Momentum Adapt: Robust Unsupervised Adaptation for Improving Temporal Consistency in Video Semantic Segmentation During Test-Time

Amirhossein Hassankhani[1]
amirhossein.hassankhani@tuni.fi

Hamed Rezazadegan Tavakoli[2]
hamed.rezazadegan_tavakoli@nokia.com

Esa Rahtu[1]
esa.rahtu@tuni.fi

[1] Tampere University
Tampere, Finland

[2] Nokia Technology
Espoo, Finland

Abstract

Generating temporally consistent outputs in video semantic segmentation is critical, especially in sensitive applications like self-driving cars. Most approaches attempt to solve the temporal inconsistency issue by using optical flow networks or altering the architecture of the network to extract relevant information from multiple input frames. This paper presents Momentum Adapt, an unsupervised online method for improving the temporal consistency in video semantic segmentation. The method uses two semantic segmentation networks with identical architecture and tries to increase the model's confidence by taking their predictions as ground truth. The first network (AuxNet) is updated by backpropagation, while the weights in the second network (MainNet) are the exponential moving average of the weights from the first network. Our extensive quantitative evaluation shows that our approach significantly improves the performance of the network without adaptation. It also outperforms the state-of-the-art algorithm, especially in more severe conditions, including domain shift and noise. These evaluations are performed on three datasets, Cityscapes, KITTI, and SceneNet RGB-D, with many state-of-the-art semantic networks used as the base network for the adaptation algorithms.

1 Introduction

Over the years, significant improvement has been made in image semantic segmentation. However, video semantic segmentation is a more complicated task. One additional complexity of video semantic segmentation is outputting consistent predictions over time. Therefore, the typical approach of applying an image-based model to video frames leads to temporally inconsistent predictions.

Many methods try to address the problem of temporal inconsistency in video semantic segmentation. Several studies [16, 23] use optical flow to make more consistent predictions in consecutive frames. However, optical flow estimation is computationally expensive and can be inaccurate, contributing to worse predictions over time. Other approaches attempt to change the model design to incorporate processing multiple frames inside the architecture, indirectly making more consistent predictions by including information from multiple frames, such as studies [11, 25].

In a more recent study [30], the authors present AuxAdapt, an unsupervised test-time adaptation technique for temporally consistent video semantic segmentation. The main idea behind AuxAdapt is for the network to learn from its previous decisions. AuxAdapt processes the frames one by one without additional computation of multiple frames. Furthermore, it employs an additional network to stabilize the process. During the adaptation process, one of the networks (MainNet) is frozen, while the other network (AuxNet), usually a smaller one, is trained via backpropagation. By keeping MainNet frozen, the performance of the model is highly dependent on the performance of AuxNet, which could lead to instability.

We hypothesize while the frozen network in AuxAdapt is a stabilizing factor, it limits the performance of the model. Therefore, We propose Momentum Adapt, a novel unsupervised online adaptation for semantic segmentation models. Momentum Adapt, which is shown in the middle of figure 1, addresses the problem of instability and limited performance by incorporating a momentum network instead of a frozen Network. In Momentum Adapt, the active network (AuxNet) is trained with backpropagation, while the momentum network (MainNet) is updated in each iteration using the weights of AuxNet. More specifically, the MainNet's weights are the exponential moving average (EMA) of AuxNet's weights. We hypothesize that by having the momentum network, the model will more easily adapt to the changes in the environment during the test time.

The momentum network concept has shown great success in other fields. For example, Momentum Contrast (MoCo) [9] keeps a momentum encoder as a dictionary for contrastive learning. Similarly, a momentum network is used for cross-domain object detection [28]. In this work, we utilize a momentum network for online adaptation in video semantic segmentation to generate temporally consistent segmentations.

The main contribution of this work includes:

- We present Momentum Adapt, a robust unsupervised online adaptation technique for improving the temporal consistency of semantic segmentation models. Additionally, Momentum Adapt employs a new loss function that works well with momentum updates.

- We do comprehensive testing to compare the effectiveness and robustness of our methods compared to the best current adaptation technique. The experiments are conducted in standard, and more severe conditions, including noise, domain shift, and lower adaptation frequency.

- We combine our approach with AuxAdapt to take advantage of having a smaller network, leading to a less expensive backward pass during adaptation for computation-sensitive tasks.

