IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Segmenting the Future

Hsu-kuang Chiu[1], Ehsan Adeli[1], Juan Carlos Niebles[1]

***Abstract*— Predicting the future is an important aspect for decision-making in robotics or autonomous driving systems, which heavily rely upon visual scene understanding. While prior work attempts to predict future video pixels, anticipate activities or forecast future scene semantic segments from segmentation of the preceding frames, methods that predict future semantic segmentation solely from the previous frame RGB data in a single end-to-end trainable model do not exist. In this paper, we propose a temporal encoder-decoder network architecture that encodes RGB frames from the past and decodes the future semantic segmentation. The network is coupled with a new knowledge distillation training framework specific for the forecasting task. Our method, only seeing preceding video frames, implicitly models the scene segments while simultaneously accounting for the object dynamics to infer the future scene semantic segments. Our results on Cityscapes and Apolloscape outperform the baseline and current state-of-the-art methods. Code will be available soon.**

I. Introduction

Prediction of dynamics in visual scenes is one of the crucial components of intelligent decision-making in robotics and autonomous driving applications [1]. To this end, learning useful representations that enable reasoning about the future has recently been of great attention. Example applications are predicting visual context [2], forecasting human dynamics [3], tracking dynamics in scenes [4], [5].

In recent years, semantic and instance segmentation of videos [6] have become the leading methods to transform the scene into its semantic components, such as street, tree, vehicles, pedestrians, and obstacles. These semantic entities provide a high-level interpretation of the scene and hence predicting them can be of great interest. We argue that prediction of pixels in the RGB space is an overly perplexing task, while predicting high-level scene properties is sufficient, can be more useful, and is easier to interpret for decision-making purposes. Towards this direction, previous work predicted future semantic segments given the segmentation of the preceding frames [4], [7], or more sparsely predicted future instance segmentation from previous frames [5]. In contrast, we (1) do not require segmentation of previous frames and (2) provide a dense forecast for all regions in the frame.

Some other motion prediction methods used in existing robotics and autonomous driving systems apply object detection and tracking algorithms first [8], [9]. Then each of the tracked object future positions is predicted by state estimation approaches such as Kalman Filters or learning-based approaches with recurrent neural networks (RNN).

This work was supported by Panasonic and Oppo.

[1]All authors are with the Computer Science Department, Stanford University, 353 Jane Stanford Way, Stanford, CA, USA {hkchiu, eadeli, jniebles}@stanford.edu

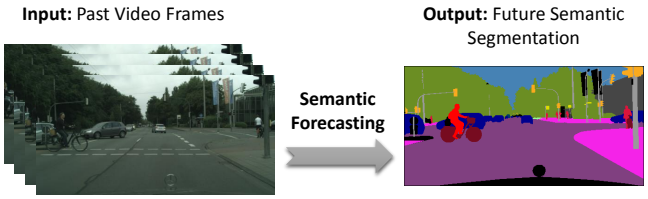

Fig. 1. We obtain future semantic segmentation directly from past frames in a single end-to-end trainable model. Our method implicitly infers the scene semantic segments while also forecasting the future configuration.

However, those methods can only forecast the motion of limited categories of objects, which can be detected and tracked. They neglect other categories such as roads and buildings. To correctly predict the position and shape of those objects relative to the ego-camera are also important for autonomous driving systems to navigate safely. Besides, the extra need for object detection and tracking modules could increase the computation overhead, system complexity, and memory usage, thus making the deployment more challenging. On the contrary, we focus on the semantic segmentation forecasting task that predicts the future high-level scene understanding for all objects. Furthermore, our proposed method does not depend on object detection or tracking, potentially avoiding the aforementioned disadvantages.

In this paper, we propose a model that predicts the future semantic segmentation in a video directly from pure RGB data of the previous frames (see Fig. 1). One relevant work [4] adopted a two-stage approach, first using the past RGB sequences to predict the future RGB frame, and then generating the future segmentation on top of that. On the contrary, one of our key observations is that future frame pixel values are *not* necessary for generating future semantic segmentations, which is itself an easier task than generating future pixel values. We propose a single stage end-to-end trainable model that learns to implicitly model the scene segments, and simultaneously account for the intrinsic dynamics of semantic maps for several object categories to predict future segmentation. In particular, this is a challenging task as objects in the semantic maps can significantly deform over the video frames due to changes in camera viewpoint, illumination, or orientation. To alleviate these challenges, our architecture encodes the sequence of input frames in a multi-resolution manner into a collective latent representation, and then decodes this representation to the future semantic map. We propose a novel knowledge distillation training framework that extracts future information to further refine the future semantic map. During the training stage, we utilize a fixed pre-trained single frame segmentation model and use it as a 'teacher network.' Taking the future frame as the input,

it predicts the future segmentation. The predicted output from the teacher network provides additional information to guide the training of our main forecasting model, denoted by 'student network.' This introduces one more training loss component, called distillation loss, which measures the difference between the outputs of the teacher and student networks. During inference, the teacher network is not used and the student network itself forecasts the future semantic segmentation using only the past RGB sequence as the input. This new distillation training framework further improves the forecasting performance.

To evaluate the performance of our method, we use the Cityscapes [10] and the Apolloscape [11] datasets under several scenarios, and compare our results with baseline methods. We predict the future semantic segmentation maps at three different temporal horizons (*i.e.*, , short-term, mid-term, and long-term) from the preceding RGB frames. Our method outperforms the previous methods although solving a much harder problem of predicting all semantic segments by only using past raw image sequences as input. Not only we define this harder problem and achieve the state-of-the-art, but we also outperform prior works under the simpler task that uses past semantic segmentation to forecast the future, even without modifying any part of our model.

