DISTORTION-AWARE NETWORK PRUNING AND FEATURE REUSE FOR REAL-TIME VIDEO SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-time video segmentation is a crucial task for many real-world applications such as autonomous driving and robot control. Since state-of-the-art semantic segmentation models are often too heavy for real-time applications despite their impressive performance, researchers have proposed lightweight architectures with speed-accuracy trade-offs, achieving real-time speed at the expense of reduced accuracy. In this paper, we propose a novel framework to speed up any architecture with skip-connections for real-time vision tasks by exploiting the temporal locality in videos. Specifically, at the arrival of each frame, we transform the features from the previous frame to reuse them at specific spatial bins. We then perform partial computation of the backbone network on the regions of the current frame that captures temporal differences between the current and previous frame. This is done by dynamically dropping out residual blocks using a gating mechanism which decides which blocks to drop based on inter-frame distortion. We validate our Spatial-Temporal Mask Generator (STMG) on video semantic segmentation benchmarks with multiple backbone networks, and show that our method largely speeds up inference with minimal loss of accuracy.

1 INTRODUCTION

Semantic segmentation (Zhao et al., 2017; 2018; Yu et al., 2018; Chen et al., 2019; Li et al., 2020a; Fan et al., 2021; Hong et al., 2021) is a fundamental task in computer vision that is crucial for many real-world applications including autonomous driving, robot control, augmented reality, surveillance system, aerial imagery, drone image analysis, and medical diagnosis (Ronneberger et al., 2015; Cordts et al., 2016; Zhou et al., 2018; Balloch et al., 2018; Azimi et al., 2019; Oh et al., 2020; Jung et al., 2021). Recently, much focus has been placed on developing segmentation models for streaming videos. This is a challenging task that requires not only high accuracy but also real-time inference speed (Hu et al., 2020a; Wang et al., 2021). However, majority of semantic segmentation models still rely on computationally heavy backbone networks (He et al., 2016) to achieve state-of-the-art performance. These heavy backbones, in most cases, become computational bottlenecks in the real-time video semantic segmentation pipeline.

One approach to tackle this challenge is to identify the key frames which are processed with heavy backbone networks and to propagate the pre-computed features of key frames for other frames (Zhu et al., 2017; Jain et al., 2019; Wang et al., 2020; Zhuang et al., 2020; Xu et al., 2018) in the video stream. However, this approach is based on the assumption that flow estimation and feature propagation are faster than running the backbone networks (Zhu et al., 2017). Since state-of-the-art lightweight architectures achieve real-time speed, this approach is not applicable to modern segmentation models for real-time speed up. Further, the approach sacrifices the segmentation accuracy of non-key frames even with accurate optical flow-based feature propagation, thus maintaining the accuracy of non-key frames in this pipeline still remains a problem to be addressed. Hu et al. (2020a) exploit temporal information in videos to tackle this problem via a temporally distributed network architecture. However, this strategy is suboptimal in that the sub-networks are designed independently of individual frames with no consideration of temporal locality across frames.

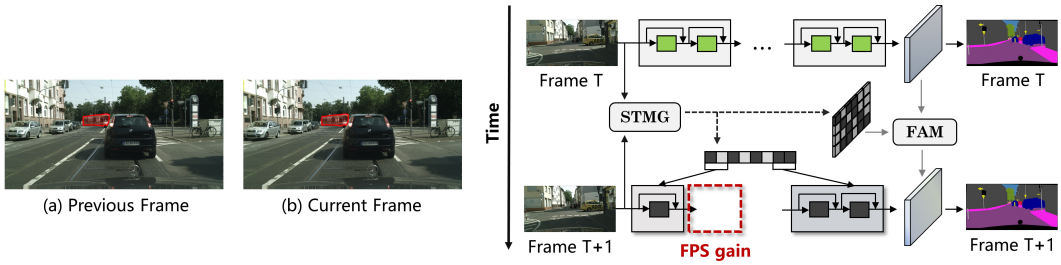


Figure 1: **Conceptual Overview.** Our proposed Spatial-Temporal Mask Generator (STMG) captures temporally evolving regions (highlighted by the red box) between two consecutive frames using spatial masks. This allows us to reuse the computed features from the previous frames for static regions. Additionally, conditioned on the distortion between the two frames, we generate a block pruning mask that determines which residual blocks to drop when running the segmentation model on the current frame. The transformed features from the previous frame are then combined with the output of the partial network computation using a feature aggregation module (FAM).

To tackle these issues, we propose a novel method to achieve practical speed-ups for real-time video semantic segmentation networks, exploiting spatial-temporal locality to create input-dependent sub-networks that preserves accuracy on both key and non-key frames at inference time. Specifically, we propose an input-dependent block-wise pruning mechanism to obtain inference time dynamic sub-networks tailored to each frame by exploiting the resiliency of residual networks to the removal of residual blocks at inference time due to their skip-connections. That is, we learn a masking mechanism that decides which residual blocks to drop and which to keep, conditioned on a given input frame and spatial-temporal information across previous frames.

While the partial computation of residual blocks produces real-time speed-ups, it still suffers from the speed-accuracy trade-off problem. To compensate for this, we further exploit the temporal locality of video frames as depicted in Figure 1. In particular, we transform the features from previous frames for reuse in the static regions across adjacent frames. We do this by generating spatial masks that capture the distortion between adjacent frames. However in most video semantic segmentation datasets such as Cityscapes (Cordts et al., 2016), not all frames are labelled hence we utilize knowledge distillation by employing a strong image segmentation model as a teacher network to train the spatial mask generator.

In addition, to avoid information loss due to repeated partial computations for sequential frames, we introduce a distortion-aware scheduling policy which determines non-key frames to apply the proposed model. Specifically, we consider frames with distortions greater than some set threshold as key frames, and invoke the full network to compute their features. Crucially, when a frame is determined to be a non-key frame, we transform the feature from the previous frame, and perform partial computation of the backbone network only in spatial regions that change from the previous frame. The resulting model, Spatial-Temporal Mask Generator (STMG) achieves a better speed-accuracy trade-offs compared to semantic and video semantic segmentation models as shown in Figure 2. Our contributions are three fold:

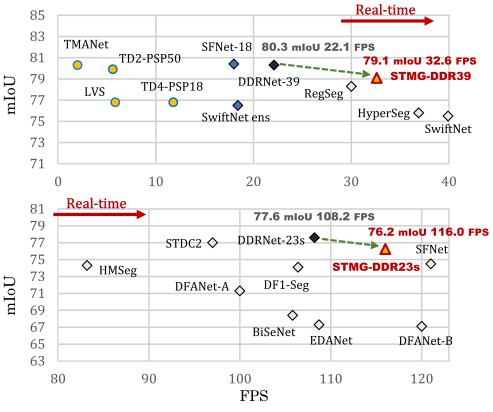


Figure 2: **Performance on Cityscapes.** Our proposed method, Spatial-Temporal Mask Generator, provides a better trade-off between mIoU and FPS.

- We propose a framework to speed up the inference of residual backbone networks for real-time video semantic segmentation by performing partial network evaluation and reusing temporally consistent features across frames.
- We propose a method to capture spatial-temporal information across frames based on inter-frame distortion without heavy optical flow computations.
- We validate our method on two benchmarks against multiple state-of-the-art real-time video segmentation models and show that our method can speed up models with residual backbones at inference time with marginal loss of accuracy.

