**International Journal of Computer Vision (IJCV) manuscript No.**
(will be inserted by the editor)

# Instance-aware Scene Layout Forecasting

**Xiaotian Qiao, Quanlong Zheng, Ying Cao, and Rynson W.H. Lau**

Received: date / Accepted: date

**Abstract** Forecasting scene layout is of vital importance in many vision applications, *e.g.*, enabling autonomous vehicles to plan actions early. It is a challenging problem as it involves understanding of the past scene layouts and the diverse object interactions in the scene, and then forecasting what the scene will look like at a future time. Prior works learn a direct mapping from past pixels to future pixel-wise labels and ignore the underlying object interactions in the scene, resulting in temporally incoherent and averaged predictions. In this paper, we propose a learning framework to forecast semantic scene layouts (represented by instance maps) from an instance-aware perspective. Specifically, our framework explicitly models the dynamics of individual instances and captures their interactions in a scene. Under this formulation, we are able to enforce instance-level constraints to forecast scene layouts by effectively reasoning about their spatial and semantic relations. Experimental results show that our model can predict sharper and more accurate future instance maps than the baselines and prior methods, yielding state-of-the-art performances on short-term, mid-term and long-term scene layout forecasting.

**Keywords** Scene layout · Instance interaction · Layout forecasting · Scene understanding

Xiaotian Qiao
Department of Computer Science, City University of Hong Kong.
E-mail: qiaoxt1992@gmail.com

Quanlong Zheng
Department of Computer Science, City University of Hong Kong.
E-mail: xiaolong921001@gmail.com

Ying Cao
Department of Computer Science, City University of Hong Kong.
E-mail: caoying59@gmail.com

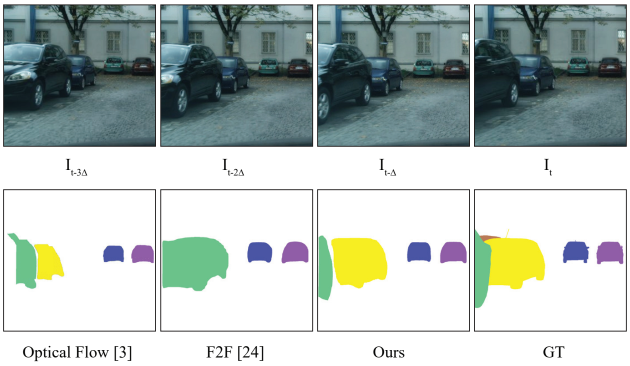
Rynson W.H. Lau
Department of Computer Science, City University of Hong Kong.
E-mail: Rynson.Lau@cityu.edu.hk

## 1 Introduction

Human beings are remarkably capable of forecasting the future states of a scene given past observations of it. We can do this mainly because we have built a mental model of scene dynamics by observing many examples of how objects move and interact in real-world scenes. A machine equipped with a similar capability to predict what a scene will look like in the near future will enable intelligent agents to plan their actions early based on past observations. For example, self-driving cars and social robots need to predict the future in order to plan ahead and react to the environment more quickly [6,34].

Developing machines' capability to anticipate the future is very challenging, as it requires understanding various appearance changes, complex motion dynamics, and diverse object interactions in a scene. Towards this objective, there has been a line of research on forecasting scene layouts (represented by semantic segmentation maps [25,32] or instance maps [24]) given the observed past frames in an input video. However, all these works adopt an end-to-end, per-pixel prediction framework of an existing segmentation model, which is extended to the temporal domain to directly map past pixels to future pixel-wise semantic labels. Such a design would lead to blurry and averaged predictions, where instances have degraded shapes and may disappear unexpectedly, especially for long-term prediction. It may also suffer from temporally incoherent predictions of instance shapes and locations, if partial occlusions are present in the input frames. Consider Figure 1 as an example. The shape of the car (marked in green) generated by the state-of-the-art F2F model [24] is inaccurate compared to that of the ground truth. In addition, the car marked in yellow is missing due to the lack of temporal consistency.

Our key observation is that to anticipate what a scene will look like at a future time, human beings would typi-


$I_{t-3\Delta}$ $I_{t-2\Delta}$ $I_{t-\Delta}$ $I_t$

Optical Flow [3] F2F [24] Ours GT

**Fig. 1** Scene layout forecasting. Given four past image frames ($I_{t-3\Delta}, I_{t-2\Delta}, I_{t-1\Delta}, I_t$) of a video as input (top row), our goal is to predict a future instance map $I_{t+n\Delta}$, where $n\Delta$ is the number of frames ahead in the future (bottom row). State-of-the-art methods, *e.g.*, F2F [24], tend to predict inaccurate object shapes and temporally incoherent objects, especially over a long time span. In contrast, our model can predict a more accurate and sharper future instance map over different time spans (see Section 4.4).

cally recognize and localize individual instances in the scene first, and then reason about their spatial and semantic interactions to make the prediction. Inspired by this observation, rather than performing direct pixel-level prediction as in existing works, we address the problem from an instance-aware perspective. We model the dynamics of individual instances separately and reason their interactions with each other explicitly. By doing so, we can enforce constraints at instance-level, which help produce sharper instance shapes and temporally stable predictions. We can also predict plausible instance motions due to the explicit modeling of subtle instance-to-instance interactions.

Based on the above idea, we propose a *separation and composition* framework to predict future instance maps (*i.e.*, scene layouts) from past image frames. As shown in Figure 2, unlike pixel-level prediction, our framework explicitly models the dynamics of individual instances and the relations among them in a scene. Given a sequence of image frames, we first generate a sequence of consistent instance maps by an off-the-shelf instance segmentation network [12]. The instance maps along with the corresponding image frames are *separated* in an instance-wise manner, generating a set of instance-wise representations. Each representation captures the structure and appearance dynamics of an instance over the time span of the input frames. The framework then learns per-instance features by encoding the instance-wise representation and modeling its spatial and semantic relations with other instances via an instance relation module. The learned per-instance features can then be used to predict instance layouts, which are finally *composed* to form the predicted future instance map. Figure 1 shows that our approach can generate much more accurate scene layouts, compared with the state-of-the-art methods.

To evaluate the performance of our method, we use the popular Cityscapes dataset [5] for short-term, mid-term and long-term future instance map predictions. Extensive quantitative and qualitative comparisons are performed against the baselines and state-of-the-art methods. Results show that our method is able to forecast sharper and more accurate scene layouts, yielding state-of-the-art performances.

In summary, the main contributions of this work are:

- To the best of our knowledge, we are the first to propose an instance-aware approach to address the scene layout forecasting problem. Our approach is able to produce sharp instance shapes and plausible instance motions.
- We propose a novel separation and composition framework for predicting future instance maps. It separately models the dynamics of individual instances and explicitly captures their interactions in a scene.
- We extensively validate our method and show that it achieves state-of-the-art performances in short-term, mid-term and long-term future predictions.

## 2 Related Work

Future forecasting, though challenging, is important to many real-world applications such as autonomous driving. Learning to forecast the future has received a significant amount of attention in the computer vision community. There has been much research effort on this problem by parameterizing the future in different forms, such as object movement trajectories [39], event categories [14], motion fields (optical flow) [37], human body poses [2], human behaviors [11], and action categories [35,8]. Unlike these works that predict future object locations, scene layout forecasting needs to jointly predict object positions, sizes and shapes [25,18,24]. Thus, all the aforementioned works are not amenable to our problem. In this section, we focus our discussion on recent works about image frame and scene layout forecasting.

