WALDO: Future Video Synthesis using Object Layer Decomposition and Parametric Flow Prediction

Guillaume Le Moing[1, 2, *] Jean Ponce[2, 3] Cordelia Schmid[1, 2]

[1]Inria [2]Département d'informatique de l'ENS (CNRS, ENS-PSL, Inria) [3]Center for Data Science New York University

Abstract

This paper presents WALDO (WArping Layer-Decomposed Objects), a novel approach to the prediction of future video frames from past ones. Individual images are decomposed into multiple layers combining object masks and a small set of control points. The layer structure is shared across all frames in each video to build dense inter-frame connections. Complex scene motions are modeled by combining parametric geometric transformations associated with individual layers, and video synthesis is broken down into discovering the layers associated with past frames, predicting the corresponding transformations for upcoming ones and warping the associated object regions accordingly, and filling in the remaining image parts. Extensive experiments on multiple benchmarks including urban videos (Cityscapes and KITTI) and videos featuring nonrigid motions (UCF-Sports and H3.6M), show that our method consistently outperforms the state of the art by a significant margin in every case. Code, pretrained models, and video samples synthesized by our approach can be found in the project webpage.[1]

1. Introduction

Predicting the future from a video stream is an important tool to make autonomous agents more robust and safe. In this paper, we are interested in the case where future frames are synthesized from a fixed number of past ones. One possibility is to build on advanced image synthesis models [20, 23, 44, 59] and adapt them to predict new frames conditioned on past ones [79]. Extending these already memory- and compute-intensive methods to our task may, however, lead to prohibitive costs due to the extra temporal dimension. Hence, the resolution of videos predicted by this approach is often limited [36, 95, 96]. Other works resort to compression [64, 82] to reduce computations [63, 97, 103], at the potential cost of poorer temporal consistency [61].

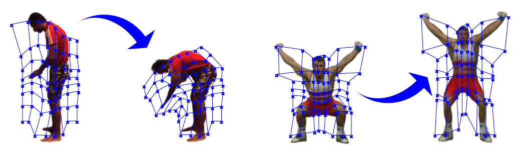


Figure 1. **WALDO** synthesizes future frames by deforming grids of control points, automatically associated with different objects.

Our work relies instead on semantic and motion cues extracted from the past to model complex dynamics in high resolution and predict the future. Wu *et al*. [99] decompose scenes into objects/background, predict affine transformations for the objects and non-affine ones for the background, and warp the last input frame to produce new ones. Bei *et al*. [7] predict dense flow maps for individual semantic regions. Geng *et al*. [29] augment classical frame reconstruction losses with flow-based correspondences between pairs of frames. Wu *et al*. [100] build on a pretrained video frame interpolation model which they adapt to future video prediction. Contrary to [99], our model automatically discovers the object decomposition without explicit supervision. Moreover, rather than directly predicting optical flow at each pixel [7, 99, 100], we use thin-plate splines (TPS) [9] as a parametric model of per layer flow for any pair of frames. This improves robustness since we predict future frames using all past ones as opposed to [7, 29, 99, 100]. In addition, TPS provide optical flow at any resolution, and allow using a lower resolution for fast training while retaining good performance with high-resolution inputs at inference. Lastly, a few time-dependent control points associated with each object and the background are sufficient to parameterize TPS. This compact yet expressive representation of motion allows us to break down video synthesis into: (a) discovering the layers associated with past frames, (b) predicting the corresponding transformations for upcoming ones while handling complex motions [13] and modeling future uncertainty if needed [1, 2], and (c) warping the associated object regions accordingly and filling in the remain-

*corresponding author: guillaume.le-moing@inria.fr
[1]url: https://16lemoing.github.io/waldo

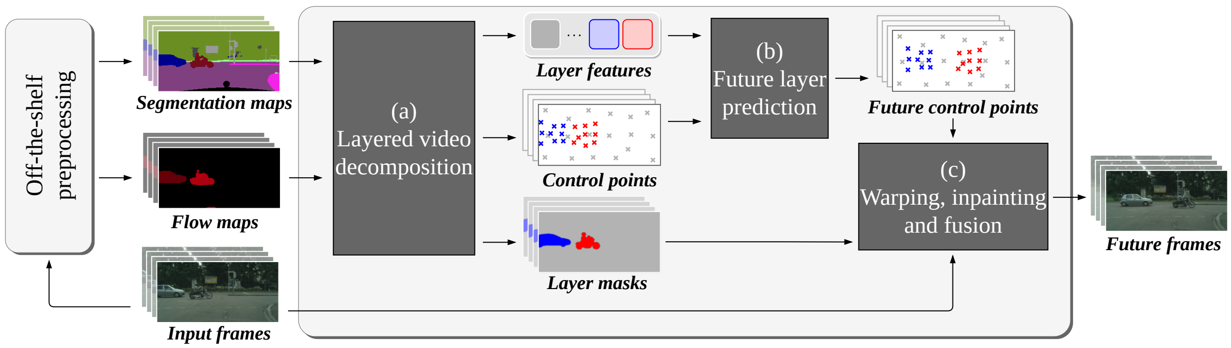


Figure 2. **Overview of WALDO.** Given an input video sequence and associated semantic segmentation and optical flow maps preprocessed by off-the-shelf models [14, 77], our approach breaks down video synthesis into **(a) layered video decomposition:** using semantic and motion cues to decompose the sequence into layers represented by both object masks and features, with spatial information in the form of a small set of control points, **(b) future layer prediction:** predicting the new position of these points in the target output frames, and **(c) warping, inpainting and fusion:** using the corresponding offset and a thin-plate spline deformation model to warp the input frames and object masks, merge the corresponding regions, and fill in the empty image parts. Our model is trained, without explicit annotations, on a set of videos of $T+K$ frames by using the first T to predict the next K. At inference, a (potentially greater) number K of frames is predicted from T input ones by repeating (a), (b) and (c) in an autoregressive fashion if needed.

ing image parts. By combining these three components, our approach (Figure 2) to predict future frames by WArping Layer-Decomposed Objects (WALDO) from past ones sets a new state of the art on diverse benchmarks including urban scenes (Cityscapes [17] and KITTI [28]), and scenes featuring nonrigid motions (UCF-Sports [67] and H3.6M [41]). Our main contributions are twofold:

• **Much broader operating assumptions:** Previous approaches to video prediction that decompose individual frames into layers assume prior foreground/background knowledge [99] or reliable keypoint detection [88, 106], and they do not allow the recovery of dense scene flow at arbitrary resolutions for arbitrary pairs of frames [1, 2, 7, 13, 29, 99, 100]. Our approach overcomes these limitations. Unlike [7, 13, 29, 99, 100], it also allows multiple predictions.

• **Novelty:** The main novelties of our approach to video prediction are (a) a layer decomposition algorithm that leverages a transformer architecture to exploit long-range dependencies between semantic and motion cues; (b) a low-parameter deformation model that allows the long-term prediction of sharp frames from a small set of adjustable control points; and (c) the effective fusion of multiple predictions from past frames using state-of-the-art inpainting to handle (dis)occlusion. Some of these elements have been used separately in the past (e.g., [61, 104, 106]), but never, to the best of our knowledge, in an integrated setting.