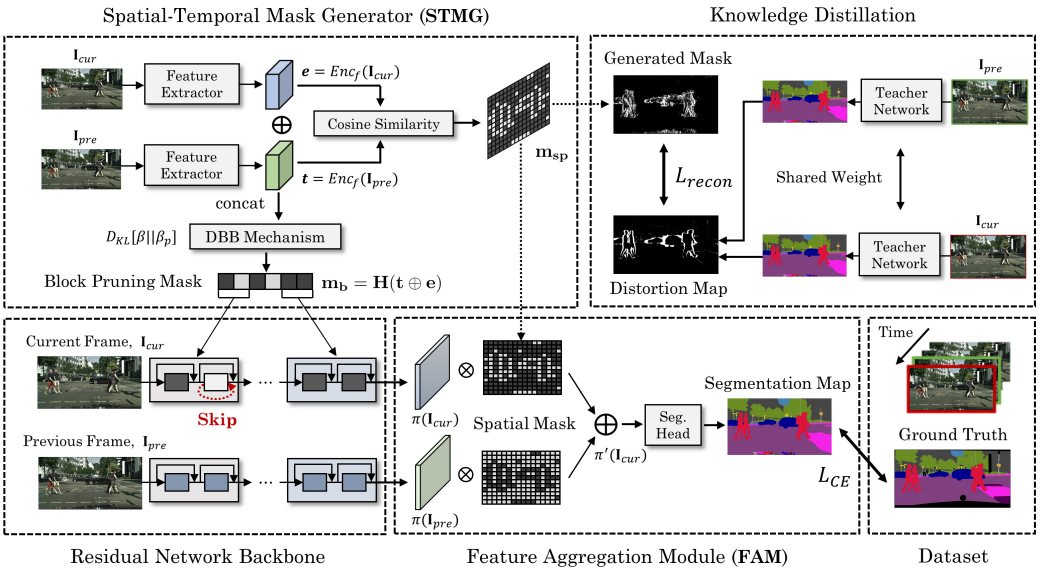


Figure 3: **The overall process for video segmentation.** Our framework exploits spatial-temporal properties by extracting features from images of adjacent frames and utilizing them to compute both spatial mask and block pruning mask. In detail, a 2D mask is generated by calculating the cosine similarity between two features and utilized as a ratio to blend the features pixel-wisely (spatial mask). These spatial-temporal properties are transferred to learn the block dropping behavior by providing the extracted features to generate the block pruning mask by concatenation (block pruning mask).

2 Related Work

Semantic Segmentation Deep convolutional neural networks have been widely used in semantic segmentation tasks by adopting a data-driven approach to improve accuracy. By providing an effective global contextual prior and fusing local and global context features, the networks can learn details and semantics simultaneously (Zhao et al., 2017). With urban scene properties where each row of the image has different statistics in terms of category distribution, each row of the feature map can be individually more focused on a specific channel using a height-driven attention map (Choi et al., 2020). Explicitly exploiting visual dependency relations, such as intra-class and global dependence among semantic entities, networks can improve generalization ability (Liu et al., 2021).

Real-time Semantic Segmentation Modern approaches to real-time semantic segmentation usually adopt a strategy of reducing the size of an input image provided to the residual backbone, as residual networks are considered to be bottlenecks in the overall computation (Zhao et al., 2018). Reducing the size of an input image generally degrades spatial resolution, which leads to a degradation of performance. Hence recent approaches adopt a strategy of designing a dual-resolution path consisting of a spatial path and a context path (Yu et al., 2018; 2021). Bilateral connections between dual resolutions are continuously repeated as the network deepens to deliver the state-of-the-art performance of the latest models with efficient information fusion (Hong et al., 2021).

Video Semantic Segmentation Video semantic segmentation approaches commonly use features from previous frames to improve accuracy or speed up inference. Zhu et al. (2017) and Li et al. (2018) introduce scheduling policies to identify key frames to apply heavy neural networks and propagate the feature across multiple frames sequentially. That is, features are reused and this reduces the computational requirements on non-key frames. Still, a fixed key frame duration is assumed in Zhu et al. (2017). Adaptive scheduling policies that select key frames based on changing video dynamics instead of heuristic policies have been proposed in Xu et al. (2018), Wang et al. (2020) and He et al. (2020). However, even if the policy is dynamic, the neural architecture that can be selected for each frame is not dynamic and still limited to two networks: fixed segmentation network and flow network. Non-dynamic characteristics of a neural network can lead to a problem of not capturing the video dynamics effectively, since video dynamics can presently only be captured by a scheduling policy.

Dynamic Network Pruning Residual networks (He et al., 2016) are composed of residual blocks and skip connections which enable them to behave like ensembles of relatively shallow networks (Veit et al., 2016). These networks have been shown to be resilient to layer dropping, usually with minimal loss of accuracy. In Wang et al. (2018) and Wu et al. (2018), this resilience to layer dropping is exploited using reinforcement learning to prevent over-parameterization in residual networks. Li et al. (2020b) exploits contextual information to develop dynamic routing methods that adapt to the scale of the input image for the semantic segmentation task. Furthermore, He et al. (2021) focuses on utilizing spatial information for channel-wise pruning of the semantic segmentation networks. However, for video semantic segmentation, such methods that exploit spatial information are insufficient since they do not consider the temporal nature of videos. Hence, to fully utilize contextual information in the video semantic segmentation task, the spatial-temporal properties have to be taken into account.

Dependent Beta-Bernoulli Process In latent feature model, the Indian buffet process (Ghahramani & Griffiths, 2005) defines a prior distribution on the binary feature indicators with potentially infinite number of features. Dependent Indian buffet process and its finite-dimensional beta-Bernoulli approximation (Williamson et al., 2010; Zhou et al., 2011; Ren et al., 2011) extend the Indian buffet process (IBP) to incorporate input covariates as follows:

$$\varphi_k \sim p(\varphi_k), \quad z_{n,k}|\varphi_k, \mathbf{x}_n \sim Ber(g(\varphi_k, \mathbf{x}_n)), \tag{1}$$

where g is an arbitrary function mapping φ_k and $\mathbf{x}_n$ to a probability. The input covariates $\mathbf{x}_n$ corresponds to the inputs of a certain layer and dropout probabilities are adjusted according to $\mathbf{x}_n$ in the dependent beta-Bernoulli dropout (Lee et al., 2018).

3 METHOD

In this section, we first introduce our dynamic block pruning mechanism based on spatial-temporal locality. We then describe a spatial-temporal mask generator for feature transformation and reuse based on the distortion between adjacent frames. The overall framework consists of two networks, the segmentation network and the mask generator. Note that the proposed method is model-agnostic and applicable to any semantic segmentation model with a residual network.

3.1 INPUT-DEPENDENT BLOCK PRUNING

In this section, we describe the input-dependent block pruning mechanism. Let $\mathbf{x}_n$ be an input and let $\mathbf{z}_n \in \{0,1\}^K$ be a binary mask vector generated for the input $\mathbf{x}_n$ where K is the number of prunable residual blocks. Note that the first block in each residual layer is not prunable. Conditioned on the input $\mathbf{x}_n$, we define an input-dependent residual block pruning probability $\boldsymbol{\varphi}(\mathbf{x}_n) \in [0,1]^K$ for $\mathbf{x}_n$ and generate a binary mask vector $\mathbf{z}_n$ for each block as follows:

$$z_{n,k} \sim Ber(\varphi_k(\mathbf{x}_n)) \text{ for } k = 1, \ldots, K. \tag{2}$$

That is, given a residual network with K prunable blocks, we model the block pruning probability of each individual block as a Bernoulli random process follows:

$$\psi(\mathbf{u}|\mathbf{x}_n) = \prod_{k=1}^{K} \varphi_k(\mathbf{x}_n)^{u_k}(1 - \varphi_k(\mathbf{x}_n))^{1-u_k}, \tag{3}$$

where $u_k \in \{0,1\}$ indicates whether to prune or keep the k-th residual block.

In this paper, we model the transformation process from images into pruning probabilities with a Bayesian approach by inducing sparsity with the batch normalization (Ioffe & Szegedy, 2015) layer of the pruning model inspired by Lee et al. (2018). To obtain a pruning mask, we impose a sparsity inducing prior on $\varphi_k(\mathbf{x}_n)$ and independently compute the block pruning probabilities as follows:

$$\varphi_k(\mathbf{x}_n, \beta_k) = \text{clamp}\left(\gamma_k \frac{g_k(\mathbf{x}_n) - \mu_k}{\sigma_k} + \beta_k, \tau\right), \tag{4}$$

where $g_k(\mathbf{x}_n)$ denotes a transformation of $\mathbf{x}_n$ using the non-linear function g for the k-th residual block, μ_k and σ_k are the estimates of the k-th component's mean and standard deviation of the transformed inputs, γ_k and β_k are scaling and shifting parameters, and $\tau > 0$ is some tolerance to prevent overflow, which is set as $1e^{-10}$ while $\text{clamp}(x, \tau) = \min(1 - \tau, \max(\tau, x))$.

