Video Scene Parsing with Predictive Feature Learning

Xiaojie Jin[1] Xin Li[2] Huaxin Xiao[2] Xiaohui Shen[3] Zhe Lin[3] Jimei Yang[3]
Yunpeng Chen[2] Jian Dong[5] Luoqi Liu[4] Zequn Jie[4] Jiashi Feng[2] Shuicheng Yan[5,2]
[1]NUS Graduate School for Integrative Science and Engineering (NGS), NUS
[2]Department of ECE, NUS [3]Adobe Research [4]Tencent AI Lab [5]360 AI Institute

Abstract

Video scene parsing is challenging due to the following two reasons: firstly, it is non-trivial to learn meaningful video representations for producing the temporally consistent labeling map; secondly, such a learning process becomes more difficult with insufficient labeled video training data. In this work, we propose a unified framework to address the above two problems, which is to our knowledge the first model to employ predictive feature learning in the video scene parsing. The predictive feature learning is carried out in two predictive tasks: frame prediction and predictive parsing. It is experimentally proved that the learned predictive features in our model are able to significantly enhance the video parsing performance by combining with the standard image parsing network. Interestingly, the performance gain brought by the predictive learning is almost costless as the features are learned from a large amount of unlabeled video data in an unsupervised way. Extensive experiments over two challenging datasets, Cityscapes and Camvid, have demonstrated the effectiveness of our model by showing remarkable improvement over well-established baselines.

1. Introduction

Video scene parsing aims to predict per-pixel semantic labels for every frame in scene videos recorded in unconstrained environments. It has drawn increasing attention as it benefits many important applications like drones navigation, autonomous driving and virtual reality.

In recent years, remarkable success has been made by deep convolutional neural network (CNN) models in image parsing tasks [3, 5, 22, 30, 31, 36, 47, 50, 18, 48]. Some of the **frame parsing models**[1] are then proposed to be used for parsing scene videos frame by frame. However, such frame parsing models suffer from noisy and inconsistent labeling results across frames, since the important temporal context cues are ignored. For example, in the second row of Figure 2, the top-left region of *building* in the frame T is incorrectly classified as *car*, which is temporally inconsistent with the parsing result of its preceding frames. More importantly, current CNN models are hungry for data and finely annotated video data for training are rather labor-intensive to collect and limited. Even in the very recent scene parsing dataset Cityscapes [4], there are only 2,975 finely annotated training samples *vs.* overall 180,000 video frames. Deep CNN models are prone to over-fitting when trained using a small training data set and thus generalize badly in real applications.

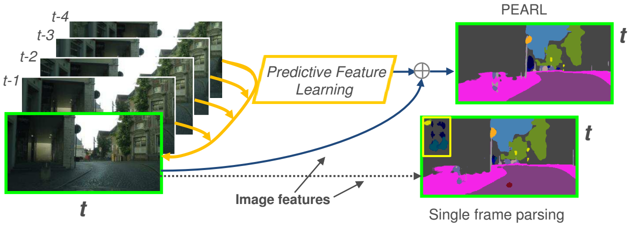

Figure 1: Illustration of the proposed PEARL. PEARL solves the video scene parsing problem through predictive feature learning that includes frame prediction (yellow arrows) and predictive parsing which learn temporal-aware features across frames to augment image features (also learned in PEARL). PEARL produces accurate and temporally consistent parsing results (top-right) compared with standard frame parsing results (bottom-right).

To tackle these two problems, we propose a novel **P**arsing with pr**E**dictive fe**A**tu**R**e **L**earning (**PEARL**) model, which is both annotation-efficient and effective for the video scene parsing task. The basic idea of PEARL is illustrated in Figure 1. Through predictive learning with a GAN-like architecture, PEARL learns powerful temporal representations to capture rich video dynamics as well as high-level video contexts which are critical for video scene parsing. By effectively utilizing the predictive features of both semantic and temporal information, PEARL substan-

[1] For clarity, we use *frame parsing model/network* to indicate *conventional image parsing model/network* which takes a single frame as input.

