Improving Video Generation for Multi-functional Applications

Bernhard Kratzwald, Zhiwu Huang, Danda Pani Paudel, Acharya Dinesh,
Luc Van Gool
{kratzwab,acharyad}@ethz.ch
{zhiwu.huang,paudel,vangool}@vision.ee.ethz.ch

ETH Zurich

Abstract. In this paper, we aim to improve the state-of-the-art video generative adversarial networks (GANs) with a view towards multi-functional applications. Our improved video GAN model does not separate foreground from background nor dynamic from static patterns, but learns to generate the entire video clip conjointly. Our model can thus be trained to generate—and learn from—a broad set of videos with no restriction. This is achieved by designing a robust one-stream video generation architecture with an extension of the state-of-the-art Wasserstein GAN framework that allows for better convergence. The experimental results show that our improved video GAN model outperforms state-of-the-art video generative models on multiple challenging datasets. Furthermore, we demonstrate the superiority of our model by successfully extending it to three challenging problems: video colorization, video inpainting, and future prediction. To the best of our knowledge, this is the first work using GANs to colorize and inpaint video clips.

1 Introduction

Viewed as a digital window into the real-life physics of our world, videos capture how objects behave, move, occlude, deform, and interact with each other. Furthermore, videos record how camera movements, scene depth or changing illumination influence a scene. Fully understanding their temporal and spatial dependencies is one of the core problems in computer vision. Teaching computers to model and interpret scene dynamics and dependencies occurring within videos is an essential step towards intelligent machines capable of interacting with their environment.

In contrast to the domain of images, the work on supervised and unsupervised learning from videos is still in its infancy. This can be attributed to the high-dimensional nature of videos. Performing large-scale supervised learning on video data requires prohibitively large amounts of labeled training samples. This can quickly become a bottleneck in supervised learning for video. The recent focus of research on videos has therefore shifted from supervised to unsupervised models. The near endless amount of unlabeled video data available on the Internet further encourages the choice of unsupervised methods [1–4].

State-of-the-art unsupervised video models are often designed to simplify the generation process by segmenting certain aspects of the video. Generative video models separate foreground from background [5], or dynamic from static patterns [6, 7]. These are architectural choices that simplify and stabilize the generation process. On the other hand they often impose certain restrictions on the training data; e.g. [5] requires stable backgrounds and non-moving cameras. Video generation in a single stream avoids such simplifications but is inherently more difficult to achieve as low frequencies span both the temporal and spatial domain. The motivation of this work is to create a robust, universal and unrestricted generative framework that does not impose any preconditioning on the input videos while at the same time producing state-of-the-art quality videos.

The task of generating videos is related to modeling and understanding the scene dynamics within them. For realistic video generation, it is essential to learn which objects move, how they move, and how they interact with each other, which *vice versa* implies an understanding of real-world semantics. A model capable of understanding these semantics is ideally not restricted to the task of video generation but can also transfer this knowledge to a broad number of other applications. Important applications include action classification, object detection, segmentation, future prediction, colorization, and inpainting.

Our paper focuses both on the robustness of our generative video framework as well as on its application to three problems. First, we design a stable architecture with no prior constraints on the training data. More precisely, we design a one-stream generation framework that does not formally distinguish between foreground and background, allowing us to handle videos with moving backgrounds/cameras. Video generation in a single-stream is a fragile task, demanding a carefully selected architecture within a stable optimization framework. We accomplish this stability by exploiting state-of-the-art Wasserstein GAN frameworks in the context of video generation. In a second step, we demonstrate the applicability of our model by proposing a general multi-functional framework dedicated to specific applications. Our extension augments the generation model with an auxiliary encoder network and an application-specific loss function. With these modifications, we successfully conduct several experiments for unsupervised end-to-end training.

The two main contributions of this paper are as follows: (i) We propose iVGAN, a robust and unrestricted one-stream video generation framework. Our experiments show that iVGAN outperforms state-of-the-art generation frameworks on multiple challenging datasets. (ii) We demonstrate the utility of the multi-functional extension of iVGAN for three challenging problems: video colorization, video inpainting, and future prediction. To the best of our knowledge, this is the first work exploiting the advantages of GANs in the domains of video inpainting and video colorization.

2 Related Work

Generative Adversarial Networks (GANs): GANs [8] have proven successful in the field of unsupervised learning. Generally, GANs consist of two neural networks: a generator network trained to generate samples and a discriminator network trained to distinguish between real samples drawn from the data distribution and fake samples produced by the generator. Both networks are trained in an adversarial fashion to improve each other. However, GANs are also known to be potentially unstable during training.

To address this problem, Radford et al. [9] introduced a class of *Deep Convolutional GANs* (DCGANs) that imposes empirical constraints on the network architecture. Salimans et al. [10] provide a set of tools to avoid instability and mode collapsing. Che et al. [11] use regularization methods for the objective to avoid the problem of missing modes. Arjovsky et al. [12] suggest minimizing the Wasserstein-1 or Earth-Mover distance between generator and data distribution with theoretical reasoning. In a follow-up paper, Gulrajani et al. [13] propose an improved method for training the discriminator – termed *critic* by [12] – which behaves stably, even with deep ResNet architectures. GANs have mostly been investigated on images, showing significant success with tasks such as image generation [9, 13–16], image super-resolution [17], style transfer [18, 19], and many others.

