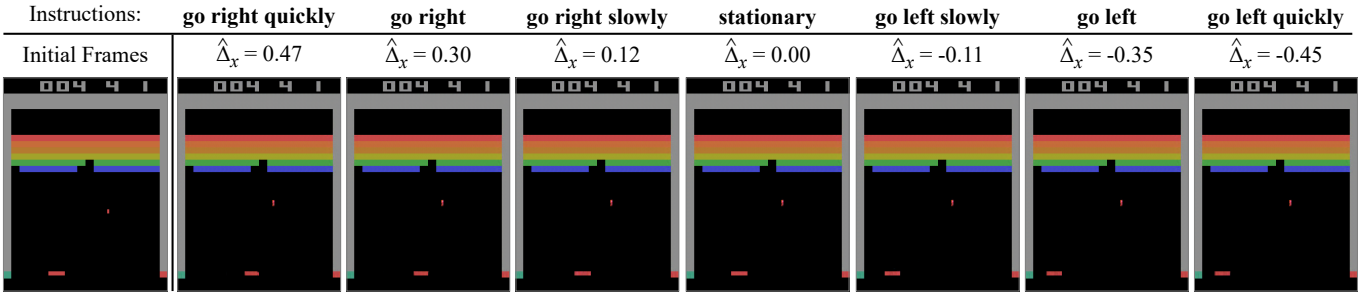


Fig. 9: Qualitative evaluation on Atari Breakout dataset. Given initial frame and different instructions, estimated low-level control signals and generated frames by CVGI are shown. $\hat{\Delta}_x$ denotes the estimated low-level control signals in 1D for base of breakout game.

CVGI comprehends instructions and controls the motion of both hands to synthesize videos depicting desired hand-based action.

As shown in Figures 8 and 9, CVGI can control generation based on instructions since, for the same initial frame, estimated low-level control signals and generated next frames differ according to instructions. In addition, generated frames also depict the desired motion. Thus, the qualitative evaluation shows that CVGI is able to control 2D and 1D motion of a single object of interest while producing realistic frames.

C. Quantitative Results

We follow the evaluation protocol proposed in [8] to evaluate the video generation quality of our framework. According to the protocol, models are used to generate frames of the test set by starting from the initial frame. Then the quality of generated frames is measured by three metrics: FID [45], FVD [46], and LPIPS [47]. They measure the similarity between two sets of samples and a lower score means more similar sets. Fréchet Inception Distance (FID) [45] measures similarity between two sets by comparing Gaussian distribution of deep features. Fréchet Video Distance (FVD) [46] is a variant of the FID metric specifically to evaluate the quality of video generation models. Learned Perceptual Image Patch Similarity (LPIPS) [47] measures the perceptual similarity between image patches. [8] chooses MoCoGAN [24], SAVP [48], and SRVP [49] as baseline. As discussed in [8], SAVP and SRVP are originally proposed to address future frame prediction which is a task to predict future frames with given previous frames. However, they can be adapted to our task without requiring major adjustments since future frame prediction is closely related to the controllable video generation. Besides, MoCoGAN which generates controllable videos of moving faces, body parts, artificial objects requires adjustment to handle the actions.

TABLE I: Comparison of video generation quality on BAIR robot pushing dataset: Results are reported as two groups. In the first group (the first four rows), models are trained to generate frames in low resolution (64x64). After frames are generated to reconstruct test videos, they are rescaled to high resolution (256x256). The second group (the last four rows) shows the results of models that are trained to generate frames in high resolution.

Models	FID [45] ↓	FVD [46] ↓	LPIPS [47] ↓
MoCoGAN [24]	198	1380	0.466
SAVP [48]	220	1720	0.433
SRVP [49]	224	3540	0.491
CVGI	**101**	**555**	**0.288**
MoCoGAN+	66.1	849	0.201
SAVP+	27.2	**303**	0.154
CADDY [8]	35.9	423	0.202
CVGI	**27.1**	376	**0.125**

With this protocol, we evaluate the video generation quality of CVGI's action generator on the video reconstruction task over BAIR robot pushing and Atari Breakout datasets and compare it with models including MoCoGAN [24], SAVP [48], SRVP [49], and CADDY [8].