In summary, our contributions are three-fold. First, we propose a single-stage end-to-end trainable model for the challenging task of predicting future semantic segmentation based on only the preceding RGB frames. Second, we propose a new knowledge distillation training framework that better uses future information. We introduce an additional distillation loss using a teacher network during training. We show that our method can uncover the relations between the previous and future frames while taking the motion into account. Third, our proposed model also outperforms the previous state-of-the-art methods on the simpler setting that uses past semantic segmentation to forecast the future.

II. Related Work

Semantic and Instance Segmentation: Several works show that intermediate visual representations, including semantic and instance segmentation, are significantly useful for robotics systems to learn better policies of indoor navigation [12], urban driving, and off-road trail traversal [13]. Semantic segmentation problems are often modeled by fully convolutional networks (FCN) [6], [14], U-net [15], [16], or by larger receptive fields [17]. On the other hand, instance segmentation often maintains a strategy to generate instance-proposal regions [18] as part of the segmentation pipeline.

Other works have explored the utilization of temporal information and consistency across frames [19], based on CRF models [20] or optical flow [21]. More recently, a number of methods utilize predictive feature learning techniques to enhance video segmentation [22].

Video Forecasting: Visual forecasting tasks were defined as extrapolating video pixels to create realistic future frames [14], [2]. Prediction future pixels can also be used in robot learning with model predictive control for manipulation tasks [23]. Although those video forecasting methods have some success in predicting the future at the pixel-level, modeling raw RGB pixel values is rather cumbersome in comparison with predicting future high-level properties of the video [24]. These high-level properties, such as semantic segmentation, can not only be sufficient for analysis in applications but also be more beneficial due to the higher level of abstraction.

TABLE I

Task setting comparison with prior work. We propose an end-to-end trainable model for forecasting future semantic segmentation (Seg) given only past RGB sequences.

Model	Input	Output
X2X [4]	RGB	RGB
S2S [4]	Seg	Seg
XS2X [4]	RGB+Seg	RGB
XS2S [4]	RGB+Seg	Seg
XS2XS [4]	RGB+Seg	RGB+Seg
ConvLSTM [7]	Seg	Seg
Ours	RGB	Seg

Recently, a few works proposed techniques for predicting semantic segmentation in videos. For instance, [4] predicted future scene segmentation either from segmentation of the preceding frames or from the combination of segmentation and RGB data of the previous frames. They also presented a two-stage approach that first predicts the future frame pixel values, and then generates segmentation maps on top of the predicted future frame. Three other relevant works [25], [7], [26] predicted the future segmentation from previous frame segmentation maps. The first [25] developed a method based on flow anticipation. The second and third [7], [26] developed convolutional LSTM (ConvLSTM) and deformable convolutional models respectively. Another work [5] developed a predictive model with Mask R-CNN for future instance segmentation. Their work can only predict the future movement for limited types of objects, but not for other critical classes of importance for autonomous driving applications, such as roads and buildings. In addition to using only the segmentation and RGB data, [27], [28] include extra ego-motion information to further improve the prediction accuracy. In summary, we mainly use X2X [4] and ConvLSTM [7] as the baselines, since they are the closest works to ours. However, their models rely on sequences of past semantic segmentation as the input [7], or a two-stage approach by forecasting the future RGB frame as an intermediate step [4] (see Table I). In contrast, we introduce an end-to-end trainable model for predicting the future segmentation solely based on the preceding RGBs.

Knowledge Distillation: Knowledge distillation [29] was originally proposed to compress the knowledge from an ensemble of models into a single model during the training. This idea was extended to distill knowledge from different data modalities, such as optical flow and depth information for action recognition and video classification tasks [30]. Different from the previous work, we propose a teacher network that takes the input from the same modality, the RGB frame, but in a different temporal range.

III. Method

Our goal is to build a model to forecast the future semantic segmentation given the past RGB sequence. Our proposed architecture involves two networks, the **student network** and the **teacher network**. The former performs our main forecasting task and the latter, during training, uses the future RGB frame to provide additional guidance to help the student network. At inference, only the student network is used to complete the semantic forecasting task, as shown in Fig. 2.

The student network, as shown in the upper half of Fig. 2, has three main components: encoder, forecasting module, and decoder. The **encoder** generates feature maps in multiple resolutions from each input frame. For a video observed up to time t, the inputs are X_{t-3d}, X_{t-2d}, X_{t-d}, and X_t, where d denotes the displacement between each pair of the preceding frames. The **forecasting module** uses the feature maps (the lowest level maps from each past frame pathway) to learn a latent-space representation by consolidating temporal dynamics across them. This module uses a temporal 3D convolution structure and acts as a predictive feature learning module integrating feature maps from the preceding frames. Finally, the **decoder** combines the spatial feature maps (through skip connections to the encoder) and the temporal features (output of the Conv3D module) to generate the final semantic segmentation of the future frame at time $t + d'$. Note that d' denotes the time delay in the future for which the semantic segmentation is sought. The ground-truth future segmentation is referred to by $S_{t+d'}$ and the prediction by $\hat{S}_{t+d'}$. The choice of d' defines how far in the future we plan to segment. We experiment on three different settings of the combinations of d and d' for short-term, mid-term, and long-term semantic segmentation forecasting.

The lower half of Fig. 2 shows how the teacher network generates the additional loss to help train the student network. Unlike X2X [4] that used the future RGB frame as an intermediate training target, our model uses the future frame in a new knowledge distillation approach during training. The teacher network can be any fixed pre-trained single frame semantic segmentation network. It uses the future frame $X_{t+d'}$ as the input to predict the future segmentation. The difference between the pre-softmax output features from the teacher network and the one from the student network is used as the additional training loss. During inference, the student network alone predicts the future segmentation as the output.