Video Generation: There has been little work on the topic of video generation so far [5–7]. In particular, Vondrick et al. [5] adapts the DCGAN model to generate videos, predict future frames and classify human actions. Their *Video GAN* (VGAN) model suggests the usage of independent streams for generating foreground and background. The background is generated as an image and then replicated over time. A jointly trained mask selects between foreground and background to generate videos. In order to encourage the network to use the background stream, a sparsity prior is added to the mask during learning. More recently, *Temporal GAN* (TGAN) [7] deals with the instability in video generation by deploying a frame-wise generation model. A generative model for image generation is used to sample frames; a temporal generator preserves temporal consistency and controls this model. Tulyakov et al. [6] also adopted a two-stream generative model that produces dynamic motion vs. static content. In particular, the static part is modeled by a fixed Gaussian when generating individual frames within the same video clip, while a recurrent network that represents the dynamic patterns models the motion part. To deal with the instability of training GANs all three models separate integral parts of a video, as foreground from background or dynamic from static patterns. We argue that it is more natural to learn these patterns and their interference conjointly. Therefore, we propose a single-streamed but robust video generation architecture in Sec. 3.

Video Colorization: Works on image and video colorization can be divided into two categories: interactive colorization that requires some kind of user input [20–25] and automatic methods [26–31]. Our approach belongs to the latter category. Most automatic methods come with restrictions that prevent them from working in general settings. For instance, [29] requires colored pictures of a similar viewing angle and [26] requires separate parameter tuning for every input picture. Methods such as [28, 30] produce undesirable artifacts. In the video domain, methods such as [27] process each frame independently, which in turn leads to temporal inconsistencies. Recently, image colorization has been combined with GANs [32], but no prior research on colorizing videos has been presented.

Video Inpainting: Inpainting is a fairly well investigated problem in the image domain [33–35]. For videos, it has been used to restore damage in vintage films [36], to remove objects [37] or to restore error concealment [38]. State-of-the-art frameworks like [39] use complex algorithms involving optical flow computation; thus demanding an optimized version to run within a feasible amount of time. Recovering big areas of an image or a video, also called *hole-filling*, is inherently a more difficult problem than the classical inpainting. Approaches like texture synthesis [40, 41] or scene completion [42] do not work for *hole-filling* [43]. While there has been some work on image inpainting with adversarial loss functions [43], we are not aware of any in the case of videos.

Future Prediction: Future prediction is the task of predicting the future frames for one/multiple given input frames. In contrast to video generation, future prediction is an elegant way of turning an unsupervised modeling problem into a supervised learning task by splitting videos into conditioning input and ground-truth future. Our method builds upon recent future prediction work e.g. [2–4, 44–49], especially that using generative models and adversarial losses [1, 5, 50, 51].

3 Our Model - iVGAN

For robust video generation, we propose a simple yet tough to beat video generation model, called *improved Video GAN* (iVGAN). Our model consists of a generator and a discriminator network in the GAN framework. Particularly, the designed generator $G : Z \rightarrow \mathcal{X}$ produces a video $\boldsymbol{x}$ from a low dimensional latent code $\boldsymbol{z}$. The proposed critic network $C : \mathcal{X} \rightarrow \mathbb{R}$ is optimized to distinguishing between real and fake samples and provides the generator updates with useful gradient information.

Distinct from [5], we design the generation framework without any prior assumptions upon the nature of the data. Two-stream architectures generate the background as an image; thereby, limit the training data to videos with static backgrounds and non-moving cameras. It is thus essential that our generator is of one-stream, without separating back- and foreground. In contrast to [6, 7] we use a simple but effective architecture which learns spatial and temporal dependencies conjointly, rather than separating them into two networks.

As studied in [9, 10, 12, 13] for image generation, it is non-trivial to train GAN models in a stable manner. Especially for video generation, it turns out to be much more challenging [7] as low frequencies also span the additional temporal domain. To address this problem, we generalize the state-of-the-art Wasserstein GAN to the context of video generation for more stable convergence. Formally, we place our network within the Wasserstein GAN framework [12] optimizing

$$\min_{G} \max_{\|C\|_L \leq 1} V(G, C) = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})} [C(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})} [C(G(\boldsymbol{z}))]. \tag{1}$$

In order to enforce the Lipschitz constraint on the critic function, we penalize its gradient-norm with respect to the input [13]. For this purpose we evaluate the critic's gradient $\nabla_{\hat{\boldsymbol{x}}} C(\hat{\boldsymbol{x}})$ with respect to points sampled from a distribution over the input space $\hat{\boldsymbol{x}} \sim p_{\hat{x}}$, and penalize its squared distance from one via

$$\mathcal{L}_{GP}(C) = \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim p_{\hat{x}}} \left[(\|\nabla_{\hat{\boldsymbol{x}}} C(\hat{\boldsymbol{x}})\|_2 - 1)^2 \right]. \tag{2}$$