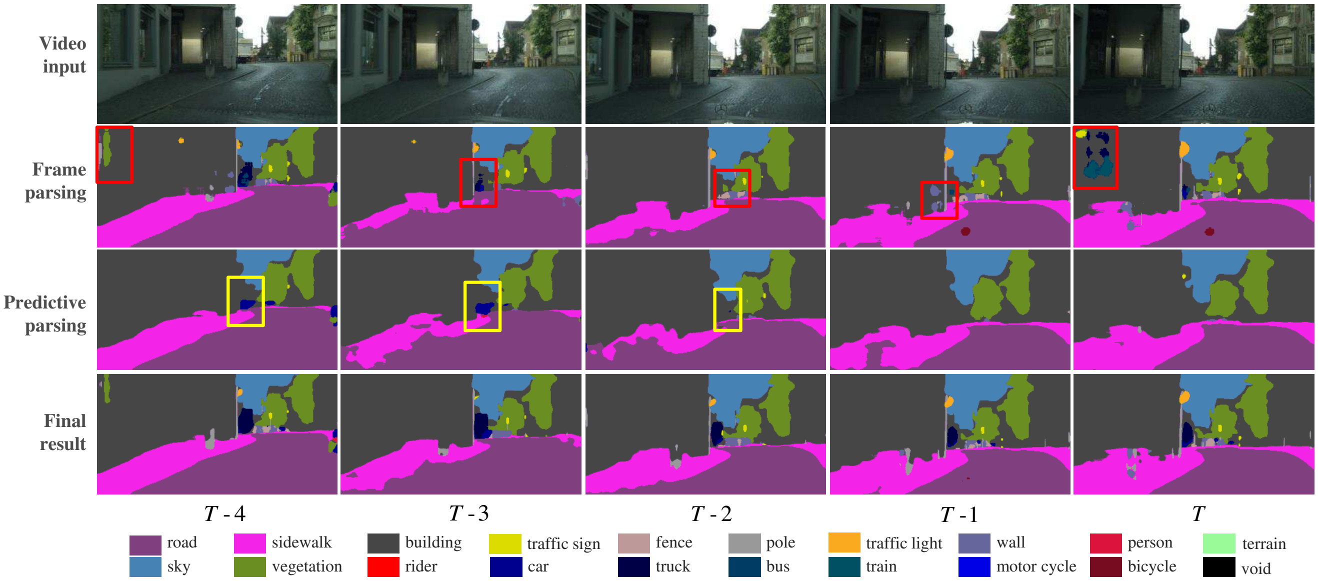

Figure 2: Illustration of the working principle of PEARL on video scene parsing. **Top**: a five-frame sequence to be parsed. **Second row**: frame parsing produced by the state-of-the-art image parsing model [3] which takes a single frame as input. Since such a model is unable to model temporal context, severe noise and inconsistency across frames can be observed within the red boxes. **Third row**: results from predictive parsing. The regions showing inconsistency in the second row are classified consistently across frames using the learned predictive features. Besides, the motion trajectories of moving objects (*cars*) are correctly captured (see yellow boxes). Note for each frame to be parsed, the input for predictive parsing is its preceding frames which are not fully presented for brevity. **Bottom**: labeling maps produced by PEARL with better accuracy and temporal consistency. PEARL combines the advantages of conventional frame parsing model (the second row) and predictive parsing (the third row).

tially improves the video scene parsing performance over the standard frame parsing models. Attractively, our predictive learning process is annotation-efficient as it learns features from unlabeled videos data which are nearly unlimited. To the best of our knowledge, PEARL is the first predictive learning model for the video scene parsing task.

To make the working principle clearer, we illustrate practical examples in Figure 2. Concretely, PEARL conducts two complementary predictive learning phases for video scene parsing. In phase I, as shown in the top row of Figure 2, given historical frames T-4 to T-1, PEARL learns to predict the future frame T using a GAN-like architecture. Such a process enables PEARL to learn discriminative video representations to capture rich temporal cues across frames. In phase II, PEARL further adapts its predictive learning component trained in phase I to the predictive parsing task, *i.e.* predicting the labeling maps of annotated frames only using their preceding frames. Through such a predictive parsing task, PEARL injects semantic information to those frame-prediction features learned in phase I, thus providing features more powerful for video scene parsing. At the same time, we apply this predictive learning network together with a standard frame parsing network, which are both end-to-end trained for video scene parsing. The predictive learning network can provide powerful high-level temporal representations to augment the standard frame-parsing network which is unable to model temporal information in videos. Consequently, the integrated model finally produces more accurate and temporally consistent parsing results (as shown in the bottom row of Figure 2).

We conduct extensive experiments on two challenging scene parsing datasets and compare PEARL with strong baselines built upon state-of-the-art VGG16 and Res101 image parsing models. Our model achieves the best results on both datasets. We also compare PEARL with optical flow methods. Extensive experiments demonstrate that it can learn stronger video dynamics features than simple optical flow and largely boost the performance of video scene parsing without requiring extra supervision information.

To summarize, we make the following contributions to video scene parsing:

- To the best of our knowledge, PEARL is the first systematic predictive learning model for video scene parsing. The proposed model presents a strong ability to learn temporal representations and high-level video context from unlabeled video data.
- We develop an effective model to utilize the predictive

learning features to produce temporally smooth and structure preserving parsing maps for video frames.

- Our model achieves state-of-the-art performance on two challenging datasets, *i.e.*, Cityscapes and Camvid.

2. Related Work

Recent image scene parsing progress is mostly stimulated by various new CNN architectures, including the fully convolutional architecture (FCN) with multi-scale or larger receptive fields [5, 22, 36, 49] and the combination of CNN with graphical models [3, 30, 48, 50, 31]. There are also some recurrent neural networks based models [12, 17, 28, 34, 41]. However, without incorporating the temporal information, directly applying them to every frame of a video leads to parsing results that commonly lack cross-frame consistency.

To utilize temporal consistency across frames, the motion and structure features in 3D data are employed by [6, 37, 46]. In addition, [9, 14, 16, 25] use CRF to model temporal context. However, those methods suffer from high computation cost as they need to perform expensive inference of CRF. Some other methods [11, 32] employ optical flow to capture the temporal consistency. Different from the above works that heavily depend on labeled data for supervised learning, our proposed model takes advantage of both the labeled and unlabeled video sequences to learn richer temporal context information.

Generative adversarial networks are firstly introduced in [8] to generate natural images from random noise, and have been widely used in many fields including image synthesis [8], future prediction [23, 26, 42, 43] and semantic inpainting [27]. Our model also uses the adversarial loss to learn more robust video representations in frame predictions. Our model is more related to [23, 26, 24] which performs frame prediction. However, different from [23, 26, 24], PEARL tackles the video scene parsing problem by utilizing the temporal information learned in frame prediction.