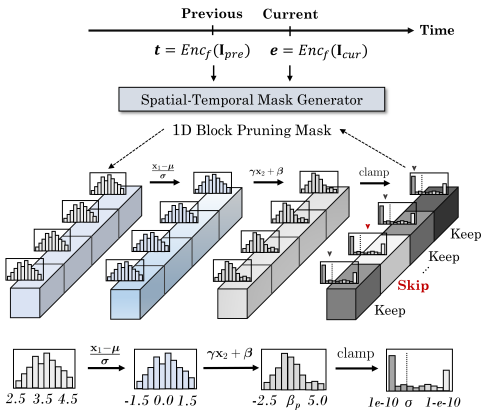


Figure 4: **Illustration of input-dependent block pruning.** We visualize how the extracted features $\mathbf{t}$ and $\mathbf{e}$ are utilized by STMG for generating the residual block pruning probabilities.

During the optimization phase, $z_{n,k}$ is sampled from the relaxed Bernoulli distribution. Specifically, by passing $\boldsymbol{\delta}_{uc} \in \mathbb{R}^K$ to a softplus function and sampling $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to provide a variability during the optimization stage, the sparsity inducing prior on $\boldsymbol{\varphi}(\mathbf{x}_n)$, which is $\mathcal{N}(\boldsymbol{\beta}_p, \rho\mathbf{I})$, and the trainable parameters of $\boldsymbol{\beta}$ are defined as follows:

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_{bn}, \boldsymbol{\delta}), \;\; \boldsymbol{\beta}_p \sim \mathcal{N}(\boldsymbol{\beta}_p, \rho\mathbf{I}), \tag{5}$$

while $\boldsymbol{\beta}_{bn}$ is obtained with batch normalization layer and $\delta_k = \epsilon_k \cdot \text{softplus}(\delta_{uc,k})$ is obtained with $\boldsymbol{\delta}_{uc}$ and $\boldsymbol{\epsilon}$ during the optimization phase.

That is, $D_{KL}[\boldsymbol{\beta} \| \boldsymbol{\beta}_p]$, the KL divergence loss for sparsity inducing prior on $\boldsymbol{\varphi}(\mathbf{x}_n)$ is defined as follows:

$$D_{KL}[\boldsymbol{\beta} \| \boldsymbol{\beta}_p] = \log \frac{\rho}{\delta} + \frac{\delta^2 + (\boldsymbol{\beta}_p - \boldsymbol{\beta}_{bn})^2}{2\rho^2}, \tag{6}$$

where ρ and $\boldsymbol{\beta}_p$ are constants for sparsity inducing prior and δ and $\boldsymbol{\beta}_{bn}$ are trainable parameters of the pruning mechanism. An illustration of the pruning method is depicted in Figure 4.

3.2 Spatial-Temporal Mask Generator

We describe the Spatial-Temporal Mask Generator (STMG) which generates two types of mask: a block pruning mask and a spatial mask, based on spatial-temporal information. An overview of the mask generator is depicted in Figure 3.

Block Pruning Mask Based on the input-dependent block pruning method previously described, we develop a block mask generator for prunable residual blocks. Specifically, the mask generator takes as input two adjacent frames, $\mathbf{I}_{pre}$ and $\mathbf{I}_{cur}$, and processes each frame using a feature extractor, Enc_f, with shared weights across all frames. We then concatenate and pass the outputs of the feature extractor to another convolution layer to generate a pruning mask with the same dimensionality as the number of residual blocks in the backbone network as follows:

$$\mathbf{t} = Enc_f(\mathbf{I}_{pre}), \;\; \mathbf{e} = Enc_f(\mathbf{I}_{cur}), \;\; \mathbf{m_b} = \mathbf{H}(\mathbf{t} \oplus \mathbf{e}), \tag{7}$$

where $\mathbf{t}$ and $\mathbf{e}$ are extracted features from the previous and current frame, $\mathbf{m_b}$ denotes a block pruning mask and $\mathbf{H}$ denotes overall transformation of the input in the pruning mechanism.

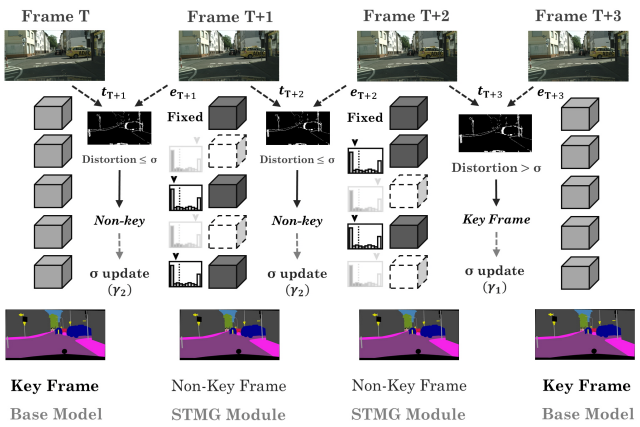
Spatial Mask In addition, to compensate for information loss due to the partial computation of residual blocks, we propose a spatial masking scheme which exploits the temporal locality of videos. Specifically, we generate a spatial mask, $\mathbf{m_{sp}}$, which predicts unchanged regions between adjacent frames together with the block pruning mask. By applying the spatial mask on the features of the previous frame, we transform the feature from the previous frame for reuse in the current frame. This is done by computing the cosine similarity between the outputs of the feature extractor from the previous frame and current frame denoted as $\mathbf{t}$ and $\mathbf{e}$ as follows:

$$m_{sp,i}(\mathbf{t}, \mathbf{e}) = -0.5 \times cos(t_i, e_i) + 0.5, \tag{8}$$

where $m_{sp,i}$ denotes a spatial mask of the i-th pixel and we scale the value to be in the range of $[0, 1]$.

Knowledge Distillation However, the cosine similarity between $\mathbf{t}$ and $\mathbf{e}$ from adjacent frames alone is not sufficient for encoding meaningful spatial-temporal information. To further improve the quality of the features, we add an auxiliary loss to match the generated spatial mask with the ground-truth distortion map. Since not all frames in the dataset are labeled, we utilize knowledge distillation (Hinton et al., 2015) to generate ground-truth distortion map as shown in Figure 6 (b). Specifically, we generate the ground-truth distortion map by subtracting the segmentation maps obtained by applying a strong image segmentation model on two adjacent frames. Following Deng et al. (2018), the loss function for spatial mask consists of the binary cross-entropy loss and dice loss, as follows:

$$L_{recon}(\mathbf{m_{sp}}, \mathbf{n}) = L_{bce}(\mathbf{m_{sp}}, \mathbf{n}) + L_{dice}(\mathbf{m_{sp}}, \mathbf{n}), \tag{9}$$



Algorithm 1 Distortion-aware scheduling

Input: Frames $\mathbf{V} = \{\mathbf{I}_0, \mathbf{I}_1, \ldots, \mathbf{I}_n\}$.
Output: Video segmentation maps $\mathbf{S} = \{\mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_n\}$.