TABLE II: Comparison of video generation quality on Atari Breakout dataset: Similar to Table I, results are reported in two resolution groups. Models in the first group (the first two rows), are trained to generate frames in low resolution and rescaled to high resolution. In the second group (the last four rows), models are trained to generate frames in 160x210 resolution and compared without requiring to rescale.

Models	FID [45] ↓	FVD [46] ↓	LPIPS [47] ↓
MoCoGAN [24]	99.9	**447**	**0.234**
SAVP [48]	**98.4**	487	0.239
MoCoGAN+	10.4	103	0.066
SAVP+	4.84	84.4	0.039
CADDY [8]	**0.72**	**5.94**	**0.008**
CVGI	9.52	23.84	0.018

In Tables I and II, we report the evaluation results as two groups according to the resolution of generated frames. The first group reported at the top shows the comparison of models that are trained with frames in low resolution (64x64 for BAIR robot pushing, 128x48 for Atari Breakout). After frames are generated, they are rescaled to high resolution (256x256 for BAIR robot pushing and 160x210 for Atari Breakout). MoCoGAN, SAVP, and SRVP are originally proposed to generate frames in low resolution and adapting them to high-resolution generation requires to improve the representation capacity of networks as discussed in [8]. The results of such improved models for MoCoGAN and SAVP are also reported in the second group where models are indicated with + sign. The second group includes CADDY which is proposed to generate frames in high resolution. Note that scores of other models in Tables I and II are reported from [8].

Table I shows the superior performance of CVGI in both resolution groups on BAIR robot push dataset. This could be attributed to use of L2 norm as reconstruction loss ($\mathcal{L}_{img}$) to train out model, which leads to better visual quality scores as discussed in [48]. On the other hand, the quality of generated videos on Atari Breakout dataset is comparable to the state-of-the-art video generation and prediction models as shown in Table II. The reason for this could be the limited training set.

Moreover, the action generation module of CVGI is capable to control the motion of the object of interest conditioned on the displacements which are continuous low-level control signals despite action space being discretized in the control signal estimator for the sake of simplicity of the control of the motion. On the other hand, CADDY framework controls the motion of the object of interest conditioned on the label of discrete actions. Thus, although the quality of generated videos by CADDY framework on Atari Breakout dataset is better as shown in Table II, controlling the motion of the object of interest with continuous low-level signals instead of discrete labels of actions fits better with the motion. And it allows a better understanding of the motion and increases the flexibility of the motion control.

The same evaluation protocol cannot be used for EPIC-Kitchens-55 because other models require major adaptations to handle complex scenes and actions of EPIC-Kitchens-55. For this reason, we evaluate the future frame prediction of CVGI by comparing it with Retrospective CycleGAN [7] which is one of the state-of-the-art future frame prediction models. Future frame prediction is a closely related research area and in the evaluation, we use Retrospective CycleGAN without adapting it to controllable video generation because such adaptation requires major modification of the networks and loss functions. Retrospective CycleGAN originally trained to predict future frames conditioned on four previous consecutive frames. Moreover, similar to CVGI, we adapt Retrospective CycleGAN to predict the future frame conditioned on the previous frame instead of four previous frames. In addition to Retrospective CycleGAN and the adapted Retrospective CycleGAN, we use Video Diffusion Models [39] as it is one of most recent and powerful model in video synthesis. However, the Video Diffusion model uses textual conditions only whereas CVGI uses visual conditions (a single frame) and textual conditions (text-based instruction) together. Thus, we adapt Video Diffusion which takes textual and visual conditions together for generating next frames. The adapted model is trained from scratch on EPIC-Kitchens-55 dataset with the default hyper-parameters of Video Diffusion.