3. Parsing with Predictive Feature Learning

PEARL aims to address two challenges in video scene parsing: 1) how to learn effective video representations to guarantee cross-frame smoothness and structure preserving in parsing results; 2) how to build effective parsing models even in presence of insufficient labeled training data.

PEARL solves these two problems through a novel predictive feature learning strategy. The predictive feature learning is performed on partially labeled videos, which are denoted as $\{\mathcal{X}, \mathcal{Y}\}$, where $\mathcal{X} = \{X_i, i = 1, \ldots, N\}$ denotes the raw video frames and $\mathcal{Y} = \{Y_j, j = 1, \ldots, M\}$ denotes the pixel-wise annotations for a subset of $\mathcal{X}$. Here $M \ll N$ as only a small portion of the video frames are annotated. $Y_j(u, v) \in \{1, \cdots, C\}$ denotes the ground truth class at pixel location (u, v) in which C is the number of classes. Correspondingly, let $\hat{X}_i$ and $\hat{Y}_i$ denote predicted frames and predicted labeling maps respectively. We use $P_i^s = \{X_{i-k}\}_{k=1}^s$ to denote the s preceding frames ahead of X_i. For the first several frames in a video, we define their preceding set as $X_{i-k} = X_1$ if $i \leq k$.

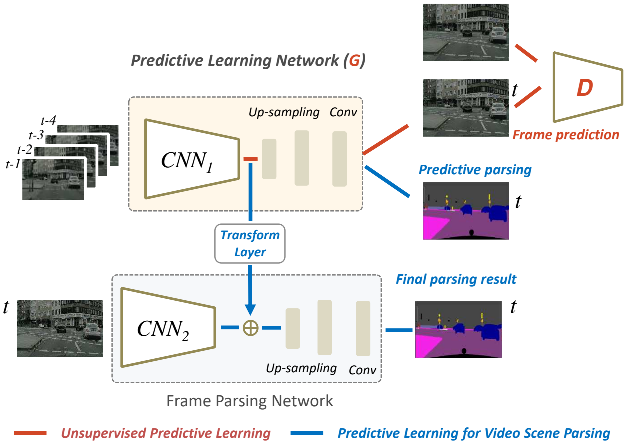

Figure 3: The framework of our proposed parsing with predictive feature learning model (PEARL). The core component is the novel predictive learning network. There are two predictive learning phases in PEARL, the working flows of which are highlighted in red and blue, respectively. In phase I (red), the predictive learning network (used as "G") is pre-trained in the video frame prediction under the GAN-like architecture [8]. In phase II, the predictive learning network is transferred to the predictive parsing task and jointly trained end-to-end with a standard frame parsing network. The symbol $\oplus$ denotes "concatenation".

3.1. Model Overview

The framework of the proposed PEARL model is illustrated in Figure 3. The core component is the novel *predictive learning network*. Abstractly, the predictive learning network learns temporal representations through predicting the pixel values or labeling maps of future frames. There are two learning phases in PEARL (highlighted by red and blue colors respectively). In phase I (unsupervised predictive learning), we pre-train a predictive learning network within a video frame prediction task to capture the variability of content and dynamics of videos. This learning process is performed in an unsupervised manner with the GAN-like architecture. In phase II (predictive learning for video scene parsing), the predictive learning network is further trained to predict the labeling maps of future frames, and at the same time used to augment the standard frame parsing network to produce more accurate and temporally consistent parsing results. To summarize, the predictive learning network plays two important roles: on one hand, it learns meaningful temporal-aware features from unlabled video data which addresses the problem of insufficient labeled training data; on the other hand, the learned features convey temporal context cues to the frame parsing network, thus enabling PEARL to produce temporally consistent parsing results on video frames.

3.2. Unsupervised Predictive Learning

In phase I, PEARL performs unsupervised predictive learning in a video frame prediction task using a GAN-like architecture which is highlighted in red in Figure 3. The learning process involves two components, *i.e.*, the predictive learning network (which plays as the generator, denoted as G) to produce frame prediction $\hat{X}_i = G(P_i^s)$ based on preceding video frames P_i^s, and the discriminator (denoted as D) which plays against G by identifying the predicted frame $\hat{X}_i$ and the real one X_i. The predictive learning network first maps the input video sequence to temporal representations via the feature encoder CNN_1 which are then spatially enlarged via up-sampling layers and finally fed to a convolutional layer to produce the pixel-wise RGB values. Since deep CNNs, *e.g.* VGGNet [35] and ResNet [10] are generally designed to take images as input, we adapt them to video inputs by using group convolution [13] for the first convolutional layer (thus no extra parameter is added), where the group number is equal to that of input frames.

The predictive learning network G and D are alternatively trained to predict frames. Denote the learnable parameters of D and G as W_D and W_G respectively. The objective for training D is to classify the input X_i into class 1 and the input $D(G(P_i^s))$ into class 0 while keeping W_G fixed. The loss function we use to train D is

$$\min_{W_D} \ell_D \triangleq -\log(1 - D(G(P_i^s); W_D)) - \log D(X_i; W_D).$$

G learns to predict future frames that look both similar to the corresponding real frame and sufficiently authentic to fool the strong competitor D. The objective loss for training G is the combination of reconstruction loss (the first term) and adversarial loss (the second term) while keeping W_D fixed:

$$\min_{W_G} \ell_G = \|X_j - \hat{X}_j\|_2 - \lambda_{adv} \log D(G(P_i^s; W_G)). \quad (1)$$

Note the GAN model used here is different from vanilla GAN [8] and tailored for video scene parsing. The key difference lies in the generator G, *i.e.* the predictive learning network that takes the past frame sequence as temporally conditioned input, instead of crafting new samples completely from random noise as vanilla GANs. Therefore, such a temporally conditioned generator could generate temporally consistent future frames w.r.t. the input past frames. More importantly, G can learn representations containing temporal cues desired for solving video scene parsing problems.