A. Student Network

This network contains three main components: past encoder, forecasting module, and future decoder to perform our main forecasting task, as shown in the upper half of Fig. 2.

Encoder: In contrast to the previous semantic segmentation methods, which are based on encoder-decoder FCN models, our encoder module contains parallel pathways, one for each input preceding frame. Each pathway contains a series of fully convolutional networks, non-linearity layers, and max-pooling layers, to generate multi-resolution feature maps. The encoder can be designed using the common image classification models, such as VGG [31] or ResNet [32], where the feature maps in different resolutions can be extracted right before each max-pooling layer. In our proposed method, we choose VGG19 with batch normalization as our encoder backbone. Refer to the appendix for the architecture details.

Forecasting Module: We introduce a forecasting module to learn predictive features and representations of temporal dynamics. For this purpose, we choose a 3D convolution network Conv3D to combine the encoded feature maps in the lowest resolution of each encoder pathway. This is a simple design choice, but proves to be more accurate than LSTM [33] or ConvLSTM [34] in our experiments. The intuition behind using a Conv3D forecasting module is that it is able to learn more various motion dynamic information. On the contrary, the LSTM-based models always encode all the past information into a fixed size embedding at every timestep. The Conv3D module can access features from any past timesteps and learn to combine them in various ways to model the scene dynamics. This motivates our design of using Conv3D as the forecasting module. In addition to the 3D convolution pathway, skip connections provide another set of interactions between the encoder and the decoder.

Decoder: The decoder takes the encoded feature maps at each resolution and the temporal dynamics representation as inputs and generates the future semantic segmentation, $\hat{S}_{t+d'}$. Different decoder structures are used in previous works. As an example, ConvLSTM [7], reused the decoder of FCN [6] to obtain the segmentation at the original resolution.

We build our decoder symmetric to the encoder module sequence. This choice gives us more computation capacity than the FCN decoder. Furthermore, the feature maps from different resolutions of the encoder can be directly concatenated with their counterparts in the decoder pathway, maintaining fine boundary information. As a result of this structure, the lower level feature representation can be fed to the transpose convolution layer, which contains trainable parameters and upsamples the lower resolution feature to a higher resolution. Then, it is concatenated with the encoder feature maps of the corresponding resolution from the latest past time-step followed by 2D convolution modules.

To construct the future segmentation from the decoder, the final convolution layer of the decoder generates a pre-softmax output tensor $O \in \mathbb{R}^{H \times W \times C}$, where H and W are the height and the width of the frames, and C is the number of semantic categories. A soft-max on O generates the predicted probability distribution tensor $P \in \mathbb{R}^{H \times W \times C}$:

$$P(h, w, c) = \frac{\exp(O(h, w, c))}{\sum_{i=1}^{C} \exp(O(h, w, i))}, \tag{1}$$

where h, w are the coordinates of a pixel, and c is the index of each semantic class. For each pixel $x = (h, w) \in \{H \times W\}$ and each semantic class c, we define the predicted probability function $p_c(x) = P(h, w, c)$, where $x = (h, w)$ represents the index of a pixel in the frame. To generate the final forecasting output, we choose the class with the highest probability as our prediction for that pixel. Hence, the predicted category for pixel $x = (h, w)$ at future time $t + d'$ is defined as: $\hat{S}_{t+d'}(x) = \mathrm{argmax}_c \, p_c(x)$.

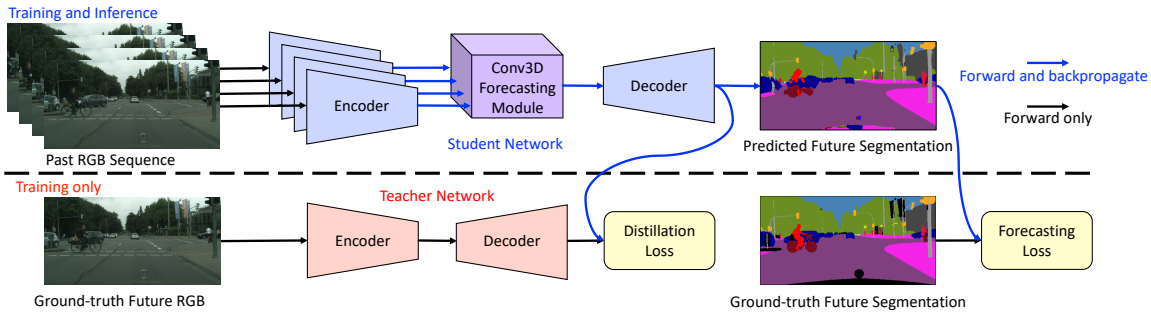

Fig. 2. Architecture overview: our student network contains encoder, forecasting module, and decoder to forecast future semantic segmentation. During training, a fixed pre-trained teacher network takes the future RGB frame as input and predicts the future segmentation. The training loss is the weighted sum of the cross-entropy forecasting loss and the mean-squared error distillation loss. During inference, the teacher network is not used and the student network alone performs the forecasting task.