2 Related Work

For many years, semantic segmentation networks based on convolutional networks were the standard methods for single-image semantic segmentation, e.g., PSPNet[32], HRNet[26],

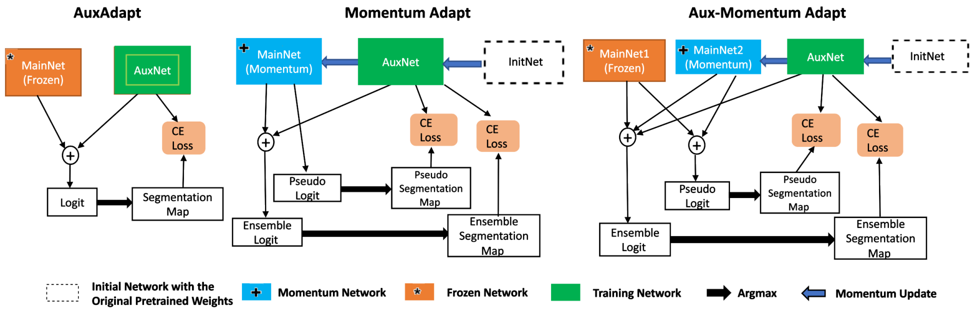

Figure 1: Conceptual Comparison of three adaptation algorithms for improving temporal consistency: Here, we show how each algorithm processes video frames individually to improve temporal consistency. **AuxAdapt**: It employs two pre-trained networks, one frozen network (MainNet), and one active network, AuxNet, shown by two borders, representing the possibility of different sizes. AuxNet is trained using the outputs from MainNet and AuxNet to increase the confidence of the model. **Momentum Adapt**: we adopt a similar approach to AuxAdapt. However, a momentum network (MainNet) with identical architecture is used instead of a frozen network. Additionally, Momentum Adapt utilizes a new weighted loss function for training AuxNet, before using momentum updates for Auxnet with the initial weights of the pre-trained network. **Aux-Momentum Adapt**: We merge AuxAdapt and Momentum Adapt to take advantage of the lower computation of a smaller active network.

DeepLab[3], and U-NET[21]. However, transformers have entirely changed the field in terms of performance and efficiency, e.g., ViT-Adapter[4], SegFormer[27], and SwinV2[14].

Despite significant advances in single-image semantic segmentation, video semantic segmentation remains to be a less explored task. Using image-based methods for videos is the most convenient approach to video semantic segmentation. However, in safety-sensitive tasks, the stability and reliability of the prediction are essential. Therefore, many studies propose approaches that directly or indirectly address the temporal inconsistency issue.

The problem of temporal inconsistency is not limited to semantic segmentation. Many studies try to improve temporal consistency in different tasks, e.g., depth estimation [12, 29], object tracking [5], and style transfer [2]. However, the approaches used for these tasks are very similar to video semantic segmentation methods. These methods often modify an image-based model to have more consistent outputs and can be categorized into two sets of approaches: modifications to model design and post-training modifications.

2.1 Modifications to Model Design

Most of these approaches try to make structural changes in the architecture of the network to improve temporal consistency. This can be done by sophisticatedly extracting relevant information from multiple frames [11, 25], incorporating an optical flow network in the architecture[16, 23], incorporating recurrent neural networks (RNN) [19, 20], or a combination of these methods[18]. The loss function is the other component of the model that can be modified. For example, [24] defines a loss function using optical flow to enforce temporal consistency between consecutive frames during training.

2.2 Post-Training Modifications

Some of the methods in this category involve fine-tuning the model using regularization loss functions based on the optical flow [13] or cross-correlation of consecutive frames [31]. Others attempt to improve the temporal consistency during test time. For example, Deep Video Prio (DVP) [10] improves the temporal consistency of a single video for lower-level processes, including image colorization and style transfer. DVP passes through a single video multiple times, making it unsuitable for online implementation. AuxAdapt [30] addresses some of the limitations of DVP and is shown to be compatible with other methods of improving temporal consistency.