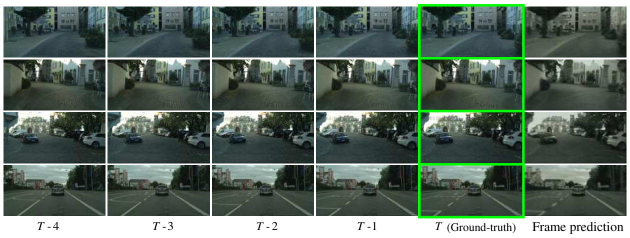

Figure 4: Example frame predictions from the predictive learning network on Cityscapes val set. In each row, the first four images are input to the predictive learning network. The fifth image (green) is the frame to be predicted. The last image is the predicted frame.

As illustrated in Figure 4, the predictive learning network produces real-looking frame predictions by learning both the content and dynamics in videos. By comparing with the ground truth frames, the predictions resemble both the structures of objects/stuff, *e.g. building/vegetation* and the motion trajectories of objects, *e.g. cars*, demonstrating that the predictive learning network learns robust and generalized temporal representations from video data.

In our experiments, we use a GoogLeNet [38] as D and G is modified from Res101 or VGG16. They are all trained from scratch using unlabeled video data. More details are given in Sec. 4.1.

3.3. Predictive Learning for Video Scene Parsing

The above predictive learning network is then used to augment the standard frame parsing network. To adapt the features learned in phase I to video scene parsing problems, we further adapt the predictive learning network to the *predictive parsing* task *i.e.*, predicting the labeling map of X_i which has pixel-wise annotations only given its preceding frames P_i^s.

As illustrated in Figure 3 (highlighted in blue), the predictive features output by CNN_1 are integrated in PEARL with a standard frame parsing network through a transform layer (*i.e.* a shallow CNN). Such a frame parsing network also consists of three components, a feature encoder (CNN_2) followed by up-sampling layers and the output convolutional layer. Through the transform layer, the predictive learning network communicates its learned temporal representations to the frame parsing network. Combining these two types of features offers two appealing properties, *i.e.*, descriptiveness for the temporal context in videos and discriminability for local pixels within a single frame. Consequently, PEARL is benefited from the predictive features and the local discriminative features, and can produce more accurate video scene parsing results.

Formally, the objective function of training PEARL is defined as the summation of the loss (ℓ_P) on the predictive learning network and the loss (ℓ_I) on the frame parsing network:

$$\min_{W,\Phi} L = \ell_I + \lambda_P \ell_P, \tag{2}$$

$$\ell_I = -\sum_{(u,v)\in X_j} f_{Y_j(u,v)}(W, X_j, T_P), \tag{3}$$

$$\ell_P = -\sum_{(p,q)\in X_j} h_{Y_j(u,v)}(\Phi, P_j^s), \tag{4}$$

where Φ and W denote the learnable parameters in the predictive learning network and the frame parsing network, respectively. T_P is the transformed feature output by the transform layer. $f_{Y_j(u,v)}(\cdot)$ and $h_{Y_j(u,v)}(\cdot)$ denote the per-pixel logarithmic probability on the ground-truth class produced by the frame parsing network and the predictive learning network, respectively. In experiments, we use the re-weighting strategy [34] which balances the effects of those two networks.

Note that during the training of phase II, all parameters except those in the output layer in the predictive learning network are initialized from W_G which is learned in the unsupervised frame prediction phase (ref. to Eqn. (1)). We observe that training Φ from scratch harms the performance. The reasons are as follows: since there are no enough data with pixel-wise annotations for training a good model free from over-fitting, it cannot directly train the predictive learning network for the predictive parsing task from scratch. Therefore, training the predictive learning network for frame prediction at first gives a good starting model for training PEARL in phase II.

Visualization of Predictive Parsing Results As shown in Figure 2, compared with those from the frame parsing network, the parsing maps from the predictive learning network present two distinct properties. First, the labeling maps are temporally smooth which are reflected in the parsing results like the *building* region where the frame parsing network produces noisy and inconsistent parsing results. This demonstrates the predictive learning network indeed learns the temporally continuous dynamics from the video data. Secondly, the predictive learning network tends to miss objects of small sizes, *e.g.*, *transport signs* and *poles*, which are captured by the frame parsing network due to its relying on locally discriminative features. One reason for missing small objects is the inevitable blurry prediction [26] since the high frequency spectrum is prone to being smoothed.

The Role of the Transform Layer Now we proceed to explain the role of the transform layer. Compared with naively combining the features from two networks (*e.g.*, concatenation), the transform layer brings following two advantages: 1) naturally normalize the feature maps to proper scales; 2) align the features of semantic meaning such that the integrated features are more powerful for parsing. Effectiveness of this transform layer is clearly validated in the ablation study in Section 4.2.1.