$\mathbf{f}_0 \leftarrow$ Key Frame
$\sigma \leftarrow |\mathbf{m_{sp;1}}| =$ distortion$(\mathbf{I_1}, \mathbf{I_0})$
for i **in** range(1, n) **do**
  if $|\mathbf{m_{sp;i}}| > \sigma$ **then**
    $\mathbf{f}_i \leftarrow$ Key Frame
    $\sigma \leftarrow \gamma_{1;i} * |\mathbf{m_{sp;i}}|$
  else
    $\mathbf{f}_i \leftarrow$ Non-Key Frame
    $\sigma \leftarrow \gamma_{2;i} * |\mathbf{m_{sp;i}}|$ $(\gamma_{2;1} = 1)$
  end if
end for

Figure 5: **(Left)**: Illustration of distortion-aware key frame selection. At the arrival of a new frame, we first calculate the distortion of adjacent frames to determine key and non-key frames. **(Right)**: Pseudo-code for distortion-aware scheduling policy. For good balance between key and non-key frames, we update the threshold value scaled by a factor γ.

where $\mathbf{n}$ denotes the ground-truth distortion map obtained by knowledge distillation. The dice loss alleviates the class-imbalance problem, as only partial spatial areas are distorted between adjacent frames. The dice loss, L_{dice}, is computed as follows:

$$L_{dice} = 1 - \frac{2\sum m_{sp,i} n_i + \kappa}{\sum m_{sp,i}^2 + \sum n_i^2 + \kappa}, \tag{10}$$

where κ is a smoothing value set as 1 to avoid zero division. L_{dice} is the cornerstone for generating crisp edges while optimizing only the cross-entropy loss leads to relatively coarse results.

Feature Aggregation Using the generated spatial mask, we aggregate features of the segmentation model from the previous frame and current frame as follows:

$$\pi'(\mathbf{I}_{cur}) = \mathbf{m_{sp}} \cdot \pi(\mathbf{I}_{cur}) + (\mathbf{I} - \mathbf{m_{sp}}) \cdot \pi(\mathbf{I}_{pre}), \tag{11}$$

where $\pi(\cdot)$ is the output of the backbone segmentation network. In particular, we intend the network to focus more on features for distorted regions, while reusing features from previous frames for static regions. Specifically, we pass the aggregated feature $\pi'(\mathbf{I}_{cur})$ to the segmentation head module to obtain the semantic segmentation map for the current frame.

3.3 Distortion-Aware Scheduling Policy

While we compensate for information loss by reusing the features from previous frames, error in the features propagate as time progresses. Hence, we propose a scheduling policy that applies the proposed module only to specific non-key frames. We do this by capturing the amount of distortion using the spatial mask for each frame. Specifically, if the distortion for a given frame is greater than some set threshold, we consider it as a key frame and compute the full network on the frame. However, if the distortion is lower than the threshold, we consider the frame a non-key frame and transform the features from the previous frame for reuse as well as perform partial computation of the residual blocks using the generated input-dependent mask.

Scaling Factor After each computation, we update the threshold value with the distortion amount of the current frame scaled by a factor γ. To prevent the full network from being applied to more than one frame in a row, we assign a value of 2 to γ_1 and update the threshold value with a multiplicative factor of γ_1 right after a key frame so that the next frame can be determined as a non-key frame. Additionally, to avoid situations where partial computations are repeated over sequential frames, we assign a value of 0.95 to γ_2 right after a non-key frame and update the threshold value with the current distortion factor multiplied by γ_2 so that the next frame can be slightly induced as a key frame. Most video semantic segmentation frameworks adopt a strategy of propagating features across multiple frames sequentially, however this require a heavy optical flow computation since errors propagate as time progresses. Since our method targets real-time speed-up without optical flow computation, we adopt a strategy of frequently switching between key and non-key frames with proposed scaling rules.

Table 1: Performance comparison with **real-time (> 30 FPS)** semantic segmentation models on Cityscapes (Cordts et al., 2016).

Model	mIoU	FPS	FLOPs	Params
PP-LiteSeg-T1 (Peng et al., 2022)	73.1	**273.6**	-	-
SwiftNetRN-18 (Orsic et al., 2019)	75.4	39.9	104.0G	11.8M
SFNet-DF2 (Li et al., 2020a)	75.8	61.0	48.5G	10.5M
BiSeNetV2-L (Yu et al., 2021)	75.8	47.3	118.5G	-
HyperSeg (Nirkin et al., 2021)	76.2	36.9	7.5G	10.1M
CABiNet (Kumaar et al., 2021)	76.6	76.5	12.0G	2.6M
STDC2-Seg75 (Fan et al., 2021)	77.0	97.0	-	-
RegSeg (Gao, 2021)	78.1	30.0	39.1G	3.3M
DDRNet-39 (Hong et al., 2021)	80.3	22.1	261.8G	32.4M
- STMG-DDR39	**79.1**	32.6	175.2G	22.9M
DDRNet-23-Slim (Hong et al., 2021)	77.6	108.2	33.8G	5.7M
- STMG-DDR23-Slim	76.2	116.0	27.2G	4.9M

Table 2: Performance comparison with video semantic segmentation models on Cityscapes (Cordts et al., 2016).

Model	mIoU	FPS	Max Latency
DVSNet (Xu et al., 2018)	63.2	30.3	-
CLK (Shelhamer et al., 2016)	64.4	6.3	198 (ms)
DFF (Zhu et al., 2017)	69.2	6.4	575 (ms)
TD4-Bise18 (Hu et al., 2020a)	75.0	47.6	21 (ms)
FANet18 (Hu et al., 2020b)	75.5	72.0	14 (ms)
PEARL (Jin et al., 2017)	76.5	1.3	800 (ms)
LVS (Li et al., 2018)	76.8	5.8	380 (ms)
TD2-PSP50 (Hu et al., 2020a)	79.9	5.6	178 (ms)
TMANet50 (Wang et al., 2021)	80.3	2.0	500 (ms)
NetWarp (Gadde et al., 2017)	**80.6**	0.3	3004 (ms)
STMG-DDR39	79.1	32.6	46 (ms)
STMG-DDR23-Slim	76.2	**116.0**	**10 (ms)**

These scaling rules provide a good balance between key and non-key frames while simultaneously capturing video dynamics and leveraging spatial-temporal locality. An illustration of this scheduling policy together with pseudo-code is provided in Figure 5.

4 Experiments

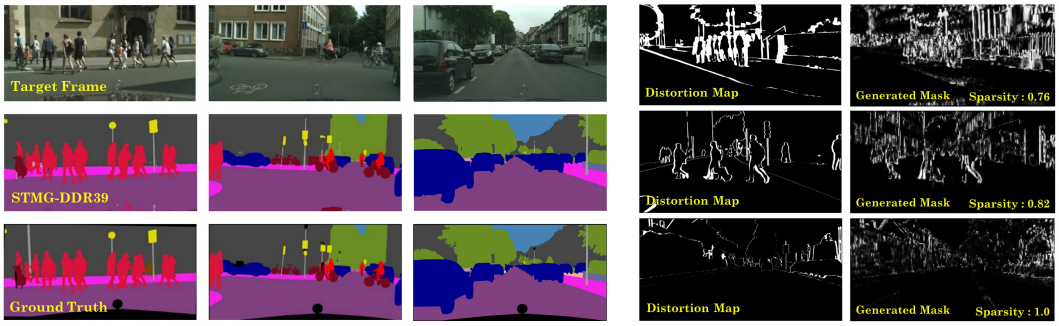
4.1 Setup and Implementation

Dataset We evaluate our method on Cityscapes (Cordts et al., 2016). Cityscapes is a dataset for urban-scene parsing consisting of 5,000 images of urban street scenes with 19 classes. The dataset contains 2,975 finely annotated images for training, 500 images for validation, and 1,525 images for testing. We evaluate our method based on the standard semantic segmentation evaluation metrics: mean class-wise intersection over union (mIoU), and Frames Per Second (FPS). Cityscapes is made up of snippets each consisting of 30 frames with the 20th frame of each snippet annotated. Hence, we train and evaluate our model on the 20th frame of each snippet with the image from the previous frame given. We also evaluate our method on CamVid (Brostow et al., 2009) which is a road-scene parsing dataset. The results for the CamVid experiments are reported in Section A of the supplementary file.