3 Method

Let X_t be a RGB frame, $X_t \in [0,1]^{H \times W \times 3}$, from a video , $\{X_t\}_{t=1}^{T}$ where T is the number of frames. Given a pre-trained semantic segmentation network, f^{main}, the prediction map for each frame, $Y_t^{main} \in \mathbb{R}^{H \times W \times K}$), K denoting number of classes, is obtained by passing X_t through f^{Main}, $Y_t^{Main} = f^{Main}(X_t)$. For a given location in the image (i,j) and class k, $Y_t^{Main}(i,j,k)$ represents the likelihood of pixel (i,j) belonging to class k. In the case of no adaptation, for the final decision of semantic segmentation, argmax is applied to the last dimension of Y_t^{Main} to get Y_t^{Seg}. This will lead to inconsistencies in the predictions of video frames over time [30].

3.1 Momentum Adapt

In Momentum Adapt, we propose a new algorithm for improving temporal consistency. Momentum Adapt has two networks with identical architecture, f^{Aux} and f^{Main}. Similar to AuxAdapt, the final decision for semantic segmentation of the frame X_t is obtained by adding the output of both networks and passing it through an argmax operation as shown by equation 1:

$$Y_t^{seg}(i,j) = argmax(f^{Aux}(X_t(i,j)) + f^{Main}(X_t(i,j))). \tag{1}$$

θ^{Aux} and θ^{Main} denote the parameters of f^{Aux} and f^{Main}, respectively. We update θ^{Main} at time t as follow:

$$\theta_t^{Main} = \alpha.\theta_{t-1}^{Main} + (1-\alpha).\theta_{t-1}^{Aux} \tag{2}$$

where $\alpha \in [0,1]$ is a a momentum coefficient for MainNet. The parameters of f^{Aux}, θ^{Aux} are updated by gradient descent as follows:

$$\theta_t^{Aux} = \theta_{t-1}^{Aux} - \gamma.\nabla_{\theta_{Aux}} \mathcal{L}_{total} \tag{3}$$

where γ denotes the learning rate. $\mathcal{L}_{total}$ is the weighted sum of two cross-entropy [1] losses .

$$\mathcal{L}_{total} = \lambda_1.\mathcal{L}_{CE}(f^{Aux}(X_t), Y^{seg}) + \lambda_2.\mathcal{L}_{CE}(f^{Aux}(X_t), argmax(f^{Main}(x_t))) \tag{4}$$

where λ_1 and λ_2 representing the importance of each loss. The second loss term is introduced to increase the stability of the algorithm since it has $f^{Aux}(X_t)$ only in its input. In each iteration, after training f^{Aux} by backpropagation, the weight of f^{Aux}, θ_t^{Aux} are updated by momentum rule, similar to equation 2, with the weights from the original network at timestep zero. The update is as follows:

$$\theta_t^{Aux} = \beta.\theta_{t-1}^{Aux} + (1-\beta).\theta_0^{Aux} \tag{5}$$

where β represents the momentum coefficient for AuxNet, and θ_0^{Aux} is the original weights of Auxnet before the adaptation.

3.2 Aux-Momentum Adapt

AuxAdapt utilizes a smaller network for backpropagation training. However, in Momentum Adapt, both networks must have the same architecture, which means more relative computation. We combine our approach with AuxAdapt to benefit from the lower computations of a smaller network. More specifically, Aux-Momentum Adapt, illustrated on the right side of figure 1, uses three networks for the adaptation process, one frozen network (MainNet1), and two identical smaller networks, MainNet2 and AuxNet. The final output of the model is generated by adding the output of all three networks and passing it through an argmax operation as follows:

$$Y_t^{seg}(i,j) = argmax(f^{Main1}(X_t(i,j)) + f^{Main2}(X_t(i,j)) + f^{Aux}(X_t(i,j)). \qquad (6)$$

The wights of MainNet2, θ^{Main2}, are updated according to equation 2. Similarly, AuxNet is trained by backpropagation according to equation 3. However, $\mathcal{L}_{total}$ is computed differently since three networks are involved. $\mathcal{L}_{total}$ is the total loss function:

$$\mathcal{L}_{total} = \lambda_1.\mathcal{L}_{CE}(f^{Aux}(X_t), Y^{seg}) + \lambda_2.\mathcal{L}_{CE}(f^{Aux}(X_t), argmax(f^{Main1}(x_t) + f^{Main2}(x_t))) \quad (7)$$

where λ_1 and λ_2 refer to the weight of the loss terms. After the backpropagation, similar to Momentum Adapt, AuxNet's weights, θ^{Aux} are updated according to equation 5.