The distribution $p_{\hat{x}}$ is defined by uniformly sampling on straight lines between points in the data distribution and points in the generator distribution. Hence, the final unconstrained objective is given by

$$\min_{G} \max_{C} V(G, C) + \lambda \mathcal{L}_{GP}(C), \tag{3}$$

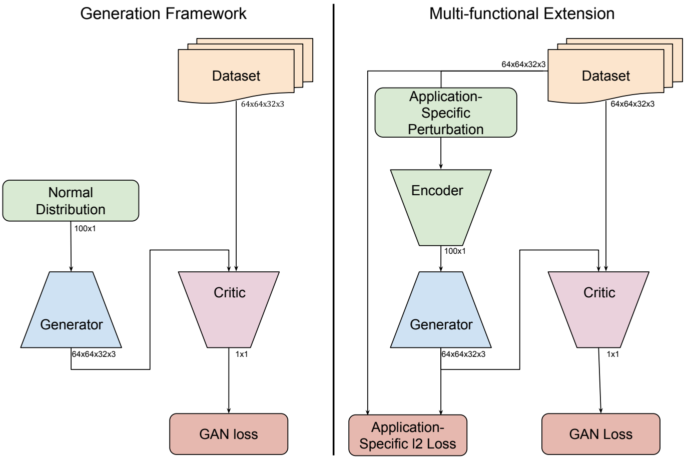

Fig. 1. iVGAN video generation framework and its multi-functional extension

where the hyperparameter λ is used to balance the GAN objective with the gradient penalty.

3.1 Generator Network

The generator takes a latent code sampled from a 100-dimensional normal distribution $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and produces an RGB video containing 32 frames of 64×64 pixels. We use a linear up-sampling layer in the first step, producing a tensor of size $2 \times 4 \times 4 \times 512$. The linear block is followed by four convolutional blocks of spatio-temporal [52] and fractionally-strided [53] convolutions. This combination has proven to be an efficient way to upsample, while preserving spatial and temporal invariances [1, 5]. All convolutional layers utilize $4 \times 4 \times 4$ kernels, a stride of $2 \times 2 \times 2$, and add a bias to the output. We found the initialization of the convolutional weights essential for stable training and faster convergence. Inspired by the ResNet architecture [54] we initialize the kernels according to He et al. [55]. Similar to DCGAN [9], all but the last layers are followed by a batch normalization layer [56]. Batch normalization stabilizes the optimization by normalizing the inputs of a layer to zero mean and unit variance, which proved critical for deep generators in early training, preventing them from collapsing [9].

The first four blocks are followed by a ReLU non-linearity after the normalization layer, while the last layer uses a hyperbolic tangent function. This is beneficial to normalize the generated videos, identically to the videos in our dataset, within the range $[-1, 1]$.

3.2 Critic Network

The critic network maps an input video to a real-valued output. It is trained to distinguish between real and generated videos, while being constrained (Eqn. 2) to yield effective gradient information for generator updates.

The critic consists of five convolutional layers and is followed by an additional linear down-sampling layer. As in [5], we use spatio-temporal convolutions with $4 \times 4 \times 4$ kernels. Again we found the initialization of kernel weights important for stability and convergence during training and used the initializion following [55]. For more expressiveness, we add a trainable bias to the output. All convolutions include a stride of $2 \times 2 \times 2$ to enable efficient down-sampling of the high-dimensional inputs.

Batch normalization correlates samples within a mini-batch by making the output for a given input $\boldsymbol{x}$ dependent on the other inputs $\boldsymbol{x'}$ within the same batch. A critic with batch normalization therefore maps a batch of inputs to a batch of outputs. On the other hand, in Eqn 2, we are penalizing the norm of the critic's gradient with respect to each input independently. For this reason, batch normalization is no longer valid in our theoretical setting. To resolve this issue, we use layer normalization [57] following [13]. Layer normalization works equivalent to batch normalization, but mean and standard deviation is calculated independently for every single sample $\boldsymbol{x_i}$ over the hidden layers. We found that layer normalization is not necessary for convergence, but essential if we optimize the generator with additional objectives, as described in the multi-functional extension in Sec. 4.

All but the last layer use a leaky ReLU [58] activation. We omit using a soft-max layer or any kind of activation in the final layer, since the critic is not trained to classify between real and fake samples, but rather trained to yield a good gradient information for generator updates.

3.3 Learning and Parameter Configuration

We optimize both networks using alternating stochastic gradient descent, more precisely we optimize the critic five times for every update step on the generator. The hyperparameter λ, controlling the trade-off between the GAN objective and the gradient penalty (Eqn. 3), is set to 10 as reported in [13]. We use Adam [59] with initial hyperparameters $\alpha = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.99$ and a batch size of 64 which has proved to work best for us after testing various alternate settings. We divide the learning rate by two after visual convergence. We train our network from scratch which usually takes four to six days on a *GeForce GTX TITAN X* GPU. The entire network is implemented in TensorFlow.

4 Multi-functional Extension

With a simple yet powerful modification, we extend our generation architecture to a multi-functional video processing framework. We choose three challenging applications to demonstrate the semantics our framework is capable of learning: (i) to successfully colorize grayscale videos our network must learn temporally consistent color semantics;

meadows e.g. have to be painted in a shade of green which should stay consistent over time (ii) inpainting, which is completing and repairing missing or damaged parts of a video, requires the network to learn spatial consistencies such as symmetries (iii) future prediction conditioned on a single input frame is the toughest application and requires our model to learn and understand which objects are plausible to move how they do so.

Fig. 1 compares the generation framework architecture with its multi-functional extension. Similar to conditional GANs [60], the generator is no longer dependent on a randomly drawn latent code $\boldsymbol{z}$ but conditioned on additional application-specific information $\boldsymbol{y}$. A convolutional network $E : \mathcal{Y} \rightarrow \mathcal{Z}$ generates a latent code $\boldsymbol{z}$ by encoding $\boldsymbol{y}$; which is in turn used to generate the desired video. To guide this generation we extend the framework by an additional application-specific loss $\mathcal{L}_{AP}$.