The generation quality is measured by three metrics: Mean-Squared Error (MSE), Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM) [50] instead of FID [45], FVD [46], and LPIPS [47]. As MSE, PSNR, and SSIM are the most widely used metrics in the evaluation of future frame prediction approaches. They measure the similarity between the generated frame and the ground truth frame directly rather than measuring the similarity of two sets of frames. Since metrics are computed for frames one by one, we report the mean of the scores. MSE is a pixel-wise metric. PSNR is also a pixel-wise metric and it is based on MSE. On the other hand, SSIM compares frames based on image patches instead of pixel-wise comparison. Whereas a lower MSE score means more similar samples like FID, FVD, and LPIPS, a higher score means more similar samples in PSNR and SSIM. In addition to MSE, PSNR, and SSIM, Inception Score (IS) [51] is used to measure the fidelity of the generated frames and a higher score means better fidelity for the generation like PSNR and SSIM.

With this evaluation, we compare the fidelity of the generated frames and the consistency between the ground truth and the generated frames that are predicted by CVGI's action generator on the video reconstruction task over EPIC-Kitchens-55 dataset. As shown in Table III CVGI is compared with the adapted Video Diffusion [39] that is indicated as Video Diffusion*, Retrospective CycleGAN, the adapted Retrospective CycleGAN indicated as Retrospective CycleGAN*, and *copy-last* that is commonly used baseline in future frame prediction means copying the last previous frame as the prediction. In this evaluation, models are trained from scratch for 500k iterations with batch size 16 as the training of CVGI and used to predict the fifth frame of every five consecutive frames of videos at the test set. In addition, we also present an ablation study where CVGI without M2M GAN is used to reconstruct the frames of the test set by taking the ground truth hand masks instead of hands masks that are generated by M2M GAN. Thus, the ablation model shows the performance of M2F GAN only.

Table III shows that CVGI is capable to generate frames

TABLE III: Comparison of CVGI with *copy-last* that is a commonly used baseline, Retrospective CycleGAN, Retrospective CycleGAN* that is an adapted Retrospective CycleGAN model to predict next frames by using only the last previous frame, Video Diffusion* that is adapted Video Diffusion model to sythesise next frames by using the last previous frame and text-based instructions, and CVGI w/o M2M GAN which is an ablation model in predicting next frames on EPIC-Kitchens-55 dataset. MSE scores are multiplied by 10^3 to emphasize the difference. Besides, note that IS of *Copy-Last* does not be included as it contains real frames and the best score and the second-best score are highlighted in bold and underlined, respectively.

Models	MSE ↓	PSNR ↑	SSIM [50] ↑	IS [51] ↑
Copy-last	7.08	13.47	0.59	-
Retrospective CycleGAN [7]	**3.24**	20.94	**0.85**	5.42
Retrospective CycleGAN*	6.99	17.22	0.68	5.07
Video Diffusion*	6.47	20.35	0.76	5.49
CVGI w/o M2M GAN	<u>4.62</u>	**23.19**	**0.85**	**5.56**
CVGI	5.45	<u>21.98</u>	<u>0.82</u>	<u>5.54</u>

TABLE IV: Comparison of CVGI with existing video generation models on a benchmark task on BAIR robot pushing dataset. The benchmark task is video prediction where 15 next frames are predicted conditioned on a given single frame. The best score and the second-best score are highlighted in bold and underlined, respectively.

Models	FVD [46] ↓
SVP-FP [52]	315.5
CDNA [35]	296.5
SV2P [53]	262.5
LVT [54]	125.8
SAVP [48]	116.4
DVD-GAN [55]	109.8
VideoGPT [56]	103.3
TrLVD-GAN-FP [57]	103.3
Transframer [58]	100
HARP [59]	99.3
CCVS [60]	99
Phenaki [37]	97
Video Transformer [61]	94
FitVid [62]	93.6
MCVD [63]	89.5
NUWA [12]	86.9
RaMViD [64]	84.2
Video Diffusion [39]	**66.92**
CVGI	<u>74.54</u>