4 Experiments

In this section, we describe the evaluation metrics that we used for comparing our method to AuxAdapt. Furthermore, We discuss the datasets and the networks that were used in different experiments and dive share the result of our experiments. These experiments include results showing the effect of noise, lower adaptation frequency, and domain shift.

4.1 Evaluation Metrics

For evaluating the accuracy of the semantic segmentation, Jaccard index [7], otherwise known as mean intersection-over-union (mIoU).

For measuring the consistency of prediction of the network for two consecutive frames, we used Temporal consistency (TC) metric introduced by [22]. While the two metrics, TC and mIoU, as shown by [22] are correlated, both are needed for a holistic evaluation of each method's performance. For example, in some of the instances in the result section 4.3, although two models have very close mIoUs, they have different TCs.

4.2 Setup

Similar to AuxAdapt, Cityscapes [6] and KITTI [8] datasets were used. For the Cityscapes dataset, the validation set, which includes 500 videos each with 30 frames of the size 1024 ×2048 is used. The KITTI dataset is used for the domain shift experiments. For this experiment, 61 raw KITTI videos from city, residential, and road categories are selected. Each video is cropped at 300 frames and resized to 384×1280.

The number of frames with ground truth labels for the Cityscapes and KITTI datasets is limited, each only having 500 and 200 frames with annotations, respectively. For addressing the limited labels issue, we used an additional simulated dataset, SceneNet RGB-D [17]. In our experiments, 60 videos from the validation split of the SceneNet RGB-D dataset were randomly sampled, each video with 300 frames of the size 320x240.

The networks used for the experiments are the state-of-the-art image-based models, such as FCN [15], HRNetV2 [26], DeepLabV3Plus [3], PSPNet [32], ANN [33], SegFormer [27]. For HRNetV2, three different versions, HRNetV2-W18-Small, HRNetV2-W18, and HRNetV2-48, are utilized in the experiments. The networks used for Cityscapes and SceneNet RGB-D experiments are trained for 40,000 and 80,000 iterations, respectively. Then, these trained networks are employed in the adaptation algorithms. Since the networks in Momentum Adapt must have the same architecture, identical networks are also used in AuxAdapt to keep the relative computation comparable. Although, this is not the case for Aux-Momentum Adapt experiments, and a smaller network, similar to the original AuxAdapt design, is utilized for AuxNet in AuxAdapt and Aux-Momentum.

There are some very important hyperparameters in our algorithms that can completely change the results of our methods. We use the typical learning rate in the fine-tuning task, 0.0001. For momentum coefficients, α is set to 0.0001 and β is set to 0.1. For weights of the loss terms, λ_1 and λ_2 are set to 15 and 1, respectively.

4.3 Results

In the following section, we show that both of our methods significantly improve the performance of the model compared to the base model with no adaptation. Furthermore, the effectiveness of our algorithms is analyzed by comparing our results to AuxAdapt's. We use many state-of-the-art single-frame semantic networks [3, 15, 26, 27, 32, 33] in different situations to illustrate the superiority of our methods.

SceneNet RGB-D and Cityscapes Evaluation: In this set of experiments, the performance of our method are compared to AuxAdapt and the base network. In Addition to TC and mIoU, the estimated number of floating-point operations per second (GFLOPS) and Memory used in one iteration are also available in Table 1 and Table 2

For Momentum Adapt experiments on both datasets, as shown by Table 1, the method makes a notable difference in both performance metrics compared to the base network. For example, on SceneNet RGB-D dataset, Momentum Adapt increases TC and mIoU for HRNetV2-w18s network by 6% and 3.7%, respectively. Furthermore, Momentum Adapt supersedes the performance of AuxAdapt in most experiments, except for the case of the SegFormer network on the SceneNet RGB-D dataset, where it has around 1% lower performance in both metrics.

More stability is the other advantage of Momentum Adapt. As shown by the red numbers in Table 1, AuxAdapt has collapsed to a single class, outputting constant predictions of only one class, thus having a very low mIoU. This failure of AuxAdapt can be due to the unchanging scenery, for example, the camera pointing to a wall in the SceneNet RGB-D dataset for many frames.

As shown by Table 2, similar to Momentum Adapt experiments, Aux-Momentum Adapt significantly improves both metrics compare to the case of no adaptation and AuxAdapt. In this set of experiments, as shown by the red figures in Table 2 on the SceneNet RGB-D dataset, AuxAdapt fails two times out of three experiments.