The choice of $\boldsymbol{y}$ and the loss function depends on the application at hand. For video colorization we encode a grayscale video we wish to colorize and use the $\ell 2$ loss between the generated and input video. For inpainting we condition on the damaged input clip and calculate the $\ell 2$ loss between reconstruction and ground-truth. To predict future frames we encode a single input frame and apply the $\ell 2$ loss between that frame and the first frame of the generated video.

We jointly optimize for the GAN value function (Eqn. 1), the gradient penalty (Eqn. 2), and the new domain-specific loss $\mathcal{L}_{AP}$, using two hyperparameters λ and ν to control the trade-off between them. To gain a deeper understanding of the interaction between GAN- and reconstruction loss, we conduct experiments with two variations of the colorization framework: In the *unsupervised* setting the reconstruction loss is calculated in grayscale color space and does therefore not penalize wrong colorization, leaving the GAN-loss solely responsible for learning color semantics. In the *supervised* setting on the other hand, the $\ell 2$-loss is calculated in RGB color space and thus penalizes both wrong colorization and wrong structure. It remains unclear what role the GAN-loss takes in the latter setting. Following Zhao et al. [61] we argue in Sec. 5.4 that the GAN-loss acts as a regularizer similar to a variational autoencoder; thus preventing the encoder-generator from learning a simple identity function.

4.1 Learning and Parameter Configuration

The encoder network consists of four strided convolutional layers, each of which is followed by a batch normalization layer and a ReLu activation function. We found it difficult to adjust the hyperparameter ν which controls the trade-off between the GAN loss and the domain-specific $\ell 2$ loss. While the latter is per definition within the range $[0, 1]$, the GAN loss is not bound as the critic output does not yield a probability anymore. We found it essential for a stable GAN loss to use layer normalization in the critic network; allowing us to monitor the losses and empirically set $\nu = 1000$.

5 Experiments

We evaluate our generation framework on multiple challenging datasets and compare our results with the two state-of-the-art video generation frameworks; namely the Video GAN (VGAN) [5] and the Temporal GAN (TGAN) [7] model. Other models such

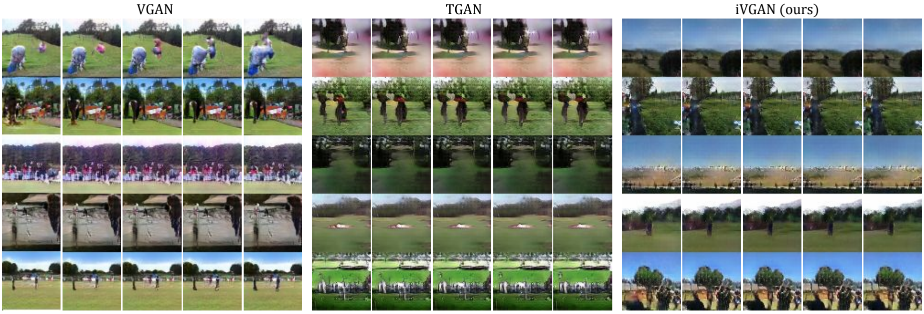

Fig. 2. Video generation results on stabilized golf clips. *Left*: Videos generated by the two-stream VGAN model. *Middle*: Videos generated by the TGAN model. *Right*: Videos generated by our one-stream iVGAN model

as [51] require supervision by one or more input frames and are hence excluded from our evaluation. For our multi-functional extension, we choose to colorize grayscale videos; inpaint damaged videos, and predict future frames from static images. Note that, for a better understanding, we also provide the readers with examples of animated generations and the source code for all our models in the supplementary material.

5.1 Datasets

We used different datasets of unlabeled but filtered video clips, which have been extracted from high-resolution videos at a natural frame rate of 25 frames per second.

Stabilized Videos: This dataset[1] was composed by [5] and contains parts of the Yahoo Flickr Creative Commons Dataset [62]. The Places2 pre-trained model [63] has been used to filter the videos by scene category *golf course*. All videos have been pre-processed to ensure a static background. Therefore, SIFT keypoints were extracted to estimate a homography between frames and minimize the background motion [5]. The task of background stabilization may very often not be valid, forcing us to renounce a significant fraction of data. Discarding scenes with non-static background significantly restricts our goal of learning real-world semantics through unsupervised video understanding.

Airplanes Dataset: We compiled a second more challenging dataset of filtered, unlabeled and unprocessed video clips. Similar to the golf dataset videos are filtered by scene category, in this case *airplanes*. Therefore, we collected videos from the YouTube-BoundingBoxes dataset [64] which have been classified containing airplanes. No pre-processing of any kind has been applied to the data and the dataset thus contains static scenes as well as scenes with moving background or moving cameras.