Models and Baselines We select two state-of-the-art real-time semantic segmentation models for our experiments: DDRNet-39 and DDRNet-23-Slim. Since DDRNet models have residual blocks both in the high and low resolution branches, we apply our dynamic block pruning mechanism in both branches. To preserve the output dimension of each residual layer, we do not drop the first residual block in each layer. This is because dropping the first block in each layer will not allow forward propagation of the features in residual networks. Specifically, there are a total of 6 and 17 prunable residual blocks in DDRNet-23-Slim and DDRNet-39 respectively.

Implementation To train our models, we use mini-batch stochastic gradient descent (SGD) with weight decay of 0.0005 and the momentum of 0.9. We use a batch size of 16 for training on the Cityscapes dataset. We set the initial learning rate as 0.01 and train the model for 1,000 epochs for DDRNet-23-Slim and 484 epochs for DDRNet-39. We also apply random cropping for data augmentation with crop size of 1024×1024. We implement all models and conduct all experiments using PyTorch. We train each model with two GeForce RTX 3090 GPUs for DDRNet-39 base models, and two GeForce RTX 2080 Ti GPUs for DDRNet-23-Slim base models to reproduce the results.

Inference Speed Measurement We conduct inference speed measurement on a single RTX 2080 Ti. Following the evaluation procedure of Si et al. (2019), Orsic et al. (2019), and Hong et al. (2021), we remove the batch normalization layers after each convolutional layer during the speed measurement. Specifically, we use the protocols established by Chen et al. (2019) for a fair comparison. Since not all frames are annotated in Cityscapes, we evaluate the speed of our model by sampling a portion of the entire video sequence to include both key and non-key frames in the annotated frames. Further, we report the maximum latency of our model along with the speed of our model as video semantic segmentation models have varying latency between key and non-key frames. We compare the speed of our model on two semantic segmentation backbone architectures: DDR-39 and DDR-23-Slim.



(a) Qualitative Results (b) Spatial Mask Visualization

Figure 6: **Qualitative results** of (a) our method and (b) visualization of the spatial mask. Black and white represent static and distorted regions, respectively. During the training phase, a distortion map is provided to STMG to better capture the spatial-temporal locality between adjacent frames. The degree of sparsity refers to the pruning ratio among prunable residual blocks in the backbone.

Table 3: **Ablation study** on spatial and block pruning mask. m_{sp} denotes a blending ratio of previous and current features while m_{avg} is the average value of $\mathbf{m_{sp}}$ in the corresponding STMG.

Pruning and Masking Methodology	mIoU	FLOPs	Params
Random Block Pruning (w/o *ft*)	71.40 ± 0.14	175.6 ± 0.4G	22.8M
Random Spatial Masking	78.06 ± 0.02	175.2G	22.9M
Uniform Blending ($m_{sp} = m_{avg}$)	78.69	175.2G	22.9M
Uniform Blending ($m_{sp} = 0.5$)	78.25	175.2G	22.9M
Ours (Spatial-Temporal Masking)	**79.09**	**175.2G**	22.9M

Table 4: **Ablation study** on distortion-aware scheduling policy. R denotes a key frame duration length for the fixed policy and γ denotes a threshold scaling factor for the distortion-aware policy.

Scheduling Policy	mIoU	FLOPs	Params
Distortion-Aware Scheduling ($\gamma = 0.95$)	79.09	**175.2G**	**22.9M**
Distortion-Aware Scheduling ($\gamma = 0.9$)	79.43	187.6G	24.2M
Fixed Scheduling ($R = 2$)	**79.73** ± 0.03	189.9G	24.5M

Table 5: FPS gain results. For a fair comparison, we select and compare the model and framework whose FPS of the base network is closest to the base network of ours.

Framework (Model)	FPS gain	mIoU drop
TD4-Bise18 (Bise34) (Hu et al., 2020a)	28.57%	1.32%
STMG-DDR39 (DDR39)	**47.51%**	1.47%

Table 6: **Ablation study** on FAM. For a fair comparison, we use the same sparsity inducing prior β_p and the same KL scale factor for per-frame block-wise pruning inspired by dependent beta-Bernoulli (DBB) dropout (Lee et al., 2018).

Model	mIoU	Drop	FPS	Gain
DDRNet-39	80.27	-	22.1	-
- Per-Frame Pruning	76.31	4.93%	41.7	88.7%
- **Ours (STMG-DDR39)**	**79.09**	1.47%	32.6	47.5%

4.2 Results

In Table 1, we report the experimental results on Cityscapes for the DDRNet-39 backbone network. The results show that our method reaches 32.6 FPS which is a 47.5% gain over the base network which runs at 22.1 FPS on high resolution images. The segmentation outputs are shown in Figure 6 (a) and in Section C of the supplementary file. We report similar results for the DDRNet-23-Slim architecture in Table 1. As shown in Table 1, our method reaches 116.0 FPS with 7.2% FPS gain compared to the base network. In both architectures, we increase the FPS with less than 2% drop in accuracy. That is, STMG provides a better trade-off between accuracy and speed compared to state-of-the-art models. We further visualized this speed-accuracy trade-off in Figure 2.

Degree of Distortion and Sparsity We visualize the generated spatial masks in Figure 6 (b). We observe that the mask generator successfully encodes meaningful spatial-temporal information and captures the distorted regions such as the edge movement of people or bicycles. As shown in Figure 6 (b), each example shows different degrees of sparsity and the results show a strong correlation between sparsity and degree of distortion. To further improve the encoding of spatial-temporal information, we add a distortion bias η to DBB (Lee et al., 2018)-inspired block pruning mechanism as follows:

$$\varphi_k(\mathbf{x}_n, \beta_k) = \text{clamp}\left(\gamma_k \frac{g_k(\mathbf{x}_n) + \eta - \mu_k}{\sigma_k} + \beta_k, \tau\right), \quad \eta = -\overline{m_{sp,i}} + 0.5, \tag{12}$$

where $\varphi_k(\cdot)$ denotes a block pruning probability of the k-th residual block. The resulting pruning mechanism, distrotion bias-based DBB (dbb-DBB) achieves a stronger correlation between sparsity and degree of distortion as shown in Figure 8. The comparison results are reported in Table 7.

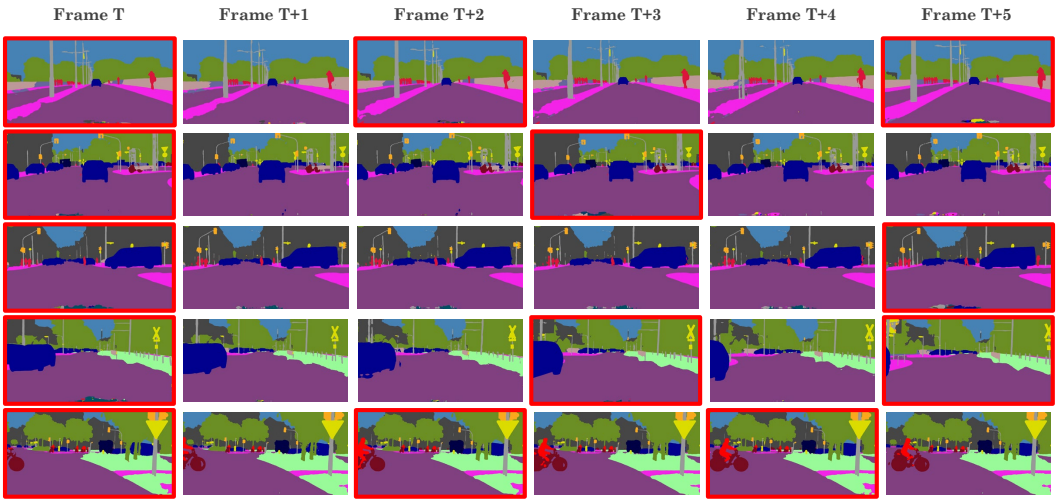


Figure 7: **Qualitative results of distortion-aware scheduling policy.** Frames highlighted in red represent key frames and other frames represent non-key frames. The third row shows that for static video streams, the scheduling policy continues to perform partial computations on non-key frames.