**Image Frame Forecasting.** There is a large body of research on predicting raw pixel values of future frames in a video sequence. Ranzato et al. [30] proposed the first unsupervised deep model for next frame prediction. Mathieu et al. [26] improved the predictions using the adversarial loss and a gradient difference loss to avoid blurry results. A similar training strategy was employed for future frame predictions in time-lapse videos [40]. Rather than predicting unconstrained pixel intensities directly, Vondrick et al. [36] learned pixel transformation and synthesized future frames by transforming pixels from existing frames. Kwon et al. [21] used a single network to predict future and past frames retrospectively to enforce the consistency of bi-directional prediction. Kim et al. [19] proposed an adaptive online updating network for future frame prediction. All these approaches suffer from blurry and averaged predic-

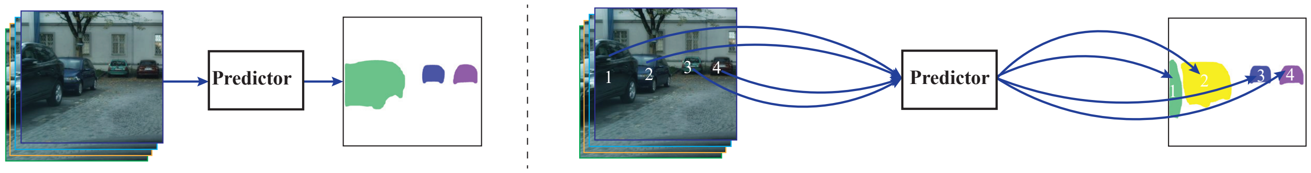


**Fig. 2** Illustration of the existing pixel-level prediction (left) and the proposed instance-aware modeling (right) for scene layout forecasting. Pixel-level prediction directly maps the input pixels to per-pixel semantic labels, while instance-aware modeling considers each instance separately and enforces spatial and temporal constraints at instance level. $R_k$ is the $k$-th instance representation, and $L_k$ is the future layout of the $k$-th instance.

tion, due to the difficulty of pixel-level prediction. In addition, pixel-level fine-grained predictions are often not necessary for intelligent systems to make decision. In contrast to these works that predict pixel intensities/colors, we focus on predicting a mid-level representation (*i.e.*, instance maps). Our work is similar in spirit to the works on future frame synthesis based on motion decomposition [28] and foreground-background separation [33]. However, instead of decomposing motion or separating the foreground and background, we decompose a scene into instances, model the interactions among these instances, and then predict a future layout of the instances.

**Scene Layout Forecasting.** Our work is in line with a series of recent works on scene layout forecasting. Instead of synthesizing (or predicting) pixel values for future frames, these works aimed to predict future image content in a more abstract way, *i.e.*, semantic scene layout. Jin et al. [17] trained a model to predict the semantic segmentation of the next frame from the preceding input frames. Luc et al. [25] directly trained a single network by taking several segmentation masks as input to predict the semantic segmentation of a future frame. Nabavi et al. [31] proposed a LSTM to model the temporal information of the input frames and predicted the semantic segmentation of a future frame. Hu et al. [15] proposed a deep learning model to jointly predict ego-motion, static scene, and the motion of dynamic agents in a probabilistic manner. While the above approaches are relevant to scene layout forecasting, they focus on producing object category labels instead of instance labels. Jin et al. [18] further divided a scene layout into three groups and proposed a multi-task learning framework to jointly predict optical flow and semantic segmentation. However, it only considered three groups in the scene and required optical flow annotations, which are difficult to obtain.

Notably, our work is closely related to [24], which also predicted future instance maps. They first predicted the features for a future frame, and then integrated the predicted features into the Mask R-CNN pipeline [12] for instance segmentation. However, their method still follows the traditional instance segmentation framework without explicitly accounting for instances. Graber et al. [10] further extended the future instance map prediction task to future panoptic segmentation by learning the dynamics of background stuff and foreground objects. However, they do not consider the interactions among instances. In contrast, our framework models individual instances and captures their spatial and semantic relationships explicitly. Our experiments in Section 4.4 show that while previous works tend to average different object classes into blurry future predictions, our method can preserve the properties of individual objects better due to its ability to exert instance-level control over the predictions, especially for mid-term and long-term predictions.

## 3 Approach

The scene layout forecasting task is to predict instance shapes, sizes as well as motions in a scene at a future time. The inputs to our method include four image frames, $I_{t-3\Delta:t} = \{I_{t-3\Delta}, I_{t-2\Delta}, I_{t-\Delta}, I_t\}$ (where $\Delta$ denotes the time interval), which capture the appearance dynamics of a scene. Our goal is to predict a future instance map $\widehat{S}_{t+n\Delta}$ (where n is the number of time intervals ahead in the future) that describes the semantic and structural information of the scene at time step $t + n\Delta$.

Figure 3 shows our framework. It has three main tasks. First, after segmenting the input frames $I_{t-3\Delta:t}$ into a sequence of instance maps $S_{t-3\Delta:t}$ using Mask R-CNN [12], we separate $I_{t-3\Delta:t}$ and $S_{t-3\Delta:t}$ in an instance-wise manner to generate disjoint sets of instance-wise frames, each of which captures the structure and appearance dynamics of an instance over the input time span $3\Delta$. We also segment the background from the input frames and the instance maps into a set of background frames. Second, we use background and instance encoders to extract background features and per-instance features, respectively, from the decomposed maps and frames. The features of each instance are then refined by the instance relation module to model its relation with other instances and the background. Third, we feed the refined instance features to LayoutNet to help predict the layouts of individual instances, which are then combined by a composition module to form a predicted scene layout. A sequence of future scene layouts can be generated by applying the framework recursively. We present the details of these three tasks below.

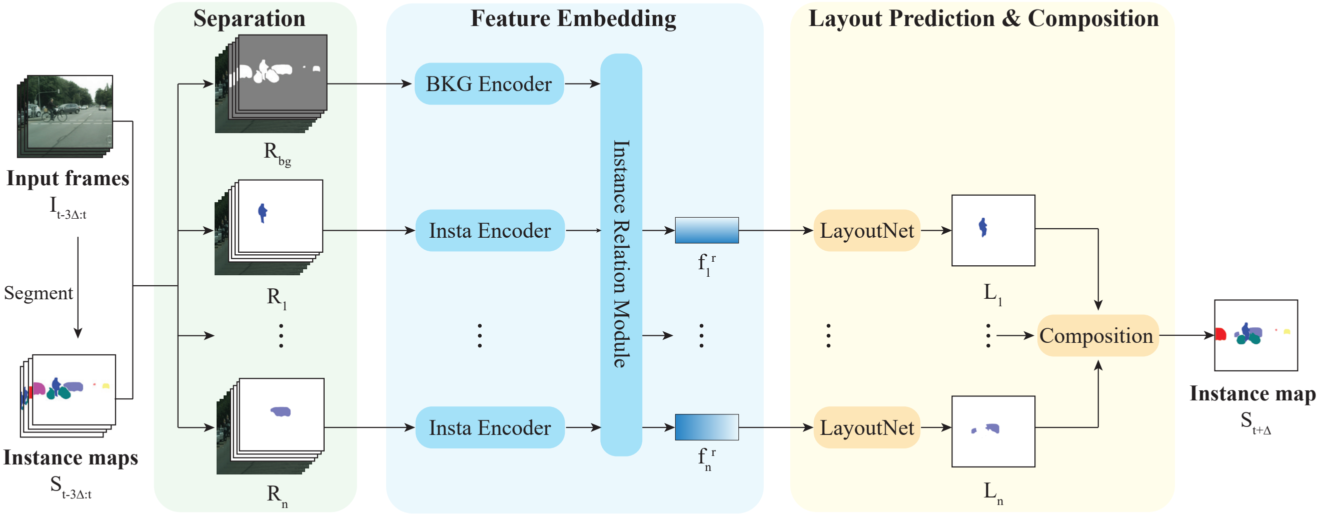


**Fig. 3** We propose an instance-aware separation and composition framework for scene layout forecasting, by modeling instances and their relations. Given a sequence of input frames $I_{t-3\Delta:t}$, we use Mask R-CNN as a pre-process to obtain a sequence of instance maps $S_{t-3\Delta:t}$. $I_{t-3\Delta:t}$ and $S_{t-3\Delta:t}$ are decomposed into instance-wise representation $R_k$ and background representation $R_{bg}$. These representations are then embedded into per-instance features and updated by the instance relation module, resulting in new features $f_k^r$ that capture interactions among the instances. $f_k^r$ are then used to predict instance layouts $L_k$, which are finally composed to output a future instance map.