B. Training Loss and Teacher Network

During the training, our loss L is defined as the combination of the forecasting loss L_f and the distillation loss L_d using weighted sum $L = L_f + \lambda L_d$, where λ is a hyperparameter. The forecasting loss L_f measures the difference between the predicted output of the model $\hat{S}_{t+d'}$ and the ground-truth semantic segmentation $S_{t+d'}$. We use the cross-entropy function to define this classification loss:

$$L_f = -\frac{1}{HW} \sum_{x \in \{H \times W\}} \log \left(p_{g(x)}(x) \right), \tag{2}$$

where $g(x) \in \{1, \ldots, C\}$ defines the ground-truth label for pixel x. The second loss term L_d is used to measure the difference between the outputs from the student network and the teacher network. The teacher network (lower half of Fig. 2) is a fixed pre-trained single frame segmentation model. The teacher network has the same encoder and decoder architecture as the student network, but without sharing the trainable parameters. The teacher network takes the future RGB frame $X_{t+d'}$ as input and generates its own predicted future semantic segmentation $\hat{S}^{te}_{t+d'}$. Instead of directly comparing the predicted semantic segmentation, we define the distillation loss L_d using the mean squared error between the pre-softmax output tensors from the student network and the teacher network:

$$L_d = \frac{1}{HWC} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{c=1}^{C} (O(h,w,c) - O_{te}(h,w,c))^2, \tag{3}$$

where O is the pre-softmax output tensor from the student network, and O_{te} is the one from the teacher network. With this distillation loss L_d, the single frame semantic segmentation teacher network provides additional regression guidance for the student network. During training, we minimize the overall loss L. During inference, the predicted output of the student network $\hat{S}_{t+d'}$ is the final output of our model.

IV. Experimental Results

Datasets: We first evaluate our method on the *Cityscapes* [10] dataset. In the training, validation, and testing sets, the dataset provides 2975, 500, and 1525 annotated frames with 19 semantic classes. In each of the video clips of length 30 (frames are indexed 0 to 29), the dataset provides fine annotations for the 19th frame. In total there are 180,000 frames of resolution of 1024×2048 pixels. Following the same setting of [7], [4], we only use the finely annotated frames and downsample the frames to resolution of 256×512. We train our model using the Adam optimizer with initial learning rate of 0.001, batch size 8, and $\lambda = 100$ (the two loss terms will have similar numerical scale in the beginning of training). We initialize the encoder of the student network with ImageNet [35] pre-trained weights.

Apolloscape dataset [11] is also used as an additional experiment, which has 140,000 frames with pixel-level annotations of 22 semantic classes. We extracted 1950 training and 380 validation sequences and down-sampled each frame to resolution of 320×384.

Settings: Following the forecasting settings in the related previous works [4], [7], we design three experimental settings on different time-ranges: short-term, mid-term, and long-term. For all settings, we always define the 19th frame in the Cityscapes sequences, denoted by S_{19}, as the forecasting target frame, since the ground-truth semantic labels for this frame are available. The input RGB frames are selected from different timesteps in the past, denoted by X_i, depending on the forecasting time-range setting For short-term forecasting, the input RGB frames are $X_{15}, X_{16}, X_{17}, X_{18}$; for mid-term forecasting $X_7, X_{10}, X_{13}, X_{16}$; and for long-term forecasting they are X_1, X_4, X_7, X_{10}, while the output for all is $\hat{S}_{19}$. The Cityscapes sequences were recorded at a frame-rate of 17Hz, so our three time-range settings aim to predict 0.06, 0.18, and 0.53 seconds into the future respectively.

Evaluation Metrics: Following previous work, we use the mean Intersection Over Union (mIOU) as the performance metric for segmentation evaluation. We also report pixel-level accuracy (pAcc) and mean per-class accuracy (mAcc). Specifically, mIoU is the pixel IOU averaged across all classes; pAcc defines the percentage of correctly classified pixels; and mAcc is the average class accuracies.

Baseline Methods: We use the ConvLSTM [7] model as the main comparison baseline, which outperforms the previous state-of-the-art S2S [4]. This model achieves the state-of-the-art performance on a slightly different segmentation forecasting task, *i.e.*, , the inputs are the past segmentation sequences. The architecture of ConvLSTM is based on the bidirectional ConvLSTM temporal module and uses the asymmetric Resnet101-FCN encoder-decoder backbone.

Additionally, [4] has an X2X architecture, which is also

TABLE II

EVALUATION OF OUR METHOD IN TERMS OF MIOU, PIXEL-LEVEL ACCURACY (PACC), AND MEAN CATEGORY ACCURACY (MACC) IN COMPARISON WITH BASELINE AND RELEVANT METHODS ON FORECASTING FUTURE SEMANTIC SEGMENTATION USING PAST RGB SEQUENCES AS THE INPUTS. IN EACH COLUMN, THE BEST OBTAINED RESULTS ARE TYPESET IN BOLDFACE AND THE SECOND BEST ARE UNDERLINED.

Model	Short-term			Mid-term			Long-term		
	mIOU	pAcc	mAcc	mIOU	pAcc	mAcc	mIOU	pAcc	mAcc
Zero-motion	58.91	91.96	69.68	48.15	87.89	59.67	36.21	81.77	47.07
Optical-flow	60.87	93.33	70.51	49.99	89.50	62.03	36.22	82.57	47.56
Two-stage	49.17	90.22	61.68	26.53	74.60	36.19	9.64	44.86	14.49
X2X* [4]	-	-	-	23.00	-	-	-	-	-
ConvLSTM** [7]	45.08	89.28	54.15	36.81	85.79	45.57	27.36	80.44	24.63
Ours	**65.08**	**93.83**	**74.36**	**56.98**	**91.38**	**67.67**	**40.81**	**86.03**	**50.13**

*X2X is not the main focus of [4], and only the mid-term mIOU result is reported in this setting.
**Our implementation of [7], using past RGB sequences as the inputs.