Distortion-Aware Scheduling Policy As shown in Figure 7, the adaptive scheduling policy successfully determines key and non-key frames by utilizing the distortion between adjacent frames. Diverse dynamic key frame paths are generated which shows that the scheduling policy does not generate a fixed configuration of key and non-key frames. For example, the 3rd row shows a repeated non-key frame as the distortion over consecutive frames are small, and the 5th row shows repeated switching between key and non-key frame when the distortion across frames becomes large.

Ablation Studies We conduct ablation studies to examine the effectiveness of each component in our framework. As described in Table 3, we conduct an ablation study on spatial masking and block pruning. The results show that our proposed spatial masking trained with knowledge distillation outperforms the model that uses random or uniform blending with features of previous frame. The masking method differs during the inference time and the identical trained weights are used during the ablation study. Further, our DBB-inspired pruning mechanism outperforms the random pruning at the same computational cost. As described in Table 4, the results show that our distortion-aware scheduling policy achieves lower FLOPs and parameters compared to the fixed scheduling policy. In Table 6, we conduct an ablation study on our proposed feature aggregation module (FAM), and we observe that the information from the previous frame can compensate for the information loss due to the dropped blocks in the backbone for the current frame.

Table 7: Performance on Cityscapes over the DDRNet-39. "+dbb-DBB" represents a pruning mechanism using the proposed distortion bias-based DBB (dbb-DBB). STMG uses a default DBB-inspired pruning.

Model	mIoU	FLOPs	Params
STMG	79.1	**175.2G**	**22.9M**
STMG + dbb-DBB	**79.3**	188.7G	24.8M

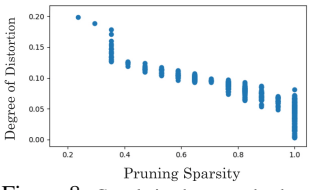


Figure 8: Correlation between the degree of distortion estimated by the spatial mask generator and pruning sparsity in dbb-DBB.

5 Conclusion

In this work, we proposed an efficient framework for real-time video semantic segmentation that exploits the spatial-temporal locality of videos. Our framework reuses the features from previous frames, and perform partial evaluation of the backbone network by taking into account the differences between two consecutive frames. This partial network evaluation is done using an input-dependent gating mechanism that decides which features and blocks to prune. Our framework is model-agnostic and can be applied to any semantic segmentation models with residual backbone networks to speed up their inference. We validated our method on two benchmark datasets with different semantic segmentation models and showed that it significantly improves inference-time speed at the expense of marginal drop in accuracy compared to baselines approaches that achieve lower inference speed-ups with similar decreases in model accuracy.

REPRODUCIBILITY STATEMENT

We provide the source code via an anonymous URL to verify the reproducibility of our work in Section B of the supplementary file. For experimental setup, we include all details in Section 4 of the main paper, and Section A and Section F of the supplementary file.

REFERENCES

Seyed Majid Azimi, Corentin Henry, Lars Sommer, Arne Schumann, and Eleonora Vig. Skyscapes fine-grained semantic understanding of aerial scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7393–7403, 2019.

Jonathan C Balloch, Varun Agrawal, Irfan Essa, and Sonia Chernova. Unbiasing semantic segmentation for robot perception using synthetic data feature transfer. *arXiv preprint arXiv:1809.03676*, 2018.

Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

Wuyang Chen, Xinyu Gong, Xianming Liu, Qian Zhang, Yuan Li, and Zhangyang Wang. Fasterseg: Searching for faster real-time semantic segmentation. *arXiv preprint arXiv:1912.10917*, 2019.

Sungha Choi, Joanne T Kim, and Jaegul Choo. Cars can't fly up in the sky: Improving urban-scene segmentation via height-driven attention networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9373–9383, 2020.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to predict crisp boundaries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 562–578, 2018.

Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9716–9725, 2021.

Raghudeep Gadde, Varun Jampani, and Peter V Gehler. Semantic video cnns through representation warping. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4453–4462, 2017.

Roland Gao. Rethink dilated convolution for real-time semantic segmentation. *arXiv preprint arXiv:2111.09957*, 2021.

Zoubin Ghahramani and Thomas Griffiths. Infinite latent feature models and the indian buffet process. *Advances in neural information processing systems*, 18, 2005.

Fei He, Naiyu Gao, Qiaozhe Li, Senyao Du, Xin Zhao, and Kaiqi Huang. Temporal context enhanced feature aggregation for video object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10941–10948, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Wei He, Meiqing Wu, Mingfu Liang, and Siew-Kei Lam. Cap: Context-aware pruning for semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 960–969, 2021.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Yuanduo Hong, Huihui Pan, Weichao Sun, Yisong Jia, et al. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv preprint arXiv:2101.06085*, 2021.

Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8818–8827, 2020a.

Ping Hu, Federico Perazzi, Fabian Caba Heilbron, Oliver Wang, Zhe Lin, Kate Saenko, and Stan Sclaroff. Real-time semantic segmentation with fast attention. *IEEE Robotics and Automation Letters*, 6(1):263–270, 2020b.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Samvit Jain, Xin Wang, and Joseph E Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8866–8875, 2019.

Xiaojie Jin, Xin Li, Huaxin Xiao, Xiaohui Shen, Zhe Lin, Jimei Yang, Yunpeng Chen, Jian Dong, Luoqi Liu, Zequn Jie, et al. Video scene parsing with predictive feature learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5580–5588, 2017.

Sanghun Jung, Jungsoo Lee, Daehoon Gwak, Sungha Choi, and Jaegul Choo. Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15425–15434, 2021.

Saumya Kumaar, Ye Lyu, Francesco Nex, and Michael Ying Yang. Cabinet: Efficient context aggregation network for low-latency semantic segmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13517–13524. IEEE, 2021.

Juho Lee, Saehoon Kim, Jaehong Yoon, Hae Beom Lee, Eunho Yang, and Sung Ju Hwang. Adaptive network sparsification with dependent variational beta-bernoulli dropout. *arXiv preprint arXiv:1805.10896*, 2018.

Xiangtai Li, Ansheng You, Zhen Zhu, Houlong Zhao, Maoke Yang, Kuiyuan Yang, Shaohua Tan, and Yunhai Tong. Semantic flow for fast and accurate scene parsing. In *European Conference on Computer Vision*, pp. 775–793. Springer, 2020a.

Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8553–8562, 2020b.

Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5997–6005, 2018.

Peiwen Lin, Peng Sun, Guangliang Cheng, Sirui Xie, Xi Li, and Jianping Shi. Graph-guided architecture search for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4203–4212, 2020.

Mingyuan Liu, Dan Schonfeld, and Wei Tang. Exploit visual dependency relations for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9726–9735, 2021.

Yuval Nirkin, Lior Wolf, and Tal Hassner. Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4061–4070, 2021.

Dokwan Oh, Daehyun Ji, Cheolhun Jang, Yoonsuk Hyun, Hong S Bae, and Sungju Hwang. Segmenting 2k-videos at 36.5 fps with 24.3 gflops: Accurate and lightweight realtime semantic segmentation network. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3153–3160. IEEE, 2020.

Marin Orsic, Ivan Kreso, Petra Bevandic, and Sinisa Segvic. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12607–12616, 2019.

Juncai Peng, Yi Liu, Shiyu Tang, Yuying Hao, Lutao Chu, Guowei Chen, Zewu Wu, Zeyu Chen, Zhiliang Yu, Yuning Du, et al. Pp-liteseg: A superior real-time semantic segmentation model. *arXiv preprint arXiv:2204.02681*, 2022.

Lu Ren, Yingjian Wang, Lawrence Carin, and David Dunson. The kernel beta process. *Advances in Neural Information Processing Systems*, 24:963–971, 2011.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *European Conference on Computer Vision*, pp. 852–868. Springer, 2016.

Haiyang Si, Zhiqiang Zhang, Feifan Lv, Gang Yu, and Feng Lu. Real-time semantic segmentation via multiply spatial fusion network. *arXiv preprint arXiv:1911.07217*, 2019.

Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. *Advances in neural information processing systems*, 29:550–558, 2016.