that are consistent with the ground truth and also it shows that the generated frames by CVGI are photo-realistic. In the comparison with Retrospective CycleGAN, CVGI has better and close scores in PSNR, SSIM, and IS whereas Retrospective CycleGAN has a better MSE score than CVGI. When we compare CVGI with Retrospective CycleGAN which is adapted to predict the next frame by using a single frame, CVGI consistently has better scores. This indicates when the supervision (the number of previous frames used to predict the next frame) in Retrospective CycleGAN decreases, its performance also decreases. Consequently, while the performance of CVGI and Retrospective CycleGAN are close to each other, CVGI has superior performance to Retrospective CycleGAN*. Despite the scores of Retrospective CycleGAN, predicting the future frame conditioned on a single frame as in CVGI and Retrospective CycleGAN* is indeed more challenging. In addition, CVGI consistently outperforms the adapted Video Diffusion model that uses both textual and visual conditions. In its comparison with Retrospective CycleGAN, it is observed that Video Diffusion has a better score in IS only. Moreover, the performance of Video Diffusion and CVGI is relatively close in IS whereas CVGI outperforms in MSE, PSNR, and SSIM with larger margins. In other words, the fidelity of generated frames by CVGI and Video Diffusion is relatively close to each other but generated frames by CVGI are more consistent with the ground truth than generated frames by Video Diffusion. The reason for this, directly using text-based instructions might be ambiguous for consistent video synthesis.

Moreover, Table III includes an ablation study. Although the scores of the ablation model and the complete model are close to each other, the performance of the ablation model is slightly better than the complete model. Because the complete model generates hand masks as well and the error in M2M GAN is accumulated. We believe these are the reasons for the performance difference. On the other hand, although the ablation model has slightly better performance than the complete model, it requires masks to control the hands' motion which is unreasonable to expect as user input. Thus, although M2M GAN causes error accumulation, it is essential for generating controllable videos.

Finally, to compare CVGI with a wider range of existing video generation models, we evaluated CVGI on the common benchmark task on BAIR robot pushing dataset (in low resolution 64x64). The task is video prediction to reconstruct test frames by synthesizing next 15 frames priming on a given single frame. To reconstruct the test frames from the given single frame, we employ the trained model of CVGI's action generator and reconstruct test frames conditioned on the displacement of the gripper. The frames are generated in a loop as CVGI is trained to generate a single next frame rather than a set of next frames. Then, by following the evaluation protocol in [54], [62], the generation quality of CVGI is measured with FVD score [46] between generated videos and ground truth videos. Note that FVD scores of others are reported from [37], [39], [54], [62]. As shown in Table IV, CVGI has the second-best score in the benchmark task on BAIR robot pushing dataset. Although FVD score of Video Diffusion is better than CVGI, Table IV shows CVGI can generate realistic videos. In addition to generating high-fidelity videos, CVGI is capable to control the action to generate novel videos.

D. Ablation Study

An ablation study is performed to analyze the effectiveness of the motion estimation layer over BAIR robot pushing and Atari Breakout datasets. This ablation study, therefore, shows the effect of using continuous signal rather than discrete signal as a condition. A new ablation model is trained which includes a GAN. This model manipulates the motion with text-based instructions directly. The generator takes the label of

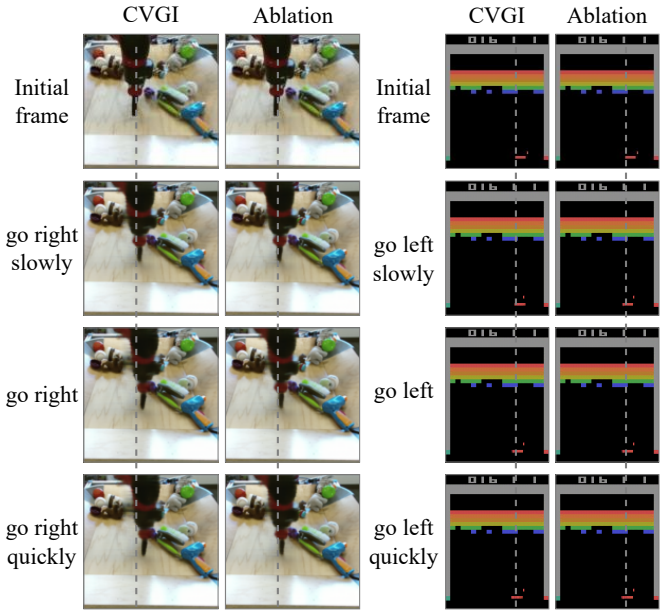


Fig. 10: Ablation results on BAIR robot pushing and Atari Breakout datasets. Initial frames and the generated frames with different textual instructions by CVGI and the ablation model that does not incorporate the motion estimation layer are shown. Gray vertical dotted lines show the position of the object of interest on the initial frame for the clarity.

motion instead of control signals along with the frames. The discriminator of the ablation model is a sequence discriminator and it has an auxiliary classifier instead of a regressor that predicts the action performed.