### 3.1 Instance-wise Separation

In order to carry out future instance map prediction in an instance-aware manner, we first decompose the input frames and instance maps into a set of instance-wise representations. As shown in Figure 3, for each instance $i$, we obtain its masked instance maps $S^i_{t-3\Delta:t}$ and masked frames $I^i_{t-3\Delta:t}$ by tracking the instance across the frames and zeroing out the values outside the instance in $S_{t-3\Delta:t}$ and $I_{t-3\Delta:t}$. We then convert the masked instance maps $S^i_{t-3\Delta:t}$ to instance-wise semantic maps $\hat{S}^i_{t-3\Delta:t}$ by filling the instance silhouette of each instance map with one-hot encoding of the instance's category. We concatenate $\hat{S}^i_{t-3\Delta:t}$ and $I^i_{t-3\Delta:t}$ to construct an *instance-wise representation* $R_i$. This instance-wise representation $R_i$ captures the structure and appearance dynamics of the instance over the time span $3\Delta$. To model the effect of background contents on the instances of interest, we also consider the background (non-instance regions) as an instance by assigning it a distinctive label, and create a background representation $R_{bg}$ similarly.

Note that since the instance maps are generated by applying Mask R-CNN to each input frame independently, there is no inherent correspondence between instances across different frames. Thus, before the separation step, we track each instance across different frames using a simple tracking method as described in the dataset processing part of Section 4.2.

### 3.2 Feature Embedding

Given the background and instance-wise representations, we use a background encoder (BKG encoder) and an instance encoder (Insta encoder) to extract background features and instance features that encode structure and appearance dynamics of the background and the instances, respectively. However, since each instance would likely plan its future states according to the behaviors of other instances in a scene, encoding each instance alone may not capture the subtle interactions among the instances, which is crucial to predicting its future state. To address this problem, we introduce an instance relation module to refine the features of each instance by modeling its spatial and semantic relation with other instances in the scene. Taking as input the background and instance features $\{f_{bg}, f_1, f_2, \ldots, f_n\}$, the relation module outputs the refined instance features $\{f_1^r, f_2^r, \ldots, f_n^r\}$ as the final embedded instance features.

**Instance Relation Module.** To predict the future state of an instance, the states of some neighboring instances might be more relevant than the others. Hence, we adopt an attention mechanism [22] to incorporate relational information into the features of each instance. Note that our framework is general, and any differentiable relation module can be used in principle. Specifically, the refined features of instance $i$ are a linear combination of the features of the other instances and the background as:

$$f_i^r = f_i + \sum_{j \neq i}^{n} \varepsilon_{ij} \cdot (W_{v1} f_j) + \varepsilon_i^{bg} \cdot (W_{v2} f_{bg}), \quad (1)$$

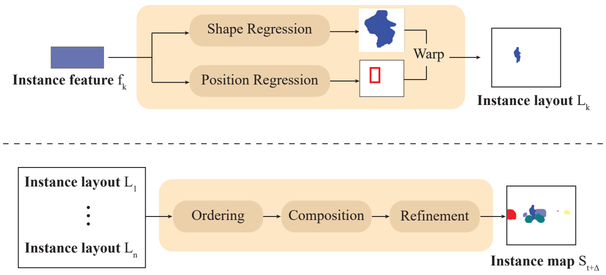


**Fig. 4** Details of the LayoutNet module (top) and the composition module (bottom).

where $W_{v1}$ and $W_{v2}$ are learnable weight matrices. $\varepsilon_{ij}$ indicates the relative importance of instance $j$ to $i$. $\varepsilon_i^{bg}$ indicates the relationship of instance $i$ to the background. Both $\varepsilon_{ij}$ and $\varepsilon_i^{bg}$ are attention weights that can be computed dynamically via an attention mechanism:

$$\varepsilon_{ij} = max(0, W_Q f_i + W_P f_j), \tag{2}$$

$$\varepsilon_i^{bg} = max(0, W_K(f_i + f_{bg})), \tag{3}$$

where $W_Q$, $W_P$ and $W_K$ are learnable weight vectors that project instance features $f_i$ and $f_j$ to scalar values. $max(0, x)$ is used to ensure the attention weights $\geqslant 0$, with 0 indicating no interaction with instance $i$. In this way, the refined features of each instance will encode not only the information of the instance itself but also the contextual information from other instances and the background.

### 3.3 Layout Prediction and Composition

Having generated the embedded features for individual instances in the scene, we use them to predict an instance map that describes the semantic layout of the scene at a future time, by first predicting a layout for each instance and then composing all the instance layouts to output a future instance map.

**Layout Prediction.** For each input instance, we use its features to compute an instance layout map by predicting a shape mask and a bounding box using the LayoutNet module [29], which is illustrated in Figure 4(top). We note that the moving patterns of different instances in a scene are different, depending on many factors such as the size of each instance and its distance from the camera. In particular, we have empirically found that directly predicting the future layouts of small instances cannot give satisfying results. It is possibly because the motion magnitudes of small objects are often very small. This may result in the loss function being dominated by those instances with large motions, causing the model to generate unreasonable predictions for the small instances.

To address the above problem, we train the LayoutNet module to additionally estimate how confident the network is in predicting the layout map of each instance. A low confidence score indicates that the network prediction result is unreliable. In particular, the input to the LayoutNet module is the instance-wise features $f_k$, which are fed into a shape regression branch to predict a soft binary shape mask $m_i$ and a position regression branch to predict a bounding box of the instance with five parameters $(x_i, y_i, w_i, h_i, p_i)$. $(x_i, y_i)$ refer to the center location of the box, $(w_i, h_i)$ refer to the width and height of the box, and $p_i$ is the confidence score of the box. Note that both rigid and non-rigid objects can be represented by the predicted shape masks. The generated shape is then warped to the position of the corresponding bounding box using bilinear interpolation [16], generating a layout map $L_k$ that represents the shape, size, position and confidence score of the instance in the scene. For the instances with low confidence scores, rather than predicting with the object layout network, we directly estimate their layouts using their motion trajectories in the instance maps. In particular, for each low-confidence instance, we compute an average scaling factor and movement vector based on the corresponding instance masks across all the input instance maps. We then apply the scaling factor and movement vector to the mask of the instance in the last instance map to obtain its future layout map.

**Layout Composition.** Given all the instance layout maps $\{L_1, L_2, \ldots, L_n\}$, the composition module aims to compose them to generate a coherent scene layout in the form of an instance map. However, partial occlusions among the instances may occur during the composition process. If two instances are found to overlap each other, we first determine their front-back order. Inspired by the element composition process in [27], we first train an ordering network to indicate if an instance should be in front of another. The inputs to the ordering network are two instance layout maps, and the output is a binary label that indicates if an instance is on top of another when they are composed together. To train the network, the ground truth order is obtained based on the depth maps from Cityscapes. Here, the network focuses on learning the prior that explains the instance-instance depth relations in natural scenes. Based on the outputs from the ordering network, we then composite these instances onto a single canvas, resulting in a scene layout. Note that some artifacts like holes and unreasonable shapes may still exist after the composition process. We further refine the composed scene layout via a cascaded refinement network [4], to remove these artifacts.