Hao Wang, Weining Wang, and Jing Liu. Temporal memory attention for video semantic segmentation. *arXiv preprint arXiv:2102.08643*, 2021.

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 409–424, 2018.

Yujiang Wang, Mingzhi Dong, Jie Shen, Yang Wu, Shiyang Cheng, and Maja Pantic. Dynamic face video segmentation via reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6959–6969, 2020.

Sinead Williamson, Peter Orbanz, and Zoubin Ghahramani. Dependent indian buffet processes. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 924–931. JMLR Workshop and Conference Proceedings, 2010.

Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8817–8826, 2018.

Yu-Syuan Xu, Tsu-Jui Fu, Hsuan-Kung Yang, and Chun-Yi Lee. Dynamic video segmentation network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6556–6565, 2018.

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 325–341, 2018.

Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129(11):3051–3068, 2021.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 405–420, 2018.

Mingyuan Zhou, Hongxia Yang, Guillermo Sapiro, David Dunson, and Lawrence Carin. Dependent hierarchical beta process for image interpolation and denoising. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 883–891. JMLR Workshop and Conference Proceedings, 2011.

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pp. 3–11. Springer, 2018.

Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2349–2358, 2017.

Jiafan Zhuang, Zilei Wang, and Bingke Wang. Video semantic segmentation with distortion-aware feature correction. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(8): 3128–3139, 2020.

APPENDIX

A EXPERIMENTS ON CAMVID

A.1 CAMVID DATASET

CamVid (Brostow et al., 2009) is a road-scene parsing dataset which is taken from the point of view of a driving automobile. The dataset consists of 701 annotated images with 11 classes at 1 Hz and in part 15Hz, in which 367, 101, and 233 images for training, validation, and testing, respectively.

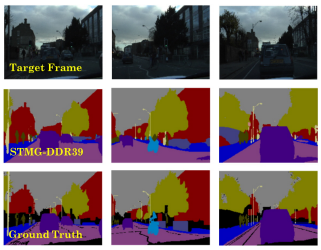


Figure 9: **Qualitative results on CamVid (Brostow et al., 2009).**

Table 8: Performance comparison with real-time semantic segmentation models on CamVid (Brostow et al., 2009).

Model	mIoU	FPS
BiSeNetV2 (Yu et al., 2021)	72.4	124.5
GAS (Lin et al., 2020)	72.8	153.1
PP-LiteSeg-T (Peng et al., 2022)	73.3	**222.3**
SFNet-RN-18 (Li et al., 2020a)	73.8	35.5
STDC2-Seg (Fan et al., 2021)	73.9	152.2
MSFNet (Si et al., 2019)	75.4	91.0
HyperSeg (Nirkin et al., 2021)	**78.4**	38.0
DDRNet-39 (Hong et al., 2021)	76.7	59.1
- STMG-DDR39	75.5	70.2

Table 9: Performance comparison with video segmentation models on CamVid (Brostow et al., 2009).

Model	mIoU	FPS
DFF (Zhu et al., 2017)	66.0	9.8
ACCEL-18 (Jain et al., 2019)	66.7	5.9
NetWarp (Gadde et al., 2017)	67.1	2.8
ACCEL-50 (Jain et al., 2019)	67.7	4.2
FANet-18 (Hu et al., 2020b)	69.0	**154**
FANet-34 (Hu et al., 2020b)	70.1	121
TD4-PSP18 (Hu et al., 2020a)	72.6	25.0
TD2-PSP50 (Hu et al., 2020a)	76.0	11.1
STMG-DDR39	75.5	70.2

A.2 SPATIAL MASKING ON CAMVID

We first learn the spatial masks using knowledge distillation as described in the main paper. However, we found that it was difficult to learn spatial masks since the amount of distortion between adjacent frames is too large in CamVid due to the low frame rate. Thus, we utilize a fixed spatial mask value to aggregate the features from previous and current frames. Specifically, the feature aggregation is computed as follows:

$$\pi'(\mathbf{I}_{cur}) = m_{sp} \cdot \pi(\mathbf{I}_{cur}) + (1 - m_{sp}) \cdot \pi(\mathbf{I}_{pre}), \tag{13}$$

where, m_{sp} is fixed to 0.8. We found that setting a higher m_{sp} is crucial to compensate for information loss due to the low frame rate by using more features from current frame than in the previous frame.

A.3 IMPLEMENTATION DETAIL

We set the initial learning rate to 0.001 and train the model for 968 epochs for DDRNet-39 and corresponding STMG module. We use a batch size of 4 with a cropping resolution of 720×960 for CamVid and conduct all experiments using PyTorch. For a fair comparison, we apply the same training setting to the corresponding STMG module. We train each model with two GeForce RTX 2080 Ti GPUs for DDRNet-39 and corresponding STMG module in the same setting. The sparsity inducing prior is set to 0.2 in the STMG module for the CamVid dataset while the KL scale factor is used in the CamVid dataset with the same value as Cityscapes. The distortion-aware scheduling policy is applied to the CamVid dataset with a gamma of 0.95, the same as the Cityscapes dataset.

A.4 EXPERIMENTAL RESULTS

We further report the experimental results on CamVid (Brostow et al., 2009) in Table 8, using DDRNet-39 backbone network (Hong et al., 2021). The results show that our method can achieve 70.2 FPS, achieving 18.8% gain over the base model. We increase the FPS with less than 2% drop in accuracy. That is, STMG provides a good trade-off comparable to state-of-the-art models on CamVid.

B CODE AND DATASET

Code and dataset are available at `https://anonymous.4open.science/r/iclrstmg`.

C ADDITIONAL RESULTS ON CITYSCAPES

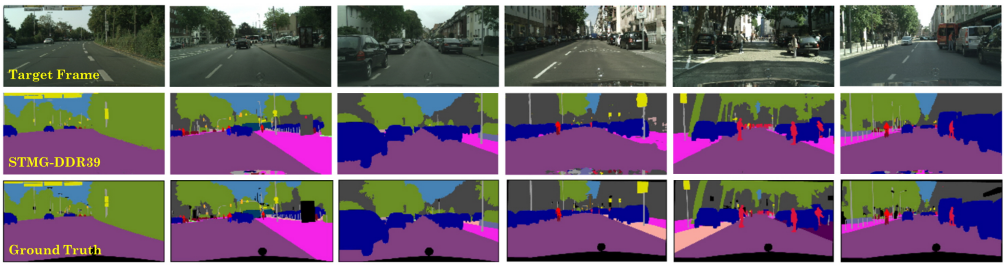


Figure 10: **Qualitative results on Cityscapes.** Visualization examples of our STMG. First row: Target Frame. Second row: STMG-DDR39. Third row: Ground Truth. Zoom in for best view.

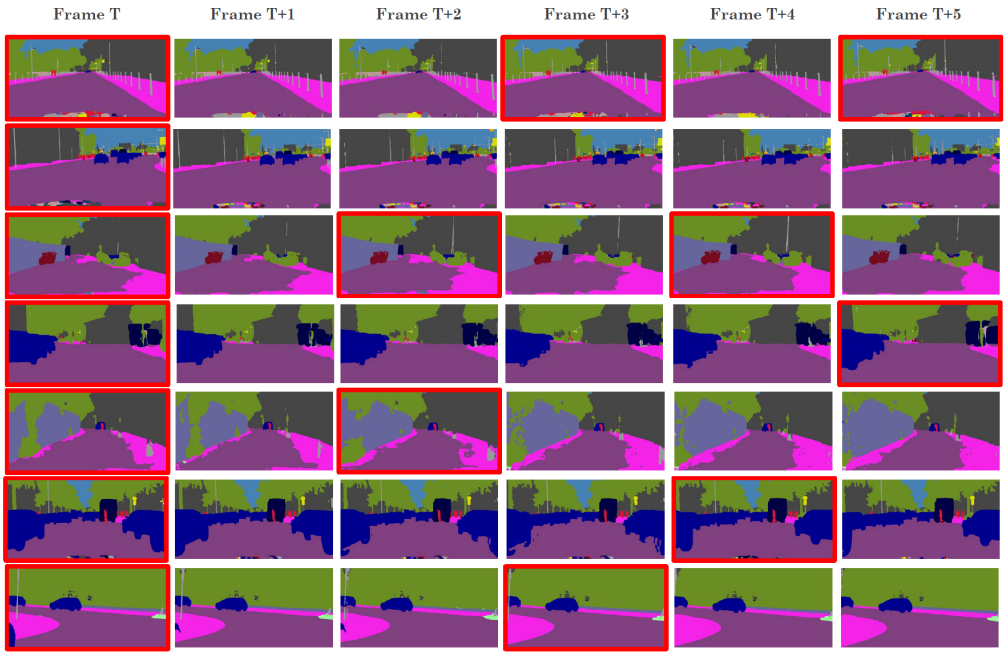


Figure 11: **Qualitative results of distortion-aware scheduling policy.** Frames highlighted in red represent key frames and other frames represent non-key frames. The fourth row shows that for static video streams, the scheduling policy continues to perform partial computations on non-key frames.