4. Experiments

In this section, we present the details and analysis the results in our experiments. We conduct extensive ablation studies to verify the effectiveness of PEARL. On both the Cityscapes and Camvid dataset, PEARL achieves the state-of-the-art performance.

4.1. Settings and Implementation Details

Datasets Since PEARL tackles the video scene parsing problem which involves temporal context, we choose Cityscapes [4] and Camvid [1] as evaluation benchmarks. Both datasets provide densely annotated frames as well as their adjacent frames, suitable for testing the temporal modeling ability. Cityscapes is a large-scale dataset containing fine pixelwise annotations on 2,975/500/1,525 train/val/test frames with 19 semantic classes and another 20,000 coarsely annotated frames. Each finely annotated frame is the 20th frame of a 30-frame video clip in the dataset which contains in total $180K$ frames. Every frame in Cityscapes has a large resolution of $1{,}024 \times 2{,}048$ pixels.

The Camvid dataset contains 701 color images with annotations on 11 semantic classes. These images are extracted from driving videos captured at daytime and dusk. Each video contains 5,000 frames on average, with a resolution of 720×960 pixels, amounting to in total $40K$ frames.

Baselines To demonstrate that PEARL can be applied with advanced deep architectures, we compare PEARL with multiple baselines which use the following two state-of-the-art deep architectures.

- **VGG16-baseline** VGG16-baseline is built upon DeepLab [3], which is the state-of-the-art image parsing model. To further enhance its ability for video scene parsing, we add three deconvolutional layers (each followed by ReLU) after fc7 to up-sample the output features. Besides, for fc7 features, we use the ParseNet contexture module which is proposed in [20] to encode global features. The details of its architecture are given in supplementary material due to space limitation.
- **Res101-baseline** The Res101-baseline is built upon the most recent Res101 [10] model following FCN [22]. Besides, like [44], we use hard training sample mining to reduce over-fitting. Architectural details are provided in supplementary material.

In our experiments, both CNN$_1$ and CNN$_2$ share the same network architectures as baseline models for fair comparison. Moreover, for each baseline model, we report

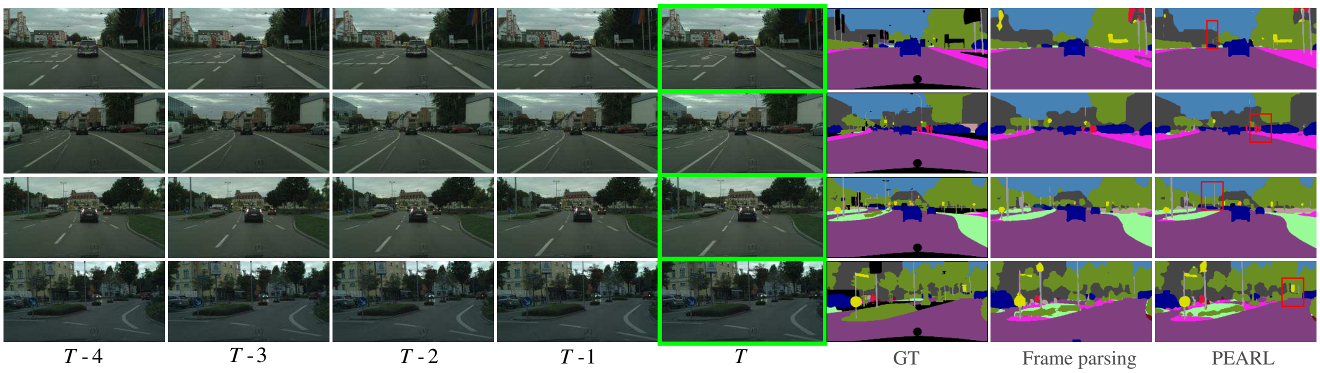

Figure 5: Examples of parsing results of PEARL on Cityscapes val set. In each row, the first five images are from a same video sequence, followed by ground-truth annotations, the respective labeling map of the baseline model and our proposed PEARL, all for frame T (highlighted in green). It is observed that PEARL produces more smooth parsing maps and shows stronger discriminability for small objects (highlighted in red boxes) compared to the baseline model.

the result of **an ensemble** of two baseline models finetuned independently (with the same number of parameters as PEARL). All results of PEARL are based on single-model single-scale testing.

Evaluation Metrics Following previous practice, we use the mean IoU (mIoU) for Cityscapes, and per-pixel accuracy (PA) and average per-class accuracy (CA) for Camvid. In particular, mIoU is defined as the pixel intersection-over-union (IOU) averaged across all classes; PA is defined as the percentage of all correctly classified pixels; and CA is the average of all class-wise accuracies.

Implementation Details Throughout the experiments, we set the number of preceding frames of each frame as 4, *i.e.*, $s = 4$ in P_i^s (ref. Sec. 3). In phase I, we randomly select frame sequences of length 4 with enough movement (the ℓ_2 distance between the raw frames is larger than a threshold 230). In this way, we finally obtain $35K/8.8K$ sequences from Cityscapes and Camvid respectively. The input frames in phase I are all normalized such that values of their pixels lie between -1 and 1. In phase II, we only perform mean value subtraction on the frames. For the predictive learning network in phase II, we select 4 frames before the annotated frames as input, where the frames are required to have sufficient motion, consistent with phase I.