Figure 10 shows the initial frames and textual instructions. In addition, it shows the different motions on generated frames conditioned on instructions by our model and the ablation model. As shown in Figure 10, in the generated frames by the ablation model, the position of the gripper and the base are approximately the same whereas in the generated frames by CVGI that incorporates the motion estimation layer, the position of the gripper and the base differs according to the given instruction. Thus, the motion estimation layer is especially essential to control the motion's speed. Consequently, using a continuous signal rather than a discrete signal such as labels of actions is better to represent the motion of the object of interest as motions are continuous as well.

V. Conclusion

In this work, we propose a controllable video generation framework that provides detailed control over the motion of the object of interest to generate novel videos with text-based instructions. It incorporates a motion estimation layer to divide the task into two sub-tasks: control signal estimation and action generation. Our model learns to plan the motion of the object of interest according to instructions in control signal estimation and generate photo-realistic action videos in action generation. Experimental results on benchmark datasets demonstrate the effectiveness of our model. In the future, we plan to extend our model into an end-to-end model.

References

[1] Y.-F. Zhou, R.-H. Jiang, X. Wu, J.-Y. He, S. Weng, and Q. Peng, "Branchgan: Unsupervised mutual image-to-image transfer with a single encoder and dual decoders," *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3136–3149, 2019.

[2] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, "Drit++: Diverse image-to-image translation via disentangled representations," *arXiv preprint arXiv:1905.01270*, 2019.

[3] B. Li, X. Qi, T. Lukasiewicz, and P. Torr, "Controllable text-to-image generation," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[4] J. Lin, Z. Chen, Y. Xia, S. Liu, T. Qin, and J. Luo, "Exploring explicit domain supervision for latent space disentanglement in unpaired image-to-image translation," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[5] B. Zhu and C.-W. Ngo, "Cookgan: Causality based text-to-image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5519–5527.

[6] J. Huang, J. Liao, and S. Kwong, "Semantic example guided image-to-image translation," *IEEE Transactions on Multimedia*, vol. 23, pp. 1654–1665, 2021.

[7] Y.-H. Kwon and M.-G. Park, "Predicting future frames using retrospective cycle gan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1811–1820.

[8] W. Menapace, S. Lathuilière, S. Tulyakov, A. Siarohin, and E. Ricci, "Playable video generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 061–10 070.

[9] M. Nawhal, M. Zhai, A. Lehrmann, L. Sigal, and G. Mori, "Generating videos of zero-shot compositions of actions and objects," in *European Conference on Computer Vision*. Springer, 2020, pp. 382–401.

[10] W. Wang, X. Alameda-Pineda, D. Xu, E. Ricci, and N. Sebe, "Learning how to smile: Expression video generation with conditional adversarial recurrent nets," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 2808–2819, 2020.

[11] R. Cui, Z. Cao, W. Pan, C. Zhang, and J. Wang, "Deep gesture video generation with learning on regions of interest," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2551–2563, 2020.

[12] C. Wu, J. Liang, L. Ji, F. Yang, Y. Fang, D. Jiang, and N. Duan, "N\" uwa: Visual synthesis pre-training for neural visual world creation," *arXiv preprint arXiv:2111.12417*, 2021.

[13] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8188–8197.

[14] A. Koksal and S. Lu, "Rf-gan: A light and reconfigurable network for unpaired image-to-image translation," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[15] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Gaugan: semantic image synthesis with spatially adaptive normalization," in *ACM SIGGRAPH 2019 Real-Time Live!*, 2019, pp. 1–1.

[16] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[17] Z. Hao, X. Huang, and S. Belongie, "Controllable video generation with sparse trajectories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7854–7863.