### 3.4 Training

We first train the ordering network to obtain the relative order of two instance layout maps. We then train our entire network end-to-end by minimizing a multi-task objective. In particular, for instance shape prediction, we use a binary cross entropy loss to penalize the pixel-wise difference between each predicted shape mask $m$ and the corresponding ground truth $\hat{m}$ as:

$$L_{shape} = -\sum_x \sum_y \hat{m}_{x,y} \log m_{x,y} + (1-\hat{m}_{x,y}) \log(1-m_{x,y}), \tag{4}$$

where $m_{x,y}$ is the presence probability of an instance at spatial location $(x, y)$.

For instance bounding box prediction, we denote the parameters and the confidence score of the predicted bounding box as $b$ and $p_{inst}$, respectively. We define the L1 loss between $b$ and the corresponding ground truth $\hat{b}$ as:

$$L_{bbox} = p_{inst} \|b - \hat{b}\|_1. \tag{5}$$

Since we do not have access to the ground truth confidence score, we apply a cross entropy loss with a constant target label of 1 on $p_{inst}$ as:

$$L_{score} = -\log(p_{inst}). \tag{6}$$

This will encourage the network to minimize the loss in Eq. 5, while allowing for a certain amount of relaxation to discount the instances with low confidence scores.

In addition, we also make use of adversarial learning [9] to encourage the generated instance map to appear realistic. Specifically, we train our model adversarially against a discriminator network $D$, which attempts to classify an input instance map as real or fake by minimizing the objective:

$$L_{adv} = \mathbf{E}_{x \sim p_{real}} \log D(x) + \mathbf{E}_{x \sim p_{fake}} \log(1 - D(x)), \tag{7}$$

where $x \sim p_{fake}$ are predicted instance maps. $x \sim p_{real}$ are the ground truth instance maps.

In summary, we train our model with a total loss:

$$L = \alpha \sum_i L^i_{shape} + \beta \sum_i L^i_{bbox} + \gamma L_{adv} + \zeta \sum_i L^i_{score}, \tag{8}$$

where $\alpha$, $\beta$, $\gamma$ and $\zeta$ are the loss weights.

## 4 Experiments

### 4.1 Implementation Details

**Training.** The proposed framework is implemented under the Pytorch framework. The network parameters are initialized using the truncated normal initializer. We downsample the original frames to a resolution of $128 \times 256$ in both training and testing stages. We optimize the parameters of our model by the Adam optimizer [20] with an initial learning rate of 0.005, $\beta_1 = 0.9$ and $\beta_2 = 0.9999$. The batch size is set to be 8. We empirically set the loss weights $\alpha$, $\beta$, $\gamma$ and $\zeta$ to 1, 1, 1, 0.1. Following [25], we train our model by sampling four frames with an interval of 3 frames (*i.e.*, $0.17s$).

**Inference.** We apply our model for three types of time spans: (1) *Short-term*: predicting the 3rd frame after $t$, *i.e.*, $S_{t+3}$ (up to $0.17s$); (2) *Mid-term*: predicting the 9th frame after $t$, *i.e.*, $S_{t+9}$ (up to $0.5s$); (3) *Long-term*: predicting the 27th frame after $t$, *i.e.*, $S_{t+27}$ (up to $1.6s$).

We only train our model for short-term prediction and adopt a recursive approach as in [25] for mid-term and long-term prediction, by predicting a future instance map given a temporal window of 4 past instance maps. Initially, the window only contains the input instance maps and the corresponding image frames. In later iterations, it contains the instance maps predicted from the previous iterations. Note that we also need to obtain the corresponding masked frames before passing them to the network. When using a predicted instance map as input to our model, for each instance on it, we estimate its masked image frame by masking the last input image frame based on the spatial transformation between its current bounding box and that in the last input frame. We empirically set the threshold of the confidence score to 0.3. When the confidence score of an instance is lower than 0.3, we use the linear motion strategy described in Section 3.3 to obtain its future layout. Otherwise, we use the network to predict the future instance layout.

### 4.2 Experimental Setup

**Dataset.** Following existing works [25,24,18], we use the Cityscapes dataset [5] for training, which consists of urban scene videos recorded from a car while driven on the street. It contains 2,975 training, 500 validation and 1,525 test video sequences. Each video has 30 frames of resolution $1024 \times 2048$. However, the ground truth instance map annotation is only available for the 20th frame of each video sequence in the dataset. To get the instance maps for all the frames, we use the Mask R-CNN model pretrained on COCO [23] and fine-tuned on Cityscapes using a ResNet-50-FPN backbone [13]. Given the image frames in the dataset, the predicted instance segmentations from Mask R-CNN are regarded as the ground-truth annotations to supervise our network.

The instance label IDs may not be consistent across frames due to the independent application of Mask R-CNN on each frame. To perform instance-level decomposition (Section 3.1) robustly, we need to filter out unreliable instances and build correspondences of the instances across frames. To this end, we propose a simple tracking method by combining multiple cues [38], including semantic consistency, spatial correlation and detection confidence. Given a sequence of 5 frames (4 inputs and 1 output), assuming that we already have a set of existing instance IDs denoted as

$\Omega$, a new instance can be assigned to an instance ID or discarded based on its confidence score. Specifically, we compute the confidence score $v_t^i$ for instance $i$ at frame $t$:

$$v_t^i = max_j[\log s_t^i + \mathbf{IoU}(b_t^i, b^j) + \delta(c_t^i, c^j)], \tag{9}$$

where $s_t^i$ is the detection score of instance $i$ from Mask R-CNN. $b_t^i$ and $c_t^i$ are its bounding box and category, respectively. $\mathbf{IoU}(\cdot)$ is the IoU between two instance bounding boxes, and $\delta(\cdot)$ is the Kronecker delta function. It is equal to 1 when $c_i$ and $c_n$ are equivalent, and 0 otherwise. If $v_t^i$ is larger than a threshold $V_{th}$, we assign it an existing ID $j$ that maximizes the value of $v_t^i$ in Eq. 9, and a new ID otherwise. We record the number of times that the confidence score of each instance ID is higher than threshold $V_{th}$. This number represents how many times the instance ID appears across different frames. We iteratively update this set of instance IDs until all the instances have been processed. Finally, those instances that appear in less than 3 frames will be discarded.

**Evaluation Metrics.** Since our objective is to predict accurate positions and shapes of the instances in the future, we employ the following metrics. We first use the Jaccard index, *i.e.*, intersection over union (IoU) between the predicted mask and ground truth mask. We also adopt the contour-based F-measure [7] to evaluate the quality of the predicted object shapes. In particular, given the pixel-wise boundaries of a predicted mask and the ground truth mask, we use a distance threshold $\rho$ to define a boundary precision $P_i$ and recall $R_i$. A predicted boundary pixel is regarded as positive only if it is within a distance of $\rho$ from any ground truth boundary pixel. The boundary-based F-measure $F_i$ is computed as:

$$F_i = \frac{(1+\beta^2) \times P_i \times R_i}{\beta^2 \times P_i + R_i}, \tag{10}$$

where $\beta^2 = 0.3$ as in [1] to emphasize the precision. It is worth noting that AP (an evaluation metric used in instance segmentation) is not applicable in our setting as it is computed by thresholding on IoU and class confidence, while our model directly predicts the future position and shape of each instance without predicting its class probabilities.