These contributions are validated through extensive experiments on urban datasets (Cityscapes and KITTI), as well as scenes featuring nonrigid motions (UCF-Sports, H3.6M), where our method outperforms the state of the art by a significant margin in *every case*

2. Related work

Video prediction ranges from unconditional synthesis [4, 16, 19, 24, 25, 47, 69, 79, 80, 86] to multi-modal and controlled prediction tasks [32, 33, 38, 61, 97]. Here, we intend to exploit the temporal redundancy of videos by tracking the trajectories of different objects. One may infer future frames by extrapolating the position of keypoints associated to target objects [88, 106], but this requires manual labeling. Some works [12, 21, 34, 37, 60] propose instead to let structured object-related information naturally emerge from the videos themselves. We use a similar strategy but also rely on off-the-shelf models to extract semantic and motion cues in the hope of better capturing the scene dynamics [7, 29, 71, 90, 99, 100]. Without access to ground-truth objects, we discover them via layered video decomposition.

Layered video decompositions, introduced in [89], have been applied to optical flow estimation [75, 101], motion segmentation [66, 104], and video editing [43, 56, 105]. They are connected to object-centric representation learning [5, 10, 22, 31, 46, 54], where the compositional structure of scenes is also essential. Like [104], we use motion in the form of optical flow maps to decompose videos into objects and background. We go beyond the single-object scenarios they tackle, and propose a decomposition scheme which works well on real-world scenarios like urban scenes, with multiple objects and complex motions. In addition, we associate with every layer a geometric transformation allowing the recovery of the flow between past and future frames.

Spatial warps, as implemented in [42], have proven useful for various tasks, *e.g.*, automatic image rectification for text recognition [72], semantic segmentation [26], and the contextual synthesis of images [62, 65, 112] or videos [1, 2, 6, 7, 27, 33, 52, 57, 87, 99, 100]. Inspired by D'Arcy Thompson's pioneering work in biology [78] as well as the shape contexts of Belongie et al. [8] , we parameterize the warp with thin-plate splines (TPS) [9], whose parameters are motion vectors sampled at a small set of control points. TPS allow optical flow recovery by finding the transformations of minimal bending energy which complies with points motion. This has several advantages: The flow is differentiable with respect to motion vectors; deformations are more general than affine ones; and the number of control points allow to trade off deformation expressivity for parameter size.

3. Proposed method

Notation. We use a subscript t in $[\![1, T{+}K]\!]$ for *time*, with frames 1 to $T{+}K$ available at train time, and the last K predicted from the first T at inference time. We use a superscript i in $[\![0, N]\!]$ for *image layer*, where $i{=}0$ represents the background and $i{>}0$ represents an object. For example, we denote by p_t^i the control points associated with layer i in frame t, by p_t those associated with all layers at time t, and by p^i those associated with layer i over all time steps.

Overview. We consider a video X consisting of $T{+}K$ RGB frames x_1 to x_{T+K} with spatial resolution $H \times W$. Each frame is decomposed into $N{+}1$ layers tracking objects and background motions over time. Layers are represented by pairs (m_t^i, p_t^i), where m_t^i is a (soft) mask indicating for every pixel the presence of object i (or background if $i{=}0$), and p_t^i is the set of control points associated with layer i at time t. Our approach (Figure 2) consists of: (a) decomposing frames 1 to T of video X into layers (Sec. 3.1); (b) predicting the decompositions up to time $T{+}K$ (Sec. 3.2); and (c) using them to warp each of the first T frames of X into the K future time steps, and finally predicting frames $T{+}1$ to $T{+}K$ by fusing the T views for each future time step and filling in empty regions (Sec. 3.3). The three corresponding modules are trained separately, starting with (a), and then using (a) to supervise (b) and (c).

Off-the-shelf preprocessing. Similar to [5, 104], we adopt a motion-driven definition of *objectness* where an object is defined as a spatially-coherent region which follows a smooth deformation over time, such that discontinuities in scene motion occur at layer boundaries. We compute for the first T frames of the video the corresponding backward flow maps $F = [f_1, \ldots, f_T]$ with an off-the-shelf method [77], where f_t is the translation map associating with every pixel of frame x_t the vector of $\mathbb{R}^2$ pointing to the matching location in x_{t-1}. We also suppose that each object has a unique semantic class out of C and compute for the first T frames the corresponding semantic segmentation maps $S = [s_1, \ldots, s_T]$ again with an off-the-shelf method [14], where s_t assigns to every pixel a label (*e.g.*, car, road, buildings, sky) represented by its index in $[\![1, C]\!]$.

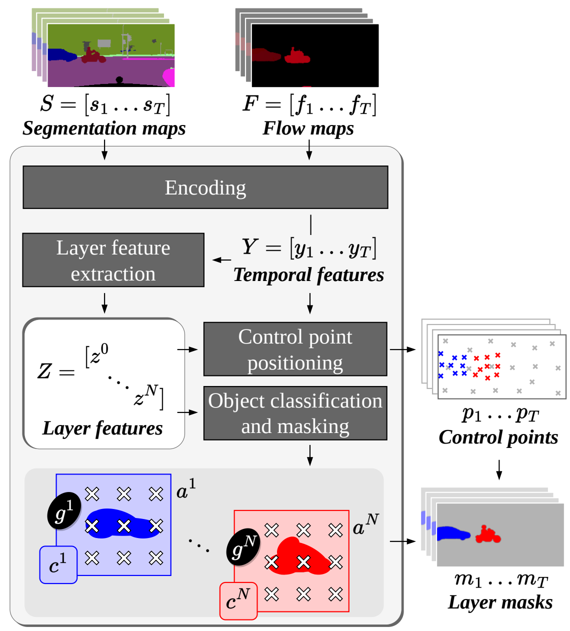


Figure 3. **Layered video decomposition.** Semantic and motion cues are mapped to temporal features with a time-independent encoder. These are combined to form layer features representing the background and the objects for the whole video. The position of individual layers at each time step is determined by a sparse set of control points, predicted from temporal features and layer features. We further associate with the layers a soft mask and a semantic class which are combined with the information of position to produce layer masks, segmenting the objects from the background.

3.1. Layered video decomposition

We associate with each layer i in $[\![0, N]\!]$ a (soft) object ($i{>}0$) or background ($i{=}0$) mask a^i of size $H^i \times W^i$ over which is overlaid a coarse $h^i \times w^i$ regular grid g^i of control points.[2] Deformed *layer masks* m_t^i are obtained by mapping the points in g^i onto their positions p_t^i at time step t and applying the corresponding TPS transformation to a^i. The decomposition module (Figure 3) maps the segmentation and flow maps S and F onto the positioned control points p_1 to p_T and the deformed layer masks m_1 to m_T.

3.1.1 Architecture

Input encoding. Input flow maps and segmentation maps F and S are fed to a time-independent encoder, implemented by a convolutional neural network (CNN) which outputs temporal feature maps $Y = [y_1, \ldots, y_T]$. Each map y_t lies in $\mathbb{R}^{d \times h \times w}$ with feature dimension d and downscaled spatial resolution $h \times w$ such that $hw \ll HW$. We denote by $l = hw$ the latent feature size, and reshape feature y_t to an $l \times d$ matrix through raster-scan reordering.

[2] In practice we use larger values of H^i, W^i for the background (same as H, W) than for object layers, and we fix the ratio H^i/h^i to 16.

Layer feature extraction. We form layer features $Z = [z^0, \ldots, z^N]$ from temporal ones Y, where, for each layer, z^i is an $l^i \times d$ matrix with $l^i = h^i w^i$. We implement this with a transformer [83], where self-attention is replaced by a binding mechanism, discovering object-centric features by iteratively grouping intra-object pixels together, as in [54].