For data augmentation, we use random cropping and random mirror for all datasets in two phases. In addition, in phase I, the temporal order of a sequence (including the frame to be predicted and 4 preceding frames) is randomly reversed to model various dynamics in videos. We set $\lambda_{adv} = 0.2$ in Eqn. (1) and $\lambda_P = 0.3$ in Eqn. (2) and the probability threshold of hard training sample mining [44] in Res101-baseline as 0.9. The values are chosen through cross-validation. SGD with momentum is employed throughout training. For other hyperparameters including weight decay, learning rate, batch size and epoch number *etc.*, please refer to the Supplementary Material.

Computational Efficiency Since no post-processing is required in PEARL, the running time of PEARL adopting Res101 architecture to obtain the parsing map of a frame with resolution $1{,}024 \times 2{,}048$ is only 0.8 seconds on a modern GPU, among the fastest models in existing works.

4.2. Results

Examples of the final parsing maps produced by PEARL on Cityscapes val set are illustrated in Figure 5, where VGG16 architecture is used in the baseline model and PEARL. We present evaluation results with more details on the two datasets as well as ablation studies below.

4.2.1 Cityscapes

Ablation Analysis We investigate the contribution of each component of the proposed PEARL model.

(1) Predictive Feature Learning. To investigate the effect of the predictive features learned in two phases on the video scene parsing performance, we conduct the following three experiments. The comparison results are listed in Table 1, where VGG16 architecture is used in both PEARL and the baseline.

First, we verify the effectiveness of the predictive features learned in phase I (learned from frame prediction). We concatenate the transformed output features of CNN$_1$ in the predictive learning network with the output features of CNN$_2$, as shown in Figure 6, and fix the parameters of CNN$_1$ during training. In this way, the predictive learning network only provides pre-computed video features to assist the video scene parsing task. As can be seen from Table 1,

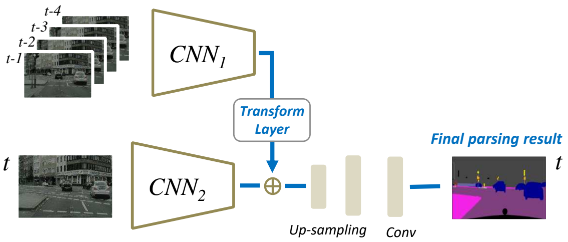

Figure 6: A variant of PEARL to verify the effectiveness of predictive features. The features of the predictive learning network are concatenated with a standard frame parsing network via the transform layer. The weights of CNN_1 are fixed during training.

Table 1: Comparative study of effects of predictive features on final performance over Cityscapes val set. VGG16 architecture is used in baseline and PEARL. $\texttt{feat}_{\text{CNN}_1}$ denotes the output features of "CNN_1" after phase I ends, while $\texttt{feat}_{\text{CNN}_1^+}$ denotes the output features after fine-tuning "CNN_1" with predictive parsing.

Methods	mIoU
VGG16-baseline	64.5
$\texttt{feat}_{\text{CNN}_1}$ + VGG16-baseline	69.2
$\texttt{feat}_{\text{CNN}_1^+}$ + VGG16-baseline	69.8
PEARL	**70.4**

by combining $\texttt{feat}_{\text{CNN}_1}$, the mIoU increases from 64.5 (of the VGG16-baseline) to 69.2, demonstrating the predictive learning network indeed learns useful temporal representations through frame prediction for video scene parsing.

Next, we investigate how adapting the predictive learning network to predictive parsing provides stronger predictive features for parsing. In this experiment, we firstly fine-tune the pre-trained predictive learning network in the predictive parsing task (see Eqn. (4)), and then similar to the above experiment, we fix the parameters of CNN_1 and concatenate the transformed output features of CNN_1^+ (CNN_1^+ is distinguished from CNN_1 by fine-tuning on predictive parsing) with the output features of CNN_2. As illustrated in Table 1, the mIoU further increases by 0.6 using $\texttt{feat}_{\text{CNN}_1^+}$ compared with using $\texttt{feat}_{\text{CNN}_1}$, demonstrating the features learned from predictive parsing are beneficial for video scene parsing.

Finally, we look into the effectiveness of joint training of the predictive parsing network and the frame parsing network. It is observed from Table 1 that the best performance is achieved using PEARL, benefiting from the joint end-to-end training strategy.

(2) Comparison with Temporal Modeling Methods. To verify the superiority of PEARL on learning the temporal representations specific for video scene parsing, we compare PEARL with other temporal context modeling methods including optical flow. First, we naively pass each of s preceding frames in P_j^s and X_j through baseline models and merge their probability maps to obtain the final parsing map of X_i. It is experimentally verified that such a method achieves worse performance (mIoU on VGG16-Baseline:

Table 2: Comparative study of PEARL with optical flow based method. `OF` means the optical flow maps augmented training data. PEARL in the upper/lower panel in the table adopts VGG16/Res101 architecture respectively.

Methods	mIoU
VGG16-baseline	64.5
`OF` + VGG16-baseline	64.7
PEARL	**70.4**
Res101-baseline	73.2
`OF` + Res101-baseline	73.5
PEARL	**76.5**

Table 3: Comparison with state-of-the-arts on Cityscapes val set. PEARL in the upper/lower panel in the table adopts VGG16/Res101 architecture respectively. Single-model, single-scale testing is used in PEARL w/o post-processing like CRF.