### 4.3 Compared Methods

Note that our objective of this work is to predict the future state of each instance in the scene (*i.e.*, a future instance map), rather than semantic segmentation as in [32, 33], which do not differentiate between different instances of each class. Thus, we compare our model with the following methods:

- **Identity.** The instance map for the future frame $S_{t+\Delta}$ is copied from the last input instance map $S_t$ directly.
- **Linear Motion.** For each object in the scene, we directly estimate its average scaling factor and movement vector from the given instance maps, and apply them to its mask in the last instance map to get its future instance map.
- **Optical Flow.** We first compute the optical flow field [3] from the last two input image frames. We then warp each instance mask in the last input instance map based on the inverted flow field on it. For long-term prediction, we recurrently apply the computed optical flow to the latest predicted instance map to make the next prediction.
- **F2F [24].** This method first predicts a future feature representation, and then send it to Mask R-CNN to predict an instance map of the future.
- **Panoptic Segmentation (PS) [10].** This is the state-of-the art method for predicting the future panoptic segmentation that contains both background stuff and foreground objects. For fair comparison, we compare our results with the foreground object prediction results in [10].
- **Oracle.** We also report the result of Mask R-CNN as a performance upper bound of our method, which is obtained by running Mask R-CNN on the future frames whose instance maps are to be predicted (recall that our model is supervised by the instance maps predicted by Mask R-CNN).

### 4.4 Results

**Qualitative Results.** We first show visual comparison with the existing methods in Figure 5 and Figure 6. For short-term prediction shown in Figure 5, we find that Optical Flow tends to produce artifacts around the instance boundaries, F2F may introduce new instances that do not exist in the input scenes, and PS may generate inaccurate instance shapes, sizes and positions. Consider Figure 5(a) as an example. Optical Flow produces some artifacts on the cars due to the inaccurate flow field estimation. Although F2F can predict plausible shape and position of the car in the red box, it introduces a truck unexpectedly inside the green box, which breaks the temporal coherency. The main reason is that its result is based on a predicted high-level, abstract future representation, which is unable to guarantee instance-level temporal consistency between the input frames and the output. In addition, without considering the relationships among the instances, PS generates inaccurate shape, size and position for the car in the red box. In contrast, our method can preserve the instances in the input frames better and generate temporally more coherent predictions. This is because our method considers each instance separately and models the interactions among these instances explicitly.

Mid-term and long-term predictions are much more challenging than short-term prediction, as the uncertainty

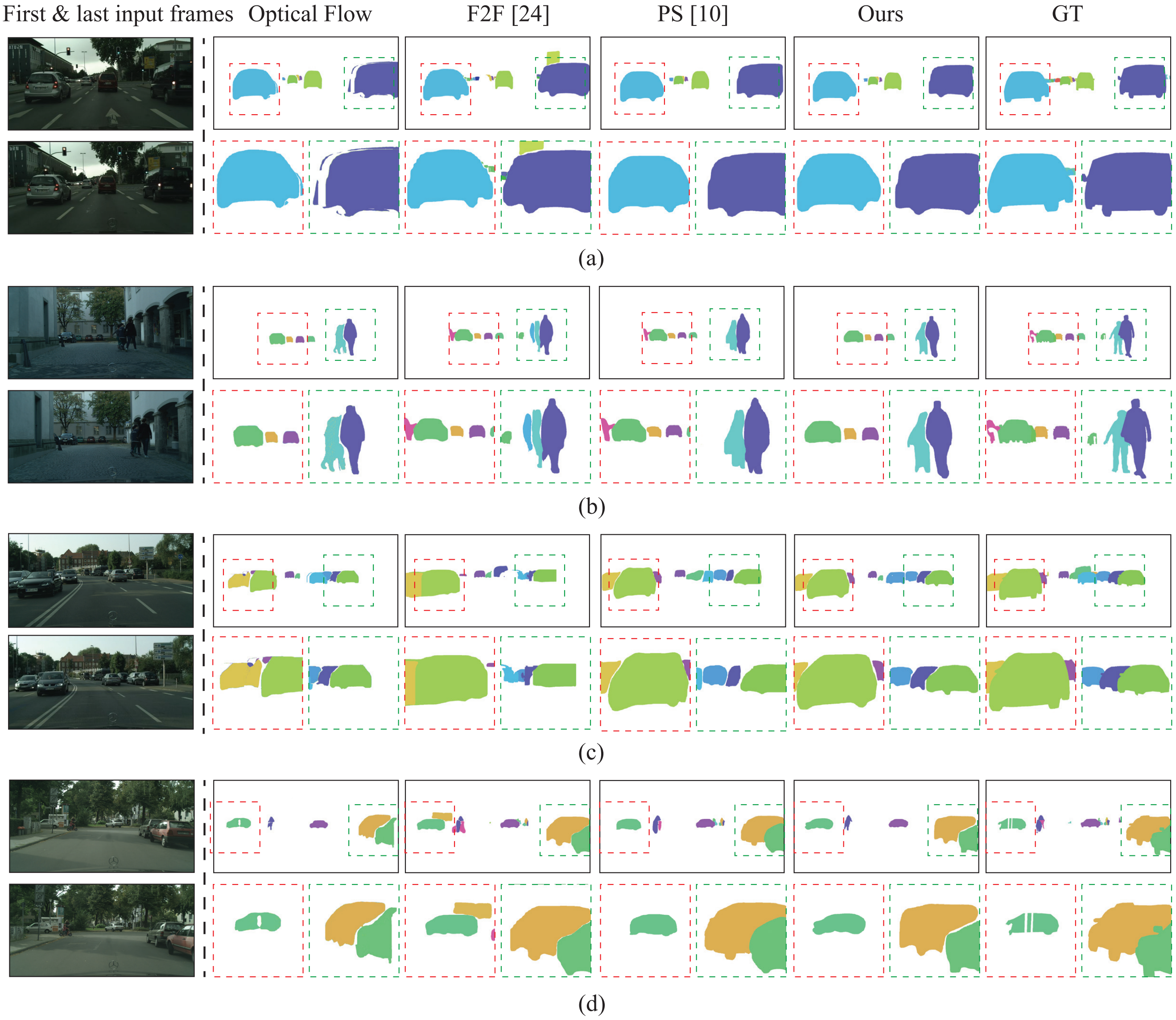


**Fig. 5** Qualitative results for short-term prediction. In each example, we show the first and last image frames of a four-frame sequence. We then show the instance map prediction results of an optical flow-based method (Optical Flow), F2F [24], PS [10]), our method (Ours), and the ground truth (GT).

increases when the time span becomes larger. For the mid-term prediction shown in Figure 6(a-c), in comparison to other methods, our method can predict sharper object shapes and more accurate object positions. For example, for the car inside the green box in Figure 6(a), Optical Flow predicts a wrong position, hiding it behind another car, while F2F produces a blurry shape. Although the result of PS is better, the shapes of the two cars are still worse than those of our method. For long-term prediction shown in Figure 6(d), we can see that Optical Flow fails miserably by having many instances diminished, due to the challenging task of estimating the motion field over a long period of time. F2F and PS tend to generate "average" predictions, where the shapes of the instances degrade considerably (*e.g.*, the cars inside the green box). In contrast, our method can still give clear instance shapes and plausible instance positions.

To further investigate the effect of different time spans (*i.e.*, from $t + 3$ to $t + 9$) on the prediction performance, we show the layout predictions of individual instances in Figure 7. We can observe that the car shapes predicted by our method closely match with the oracle [12], while Optical Flow, F2F and PS tend to produce degraded car shapes over time. These results confirm the distinctive advantage of our instance-aware modeling over the pixel-level prediction in generating high-quality object shapes.