Control point positioning. We add a third dimension to the control points p_t^i positioned in the frame x_t to record the corresponding layer "depth" ordering $o_t^i \geq 0$, with $o_t^i = 0$ for the background. In practice, the 3D vectors associated with the points p_t are once again predicted by a transformer, from the set Z of layer features and the temporal feature y_t, thus accounting for possible interactions between layers. Control points are typically close to the associated objects, but it is still fine if some end up outside an object mask. Their role is to recover a dense deformation field through TPS interpolation, and what matters is only that the part of this field within the object mask is correct. Figure 1 shows grids of control points extracted by our approach. Note how their structure automatically adapts to fit precise motions.

Layer mask prediction. A CNN maps layer features z^i onto soft masks a^i with values in $[0, 1]$ corresponding to opaqueness and defined in their own "intrinsic" coordinate systems, in which points associated with p_t^i lie on a regular grid g^i. The background is fully opaque ($a^0 = \mathbf{1}$). At time t, a^i is warped onto the corresponding layer mask m_t^i using the TPS transformation w_t^i mapping the grid points of g^i onto their positions p_t^i in frame x_t. We then improve object contours by semantic refinement using segmentation maps S. Concretely, given a mask m_t^i corresponding to an object layer ($i > 0$), a soft class assignment c^i in $[0, 1]^C$ is obtained from z^i with a fully-connected layer, then used to update m_t^i by comparing c^i to the actual class in s_t. We finally use the ordering scores o_t and a classical occlusion model [43, 62] to filter non-visible layer parts. More details about this process are in Appendix A-D.

3.1.2 Training procedure

The goal of the decomposition module is to discover layers whose associated masks and control points best reconstruct scene motion. We achieve this by minimizing the objective:

$$\mathcal{L}_d = \lambda_o \mathcal{L}_o + \lambda_f \mathcal{L}_f + \lambda_r \mathcal{L}_r, \tag{1}$$

with an object discovery loss $\mathcal{L}_o$ to encourage objects from different layers to occupy moving foreground regions; a flow reconstruction loss $\mathcal{L}_f$ to ensure the temporal consistency of the learned decompositions; and a regularization loss $\mathcal{L}_r$. At train time, we extract layer features from the first T frames, but also position layers in the K subsequent ones to have a supervision signal for later stages (Secs. 3.2 and 3.3). At inference, only the first T frames are used.

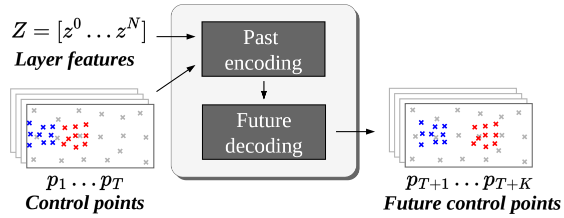


Figure 4. **Future layer prediction.** We encode past control point positions and object-specific knowledge (*e.g.*, shape and semantics) in the form of layer features to produce future positions.

Object discovery. Without ground-truth objects for training, we discover reasonable candidates using semantic and motion cues, f_t and s_t, by positioning objects in $\mathcal{M}(s_t, f_t)$, a binary mask indicating *foreground* regions (*e.g.*, cars) with significant motion compared to *stuff* regions (*e.g.*, road), written $\mathcal{S}(s_t)$. We get the mask $\mathcal{M}(s_t, f_t)$ by extrapolating the full background flow from f_t in $\mathcal{S}(s_t)$, and by thresholding with a constant τ_m the L_1 distance between background and foreground flows. The object discovery loss is:

$$\mathcal{L}_o = \sum_t (k_s \mathcal{S}(s_t) - k_m \mathcal{M}(s_t, f_t)) \odot (\max_{i>0} m_t^i), \tag{2}$$

where k_m and k_s are positive scalars, weighting the attraction of discovered objects $m_t^{i>0}$ towards moving *foreground* regions ($\mathcal{M}$) and the repulsion from *stuff* regions ($\mathcal{S}$).

Flow reconstruction. We reconstruct the backward flow $w_{t_2 \leftarrow t_1}$ between consecutive time steps t_1 and t_2, by considering the individual layer warps, denoted $w_{t_2 \leftarrow t_1}^i$ and computed as $w_{t_2}^i \circ {w_{t_1}^i}^{-1}$ where $w_{t_1}^i$ and $w_{t_2}^i$ are obtained through the TPS transformation associated with $p_{t_1}^i$ and $p_{t_2}^i$. We recover $w_{t_2 \leftarrow t_1}$ by compositing layer warps, *i.e.*, $w_{t_2 \leftarrow t_1} = \sum_i m_{t_2}^i \odot w_{t_2 \leftarrow t_1}^i$, where the mask $m_{t_2}^i$ determines layer transparency. The flow reconstruction loss is:

$$\mathcal{L}_f = \sum_t \|f_t - \hat{f}_t\|_1, \text{ with } \hat{f}_t = w_{t \leftarrow t-1}. \tag{3}$$

Regularization. The last objective $\mathcal{L}_r$ is composed of: an entropy term applied to layer mask m_t to ensure that a single layer prevails for every pixel, and an object initialization term which is the L_2 distance from regions of interest $\mathcal{M}(s_t, f_t)$ which are still empty (as per $m_t^{i>0}$) to the control points p_t^i of the nearest object ($i > 0$).

3.2. Future layer prediction

Thanks to our layered video decomposition, the prediction of future layers (Figure 4) is reduced to inferring, from past control points $[p_1, \ldots, p_T]$, the position of future ones.

Architecture. For each time step $t \leq T$ and each layer i, p_t^i is mapped onto a vector in $\mathbb{R}^d$ by a linear layer. Likewise, each z^i is also mapped onto a vector in $\mathbb{R}^d$. We concatenate

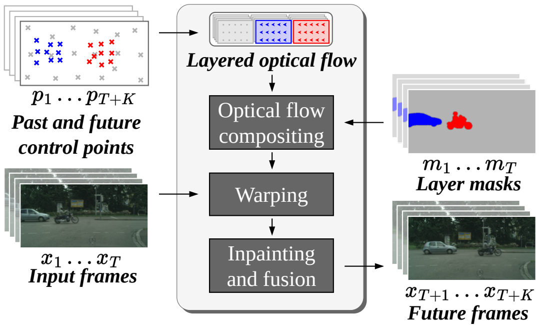


Figure 5. **Warping, fusion and inpainting.** From past and future control point positions we compute the warps associated with each layer and composite them to recover the dense scene flow by leveraging past layer masks for setting their transparency. We then warp frames from the past to form multiple views of future ones, merge these together, and fill in empty regions to produce final frames.

and feed these vectors to a two-stage transformer to construct representations for the K future time steps by combining self-attention modules applied to intermediate future representations and cross-attention modules between past and future ones. The prediction of the control points p_t^i associated with each layer in the K future time steps is done with a linear layer, which does not output their position directly, but rather their displacement with respect to p_T, with T being the last known time step from the past context.

Training procedure. We extract training data for past and future time steps by decomposing videos of length $T{+}K$ as described in Sec. 3.1. We then mask the control points corresponding to the last K time steps and train the future prediction module to reconstruct them by minimizing $\mathcal{L}_p$, the L_1 distance between extracted and reconstructed control points for time steps $T{+}1$ to $T{+}K$. Under uncertainty, $\mathcal{L}_p$ makes points converge towards an *average* future trajectory. When interested in predicting multiple futures, we add noise (as input and in attention modules) and an adversarial term [30] to $\mathcal{L}_p$. More details are in Appendix E.