[1] We downloaded the dataset from `http://carlvondrick.com/tinyvideo/`

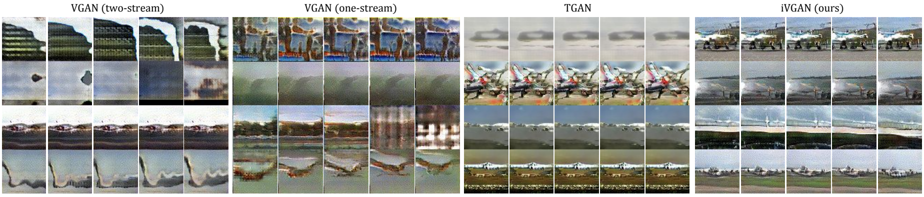

Fig. 3. Video generation results on unstabilized airplane videos. Comparing videos generated using the one and two stream VGAN as well as the TGAN model, against our iVGAN framework

5.2 Qualitative Evaluation

Fig. 2 qualitatively compares results of the VGAN, TGAN and our iVGAN generator; where all three models were trained on the golf dataset. More animated samples are available in the supplementary material. There is no formal concept of foreground or background in the iVGAN model since the entire clip is generated in a single stream. Our model nonetheless naturally learns from the data to generate clips with a static background and moving foreground. Despite the fact that the background is not generated as an image (VGAN) it looks both sharp and realistic in the majority of samples. The foreground suffers from the same flaws as the VGAN and TGAN model: it is blurrier than the background, with people and other foreground objects turning into blobs. The network correctly learns which objects should move, and generates plausible motions. Some samples are close to reality, a fraction of samples collapse during training. Overall, the network learns correct semantics and produces scenes with a sharp and realistic looking background but blurry and only fairly realistic foreground-motion.

We conducted four independent experiments using the VGAN generator on the airplanes dataset, varying the learning rate between 0.00005 and 0.0002, and the sparsity penalty on the foreground mask between 0.1 and 0.15. In all runs, without exception, the generator collapsed and failed to produce any meaningful results. One might argue that it is unfair to evaluate a two-stream generation model, which assumes a static background, on a dataset violating this assumption. Therefore, we repeated a series of experiments using the one-streamed VGAN model, which does not separate foreground and background. A one-stream model should theoretically be powerful enough to converge on this dataset. Regardless of that, the one-stream version of VGAN collapsed as well in all experiments and failed to generate meaningful videos; indicating the difficulty of video generation with unstabilized videos. The more stable TGAN model does not collapse but fails to produce videos with moving backgrounds or camera motions.

Fig. 3 qualitatively compares generations from the two- and one-stream VGAN as well as the TGAN model against our iVGAN generator. Although the quality of our samples is lower compared to the stabilized golf videos, our generator did in no single experiment collapse. The iVGAN model – unlike any other generative model – produces both: videos with static background, as well as videos with moving background or camera motion. A fraction of the generated videos collapsed to meaningless colored noise, nonetheless. Nevertheless, it is clear that the network does learn important se-

Table 1. Quantitative Evaluation on Amazon Mechanical Turk: We show workers two pairs of videos and ask them which looks more realistic. We show the percentage of times workers prefer our model against real videos, VGAN and TGAN samples on two datastes

”Which video is more realistic?”	**Percentage of Trials**
Random Preference	50
Prefer iVGAN over Real (Golf)	23.3
Prefer iVGAN over VGAN (Golf)	59.3
Prefer iVGAN over TGAN (Golf)	57.6
Prefer iVGAN over Real (Airplanes)	15.4
Prefer iVGAN over TGAN (Airplanes)	59.7

mantics since a significant number of videos shows blurry but realistic scenes, objects, and motions.

5.3 Quantitative Results:

We used Amazon Mechanical Turk for a quantitative evaluation. Following [5] we generated random samples from all three models as well as the original dataset. We showed workers a pair of videos drawn from different models and asked them: *“Which video looks more realistic?”*. We paid workers one cent per comparison and required them to historically have a 95% approval rating on Amazon MTurk. We aggregated results from more than 9000 opinions by 130 individual workers and show them in Tab. 1. Our results show that workers can clearly distinguish between real and fake videos; the distinction seems easier on the more challenging airplane dataset. Furthermore, workers asses that videos generated by our iVGAN model look significantly more realistic than those generated by the VGAN or TGAN model; hence, our iVGAN model clearly outperforms the state-of-the-art methods on both the golf and the airplane datasets. Since the VGAN model did not produce meaningful results on the airplane dataset we omitted the trivial comparison on this dataset.

5.4 Colorization

Fig. 4 qualitatively compares our framework with the state-of-the-art *Colorful Image Colorization* (CIC) model [31]. The CIC model colorizes videos in their original resolution frame by frame. Our model, on the other hand, colorizes the entire clip at once but is restricted to in- and outputs of 64×64 pixels. Frame-wise colorization is known to suffer from temporal inconsistencies [27]. Fig. 4 illustrates e.g. how the CIC colorized jacket changes its color over time while our colorization stays consistent. Our network overall learns correct color semantics: areas in the input are selected, “classified” and then painted accordingly. The sky e.g. is colorized in shades of blue or gray-white and trees are painted in a darker green than the grass. Therefore, we argue that the network not only selects the trees, but also recognizes (classifies) them as such, and paints them according to their class. The quality of the segmentation depends on the sharpness of the edges in the grayscale input. Colorized videos are blurrier compared to the grayscale

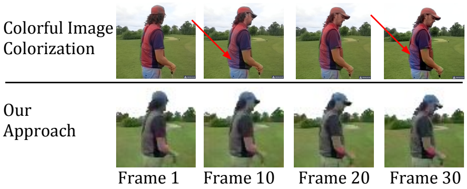

Fig. 4. Color consistency over time with different colorization approaches. Red arrows mark spots where color is inconsistent over time

input. This is mainly due to the fact that we do not keep the spatial resolution of the videos but encode them to a latent code, from which the colorized videos are then generated. Furthermore, using the mean squared error function to guide reconstructions is known to generate blurry results [50].