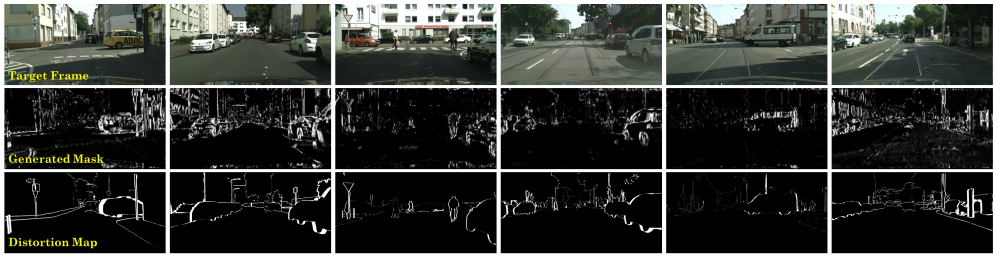


Figure 12: **Qualitative results of generated spatial mask visualization on Cityscapes.** First row: Target Frame. Second row: Generated Mask. Third row: Distortion Map. Zoom in for best view.

D ADDITIONAL RESULTS ON CAMVID

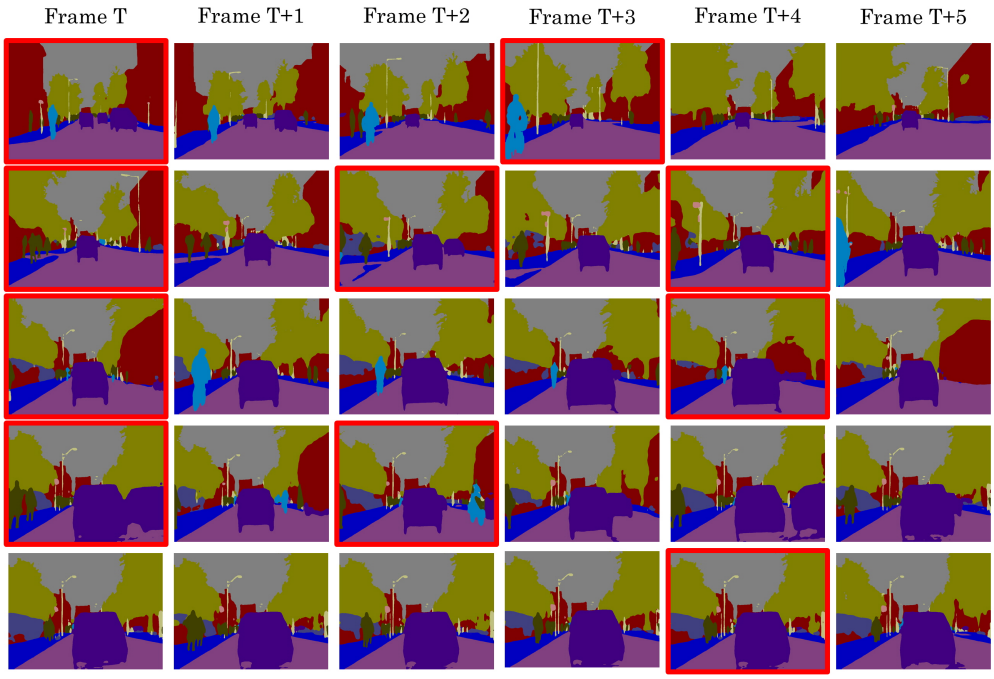


Figure 13: **Qualitative results of distortion-aware scheduling policy on CamVid.** Frames highlighted in red represent key frames and other frames represent non-key frames. The fifth row shows that the scheduling policy continues to perform partial computations for static video streams.

E DISCUSSION

In this section, we discuss the limitation and potential negative societal impacts of our work.

Potential Societal Impacts A potential ethical threat is that our proposed method could be exploited with malicious purpose to harm a society such as terrorism with limited computing resources (e.g. drones for terrorism). Further, our method could be exploited in mobile applications that can generate antisocial content, including real-time portrait matting for unethical deepfake videos. We hope that our proposed method would not be utilized for such malicious purposes to harm the society.

Limitation In this work, we propose a novel framework for real-time speed up of video semantic segmentation models by exploiting the spatial-temporal locality of video. However, since our method targets real-time speed up and state-of-the-art real-time segmentation models are much faster than optical flow estimation, our method does not utilize flow-based feature propagation. Due to these properties, our method adopts a scheduling policy with a relatively short duration between key and non-key frames compared to previous video segmentation frameworks, as errors propagate when non-key frames are repeated. Further, since DBB (Lee et al., 2018)-inspired pruning method optimizes the segmentation network while learning the pruning mask, our method uses different weights for key and non-key frames. This results in memory inefficiency both in inference time and training time, as two networks have to be loaded on the GPU. Because training a segmentation network requires a large memory, training is performed only on two adjacent frames: a fixed key frame for the previous frame and a trainable non-key frame for the current frame. We leave this limitation as future work.

F HYPERPARAMETER SETTING

Table 10: Hyperparameter setting of STMG-DDR39 on Cityscapes (Cordts et al., 2016).

Hyperparameter	Value
Image Size	1024 x 1024
Base Size	2048
Batch Size	16
OHEM Threshold	0.9
OHEM Keep	131072
Balance Weights	0.4
Learning Rate	0.01
Weight Decay	0.0005
Momentum	0.9
KL-Divergence Scaling Factor	16
Beta Prior Mean	1
Beta Prior Variance	$\sqrt{5}$
Delta UC Initial Value	0.5413
Training Epoch	484

Table 11: Hyperparameter setting of STMG-DDR39 on CamVid (Brostow et al., 2009).

Hyperparameter	Value
Image Size	960 x 720
Base Size	960
Batch Size	4
OHEM Threshold	0.9
OHEM Keep	131072
Balance Weights	0.4
Learning Rate	0.001
Weight Decay	0.0005
Momentum	0.9
KL-Divergence Scaling Factor	16
Beta Prior Mean	0.2
Beta Prior Variance	$\sqrt{5}$
Delta UC Initial Value	0.5413
Training Epoch	968

G DISTRIBUTION OF PRUNING MASK

Table 12: Distribution of binary pruning mask of STMG-DDR39 on Cityscapes (Cordts et al., 2016) during the test phase.

Residual Block Pruning Mask	Percentage
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1	30.6
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1	29.4
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1	16.6
1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1	8.6
1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1	6.4
1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1	4.8
1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1	2.2
1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1	0.4

Each binary value refers to keep or skip the corresponding residual block, and it can be seen that the dynamic inference paths are generated with various distributions during testing. Note that there are 17 prunable residual blocks in DDR39 since the first block in each residual layer is not prunable. This is because pruning the first block in each layer will not allow preservation of the output dimension of each layer. The mean $\boldsymbol{\beta}_p$ for sparsity inducing prior $\mathcal{N}(\boldsymbol{\beta}_p, \rho\mathbf{I})$ is set to 1 as a hyperparameter.