3.3. Warping, fusion and inpainting

Last comes the actual synthesis of future frames from past ones (Figure 5), using the layer decompositions extracted from the past, and the ones predicted into the future.

Architecture. Given the predicted control points, we compute, for every layer i and pair of time steps (t_1, t_2) in $[\![1, T]\!] \times [\![T{+}1, T{+}K]\!]$, the layer warp $w^i_{t_2 \leftarrow t_1}$ from t_1 to t_2 as described in Sec. 3.1. We construct $m^i_{t_2}$ by warping $m^i_{t_1}$ with $w^i_{t_2 \leftarrow t_1}$. We compute $w_{t_2 \leftarrow t_1}$ from layer warps and future masks by $w_{t_2 \leftarrow t_1} = \sum_i m^i_{t_2} \odot w^i_{t_2 \leftarrow t_1}$ as before, and warp past frames to produce multiple views of future ones, one for each pair of time steps (t_1, t_2). We obtain T views for each future frame, with different missing regions due to disocclusion. A fusion network, implemented by a U-Net [68], merges these views together according to pixel-level scores predicted for each of them. Regions which remain empty are then filled with an off-the-shelf inpainting network [51], to obtain future frame predictions $\hat{x}_{T+1}$ to $\hat{x}_{T+K}$. We ensure temporal consistency by filling frames one at a time and by using predicted motions to propagate the newly filled content onto the next frames.

Training procedure. The inpainting model [51], trained on 8M images from the Places dataset [111], is kept frozen and the objective to train the fusion U-Net is the weighted L_1 distance in pixel space and between features $\mathcal{F}$ extracted using the VGG [74] classification network trained on [18]:

$$\mathcal{L}_u = \sum_{t=T+1}^{T+K} \lambda_p \|x_t - \hat{x}_t\|_1 + \lambda_v \|\mathcal{F}(x_t) - \mathcal{F}(\hat{x}_t)\|_1, \quad (4)$$

4. Experiments

Datasets. We train WALDO on two urban datasets, Cityscapes [17], which contains 2975 30-frame video sequences for training and 500 for testing captured at 17 FPS, and KITTI [28], with a total of 156 longer video sequences ($\sim$340 frames each) including 4 for testing. We use suitable resolutions and train / test splits for fair comparisons with prior works (setup from [7] in Table 1 and from [1] in Table 2). We also train on nonrigid scenes from UCF-Sports [67] (resp. H3.6M [41]) using the splits from [13] consisting of 6288 sequences (resp. 73404) for training and 752 (resp. 8582) for testing with roughly 10 frames per sequence. We extract semantic and motion cues using pre-trained models, namely, DeepLabV3 [14] and RAFT [77].

Evaluation metrics. We evaluate the different methods with the multi-scale structure similarity index measure (SSIM) [94], the learned perceptual image patch similarity (LPIPS) [109], and the peak signal-to-noise ratio (PSNR), all standard image reconstruction metrics for evaluating video predictions. We also use the Fréchet video distance (FVD) [81] to estimate the gap between real and synthetic video distributions. We report in Tables 1-3 the mean and standard deviation for 3 randomly-seeded training sessions.

Implementation details. For reproducibility, code and pretrained models are available on our project webpage.[3] WALDO is trained with the ADAM optimizer [45] and a learning rate of 10^{-4} on 4 NVIDIA V100 GPUs for about a week. We set $(\lambda_o, \lambda_f, \lambda_r, \lambda_p, \lambda_v){=}(1, 100, 1, 1, 1)$, $(k_s, k_m){=}(0.25, 1)$ and $\tau_m{=}0.005$ by validating the performance on a random held-out subset of the training data. For example, we use spatial resolutions of 128×256 for training (a) the layered video decomposition, and (b) the future layer prediction module; and 512×1024 for training (c) the

[3] https://16lemoing.github.io/waldo

Table 1. Comparison to state-of-the-art deterministic methods on Cityscapes and KITTI test sets. We compute multi-scale SSIM ($\times 10^3$) and LPIPS ($\times 10^3$) for the k^{th} future frame and average for k in $[\![1, K]\!]$. We indicate if methods use semantic or flow ground truths for training.

Method	Sem.	Flow	(a) Cityscapes (512×1024) $K=1$ SSIM ↑	$K=1$ LPIPS ↓	$K=5$ SSIM ↑	$K=5$ LPIPS ↓	$K=10$ SSIM ↑	$K=10$ LPIPS ↓	$K=10$ FVD ↓	(b) KITTI (256×832) $K=1$ SSIM ↑	$K=1$ LPIPS ↓	$K=3$ SSIM ↑	$K=3$ LPIPS ↓	$K=5$ SSIM ↑	$K=5$ LPIPS ↓	$K=5$ FVD ↓
PredNet [55]			840	260	752	360	663	522	-	563	553	514	586	475	629	-
MCNet [85]			897	189	706	373	597	451	-	753	240	635	317	554	373	-
VFlow [53]			839	174	711	288	634	366	-	539	324	469	374	426	415	-
VEST [110]			-	-	-	-	-	-	-	-	156	-	344	-	447	-
VPVFI [100]		✓	945	064	804	178	700	278	159	827	<u>123</u>	695	<u>203</u>	611	264	050
VPCL [29]		✓	928	085	<u>839</u>	150	<u>751</u>	<u>217</u>	129	820	172	<u>730</u>	220	<u>667</u>	<u>259</u>	075
Vid2vid [90]	✓	✓	882	106	751	201	669	271	-	-	-	-	-	-	-	-
OMP [99]	✓	✓	891	085	757	165	674	233	<u>113</u>	792	185	676	246	607	304	<u>047</u>
SADM [7]	✓	✓	**959**	<u>076</u>	835	<u>149</u>	-	-	-	<u>831</u>	144	724	246	647	312	-
WALDO	✓	✓	<u>957</u>$_{\pm 2}$	**049**$_{\pm 1}$	**854**$_{\pm 1}$	**105**$_{\pm 1}$	**771**$_{\pm 1}$	**158**$_{\pm 1}$	**055**$_{\pm 1}$	**867**$_{\pm 1}$	**108**$_{\pm 1}$	**766**$_{\pm 4}$	**163**$_{\pm 2}$	**702**$_{\pm 6}$	**206**$_{\pm 3}$	**042**$_{\pm 2}$

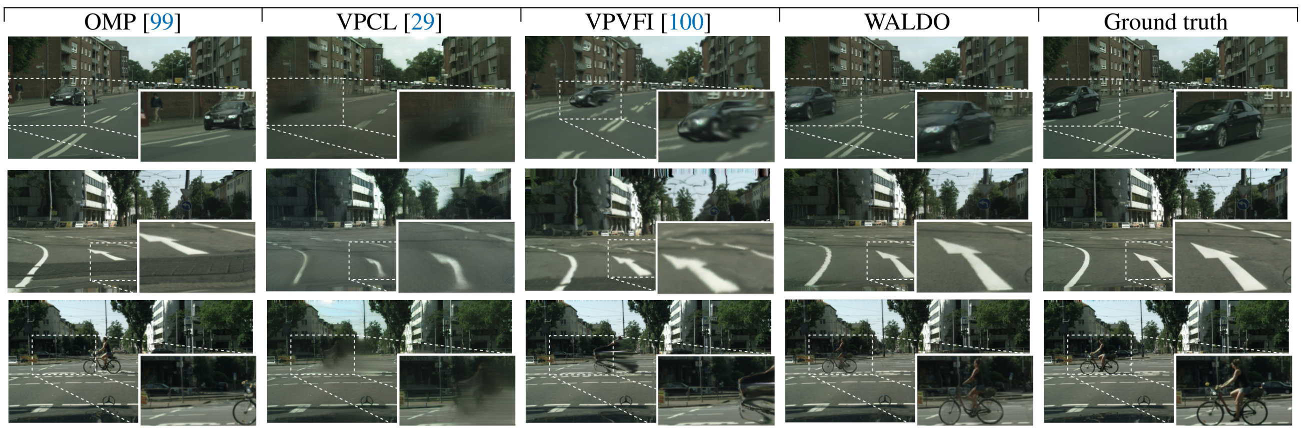


Figure 6. Comparison with [29, 99, 100] on the Cityscapes test set at time step $T{+}10$. WALDO better extracts objects from the background, better predicts their motion, and is more robust to occlusion. We strongly encourage readers to watch videos in the project webpage.