[18] T. Marwah, G. Mittal, and V. N. Balasubramanian, "Attentive semantic video generation using captions," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1426–1434.

[19] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "Animating arbitrary objects via deep motion transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2377–2386.

[20] Y. Yan, B. Ni, W. Zhang, J. Xu, and X. Yang, "Structure-constrained motion sequence generation," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1799–1812, 2019.

[21] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5933–5942.

[22] P. Esser, J. Haux, T. Milbich *et al.*, "Towards learning a realistic rendering of human behavior," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[23] P. Esser, E. Sutter, and B. Ommer, "A variational u-net for conditional appearance and shape generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8857–8866.

[24] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1526–1535.

[25] O. Wiles, A. Koepke, and A. Zisserman, "X2face: A network for controlling face generation using images, audio, and pose codes," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 670–686.

[26] P. Ardino, M. De Nadai, B. Lepri, E. Ricci, and S. Lathuilière, "Click to move: Controlling video generation with sparse motion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 749–14 758.

[27] S. W. Kim, Y. Zhou, J. Philion, A. Torralba, and S. Fidler, "Learning to simulate dynamic environments with gamegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1231–1240.

[28] O. Gafni, L. Wolf, and Y. Taigman, "Vid2game: Controllable characters extracted from real-world videos," *arXiv preprint arXiv:1904.08379*, 2019.

[29] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The epic-kitchens dataset," in *European Conference on Computer Vision (ECCV)*, 2018.

[30] F. Ebert, C. Finn, A. X. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections." in *CoRL*, 2017, pp. 344–356.

[31] J. Walker, K. Marino, A. Gupta, and M. Hebert, "The pose knows: Video forecasting by generating pose futures," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3332–3341.

[32] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *international conference on machine learning*. PMLR, 2017, pp. 3560–3569.

[33] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[34] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," *Advances in neural information processing systems*, vol. 28, 2015.

[35] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," *Advances in neural information processing systems*, vol. 29, 2016.

[36] W. Hong, M. Ding, W. Zheng, X. Liu, and J. Tang, "Cogvideo: Large-scale pretraining for text-to-video generation via transformers," *arXiv preprint arXiv:2205.15868*, 2022.

[37] R. Villegas, M. Babaeizadeh, P.-J. Kindermans, H. Moraldo, H. Zhang, M. T. Saffar, S. Castro, J. Kunze, and D. Erhan, "Phenaki: Variable length video generation from open domain textual description," *arXiv preprint arXiv:2210.02399*, 2022.

[38] M. Saito, S. Saito, M. Koyama, and S. Kobayashi, "Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan," *International Journal of Computer Vision*, vol. 128, no. 10, pp. 2586–2606, 2020.

[39] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," *arXiv preprint arXiv:2204.03458*, 2022.

[40] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni *et al.*, "Make-a-video: Text-to-video generation without text-video data," *arXiv preprint arXiv:2209.14792*, 2022.

[41] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[42] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5686–5696, 2019.

[43] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *ECCV*, 2018.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[45] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. USA: Curran Associates Inc., 2017, pp. 6629–6640. [Online]. Available: http://dl.acm.org/citation.cfm?id=3295222.3295408

[46] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Towards accurate generative models of video: A new metric & challenges," *arXiv preprint arXiv:1812.01717*, 2018.

[47] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[48] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *ArXiv*, vol. abs/1804.01523, 2018.

[49] J.-Y. Franceschi, E. Delasalles, M. Chen, S. Lamprier, and P. Gallinari, "Stochastic latent residual video prediction," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3233–3246.

[50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[51] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2234–2242. [Online]. Available: http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf

[52] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," in *International conference on machine learning*. PMLR, 2018, pp. 1174–1183.

[53] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," *arXiv preprint arXiv:1710.11252*, 2017.

[54] R. Rakhimov, D. Volkhonskiy, A. Artemov, D. Zorin, and E. Burnaev, "Latent video transformer," *arXiv preprint arXiv:2006.10704*, 2020.

[55] A. Clark, J. Donahue, and K. Simonyan, "Adversarial video generation on complex datasets," *arXiv preprint arXiv:1907.06571*, 2019.