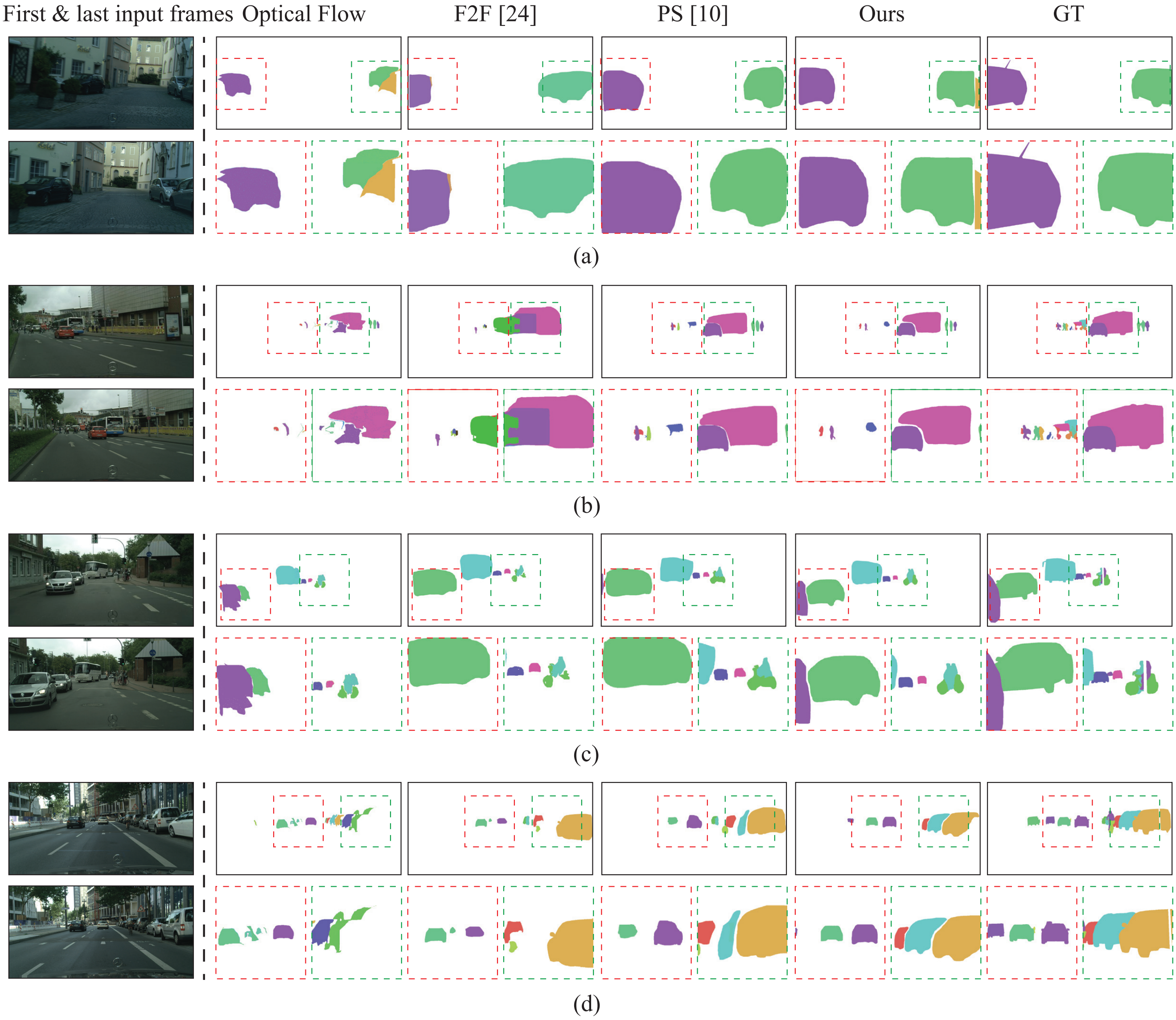


**Fig. 6** Qualitative results for mid-term (a-c) and long-term (d) predictions. In each example, we show the first and last image frames of a four-frame sequence. We then show the instance map prediction results of an optical flow-based method (Optical Flow), F2F [24], PS [10], our method (Ours), and the ground truth (GT).

**Quantitative Results.** Table 1 shows the quantitative results over different time spans. Our method performs significantly better than the baselines, F2F and PS. For short-term prediction, F2F works well and is close to our method on the Jaccard index. Predicting the immediate future is a relatively easier task since the pixels of immediate future are strongly correlated with the past pixels. Thus, pixel-level prediction models can give good results, as demonstrated by the reasonably good performances of Identity and Optical Flow, along with the modest performance gaps between the Oracle and different methods. However, it is worth noting that our results have consistently better boundary accuracy (*i.e.*, higher F-measure values) across different time spans, which confirms the distinctive advantage of our instance-wise framework in instance shape prediction. For mid-term and long-term predictions, despite the significant performance drops of all the methods, PS performs better than F2F and other baselines by modeling the motion and appearance history of individual instances. However, our method still outperforms PS by a large margin. This demonstrates the superiority of our method, particularly for a relatively long time span.

We further investigate the performances of our method on some individual object categories. In Table 2, we show quantitative results on car (a rigid object) and person (a highly deformable object). Our method consistently outper-

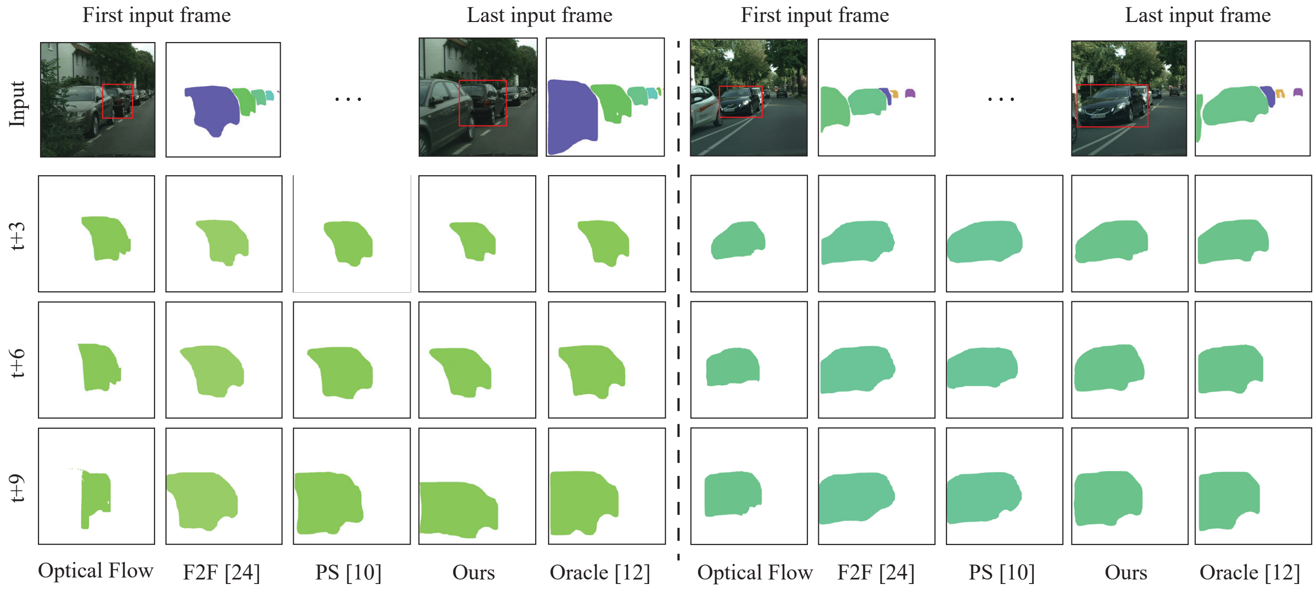


**Fig. 7** Layout predictions of individual instances by different methods over different time spans. For each case, we show the first and last RGB frames of a four-frame sequence along with their respective instance maps computed by [12]. We show the instance layouts predicted by an optical flow based algorithm (Optical Flow), F2F [24], PS [10], our method (Ours), and oracle [12] at time steps $t + 3$ (2nd row), $t + 6$ (3rd row) and $t + 9$ (4th row).