Table 2. Comparison to state-of-the-art stochastic methods ("†") on 20-frame future prediction on Cityscapes and KITTI. We report the best SSIM ($\times 10^3$), PSNR ($\times 10$), and LPIPS ($\times 10^3$) out of 100 trajectories sampled from each test sequence (except for our deterministic variant). We use 4 frames as input but others use 10.

Method	(a) Cityscapes (128×256) SSIM ↑	PSNR ↑	LPIPS ↓	(b) KITTI (92×310) SSIM ↑	PSNR ↑	LPIPS ↓
SVG† [19]	606	204	340	329	127	594
SRVP† [25]	603	210	447	336	134	635
HierVRNN† [11]	618	214	260	379	142	372
SLAMP† [1]	<u>649</u>	217	294	337	135	537
SLAMP-3D† [2]	643	214	306	383	143	501
WALDO	638$_{\pm 1}$	<u>220</u>$_{\pm 1}$	<u>158</u>$_{\pm 1}$	<u>410</u>$_{\pm 1}$	<u>145</u>$_{\pm 1}$	<u>348</u>$_{\pm 1}$
WALDO†	**653**$_{\pm 1}$	**224**$_{\pm 1}$	**147**$_{\pm 1}$	**418**$_{\pm 2}$	**147**$_{\pm 1}$	**340**$_{\pm 2}$

compositing/inpainting module on Cityscapes. As in prior works [7, 29, 90, 99], we use a context of $T{=}4$ past frames. We set the feature vector size to $d = 512$. In (a), the encoder extracts one such vector for each 16×16 patch from the input. The grids g^i of resolution 4×4 for each of the $N{=}16$ object layers and 8×16 for the background add up to a total

Figure 7. Comparison to SLAMP [1] on 50-frame prediction on the Cityscapes test set. See project webpage for videos.

of $N_c{=}384$ control points. More details are in Appendix F.

4.1. Evaluation with the state of the art

Deterministic prediction. WALDO sets a new state of the art on two urban datasets. On Cityscapes (Table 1(a)), it yields a better than 27% relative gain for LPIPS across all predicted time windows, and a significant margin for $K{>}1$ with SSIM and FVD. On KITTI (Table 1(b)), WALDO out-

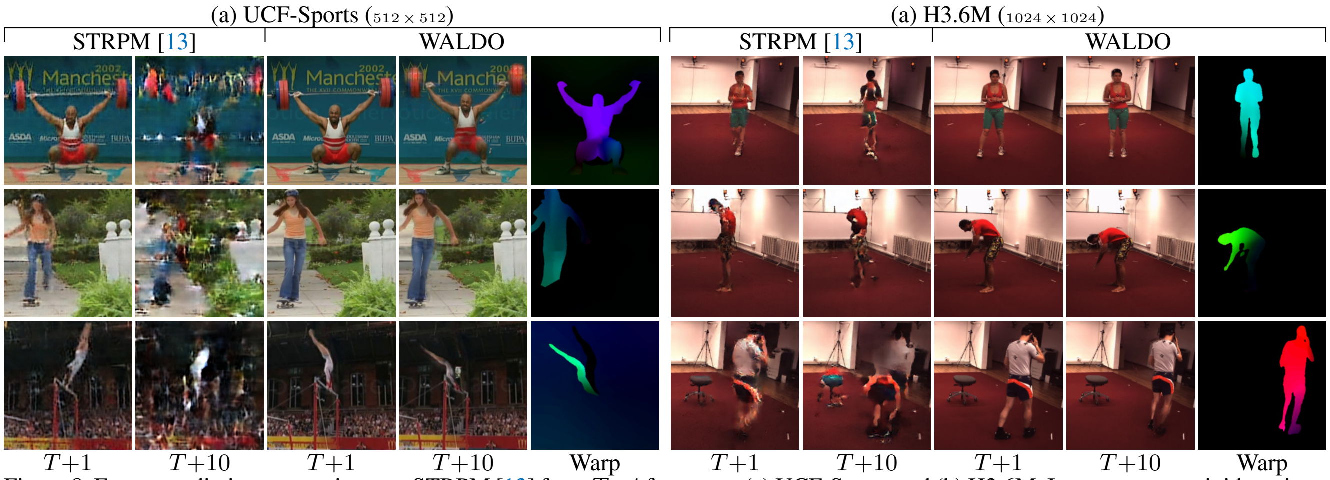


Figure 8. Future prediction comparisons to STRPM [13] from T=4 frames on (a) UCF-Sports and (b) H3.6M. In our case, nonrigid motions can be visualized by the associated warps, predicted from the control points between T and T+10 (colors represent different directions).

Table 3. Comparison with methods designed for nonrigid motions. We compute PSNR ($\times 10$), LPIPS ($\times 10^3$) for the k^{th} future frames synthesized from 4 past ones on UCF-Sports and H3.6M test sets.

Method	(a) UCF-Sports (512 × 512) PSNR ↑		(a) UCF-Sports LPIPS ↓		(b) H3.6M (1024 × 1024) PSNR ↑		(b) H3.6M LPIPS ↓	
	k=1	k=6	k=1	k=6	k=1	k=4	k=1	k=4
BMSE [58]	264	185	290	553	-	-	-	-
PRNN [93]	272	197	281	553	319	257	126	140
PRNN++ [91]	273	197	268	568	321	275	138	150
SAVP [49]	274	199	255	499	-	-	-	-
SV2P [4]	274	200	259	513	319	273	139	150
HFVP [84]	-	-	-	-	321	273	134	145
ELSTM [92]	280	203	251	478	324	277	131	139
CGAN [48]	280	200	229	449	328	283	102	110
CrevNet [108]	282	203	239	481	332	283	115	124
MRNN [98]	277	200	242	492	322	280	121	133
STRPM [13]	<u>285</u>	<u>206</u>	<u>207</u>	<u>411</u>	<u>333</u>	<u>290</u>	<u>097</u>	<u>104</u>
WALDO	**292**$_{\pm 2}$	**235**$_{\pm 1}$	**090**$_{\pm 1}$	**183**$_{\pm 1}$	**363**$_{\pm 3}$	**314**$_{\pm 1}$	**058**$_{\pm 2}$	**071**$_{\pm 1}$

performs prior methods in a quite challenging setting, with a frame rate half that of Cityscapes and a quality of precomputed semantic and motion cues poorer on KITTI. Comparisons with latest methods [29, 99, 100][4] on Cityscapes (Figure 6) show that WALDO successfully models complex object (*e.g.*, cars, bikes) and background (*e.g.*, road markings) motions with realistic and temporally coherent outputs, whereas others produce stationary or blurry videos.