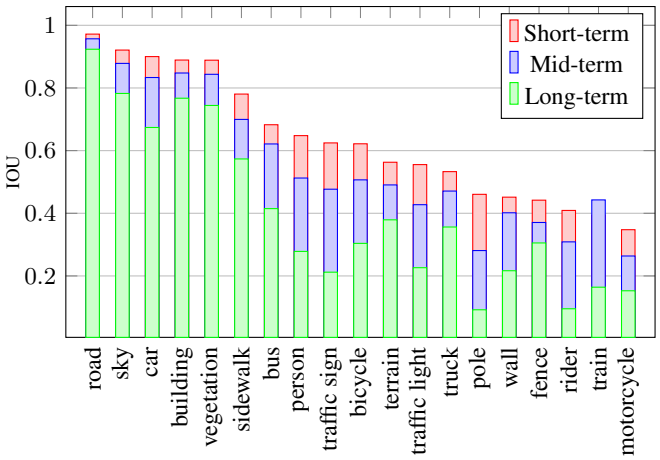

Fig. 3. Per-class IOU for all 19 classes with respect to short, mid, and long-term forecasting. The forecasting performance varies a lot across different classes, implying that some classes are more difficult to correctly classify.

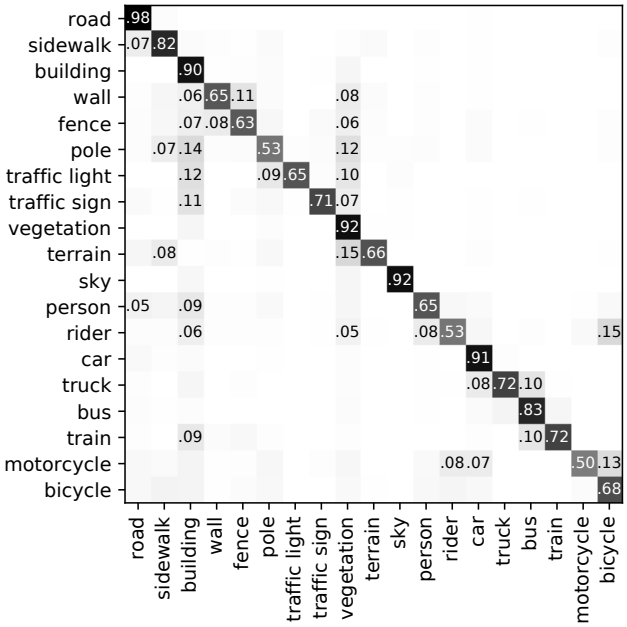

Fig. 4. Confusion matrix of the mid-term semantic segmentation forecasting for all 19 classes in the Cityscapes dataset. The x-axis refers to the predicted class labels and the y-axis represents the ground-truth class labels. For instance, the confusion matrix shows that for some cases, the label 'motorcycle' is misclassified as 'bicycle', 'rider', and 'car'.

used as our baseline. It primarily focused on the same problem setting as the one of ConvLSTM [7]. But, they also presented an X2X [4] architecture that uses the past RGB sequences to forecast the future RGB frame and then generate the future segmentation. We argue that it is not necessary to generate the future RGB frame as an intermediate step. Since [4] only presents the result in the mid-term time-range setting, we implement a two-stage model based on the same idea. Another baseline is denoted by 'zero-motion.' This is the case that no motion is anticipated in the video and the future frame semantic segmentation is identical to that of the last observed frame. Although this is a very naïve baseline, it poses as a very challenging one [36], [3], especially for short-term forecasting. To calculate this metric, we first train a single frame semantic segmentation model. Then, we apply it to the last input frame and use the segmentation map as the predicted result. We also include another baseline by warping the last input frame using the optical flow, followed by applying a single frame segmentation model.

A. Quantitative Results

Table II shows the results of our method on the Cityscapes [10] dataset, compared with the previous state-of-the-art. Note that ConvLSTM [7] focused on predicting future segmentation directly from past semantic segmentation as the inputs. Therefore, we re-implemented their model, but training and testing using past RGB sequences as the inputs. As can be seen in Table II, our model outperforms ConvLSTM [7] by a large margin in all the three time-ranges. That supports our design choice of using the symmetric encoder-decoder backbone and the Conv3D temporal module. We can also see the significant performance difference between the two-stage models, including X2X [4], and our proposed single-stage model. The performance difference supports our argument that forecasting future RGB frame is not necessary for forecasting future semantic segmentation. Besides, our method outperforms the zero-motion and optical-flow methods for all three time-ranges. Interestingly, those two simple baselines still outperform than X2X [4] and ConvLSTM [7].

In addition to quantitative comparisons with previous works, we analyze our forecasting results for each of the 19 classes of objects. Fig. 3 shows the IOU comparison for all classes over three different forecasting time-ranges, sorted in descending orders. One can notice that the prediction accuracy varies a lot across different classes. To better understand the reasons why certain classes have lower IOUs, we calculate the confusion matrix, as shown in Fig. 4. The x-axis in this figure refers to the predicted class labels and the

TABLE III

ABLATION RESULTS: EVALUATION OF OUR METHOD IN TERMS OF MIOU, PIXEL-LEVEL ACCURACY (PACC), AND MEAN CATEGORY ACCURACY (MACC) IN COMPARISON WITH VARIATIONS OF OUR METHOD ON FORECASTING FUTURE SEMANTIC SEGMENTATION USING PAST RGB SEQUENCES AS THE INPUTS. IN EACH COLUMN, THE BEST OBTAINED RESULTS ARE TYPESET IN BOLDFACE AND THE SECOND BEST ARE UNDERLINED.