Methods	mIoU
VGG16-baseline (ours)	64.5
FCN (*CVPR-15*) [22]	61.7
Pixel-level Encoding (*CVPRW-16*) [40]	64.3
DPN (*ECCV-15*) [21]	66.8
Dilation10 (*ICLR-16*) [45]	67.1
DeepLab-VGG16 (*Arxiv-16*) [2]	62.9
Deep Structure (*Arxiv-16*) [19]	68.6
Clockwork FCN (*ECCVW-16*) [33]	64.4
PEARL (ours)	**70.4**
Res101-baseline (ours)	73.2
DeepLab-Res101 (*Arxiv-16*) [2]	71.4
PEARL (ours)	**76.5**

Table 4: Comparison with state-of-the-arts on Cityscapes *test* set. Res101 architecture is used in PEARL. Note for fast inference, single-model, single-scale testing is used in PEARL without any post-processing like CRF.

Methods	mIoU
FCN_8s (*CVPR-15*) [22]	65.3
DPN (*ECCV-15*) [21]	66.8
Dilation10 (*ICLR-16*) [45]	67.1
DeepLab (*Arxiv-16*) [2]	70.4
Deep Structure (*Arxiv-16*) [19]	71.6
LRR-4X (*ECCV-16*) [7]	71.8
RefineNet (*CVPR-17*) [18]	73.6
PEARL (ours)	**75.2**

60.9 versus 64.5; mIoU on Res101-Baseline: 70.3 versus 73.2) than baseline models due to its weakness in utilizing temporal information and the noisy probability maps produced for each frame.

Since optical flow is naturally capable of modeling the temporal information in videos, we use it as a strong baseline to compete with PEARL. Firstly, the epic flow [29] is employed to compute all optical flows. Then we concatenate the optical flow maps calculated from X_{j-1} to X_j with X_j to form 5-channel raw training data $\bar{X}_j$ (RGB plus X/Y channels of optical flow). Using the optical flow augmented training data $\{(\bar{X}_j, Y_j), j = 1, \ldots, M\}$, we re-train baseline models. During training, each kernel in the first convolutional layer of baseline models is randomly initialized for the weights corresponding to the X/Y channels of optical flow. This method is referred to as "`OF` + Baseline". The comparison results of "`OF` + Baseline" and PEARL are shown in Table 2. From the results, one can observe "`OF` + Baseline" achieves higher performance than baselines as it models temporal context during training. Notably, PEARL beats "`OF` + Baseline" on both network architectures, proving its capability of modeling temporal information for video scene parsing problems.

(3) Ablation Study of the GAN Loss in Phase I We conduct experiments to evaluate how GAN loss (adversarial loss) contributes. The results show that GAN loss indeed enhances PEARLs performance. Compared with PEARL w/o GAN loss, PEARL w/ GAN loss improves the mIoU on Cityscape val by 1.0 and 0.6 for VGG16 and Res101 backbone architectures respectively. This is because with the GAN loss, the generator (CNN$_1$) in the predictive learning network learns more descriptive features for video contents and dynamics and produces more realistic frames. Such features are critical to produce temporally consistent video parsing results.

(4) Ablation Study of the Transform Layer As introduced in Sec. 3.3, the transform layer improves the performance of PEARL by learning the latent feature space transformations from $\texttt{feat}_{\text{CNN}_1}$ to $\texttt{feat}_{\text{CNN}_2}$. In our experiments, the transform layer contains one residual block [10] which has been widely used due to its good performance and easy optimization. Details of the residual block used in our experiments are deferred to supplementary material. Compared to the PEARL w/o the transform layer, adding the transform layer brings $1.2/0.5$ mIoU improvements for PEARL adopting VGG16 and Res101 architecture respectively. We also conduct experiments by stacking more residual blocks, but only observe marginal improvements at larger computational cost.

Comparison with State-of-the-arts The comparison of PEARL with other state-of-the-arts on Cityscapes val set is listed in Table 3, from which one can observe PEARL achieves the best performance among all compared methods on both network architectures. Note loss re-weighting is not used on this dataset.

Specifically, PEARL adopting VGG16/Res101 architecture significantly improve the corresponding baseline models by $5.9/3.3$ mIoU, respectively. Notably, compared with [33] which proposed a temporal skip network based on VGG16 for video scene parsing, PEARL beats it by 6.0 in terms of mIoU. We also note that different from other methods which extensively modify VGG16 networks to enhance the discriminative power for frame parsing, *e.g.* [2, 19], PEARL is built on the vanilla VGG16 architecture. Thus it is reasonable to expect further improvement on the performance by using more powerful front CNN architectures. Furthermore, we compare PEARL adopting Res101 architecture with other state-of-the-arts on Cityscapes test set. As shown in Table 4, our method achieves the best performance among all top methods.

4.2.2 Camvid

We report the comparison results of PEARL with state-of-the-arts in Table 5. Due to limited space, more experimental details are deferred to Supplementary Material.

Table 5: Comparison with the state-of-the-art on CamVid. Res101 architecture is used in PEARL.