Stochastic prediction. We adapt WALDO to the prediction of multiple futures by injecting noise inputs to the layer prediction module and using (at train time only) a discriminator to capture multiple modes of the distribution of synthetic trajectories. This simple procedure results in significant improvements in the stochastic setting (Table 2). In particular, WALDO outperforms other stochastic methods on the same datasets by a large margin. Interestingly, even our deterministic variant compares favourably to these approaches. Minimizing the L_1 reconstruction error (in this variant) make predictions average over all possible futures. Because we predict positions and not pixel values directly, and since averaging still yields valid positions, we obtain sharp images. On the other hand, averaging over pixels inevitably blur them out. This critical distinction is illustrated in Table 4(e). Although the loss function introduced in VPCL [29] aims at sharpening pixel predictions, the examples in Figure 6 show that WALDO is more successful.

[4]best performing method(s) in the corresponding benchmark(s) with predicted samples available, or pretrained checkpoints to synthesize them.

Long-term prediction. WALDO produces arbitrary long videos, without additional training, when used in an autoregressive mode. In 20-frame prediction (Table 2), it significantly outperforms prior works, without using its full potential since a lower resolution is used to match those of other methods. Visual comparisons with SLAMP [1][4] (Figure 7) at full resolution on 50-frame prediction (longer than the 30-frame videos in Cityscapes) are also striking.

Nonrigid prediction. To show that WALDO can handle complex motions, we retrain it on data with significant nonrigid objects, namely UCF-Sports and H3.6M datasets. Since our approach relies on TPS transformations, it can represent arbitrarily nonrigid motions when the number of control points is set accordingly [9]. By increasing the total number of per object points from 16 to 64, WALDO produces realistic videos even for deformable bodies such as human beings (Figure 8). As highlighted by the predicted warps, it successfully handles fine-grained motions covering a variety of human activities such as weight-lifting, doing gymnastics, walking sideways or leaning forward. Moreover, WALDO yields much more realistic outputs than STRPM [13][4] by producing high quality frames longer into

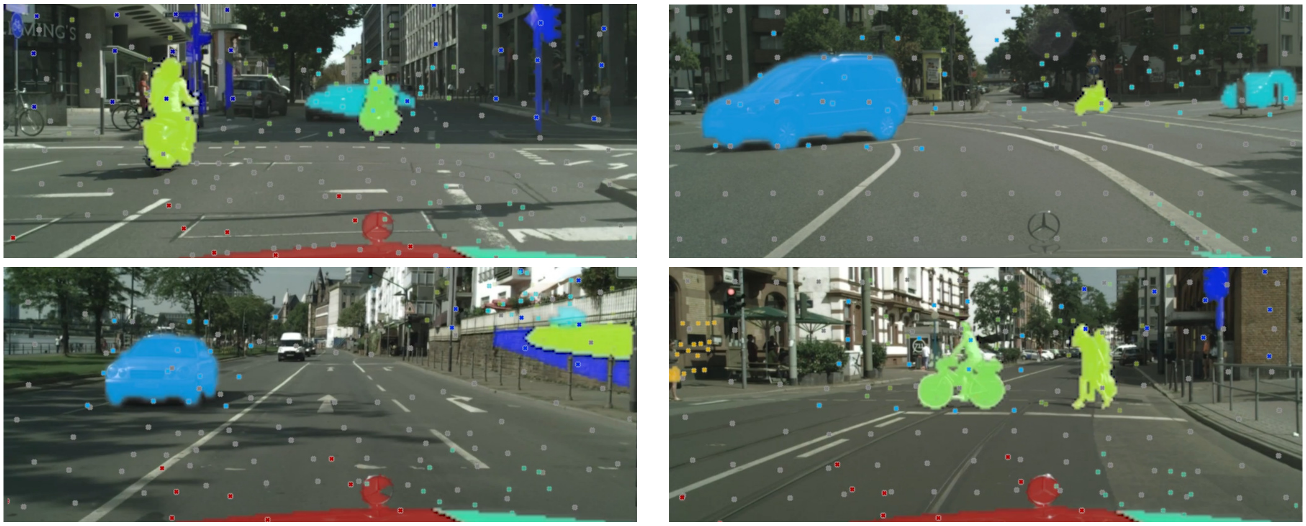


Figure 9. Visualization of control points and layer masks with different colors for each layer. See project webpage for videos.

Table 4. Ablation study on the Cityscapes test set. (a) **Layered video decomposition**: We evaluate decompositions in terms of flow reconstruction and object discovery as captured by $\mathcal{L}_f$ and $\mathcal{L}_o$. (b) **Future layer prediction**: We measure the accuracy of predicted control points. (c) **Video synthesis**: We evaluate the image reconstruction quality. (d) An example without (left) and with (right) semantic refinement. (e) L_1 reconstructions of an image in pixel space (left) and using control points (right) assuming Gaussian uncertainty over the horizontal position of the object. The former is blurry due to the plurality of potential positions. The latter singles out one position and preserves the object appearance.

(a) Layered video decomposition.

N	Input	Ref.	N_c	$\mathcal{L}_f \downarrow$	$\mathcal{L}_o \downarrow$
0	X		128	4.42	0.00
1	X		144	6.06	-3.69
8	X		256	4.47	-5.96
16	X		384	3.97	-5.79
16	S		384	3.90	-7.59
16	S+F		384	2.69	-7.56
16	S+F	✓	24	7.44	0.00
16	S+F	✓	72	7.01	-3.26
16	S+F	✓	176	4.17	-4.78
16	S+F	✓	288	3.16	-7.93
16	S+F	✓	384	**2.59**	**-8.16**

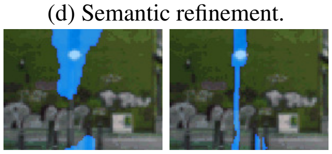
(b) Future layer prediction.

Arch.	Input	Δp_T	$\mathcal{L}_p \downarrow$
MLP	P		.514
T	P		.178
T	P	✓	.150
T	P+Z	✓	**.144**

(c) Video synthesis.

VGG	HR	Ctxt.	SSIM $\uparrow$
		1	812.1
✓		1	815.7
✓	✓	1	847.2
✓	✓	4	**848.0**

(d) Semantic refinement.



(e) Prediction strategy.

the future and with much less synthesis artifacts. This is confirmed by quantitative evaluations on both datasets (Table 3), with large relative improvements over the state of the art ranging from 1 to 3dB in PSNR and 31 to 56% in LPIPS.

4.2. Ablation studies

We conduct a detailed analysis on the Cityscapes test set to compare and highlight the key novelties of WALDO. The set of hyperparameters validated through this study have proven to work well on other datasets without any tuning.

Layered video decomposition is illustrated in Figure 9 and evaluated in Table 4(a) through the lens of our object discovery criterion ($\mathcal{L}_o$) and flow reconstruction ($\mathcal{L}_f$). We observe that the more object layers (N) the better, except for N=1 where fitting multiple objects with a single layer is suboptimal. Comparing inputs, we find using segmentation maps (S) better than RGB frames (X) alone for object discovery, and that using flow maps (F) helps motion reconstruction. Semantic refinement (Ref.) yields further gains, especially to segment thin objects like traffic lights, see Table 4(d). Finally, increasing the number of control points (N_c) allows us to capture finer motions, so we use as much as fits into memory (384 per time step). Our motion representation is scalable, with parameter size reduced from a quadratic (TK) to a linear (T+K) dependency on time compared to methods relying on optical flow directly.