Model	Short-term			Mid-term			Long-term		
	mIOU	pAcc	mAcc	mIOU	pAcc	mAcc	mIOU	pAcc	mAcc
Ours w/o symmetric backbone	47.80	89.65	57.31	39.37	87.03	47.72	28.22	81.51	35.95
Ours w/ LSTM	48.22	90.26	58.39	39.57	86.93	49.18	26.61	81.46	33.19
Ours w/ ConvLSTM	58.62	92.92	68.85	47.96	89.29	58.24	34.44	84.21	43.09
Ours w/ Multi-Res Conv3D	59.24	93.24	69.44	49.33	90.38	59.39	36.96	85.42	45.98
Ours w/o distillation loss	63.60	93.76	73.01	55.97	91.17	66.59	40.32	85.84	49.69
Ours w/ 2 input frames	63.85	93.67	73.30	55.67	91.15	66.58	40.37	85.75	50.02
Ours w/ 3 input frames	64.10	93.72	74.18	56.52	91.27	67.26	40.77	85.91	50.11
Ours	**65.08**	**93.83**	**74.36**	**56.98**	**91.38**	**67.67**	**40.81**	**86.03**	**50.13**

TABLE IV

COMPARISON OF MIOU WITH S2S [4] AND CONVLSTM [7] ON FORECASTING SEGMENTATION USING PAST SEGMENTATION AS INPUT.

Model	Short-term	Mid-term	Long-term
S2S	62.60	59.40	47.80
ConvLSTM	71.37	60.06	-
Ours	**72.43**	**65.53**	**50.52**

TABLE V

MIOU RESULTS ON APOLLOSCAPE DATASET.

Model	Short-term	Mid-term	Long-term
Zero-motion	29.87	25.44	19.56
ConvLSTM**	28.27	23.29	17.30
Ours w/o distillation loss	32.26	25.72	20.26
Ours	**32.58**	**26.09**	**20.47**

**Our implementation of ConvLSTM with RGB inputs.

y-axis represents the ground-truth class labels. For instance, the 'motorcycle' class, which had the lowest IOU in Fig. 3 is mainly confused with the 'bicycle', 'rider', and 'car' classes. Additionally, we can see two light gray vertical lines for the 'building' and 'vegetation' predicted classes. This shows that other classes are often mistaken with these two classes.

Ablation: Table III shows the ablation analysis results to examine where the performance improvements derive from. The main differences between our model and previous work are the symmetric encoder-decoder backbone, the Conv3D temporal forecasting module, and the distillation training.

First, to evaluate the symmetric encoder-decoder backbone design, we create another model by replacing our backbone architecture with the asymmetric one as in ConvLSTM [7], which simply uses the decoder of FCN (first row of table III). All the evaluation metrics are significantly worse than our proposed model. Our proposed symmetric backbone decoder is designed with more computational capacity compared to FCN decoder. In our problem setting, the inputs and outputs represent two different types of information (image and segmentation). They are potentially far away from each other in the latent representation space. Therefore, more computation capacity is required, compared with previous works [4], [7] whose inputs and outputs are all segmentation.

Next we analyze the impact of the temporal structures. We implement three other models by replacing our Conv3D temporal module with LSTM [33], ConvLSTM [34] and the multi-resolution Conv3D. The results are reported in the second, third, and the fourth rows of Table III. The LSTM module has significantly negative impact. ConvLSTM performs much better than LSTM, but still worse than our proposed Conv3D temporal module. The multi-resolution Conv3D module contains total five Conv3D layers, each of which takes the past feature maps from different resolution and generates the dynamic information for the decoder. Empirically, we observe worse performance. One possible reason is that the larger number of trainable parameters makes the model prone to over-fitting. Furthermore, it requires more memory, which forces the model to operate on a smaller batch size. Therefore, the regularization effect of batch normalization becomes less effective.

Finally, we analyze the impact of the student-teacher architecture and the distillation training loss. Without the teacher network and the distillation training loss, results of our student-only model are shown in the fifth row of Table III. Our student-only model already outperforms other previous works shown in Table II. Using the distillation loss of Eq. (3), the mIOU scores further improve by 0.5% to 1.5% mIOU scores for all the three time-range settings

Furthermore, we also experiment on using fewer numbers of input frames, and the results are shown in the sixth and seventh rows in Table III. Using fewer numbers of input frames decreases the mIOU scores by 0.04% to 1.31%. Note that we are unable to experiment on using more than four input frames due to the limitation of Cityscapes [10] dataset.

Forecasting Segmentation from Past Segmentation: We also experiment on a simpler task that forecasts future segmentation from past segmentation. This task is the exact problem setting that ConvLSTM [7] achieves the state-of-the-art. Our model still outperforms ConvLSTM[7] and another strong baseline, S2S[4], as in Table IV.

Quantitative Results on Apolloscape Dataset: The previous works [4], [7] only experimented on Cityscapes[10]. Additionally, we further experiment on Apolloscape[11]. As shown in Table V, our proposed model still outperforms ConvLSTM [7]. Notice that Apolloscape is more challenging than Cityscapes , therefore we see smaller performance gains.

B. Qualitative Results

Mid-term Forecasting: We start this section with the mid-term forecasting results, as this is the most widely used setting in the previous works. Fig. 5(b) shows the qualitative results, which uses the past RGB sequence $X_7, X_{10}, X_{13}, X_{16}$ to forecast the future semantic segmentation at time-step 19, denoted as S_{19}. In this figure, each row is a separate sample sequence, and the left most column is the past RGB