We evaluated the sharpness of the colorization quantitatively by the *Peak Signal to Noise Ratio* (PSNR) in gray-space. PSNR correlates better with visual perception than the $\ell 2$-loss. For the colorization quality we asked workers on Amazon MTurk to rate how realistic the video looks on a linear scale from 5 (very realistic) to 1 (very unrealistic). We generated random samples from each model and used random clips from the dataset as a reference value. The mean score for each model was calculated from more than 7000 ratings. We trained our models on 95% of the golf dataset and evaluated them on 5% hold-out data as well as on the out-of-domain airplane dataset. Notably even though we trained on stabilized video clips, our model is able to colorize clips with moving cameras and camera motion. The quantitative evaluation is shown in Tab. 2, animated results are available in the supplementary material.

Table 2. Quantitative evaluation of video colorization and inpainting frameworks. Left: Average user rating of their realism from 1 (very unrealistic) to 5 (very realistic). Right: Peak signal to noise ratio between generated videos, and grayscale input (colorization) or ground-truth videos (inpainting)

Model	**MTurk average rating**	**PSNR hold-out data**	**PSNR out-of-domain data**
Video Colorization			
supervised	2.45	25.2 dB	23.4 dB
unsupervised	2.95	25.6 dB	24.2 dB
Video Inpainting			
salt & pepper	3.63	29.2 dB	25.4 dB
boxes (fixed)	3.37	25.3 dB	22.9 dB
boxes (random)	3.43	24.7 dB	22.7 dB

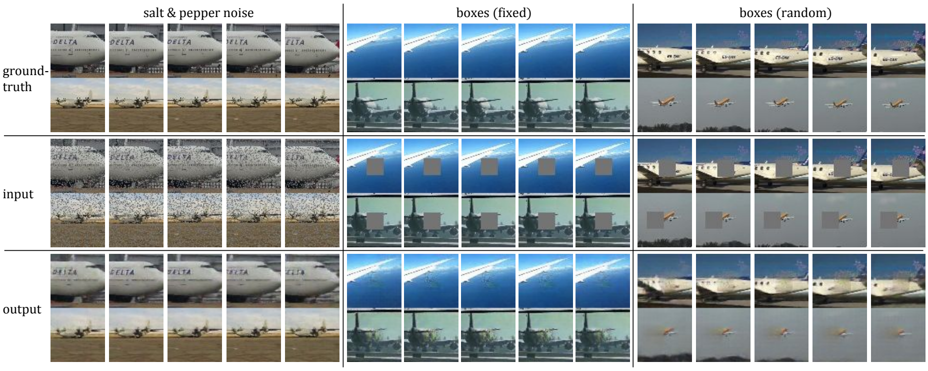

Fig. 5. Comparison of ground-truth videos with the reconstructions of salt&pepper noise, missing holes in the center and at random positions

To investigate the interplay between the GAN-loss and encoder-generator reconstruction loss we compare two variations of our model. As described in Sec. 4, the *supervised* model calculates the reconstruction loss in RGB color space, while the *unsupervised* model calculates the loss in grayscale color space. Our experiments indicate that the supervised colorization network, having a stronger objective, tends to overfit. Although they perform equally well on the training data, the unsupervised network outperforms the supervised network on hold-out and out-of-domain data as quantitatively shown in Tab. 2. The unsupervised model relies strongly on the GAN loss, which we argue – following Zhao et al. [61] – acts as a regularizer preventing the encoder-generator network from learning identity functions.

5.5 Inpainting

We corrupt inputs in various ways and observe the reconstruction quality of our network: 25% salt and pepper noise, 20 × 20 pixel holes in the center of the clip, and 20 × 20 pixel holes at random positions. We trained our network on stabilized golf videos, and evaluate it on the unstabilized airplane dataset as shown in Fig. 5.

Denoising salt and pepper corruptions is a well-studied problem, going back many years [65]. State-of-the-art approaches operate on noise levels as high as 70% [66]. The denoised reconstructions generated by our model are sharp and accurate. We can use our model – which has been trained on stabilized videos – to denoise clips with moving cameras or backgrounds, which would not be possible with a two-stream architecture. The reconstructed output is slightly blurrier than the ground-truth, which we attribute to the fact that we generate the entire video from a latent encoding and do not keep the undamaged parts of the input.

The task of hole-filling is more challenging since the reconstructions have to be consistent in both space and time. While we do not claim to compete with the state-of-the-art, we use it to illustrate that our network learns advanced spatial and temporal dependencies. For instance, in the second clip and second column of Fig. 5 we can see that, although the airplane's pitch elevator is mostly covered in the input, it is reconstructed almost perfectly and not split into two halves. This usually works best when the object covered is visible on more than one side of the box. We sometimes observe that such objects disappear although we could infer their existence from symmetry (e.g. one airplane wing is covered and not reconstructed). Our model learns temporal dependencies, as objects which are covered in some—but not all frames—are reconstructed consistently over time. The overall quality does not suffer significantly when randomizing the locations of the boxes.

Our quantitative evaluations results are shown in Tab. 2. We asked workers on Amazon MTurk to rate how realistic reconstructions look. Consistently with our quantitative findings, users rate the salt & pepper reconstructions with a score of 3.63 very high (real videos score 4.10). The margin between boxes at fixed and random positions is very small and not significant. Furthermore, we calculate the peak signal to noise ratio between ground-truth videos and their reconstructed counterparts. Salt and pepper reconstructions achieve again the best score. The margin between boxes at fixed and boxes at random positions is too small to rank the models. All three models perform better on hold-out data than on the out-of-domain data.