Future layer prediction is illustrated in Figure 10 and evaluated in Table 4(b) in terms of trajectory reconstruction. Our baseline is a multi-layer perceptron (MLP), which maps past control points P=$\{p_t\}_{t=1}^T$ to future ones. Our actual architecture relies on a transformer (T) [83], which performs much better. Further gains are obtained by not predicting control points directly but rather their relative position (Δp_T) with respect to time step T, and by using layer features Z=$\{z^i\}_{i=0}^N$ as input.

Warping, fusion and inpainting is illustrated in Figure 11 and evaluated in Table 4(c) in terms of SSIM. Control points, obtained by the layer decomposition module, are

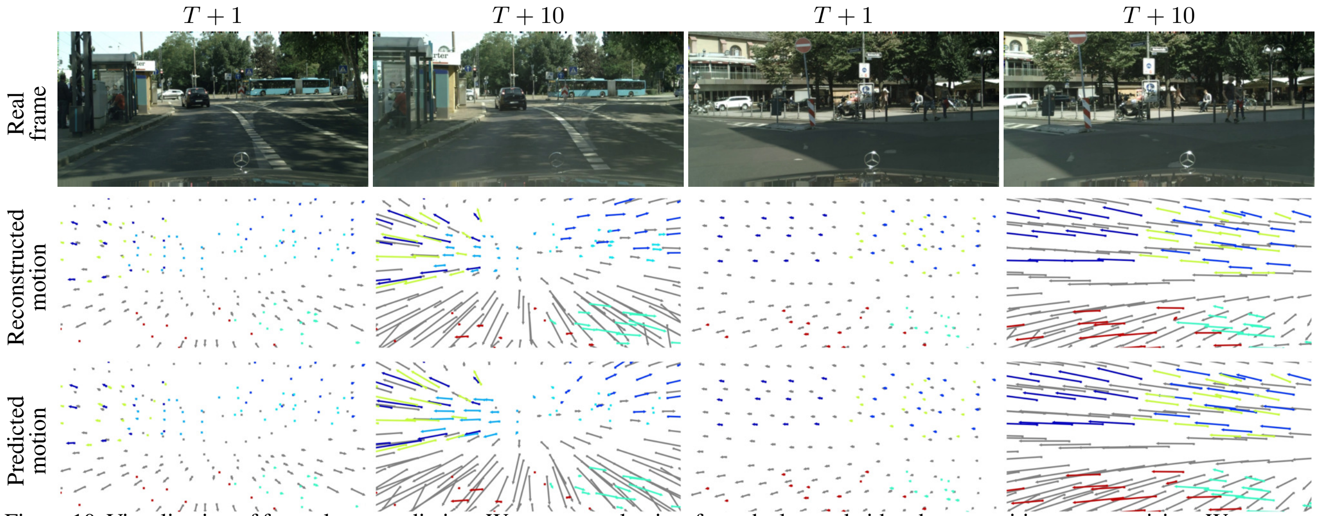


Figure 10. Visualization of future layer prediction. We use control points from the layered video decomposition as supervision. We compare motion vectors reconstructed from these points to the ones predicted for up to time step $T+10$ from a context of T=4 past frames. The motion vectors are computed between time step T and time step t in $\{T+1, T+10\}$. Different colors correspond to different layers.

Figure 11. Visualization of the synthesis process: warped visible regions and inpainted disoccluded ones between T and T+10.

used to warp, fuse and inpaint different views of the past frames to reconstruct future ones. We find using the feature distance (VGG) to be slightly better than the pixel one alone. The flexibility of WALDO, which trains at low resolution (128×256) but produces dense motions well suited for high resolution (HR) inputs (512×1024), results in higher SSIM than keeping the same resolution than for training during inference. We further improve SSIM by warping not only one but all of the four past context frames (Ctxt.).

Off-the-shelf models. We compare standard approaches in Appendix H, and find that WALDO is robust to the choice of the pretrained segmentation [14, 15, 70] and optical flow models [76, 77]. We also show in Appendix I that although we use an inpainting method [51] pretrained on external data [111] to produce realistic outputs in filled-in regions, it does not provide quantitative advantage to WALDO with at best marginal improvements in SSIM, LPIPS and FVD.

5. Conclusion

We have introduced WALDO, an approach to video synthesis which automatically decomposes frames into layers and relies on a compact representation of motion to predict their future deformations. Our method outperforms the state of the art for video prediction on various datasets. Future work includes exploring extensions of WALDO for applications from motion segmentation to video compression.

Limitations. Our performance depends on the accuracy of the layer decomposition. Failure cases include objects moving in different directions but merged into the same layer, or segmentation failures, where parts of an object are missed.

Acknowledgements

We thank Daniel Geng and Xinzhu Bei for clarifications on the evaluation process, and Pauline Sert for helpful feedback. This work was granted access to the HPC resources of IDRIS under the allocation 2021-AD011012227R1 made by GENCI. It was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), and the ANR project VideoPredict, reference ANR-21-FAI1-0002-01. JP was supported in part by the Louis Vuitton/ENS chair in artificial intelligence and a Global Distinguished Professorship at the Courant Institute of Mathematical Sciences and the Center for Data Science at New York University.

References

[1] Adil Kaan Akan, Erkut Erdem, Aykut Erdem, and Fatma Güney. SLAMP: Stochastic latent appearance and motion prediction. In *ICCV*, 2021. 1, 2, 5, 6, 7

[2] Adil Kaan Akan, Sadra Safadoust, Erkut Erdem, Aykut Erdem, and Fatma Güney. Stochastic video prediction with structure and motion. *arXiv preprint*, 2022. 1, 2, 6

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017. 16

[4] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *ICLR*, 2018. 2, 7

[5] Zhipeng Bao, Pavel Tokmakov, Allan Jabri, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert. Discovering objects that can move. In *CVPR*, 2022. 2, 3

[6] Amir Bar, Roei Herzig, Xiaolong Wang, Anna Rohrbach, Gal Chechik, Trevor Darrell, and Amir Globerson. Compositional video synthesis with action graphs. In *ICML*, 2021. 2

[7] Xinzhu Bei, Yanchao Yang, and Stefano Soatto. Learning semantic-aware dynamics for video prediction. In *CVPR*, 2021. 1, 2, 5, 6, 19

[8] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *TPAMI*, 2002. 3

[9] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *TPAMI*, 1989. 1, 3, 7, 14

[10] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint*, 2019. 2

[11] Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional VRNNs for video prediction. In *ICCV*, 2019. 6

[12] Michael B Chang, Tomer Ullman, Antonio Torralba, and Joshua B Tenenbaum. A compositional object-based approach to learning physical dynamics. In *ICLR*, 2016. 2

[13] Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. STRPM: A spatiotemporal residual predictive model for high-resolution video prediction. In *CVPR*, 2022. 1, 2, 5, 7

[14] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2, 3, 5, 9, 19, 22

[15] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint*, 2022. 9, 19

[16] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint*, 2019. 2

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 5, 16

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[19] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *ICML*, 2018. 2, 6

[20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021. 1

[21] Sébastien Ehrhardt, Oliver Groth, Aron Monszpart, Martin Engelcke, Ingmar Posner, Niloy J. Mitra, and Andrea Vedaldi. RELATE: Physically plausible multi-object scene synthesis using structured latent spaces. In *NeurIPS*, 2020. 2