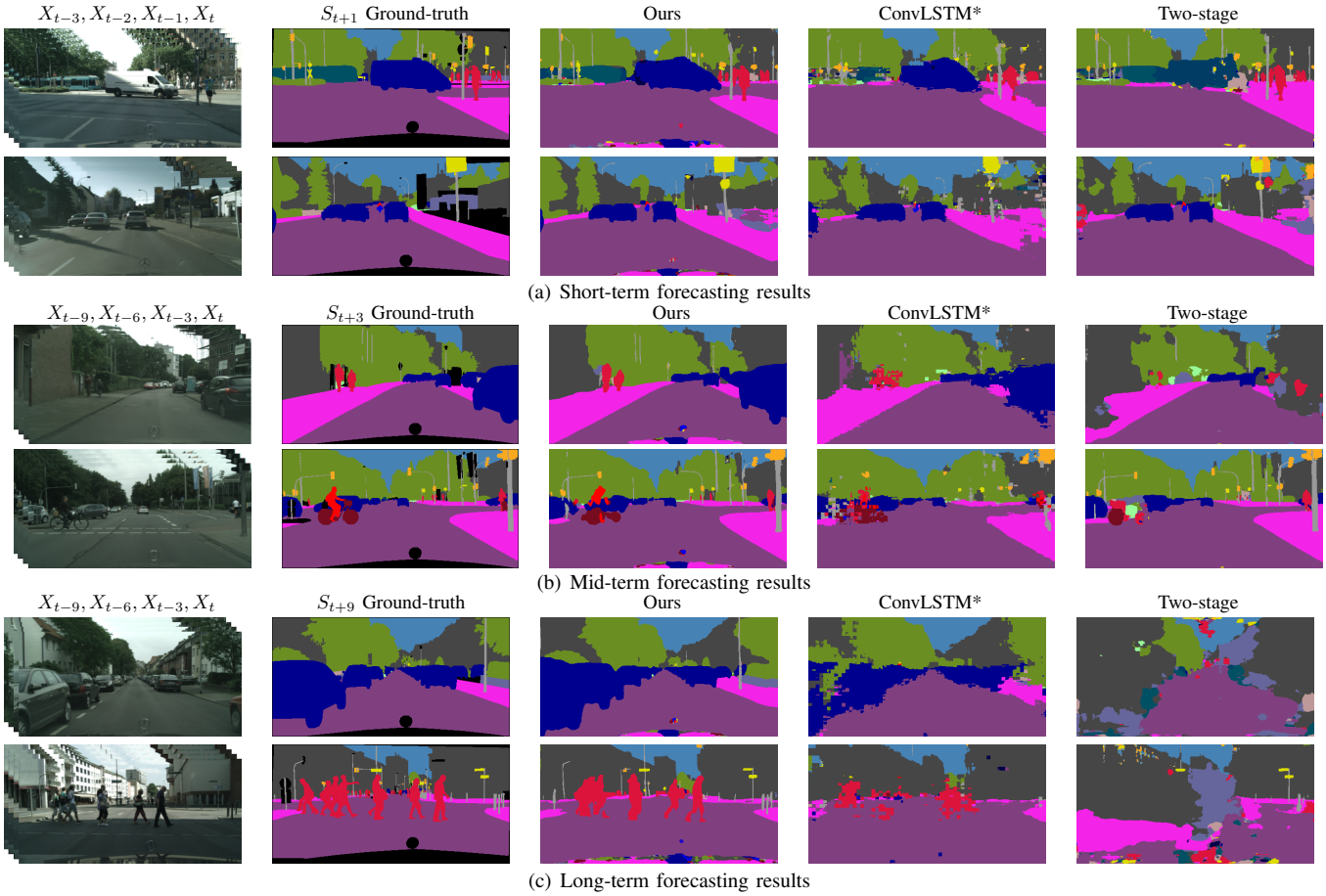

Fig. 5. Qualitative results. X denotes the input RGB frames and S the ground-truth segmentation. See the appendix for more results. (*Our implementation of [7], using past RGB sequences as the inputs.)

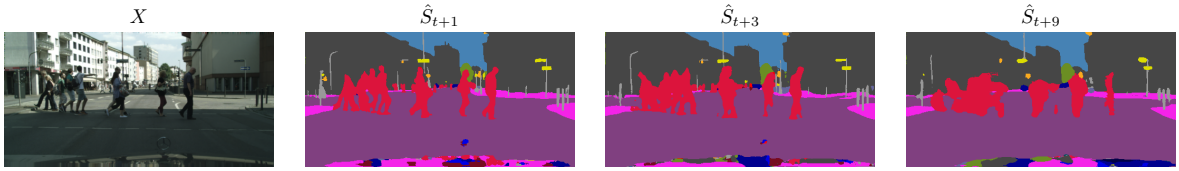

Fig. 6. Forecasting results of three different time-range settings for the same sample. X is the last RGB frame; $\hat{S}_{t+1}$, $\hat{S}_{t+3}$, and $\hat{S}_{t+9}$ denote the predicted segmentation of short-, mid-, and long-term settings, corresponding to the same X.

input sequence, followed by the ground-truth future semantic segmentation, our prediction result and two baselines.

The first row shows examples where the camera is moving forward. Our model accurately captures the relative motion dynamic between the camera and all the objects in the scene. Our prediction results show that the right-side street-parking car segmentation moves toward right further, similar to the ground-truth. The second example shows that our model can capture and predict the future based on different motion patterns. Specifically, a biker and a car are moving toward each other. Our model can predict these two segments will intersect in the future frame, but was unable to figure out which one should be in the foreground due to the lack of depth information. This problem can be potentially solvable by including additional depth information if provided. See the appendix for more qualitative results.

Short-term Forecasting: In the short-term setting, we use the past RGB sequence $X_{15}, X_{16}, X_{17}, X_{18}$ to forecast the future semantic segmentation S_{19}. Our model precisely predicts both the directions and the magnitude of the movements for the cars, as shown in Fig. 5(a). In the first row, the car is moving toward right, and the predicted car position in the future frame is the same as that in the ground-truth. In the second example, the left parked car is moving out of the frame due to the camera motion. Again, our model captures the exact relative motion dynamic information and precisely forecasts the same shape, size, and location of the same parked car in the future frame. One interesting finding is that the ground-truth annotation actually misses the circle direction sign (yellow color in the segmentation map) while our model is capable to detect that direction sign segment and place it in the right position in the future frame.