**Table 1** Quantitative results of predicting instance maps for short-term, mid-term and long-term future. The best results are highlighted in bold.

| | Short-term | | Mid-term | | Long-term | |
|---|---|---|---|---|---|---|
| Method | Jaccard index↑ | F-measure↑ | Jaccard index↑ | F-measure↑ | Jaccard index↑ | F-measure↑ |
| Oracle [12] | 64.7 | 60.3 | 64.7 | 60.3 | 64.7 | 60.3 |
| Identity | 45.7 | 28.1 | 29.1 | 7.6 | 6.9 | 2.9 |
| Linear motion | 53.7 | 34.2 | 32.6 | 9.2 | 7.5 | 3.1 |
| Optical Flow | 58.8 | 39.8 | 41.4 | 13.2 | 10.6 | 7.7 |
| F2F [24] | 61.2 | 41.9 | 41.2 | 20.3 | 15.8 | 10.3 |
| PS [10] | 60.3 | 40.8 | 42.1 | 22.9 | 17.7 | 13.2 |
| Ours | **61.3** | **42.3** | **42.8** | **24.3** | **19.2** | **15.1** |

**Table 2** Quantitative results on car and person. The best results are highlighted in bold.

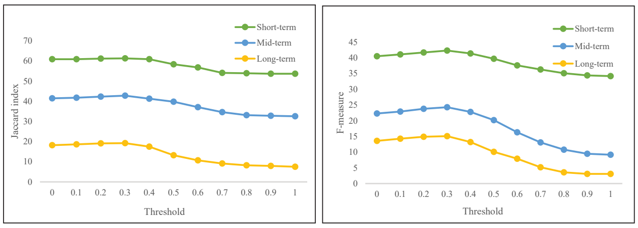
| | Short-term | | | | Mid-term | | | | Long-term | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Jaccard index↑ | | F-measure↑ | | Jaccard index↑ | | F-measure↑ | | Jaccard index↑ | | F-measure↑ | |
| Method | Car | Person | Car | Person | Car | Person | Car | Person | Car | Person | Car | Person |
| Identity | 53.3 | 38.6 | 33.7 | 20.1 | 35.5 | 21.9 | 10.5 | 4.9 | 8.1 | 5.3 | 3.4 | 2.1 |
| Linear Motion | 61.2 | 47.3 | 40.7 | 28.3 | 39.8 | 25.1 | 12.9 | 5.9 | 9.7 | 6.8 | 3.9 | 2.0 |
| Optical Flow | 65.9 | 52.9 | 46.1 | 33.2 | 47.9 | 36.2 | 17.1 | 9.6 | 13.4 | 7.9 | 9.5 | 5.8 |
| F2F [24] | 68.1 | **54.8** | 48.3 | 35.7 | 48.3 | 36.1 | 25.4 | 15.8 | 18.3 | 13.1 | 12.7 | 7.8 |
| PS [10] | 68.3 | 54.5 | 49.1 | 35.6 | 48.9 | 36.8 | 27.6 | 17.1 | 20.1 | 14.8 | 14.3 | 8.6 |
| Ours | **68.7** | 54.3 | **49.8** | **35.9** | **49.3** | **37.1** | **29.2** | **18.2** | **23.4** | **15.7** | **17.9** | **10.9** |

forms the other methods on both categories. We note that our performance gains are more pronounced on rigid objects (*i.e.*, car) than on non-rigid objects (*i.e.*, person), as non-rigid objects can deform in numerous ways, making it difficult to predict their motions and shapes accurately.

When predicting the future layout of an instance, we chose to threshold on its confidence score to determine if we use the network or a naive linear motion predictor for prediction. Here, we experiment with different confidence score thresholds to study the effectiveness of this design choice. When the threshold is set to 0 or 1, we only use our network or the linear motion predictor, respectively, for prediction. When the threshold is in (0, 1), we use the two prediction models simultaneously and determine which one to use for an instance based on its predicted confidence score. Figure 8 shows the results. We can see that as the threshold

**Table 3** Results of the ablation study. The best results are highlighted in bold.

| | Short-term | | Mid-term | | Long-term | |
|---|---|---|---|---|---|---|
| Method | Jaccard index↑ | F-measure↑ | Jaccard index↑ | F-measure↑ | Jaccard index↑ | F-measure↑ |
| w/o image frames | 52.5 | 41.6 | 38.9 | 21.3 | 18.8 | 12.8 |
| w/o background representation | 53.4 | 41.8 | 39.6 | 21.8 | 18.9 | 13.5 |
| w/o instance relation module | 50.9 | 40.6 | 35.2 | 19.8 | 12.7 | 10.1 |
| w/o ordering network | 54.8 | 40.3 | 40.8 | 20.6 | 18.8 | 11.3 |
| w/o adversarial loss | 57.7 | 41.8 | 41.9 | 23.5 | 18.9 | 14.3 |
| Our model | **61.3** | **42.3** | **42.8** | **24.3** | **19.2** | **15.1** |



(a) (b)

**Fig. 8** Performance vs. Threshold. We change the confidence score threshold from 0 to 1, and use Jaccard index and F-measure to evaluate the performance over different time spans. Higher scores indicate better performances.

increases from 0 (*i.e.*, as we add the linear motion predictor to handle instances with relatively low confidence scores), the performance improves gradually for all the time spans, until when the threshold reaches around 0.3 where the performance drops significantly. This confirms the necessity of our confidence-based dynamic model choice strategy.

### 4.5 Ablation Study

To investigate the effectiveness of our design choices, we compare our model with its ablated versions:

- *w/o image frames*: We use a sequence of instance maps as input to our model.
- *w/o background representation*: We remove the background encoder from our model.
- *w/o instance relation module*: We remove the instance relation module from our model.
- *w/o ordering network*: We remove the ordering network from our model.
- *w/o adversarial loss*: We train the model without using the adversarial loss.

The results of the ablation study are shown in Table 3. Without utilizing image frames or background representation, the performance drops. This indicates that the visual appearances of objects and the background are useful signals for forecasting scene layouts. When the instance relation module is removed, the performance becomes worse, which implies that modeling instance-instance interactions is crucial to predicting the future motions of instances in a scene. In particular, we note that the performance for long-term prediction drops more significantly without this module. This indicates that modeling instance-instance interactions is of particular importance to long-term prediction. Without the ordering network, the F-measure is affected more than the Jaccard index. This is possibly because the incorrect instance order would significantly change some instance boundaries. Finally, without using the adversarial loss, the performance also drops slightly. This indicates that the adversarial training process could help generate instance maps with more realistic details.

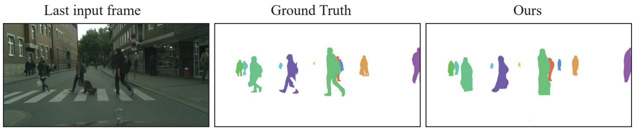


**Fig. 9** Failure case in short-term prediction. When a scene contains many people with diverse poses, our model may not be able to predict plausible shapes due to the inherent ambiguity of predicting the shapes of highly non-rigid objects.