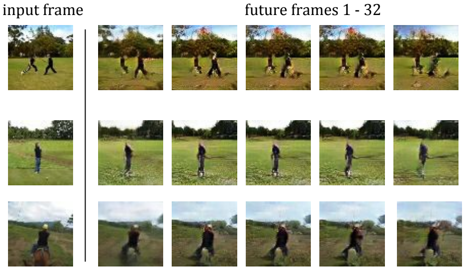

Fig. 6. Future prediction results: Generated videos and the input frames the generations were conditioned on. The *first* row shows two people who seem to fight. In person in the *second* row seems to start walking. The Person in the *third* row rides a horse; the horse is dropped in the future frames but the person moves

5.6 Future Prediction

We qualitatively show results of our future prediction network in Fig. 6. Future frames are blurrier, compared to the inpainting and colorization results, which we attribute to the fact that the reconstruction loss only guides the first frame of the generated clip – not the entire clip.

Although in many cases the network fails to generate a realistic future, it often learns which objects should move and generates fairly plausible motions. Since we use only one frame guiding the generation and omit to use the ground-truth future, these semantics are solely learned by the adversarial loss function. We emphasize that, to the best of our knowledge, this work and [5] are the only two approaches using a single input frame to generate multiple future frames. We suffer from the same problems as [5], such as hallucinating or omitting objects. For example, the horse in the bottommost clip in Fig. 6 is dropped in future frames. Unsupervised future prediction from a single frame is a notoriously hard task. Nonetheless, our network learns which objects are likely to move, and to generate fairly plausible motions.

6 Conclusion and Outlook

This paper proposed a robust video generation model that generalizes the state-of-the-art Wasserstein GAN technique to videos, by designing a new one-stream generative model. Our extensive qualitative and quantitative evaluations show that our stable one-stream architecture outperforms the Video GAN and Temporal GAN models on multiple challenging datasets. Further, we have verified that one-stream video generation can work within a suitable framework and stable architecture. The proposed iVGAN model does not need to distinguish between foreground and background or dynamic and static patterns and is the only architecture able to generate videos with moving camera/background, as well as those with a static background. Although our architecture does not explicitly model the fact that our world is stationary, it correctly learns which objects might plausibly move and how.

Additionally, dropping the assumption of a static background frees our model to handle data that is not background-stabilized, thus significantly broadening its applicability. We emphasized the superiority of our model by demonstrating that our proposed multi-functional extension is applicable to several distinct applications, each of them requiring our network to learn different semantics. Our video colorization experiments indicate that the model is able to select individual parts of a scene, recognize them, and paint them accordingly. The inpainting experiments show that our model is able to learn and recover important temporal and spatial dependencies by filling the damaged holes consistently, in both space and time. We trained our models on stabilized input frames in both applications and successfully applied them to unprocessed videos. A two-stream model would by design not be able to colorize or inpaint clips exhibiting background or camera motion.

Although unsupervised understanding of videos is still in its infancy, we have presented a more general and robust video generation model that can be used as a multi-functional framework. Nevertheless, we believe that the quality of the generated videos can be further improved by using deeper architectures like ResNet [54] or DenseNet [67], or by employing recent progressive growing techniques of GANs [14].

References

1. Vondrick, C., Torralba, A.: Generating the future with adversarial transformers. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
2. Walker, J., Gupta, A., Hebert, M.: Dense optical flow prediction from a static image. In: 2015 IEEE International Conference on Computer Vision (ICCV). (2015) 2443–2451

3. Walker, J., Doersch, C., Gupta, A., Hebert, M.: An uncertain future: Forecasting from static images using variational autoencoders. In: European Conference on Computer Vision, Springer (2016) 835–851
4. Walker, J., Gupta, A., Hebert, M.: Patch to the future: Unsupervised visual prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 3302–3309
5. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: Advances In Neural Information Processing Systems. (2016) 613–621
6. Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: Mocogan: Decomposing motion and content for video generation. arXiv preprint arXiv:1707.04993 (2017)
7. Saito, M., Matsumoto, E., Saito, S.: Temporal generative adversarial nets with singular value clipping. In: IEEE International Conference on Computer Vision (ICCV). (2017) 2830–2839
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680
9. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
10. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems. (2016) 2234–2242
11. Che, T., Li, Y., Jacob, A.P., Bengio, Y., Li, W.: Mode regularized generative adversarial networks. arXiv preprint arXiv:1612.02136 (2016)
12. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
13. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems. (2017) 5769–5779
14. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
15. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Advances in neural information processing systems. (2015) 1486–1494
16. Im, D.J., Kim, C.D., Jiang, H., Memisevic, R.: Generating images with recurrent adversarial networks. arXiv preprint arXiv:1602.05110 (2016)
17. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. arXiv preprint (2016)
18. Jurie, F.: A new log-polar mapping for space variant imaging.: Application to face detection and tracking. Pattern Recognition (1999) 865–875
19. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593 (2017)
20. Chia, A.Y.S., Zhuo, S., Gupta, R.K., Tai, Y.W., Cho, S.Y., Tan, P., Lin, S.: Semantic colorization with internet images. In: ACM Transactions on Graphics (TOG), ACM (2011) 156
21. Huang, Y.C., Tung, Y.S., Chen, J.C., Wang, S.W., Wu, J.L.: An adaptive edge detection based colorization algorithm and its applications. In: Proceedings of the 13th annual ACM international conference on Multimedia, ACM (2005) 351–354
22. Ironi, R., Cohen-Or, D., Lischinski, D.: Colorization by example. In: Rendering Techniques. (2005) 201–210
23. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: ACM transactions on graphics (tog), ACM (2004) 689–694