Long-term Forecasting: Fig. 5(c) shows the long-term forecasting results. Our model uses the past RGB sequence

X_1, X_4, X_7, X_{10} to generate the future semantic segmentation S_{19}. Such setting is more challenging but our model can still accurately predict the moving directions of the parked cars and pedestrians in the first and second examples respectively. However, the magnitude of the movements seem to be smaller than the ground-truth. This may be due to changes in the speeds of the objects in the frames that are not observed by our model.

Time-horizon Comparison: Fig. 6 shows the forecasting results for our three time horizons. For all these settings, we use the same frame, namely X_{16}, as the last input frame. We forecast the future segmentation at three different time horizon, namely $\hat{S}_{17}$, $\hat{S}_{19}$, and $\hat{S}_{25}$. The short-term forecasting result provides the best visual quality. From the mid-term result, the segmentation boundary of the pedestrians are still reasonable. But for the long-term result, their shapes start to deform away from regular pedestrian appearance. However, we can still see that different groups of the pedestrians are moving toward their destination in the correct directions, *e.g.*, the left most pedestrian is moving toward left, and the right most one is moving toward right.

V. Conclusion

We proposed a single-stage end-to-end trainable model for the challenging problem of predicting future frame semantic segmentation having only observed the preceding frames RGB data. This is a practical setting for autonomous systems to directly reason about the near future based on current video data without the need to acquire any other forms of meta-data. Our proposed model for solving this task included several encoding pathways to encode the past, a temporal 3D convolution structure for capturing the scene dynamics and predictive feature learning, and finally a decoder to reconstruct the future semantic segmentation. We further proposed a teacher network coupled with a distillation loss for training the network to improve the overall forecasting performance. The results on the popular Cityscapes and Apolloscape datasets indicate that our method can predict future segmentation and outperform several baseline and state-of-the-art methods.

VI. Acknowledgement

This work was partially funded by Panasonic and Oppo. The authors would like to thank Jingwei Ji and Damian Mrowca for their feedback on the paper.

References

[1] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *CVPR*, 2018.

[2] K.-H. Zeng, W. B. Shen, D.-A. Huang, M. Sun, and J. C. Niebles, "Visual forecasting by imitating dynamics in natural sequences," in *ICCV*, 2017.

[3] H.-k. Chiu, E. Adeli, B. Wang, D.-A. Huang, and J. C. Niebles, "Action-agnostic human pose forecasting," in *WACV*, 2019.

[4] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun, "Predicting deeper into the future of semantic segmentation," in *ICCV*, 2017.

[5] P. Luc, C. Couprie, Y. Lecun, and J. Verbeek, "Predicting future instance segmentations by forecasting convolutional features," 2018.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[7] S. S. Nabavi, M. Rochan, and Y. Wang, "Future semantic segmentation with convolutional lstm," in *BMVC*, 2018.

[8] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *ICRA*, 2019.

[9] B. Liu, E. Adeli, Z. Cao, K.-H. Lee, A. Shenoi, A. Gaidon, and J. C. Niebles, "Spatiotemporal relationship reasoning for pedestrian intent prediction," in *ICRA*, 2020.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[11] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," in *CVPR*, 2018.

[12] A. Sax, B. Emi, A. Zamir, L. Guibas, S. Savarese, and J. Malik, "Learning to navigate using mid-level visual priors," in *CoRL*, 2019.

[13] B. Zhou, P. Kraehenbuehl, and v. Koltun, "Does computer vision matter for action?" in *Science Robotics*, 2019.

[14] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *ICLR*, 2016.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.

[17] S. Zhou, D. Nie, E. Adeli, Y. Gao, L. Wang, J. Yin, and D. Shen, "Fine-grained segmentation using hierarchical dilated neural networks," in *MICCAI*, 2018.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.

[19] L. Shao, P. Shah, V. Dwaracherla, and J. Bohg, "Motion-based object segmentation based on dense rgb-d scene flow," in *IROS*, 2018.

[20] A. Kundu, V. Vineet, and V. Koltun, "Feature space optimization for semantic video segmentation," in *CVPR*, 2016.

[21] Y.-S. Xu, T.-J. Fu, H.-K. Yang, and C.-Y. Lee, "Dynamic video segmentation network," in *CVPR*, 2018.

[22] X. Jin, X. Li, H. Xiao, X. Shen, Z. Lin, J. Yang, Y. Chen, J. Dong, L. Liu, Z. Jie, *et al.*, "Video scene parsing with predictive feature learning," in *ICCV*, 2017.

[23] F. Ebert, C. Finn, A. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections," in *CoRL*, 2017.

[24] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," in *NeurIPS*, 2018.

[25] X. Jin, H. Xiao, X. Shen, J. Yang, Z. Lin, Y. Chen, Z. Jie, J. Feng, and S. Yan, "Predicting scene parsing and motion dynamics in the future," in *NeurIPS*, 2017.

[26] J. Saric, M. Orsic, T. Antunovic, S. Vrazic, and S. Segvic, "Single level feature-to-feature forecasting with deformable convolutions," in *GCPR*, 2019.

[27] S. Vora, R. Mahjourian, S. Pirk, and A. Anelia, "Future semantic segmentation leveraging 3d information," in *ECCV Workshop*, 2018.

[28] A. Bhattacharyya, M. Fritz, and B. Schiele, "Bayesian prediction of future street scenes using synthetic likelihoods," in *ICLR*, 2019.

[29] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NeurIPS workshop*, 2014.

[30] Z. Luo, J.-T. Hsieh, L. Jiang, J. C. Niebles, and L. Fei-Fei, "Graph distillation for action detection with privileged information," in *ECCV*, 2018.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2015.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[33] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[34] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *NeurIPS*, 2015.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[36] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *CVPR*, 2017.