Methods	PA(%)	CA(%)
Res101-baseline (ours)	92.7	80.8
Ladicky *et al.*(*ECCV-10*) [15]	83.8	62.5
SuperParsing(*ECCV-10*) [39]	83.9	62.5
DAG-RNN (*CVPR-16*) [34]	91.6	78.1
MPF-RNN (*AAAI-17*) [12]	92.8	82.3
Liu *et al.* (*ECCV-15*) [21]	82.5	62.5
RTDF (*ECCV-16*) [16]	89.9	80.5
PEARL (ours)	**94.4**	**83.2**

5. Conclusion

We proposed a new predictive feature learning model for effective video scene parsing. It contains two learning phases. The first phase learns temporal representations in an unsupervised manner by predicting future frames from unlabeled video data. The second phase integrates the predictive learning network and a standard frame parsing network to produce temporally smooth and structure preserving results. Extensive experiments on Cityscapes and Camvid fully demonstrated the effectiveness of our model.

Acknowledgement The work of Jiashi Feng was partially supported by NUS startup R-263-000-C08-133, MOE Tier-I R-263-000-C21-112 and IDS R-263-000-C67-646.

References

[1] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*. 2008. 5

[2] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016. 7, 8

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 1, 2, 3, 5

[4] 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. *arXiv preprint arXiv:1604.01685*, 2016. 1, 5

[5] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1915–1929, 2013. 1, 3

[6] G. Floros and B. Leibe. Joint 2d-3d temporally consistent semantic segmentation of street scenes. In *CVPR*, 2012. 3

[7] G. Ghiasi and C. C. Fowlkes. Laplacian reconstruction and refinement for semantic segmentation. In *ECCV*, 2016. 7

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 3, 4

[9] B. L. . X. H. . S. Gould. Multi-class semantic video segmentation with exemplar-based object reasoning. In *WACV*, 2016. 3

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 4, 5, 8

[11] J. Hur and S. Roth. Joint optical flow and temporally consistent semantic segmentation. In *ECCV*, 2016. 3

[12] X. Jin, Y. Chen, J. Feng, Z. Jie, and S. Yan. Multi-path feedback recurrent neural network for scene parsing. In *AAAI*, 2017. 3, 8

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 4

[14] A. Kundu, V. Vineet, and V. Koltun. Feature space optimization for semantic video segmentation. In *CVPR*, 2016. 3

[15] L. Ladickỳ, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What, where and how many? combining object detectors and crfs. In *ECCV*. 2010. 8

[16] P. Lei and S. Todorovic. Recurrent temporal deep field for semantic video labeling. In *ECCV*, 2016. 3, 8

[17] M. Liang, X. Hu, and B. Zhang. Convolutional neural networks with intra-layer recurrent connections for scene labeling. In *NIPS*, 2015. 3

[18] G. Lin, A. Milan, C. Shen, and I. D. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 1, 7

[19] G. Lin, C. Shen, A. v. d. Hengel, and I. Reid. Exploring context with deep structured models for semantic segmentation. *arXiv preprint arXiv:1603.03183*, 2016. 7, 8

[20] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 5

[21] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ECCV*, 2015. 7, 8

[22] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 3, 5, 7

[23] W. Lotter, G. Kreiman, and D. Cox. Unsupervised learning of visual structure using predictive generative networks. *arXiv preprint arXiv:1511.06380*, 2015. 3

[24] W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *ICLR*, 2017. 3

[25] B. Mahasseni, S. Todorovic, and A. Fern. Approximate policy iteration for budgeted semantic video segmentation. *CoRR*, abs/1607.07770, 2016. 3

[26] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 3, 5

[27] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. *arXiv preprint arXiv:1604.07379*, 2016. 3

[28] P. H. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene parsing. *arXiv preprint arXiv:1306.2795*, 2013. 3

[29] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015. 8

[30] A. Roy and S. Todorovic. Scene labeling using beam search under mutex constraints. In *CVPR*, 2014. 1, 3

[31] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015. 1, 3

[32] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. Optical flow with semantic segmentation and localized layers. *arXiv preprint arXiv:1603.03911*, 2016. 3

[33] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell. Clockwork convnets for video semantic segmentation. In *ECCVW*, 2016. 7, 8

[34] B. Shuai, Z. Zuo, G. Wang, and B. Wang. Dag-recurrent neural networks for scene labeling. *arXiv preprint arXiv:1509.00552*, 2015. 3, 5, 8

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[36] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *ICML*, 2011. 1, 3

[37] P. Sturgess, K. Alahari, L. Ladicky, and P. H. Torr. Combining appearance and structure from motion features for road scene understanding. In *BMVC*, 2009. 3

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 4

[39] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*. 2010. 8

[40] J. Uhrig, M. Cordts, U. Franke, and T. Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. In *CVPRW*, 2016. 7

[41] F. Visin, K. Kastner, A. C. Courville, Y. Bengio, M. Matteucci, and K. Cho. Reseg: A recurrent neural network for object segmentation. *CoRR*, abs/1511.07053, 2015. 3

[42] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, 2016. 3

[43] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. In *ICCV*, 2015. 3

[44] Z. Wu, C. Shen, and A. van den Hengel. High-performance semantic segmentation using very deep fully convolutional networks. *CoRR*, abs/1604.04339, 2016. 5, 6

[45] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 7

[46] C. Zhang, L. Wang, and R. Yang. Semantic segmentation of urban scenes using dense depth maps. In *ECCV*, 2010. 3

[47] R. Zhang, S. Tang, M. Lin, J. Li, and S. Yan. Global-residual and local-boundary refinement networks for rectifying scene parsing predictions. In *IJCAI*, 2017. 1

[48] Y. Zhang and T. Chen. Efficient inference for fully-connected crfs with stationarity. In *CVPR*, 2012. 1, 3

[49] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016. 3

[50] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 1, 3