24. Luan, Q., Wen, F., Cohen-Or, D., Liang, L., Xu, Y.Q., Shum, H.Y.: Natural image colorization. In: Proceedings of the 18th Eurographics conference on Rendering Techniques, Eurographics Association (2007) 309–320
25. Yatziv, L., Sapiro, G.: Fast image and video colorization using chrominance blending. IEEE Transactions on Image Processing (2006) 1120–1129
26. Charpiat, G., Hofmann, M., Schölkopf, B.: Automatic image colorization via multimodal predictions. In: European conference on computer vision, Springer (2008) 126–139
27. Gupta, R.K., Chia, A.Y.S., Rajan, D., Zhiyong, H.: A learning-based approach for automatic image and video colorization. arXiv preprint arXiv:1704.04610 (2017)
28. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM (2001) 327–340
29. Liu, X., Wan, L., Qu, Y., Wong, T.T., Lin, S., Leung, C.S., Heng, P.A.: Intrinsic colorization. ACM Transactions on Graphics (TOG) **27**(5) (2008) 152
30. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. In: ACM Transactions on Graphics (TOG), ACM (2002) 277–280
31. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. European Conference on Computer Vision (2016) 649–666
32. Koo, S.: Automatic colorization with deep convolutional generative adversarial networks (2016)
33. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. (2000) 417–424
34. Komodakis, N.: Image completion using global optimization. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 1., IEEE (2006) 442–452
35. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., Li, H.: High-resolution image inpainting using multi-scale neural patch synthesis. arXiv preprint arXiv:1611.09969 (2016)
36. Tang, N.C., Hsu, C.T., Su, C.W., Shih, T.K., Liao, H.Y.M.: Video inpainting on digitized vintage films via maintaining spatiotemporal continuity. IEEE Transactions on Multimedia (2011) 602–614
37. Granados, M., Kim, K.I., Tompkin, J., Kautz, J., Theobalt, C.: Background inpainting for videos with dynamic objects and a free-moving camera. In: European Conference on Computer Vision, Springer (2012) 682–695
38. Ebdelli, M., Le Meur, O., Guillemot, C.: Video inpainting with short-term windows: application to object removal and error concealment. IEEE Transactions on Image Processing (2015) 3034–3047
39. Le, T.T., Almansa, A., Gousseau, Y., Masnou, S.: MOTION-CONSISTENT VIDEO INPAINTING. In: ICIP 2017: IEEE International Conference on Image Processing. ICIP 2017: IEEE International Conference on Image Processing, Beijing, China (September 2017)
40. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Trans. Graph. **28**(3) (2009) 24–1
41. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision. Volume 2., IEEE (1999) 1033–1038
42. Hays, J., Efros, A.A.: Scene completion using millions of photographs. In: ACM Transactions on Graphics (TOG), ACM (2007) 4
43. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2536–2544

44. Chao, Y.W., Yang, J., Price, B., Cohen, S., Deng, J.: Forecasting human dynamics from static images. In: IEEE Conference on Computer Vision and Pattern Recognition. (2017)
45. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction. In: Advances in neural information processing systems. (2016) 64–72
46. Fragkiadaki, K., Levine, S., Malik, J.: Recurrent network models for kinematic tracking. CoRR, abs/1508.00271 (2015) 4
47. Kalchbrenner, N., Oord, A.v.d., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A., Kavukcuoglu, K.: Video pixel networks. arXiv preprint arXiv:1610.00527 (2016)
48. Luo, Z., Peng, B., Huang, D.A., Alahi, A., Fei-Fei, L.: Unsupervised learning of long-term motion dynamics for videos. arXiv preprint arXiv:1701.01821 **2** (2017)
49. Xue, T., Wu, J., Bouman, K., Freeman, B.: Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In: Advances in Neural Information Processing Systems. (2016) 91–99
50. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. arXiv preprint arXiv:1511.05440 (2015)
51. Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., Chopra, S.: Video (language) modeling: a baseline for generative models of natural videos. arXiv preprint arXiv:1412.6604 (2014)
52. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence (2013) 221–231
53. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2010) 2528–2535
54. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
55. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. (2015) 1026–1034
56. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. (2015) 448–456
57. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
58. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)
59. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
60. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
61. Zhao, J., Mathieu, M., LeCun, Y.: Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126 (2016)
62. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: Yfcc100m: The new data in multimedia research. Communications of the ACM (2016) 64–73
63. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Advances in neural information processing systems. (2014) 487–495
64. Real, E., Shlens, J., Mazzocchi, S., Pan, X., Vanhoucke, V.: Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. arXiv preprint arXiv:1702.00824 (2017)

65. Chen, T., Ma, K.K., Chen, L.H.: Tri-state median filter for image denoising. IEEE Transactions on Image processing (1999) 1834–1838
66. Lu, C.T., Chou, T.C.: Denoising of salt-and-pepper noise corrupted image using modified directional-weighted-median filter. Pattern Recognition Letters (2012) 1287–1295
67. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks. arXiv preprint arXiv:1608.06993 (2016)