[22] Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *ICLR*, 2020. 2

[23] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 1

[24] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *NeurIPS*, 2016. 2

[25] Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. In *ICML*, 2020. 2, 6

[26] Aditya Ganeshan, Alexis Vallet, Yasunori Kudo, Shin-ichi Maeda, Tommi Kerola, Rares Ambrus, Dennis Park, and Adrien Gaidon. Warp-refine propagation: Semi-supervised auto-labeling via cycle-consistency. In *ICCV*, 2021. 2

[27] Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, and Trevor Darrell. Disentangling propagation and generation for video prediction. In *ICCV*, 2019. 2

[28] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *IJRR*, 2013. 2, 5

[29] Daniel Geng, Max Hamilton, and Andrew Owens. Comparing correspondences: Video prediction with correspondence-wise losses. In *CVPR*, 2022. 1, 2, 6, 7, 19

[30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 5, 15

[31] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *ICML*, 2019. 2

[32] Ligong Han, Jian Ren, Hsin-Ying Lee, Francesco Barbieri, Kyle Olszewski, Shervin Minaee, Dimitris Metaxas, and Sergey Tulyakov. Show me what and tell me how: Video synthesis via multimodal conditioning. In *CVPR*, 2022. 2

[33] Zekun Hao, Xun Huang, and Serge Belongie. Controllable video generation with sparse trajectories. In *CVPR*, 2018. 2

[34] Paul Henderson and Christoph H. Lampert. Unsupervised object-centric video generation and decomposition in 3D. In *NeurIPS*, 2020. 2

[35] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint*, 2016. 16

[36] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint*, 2022. 1

[37] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *NeurIPS*, 2018. 2

[38] Yaosi Hu, Chong Luo, and Zhenzhong Chen. Make it move: controllable image-to-video generation with text descriptions. In *CVPR*, 2022. 2

[39] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*, 2022. 19

[40] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 19

[41] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 2013. 2, 5

[42] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NeurIPS*, 2015. 2, 15

[43] Nebojsa Jojic and Brendan J Frey. Learning flexible sprites in video layers. In *CVPR*, 2001. 2, 4

[44] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *NeurIPS*, 2021. 1

[45] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[46] Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional Object-Centric Learning from Video. In *ICLR*, 2022. 2

[47] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. VideoFlow: A conditional flow-based model for stochastic video generation. In *ICLR*, 2020. 2

[48] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective Cycle GAN. In *CVPR*, 2019. 7

[49] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint*, 2018. 7

[50] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019. 19

[51] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. MAT: Mask-aware transformer for large hole image inpainting. In *CVPR*, 2022. 5, 9, 20

[52] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *ECCV*, 2018. 2

[53] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, 2017. 6

[54] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020. 2, 4

[55] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *ICLR*, 2017. 6

[56] Erika Lu, Forrester Cole, Tali Dekel, Andrew Zisserman, William T Freeman, and Michael Rubinstein. Omnimatte: associating objects and their effects in video. In *CVPR*, 2021. 2

[57] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data. *arXiv preprint*, 2020. 2

[58] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. 7

[59] Jacob Menick and Nal Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. In *ICLR*, 2019. 1

[60] Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin P Murphy, and Honglak Lee. Unsupervised learning of object structure and dynamics from videos. In *NeurIPS*, 2019. 2

[61] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. CCVS: Context-aware controllable video synthesis. In *NeurIPS*, 2021. 1, 2

[62] Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry. Unsupervised layered image decomposition into object prototypes. In *ICCV*, 2021. 2, 4

[63] Charlie Nash, João Carreira, Jacob Walker, Iain Barr, Andrew Jaegle, Mateusz Malinowski, and Peter Battaglia. Transframer: Arbitrary frame prediction with generative models. *arXiv preprint*, 2022. 1

[64] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. In *ICML*, 2021. 1

[65] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3D view synthesis. In *CVPR*, 2017. 2

[66] M Pawan Kumar, Philip HS Torr, and Andrew Zisserman. Learning layered motion segmentations of video. In *ICCV*, 2008. 2

[67] Mikel D Rodriguez, Javed Ahmed, and Mubarak Shah. Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 2, 5

[68] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 5, 17

[69] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal GAN. *IJCV*, 2020. 2

[70] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 9, 19

[71] Karl Schmeckpeper, Georgios Georgakis, and Kostas Daniilidis. Object-centric video prediction without annotation. In *ICRA*, 2021. 2

[72] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *CVPR*, 2016. 2

[73] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS*, 2019. 19

[74] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 5

[75] Deqing Sun, Jonas Wulff, Erik B Sudderth, Hanspeter Pfister, and Michael J Black. A fully-connected layered model of foreground and background flow. In *CVPR*, 2013. 2

[76] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 9, 19

[77] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 2, 3, 5, 9, 19, 22

[78] D'Arcy Wentworth Thompson. On growth and form. *Nature*, 1917. 3

[79] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N. Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *ICLR*, 2021. 1, 2

[80] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*, 2018. 2

[81] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation. In *ICLRw*, 2019. 5

[82] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017. 1

[83] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4, 8

[84] Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. In *NeurIPS*, 2019. 7

[85] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *ICLR*, 2017. 6

[86] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016. 2

[87] Carl Vondrick and Antonio Torralba. Generating the future with adversarial transformers. In *CVPR*, 2017. 2

[88] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *ICCV*, 2017. 2

[89] John YA Wang and Edward H Adelson. Representing moving images with layers. *TIP*, 1994. 2

[90] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. 2, 6, 19

[91] Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and S Yu Philip. PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *ICML*, 2018. 7

[92] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3D LSTM: A model for video prediction and beyond. In *ICLR*, 2019. 7

[93] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. PredRNN: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *NeurIPS*, 2017. 7

[94] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 5

[95] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *ICLR*, 2020. 1

[96] Bohan Wu, Suraj Nair, Roberto Martin-Martin, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. In *CVPR*, 2021. 1

[97] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pretraining for neural visual world creation. *arXiv preprint*, 2021. 1, 2

[98] Haixu Wu, Zhiyu Yao, Jianmin Wang, and Mingsheng Long. MotionRNN: A flexible model for video prediction with spacetime-varying motions. In *CVPR*, 2021. 7

[99] Yue Wu, Rongrong Gao, Jaesik Park, and Qifeng Chen. Future video synthesis with object motion prediction. In *CVPR*, 2020. 1, 2, 6, 7, 19, 21

[100] Yue Wu, Qiang Wen, and Qifeng Chen. Optimizing video prediction via video frame interpolation. In *CVPR*, 2022. 1, 2, 6, 7, 19

[101] Jonas Wulff and Michael J Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *CVPR*, 2015. 2

[102] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. UPSNet: A unified panoptic segmentation network. In *CVPR*, 2019. 19

[103] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video generation using VQ-VAE and transformers. *arXiv preprint*, 2021. 1

[104] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *ICCV*, 2021. 2, 3

[105] Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. Deformable sprites for unsupervised video decomposition. In *CVPR*, 2022. 2

[106] Yufei Ye, Maneesh Singh, Abhinav Gupta, and Shubham Tulsiani. Compositional video prediction. In *ICCV*, 2019. 2

[107] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018. 19

[108] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. In *ICLR*, 2020. 7

[109] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5

[110] Yunzhi Zhang and Jiajun Wu. Video extrapolation in space and time. In *ECCV*, 2022. 6

[111] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 2017. 5, 9, 20

[112] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *ECCV*, 2016. 2

[113] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *CVPR*, 2019. 19