[56] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, "Videogpt: Video generation using vq-vae and transformers," *arXiv preprint arXiv:2104.10157*, 2021.

[57] P. Luc, A. Clark, S. Dieleman, D. d. L. Casas, Y. Doron, A. Cassirer, and K. Simonyan, "Transformation-based adversarial video prediction on large-scale data," *arXiv preprint arXiv:2003.04035*, 2020.

[58] C. Nash, J. Carreira, J. Walker, I. Barr, A. Jaegle, M. Malinowski, and P. Battaglia, "Transframer: Arbitrary frame prediction with generative models," *arXiv preprint arXiv:2203.09494*, 2022.

[59] Y. Seo, K. Lee, F. Liu, S. James, and P. Abbeel, "Harp: Autoregressive latent video prediction with high-fidelity image generator," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 3943–3947.

[60] G. Le Moing, J. Ponce, and C. Schmid, "Ccvs: Context-aware controllable video synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 042–14 055, 2021.

[61] D. Weissenborn, O. Täckström, and J. Uszkoreit, "Scaling autoregressive video models," *arXiv preprint arXiv:1906.02634*, 2019.

[62] M. Babaeizadeh, M. T. Saffar, S. Nair, S. Levine, C. Finn, and D. Erhan, "Fitvid: Overfitting in pixel-level video prediction," *arXiv preprint arXiv:2106.13195*, 2021.

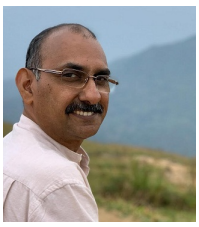
[63] V. Voleti, A. Jolicoeur-Martineau, and C. Pal, "Masked conditional video diffusion for prediction, generation, and interpolation," *arXiv preprint arXiv:2205.09853*, 2022.

[64] T. Höppe, A. Mehrjou, S. Bauer, D. Nielsen, and A. Dittadi, "Diffusion models for video prediction and infilling," *arXiv preprint arXiv:2206.07696*, 2022.

Ali Köksal received his B.Sc. and M.Sc. degrees in Computer Engineering from Izmir Institute of Technology, Turkey. He is a Ph.D. graduand of the School of Computer Science and Engineering at Nanyang Technological University, Singapore. He is working as a Scientist in the Visual Intelligence Department, Institute for Infocomm Research, A*STAR, Singapore.

Kenan Emir Ak received his B.Sc. and M.Sc. degrees in Electrical and Electronics Engineering from Isik University, Turkey and Bogazici University, Turkey. His Ph.D. degree in Electrical and Computer Engineering from National University of Singapore. He is a Scientist in the Visual Intelligence Department, Institute for Infocomm Research, A*STAR, Singapore.

Ying Sun received her B.Eng. degree from Tsinghua University, her M.Phil. degree from Hong Kong University of Science and Technology, and her Ph.D. degree from Carnegie Mellon University. She is a Senior Scientist with the Visual Intelligence Department, Institute for Infocomm Research, and Centre for Frontier AI Research, A*STAR, Singapore.



Deepu Rajan is an Associate Professor in the School of Computer Science and Engineering at Nanyang Technological University, Singapore. He received his Bachelor of Engineering degree in Electronics and Communication Engineering from Birla Institute of Technology, Ranchi (India), M.S. in Electrical Engineering from Clemson University (USA), and Ph.D. from Indian Institute of Technology, Bombay (India). From 1992 till 2002, he was a Lecturer in the Department of Electronics at Cochin University of Science and Technology, India. His research interests include image processing, computer vision, and multimedia signal processing.

Joo-Hwee Lim received B.Sc. (Hons I) and M.Sc. degrees in Computer Science from NUS Singapore and Ph.D. degree in Computer Science & Engineering from UNSW, Australia. He is Principal Scientist and Department Head (Visual Intelligence) at the Institute for Infocomm Research, A*STAR, Singapore. He has published 290 international refereed journal and conference papers in connectionist expert systems, neural-fuzzy systems, content-based image retrieval, medical image analysis, human robot collaboration.