## 5 Conclusion

In this work, we have studied the problem of forecasting scene layouts with an instance-aware approach. We have presented a learning framework to forecast the instance map of a scene at a future time from past image frames, by explicitly modeling the motion dynamics of the instances in the scene as well as instance-instance interactions. Through extensive experiments, we show that learning with instance-wise formulation is able to produce more accurate and sharp predictions, as compared with prior methods, and yields state-of-the-art performances.

Despite promising results, our method may fail to predict plausible future shapes for some highly non-rigid objects in a scene (*e.g.*, people as shown in Figure 9). This is because these objects can deform in many different ways, making it very challenging to predict their shapes. A possible solution to this problem is to group these highly deformable objects according to their semantic labels (*e.g.*, people), and design a branch to especially model their shape dynamics, which can be an interesting future work.

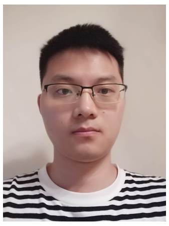
**Acknowledgements** This work was supported by the Research Grants Council of Hong Kong (Grant No.: 11205620).

## References

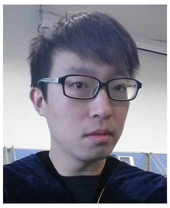
1. Achanta, R., Hemami, S., Estrada, F., Süsstrunk, S.: Frequency-tuned salient region detection. In: CVPR (2009)
2. Chao, Y.W., Yang, J., Price, B., Cohen, S., Deng, J.: Forecasting human dynamics from static images. In: ICCV (2017)
3. Chen, Q., Koltun, V.: Full flow: Optical flow estimation by global optimization over regular grids. In: CVPR (2016)
4. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: ICCV (2017)
5. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
6. Dosovitskiy, A., Koltun, V.: Learning to act by predicting the future. In: ICLR (2017)
7. Ehrig, M., ér ˆome Euzenat, J.: Relaxed precision and recall for ontology matching. In: Integrating Ontologies Workshop Proceedings (2005)
8. Gammulle, H., Denman, S., Sridharan, S., Fookes, C.: Predicting the future: A jointly learnt model for action anticipation. In: ICCV (2019)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
10. Graber, C., Tsai, G., Firman, M., Brostow, G., Schwing, A.: Panoptic segmentation forecasting. arXiv:2104.03962 (2021)
11. Guan, J., Yuan, Y., Kitani, K.M., Rhinehart, N.: Generative hybrid representations for activity forecasting with no-regret learning. In: CVPR (2020)
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
14. Hoai, M., De la Torre, F.: Max-margin early event detectors. IJCV (2014)
15. Hu, A., Cotter, F., Mohan, N., Gurau, C., Kendall, A.: Probabilistic future prediction for video scene understanding. In: ECCV (2020)
16. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: NeurIPS, pp. 2017–2025 (2015)
17. Jin, X., Li, X., Xiao, H., Shen, X., Lin, Z., Yang, J., Chen, Y., Dong, J., Liu, L., Jie, Z., et al.: Video scene parsing with predictive feature learning. In: ICCV (2017)
18. Jin, X., Xiao, H., Shen, X., Yang, J., Lin, Z., Chen, Y., Jie, Z., Feng, J., Yan, S.: Predicting scene parsing and motion dynamics in the future. In: NeurIPS (2017)
19. Kim, W., Tanaka, M., Okutomi, M., Sasaki, Y.: Adaptive future frame prediction with ensemble network. arXiv:2011.06788 (2020)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. Kwon, Y.H., Park, M.G.: Predicting future frames using retrospective cycle gan. In: CVPR (2019)
22. Li, W.H., Hong, F.T., Zheng, W.S.: Learning to learn relation for important people detection in still images. In: CVPR (2019)
23. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
24. Luc, P., Couprie, C., Lecun, Y., Verbeek, J.: Predicting future instance segmentation by forecasting convolutional features. In: ECCV (2018)
25. Luc, P., Neverova, N., Couprie, C., Verbeek, J., LeCun, Y.: Predicting deeper into the future of semantic segmentation. In: ICCV (2017)
26. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: ICLR (2016)
27. Qi, X., Chen, Q., Jia, J., Koltun, V.: Semi-parametric image synthesis. In: CVPR, pp. 8808–8816 (2018)
28. Qi, X., Liu, Z., Chen, Q., Jia, J.: 3d motion decomposition for rgbd future dynamic scene synthesis. In: CVPR (2019)
29. Qiao, X., Zheng, Q., Cao, Y., Lau, R.W.: Tell me where i am: Object-level scene context prediction. In: CVPR, pp. 2633–2641 (2019)
30. Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., Chopra, S.: Video (language) modeling: a baseline for generative models of natural videos. arXiv:1412.6604 (2014)
31. Rochan, M., et al.: Future semantic segmentation with convolutional lstm. In: BMVC (2018)
32. Šarić, J., Oršić, M., Antunović, T., Vražić, S., Šegvić, S.: Single level feature-to-feature forecasting with deformable convolutions. In: GCPR (2019)
33. Saric, J., Orsic, M., Antunovic, T., Vrazic, S., Segvic, S.: Warp to the future: Joint forecasting of features and feature motion. In: CVPR (2020)
34. Shalev-Shwartz, S., Ben-Zrihem, N., Cohen, A., Shashua, A.: Long-term planning by short-term prediction. arXiv:1602.01580 (2016)
35. Vondrick, C., Pirsiavash, H., Torralba, A.: Anticipating the future by watching unlabeled video. In: CVPR (2016)
36. Vondrick, C., Torralba, A.: Generating the future with adversarial transformers. In: CVPR (2017)
37. Walker, J., Gupta, A., Hebert, M.: Dense optical flow prediction from a static image. In: ICCV (2015)
38. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: ICCV (2019)
39. Yuen, J., Torralba, A.: A data-driven approach for event prediction. In: ECCV (2010)
40. Zhou, Y., Berg, T.L.: Learning temporal transformations from time-lapse videos. In: ECCV (2016)

## Author Biographies

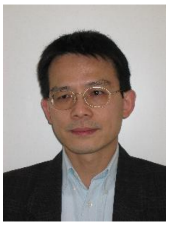
**Xiaotian Qiao** received the B.Eng. and M.Sc. degrees in information and communication engineering from Zhejiang University, China, and the Ph.D. degree in computer science from City University of Hong Kong. He is now a Postdoctoral Researcher in the Department of Computer Science, City University of Hong Kong. His research interests include computer vision and computer graphics.



**Quanlong Zheng** received the B.S. degree from BUAA in 2015. He is currently a PhD student at City University of Hong Kong. His research interest focuses on computer vision.



**Ying Cao** received the Ph.D. degree in computer science from the City University of Hong Kong, and the M.Sc. and B.Eng. degrees in software engineering from Northeastern University, China. His research generally lies in computer graphics and computer vision. His primary research interests is data-driven graphic design.



**Rynson W.H. Lau** received his Ph.D. degree from University of Cambridge. He was on the faculty of Durham University and is now with City University of Hong Kong.

Rynson serves on the Editorial Board of International Journal of Computer Vision (IJCV) and Computer Graphics Forum. He has served as the Guest Editor of a number of journal special issues, including ACM Trans. on Internet Technology, IEEE Trans. on Multimedia, IEEE Trans. on Visualization and Computer Graphics, and IEEE Computer Graphics & Applications. He has also served in the committee of a number of conferences, including Program Co-chair of ACM VRST 2004, ACM MTDL 2009, IEEE U-Media 2010, and Conference Co-chair of CASA 2005, ACM VRST 2005, ACM MDI 2009, ACM VRST 2014. Rynson's research interests include computer graphics and computer vision.