Adaptation	Base Network	TC	mIoU	GFLOPs	Memory(MB)
SceneNet RGB-D Dataset					
No Adaptation	HRNetV2-w18s [26]	53.44	44.32	2.8	1341
AuxAdapt [30]		57.45	45.00	9.8	1997
Momentum Adapt (ours)		**59.47**	**48.03**	10	2017
No Adaptation	HRNetV2-w18 [26]	56.88	45.75	5.4	1363
AuxAdapt [30]		61.98	48.48	18.9	2655
Momentum Adapt (ours)		**65.20**	**51.15**	19.3	2697
No Adaptation	HRNetV2-w48 [26]	59.17	54.47	27.6	1841
AuxAdapt [30]		45.97	2.32	96.6	5053
Momentum Adapt (ours)		**65.87**	**60.3**	97	5321
No Adaptation	SegFormer-b5 [27]	60.61	56.46	15.2	1795
AuxAdapt [30]		**64.15**	**58.80**	53.2	9057
Momentum Adapt (ours)		63.87	57.19	53.7	9421
Cityscapes Dataset					
No Adaptation	Unet-s5-d16 [21]	64.02	66.98	405.7	7099
AuxAdapt [30]		65.74	67.16	1420	23066
Momentum Adapt (ours)		**67.85**	**67.52**	1421	23173
No Adaptation	HRNetV2-w18s [26]	73.08	72.20	19.3	2037
AuxAdapt [30]		77.07	72.80	67.6	4537
Momentum Adapt (ours)		**77.70**	**74.09**	68	4595
No Adaptation	HRNetV2-w18 [26]	75.35	75.84	37	2059
AuxAdapt [30]		78.82	75.85	129.5	6897
Momentum Adapt (ours)		**79.27**	**76.74**	130	6977
No Adaptation	HRNetV2-w48 [26]	76.31	77.12	187.2	3149
AuxAdapt [30]		78.95	77.46	655.2	13105
Momentum Adapt (ours)		**79.13**	**78.19**	655.7	13381
No Adaptation	DeepLabV3-r50-d8 [3]	75.64	76.91	539.4	3229
AuxAdapt [30]		78.92	76.67	1888	13339
Momentum Adapt (ours)		**79.16**	**77.67**	1889	13659

Table 1: Comparison between AuxAdapt and Momentum Adapt evaluated on SceneNet RGB-D and Cityscapes dataset.

Method	TC	mIoU	GFLOPs	Memory(MB)
SceneNet RGB-D Dataset				
HRNet-w18s [26] (AuxNet) w/o Adaptation	53.44	44.32	2.8	1341
HRNet-w18 [26] (MainNet) w/o Adaptation	56.88	45.75	5.4	1363
w/ AuxAdapt [30]	25.60	1.88	12.4	2019
w/ Aux-Momentum Adapt (ours)	**62.10**	**51.20**	15,4	2055
HRNet-w18 [26] (AuxNet) w/o Adaptation	56.88	45.75	5.4	1363
HRNet-w48 [26] (MainNet) w/o Adaptation	59.17	54.47	27.6	1841
w/ AuxAdapt [30]	63.64	55.42	41.1	2917
w/ Aux-Momentum Adapt (ours)	**64.74**	**56.51**	46,9	2999
HRNet-w18 [26] (AuxNet) w/o Adaptation	56.88	45.75	5.4	1363
SegFormer-b5 [27] (MainNet) w/o Adaptation	60.61	56.46	15.2	1795
w/ AuxAdapt [30]	23.65	0.63	28.7	3063
w/ Aux-Momentum Adapt (ours)	**66.44**	**58.39**	34	3063
Cityscapes Dataset				
HRNet-W18s [26] (AuxNet) w/o Adaptation	73.08	72.20	19.3	2037
HRNet-W18 [26] (MainNet) w/o Adaptation	75.35	75.84	37	2059
w/ AuxAdapt [30]	78.99	75.36	85.2	4559
w/ Aux-Momentum Adapt (ours)	**79.27**	**76.31**	104.5	4683
HRNet-W18 [26] (AuxNet) w/o Adaptation	75.35	75.84	37	2059
HRNet-W48 [26] (MainNet) w/o Adaptation	76.31	77.12	187.2	3149
w/ AuxAdapt [30]	79.55	77.01	279.7	7283
w/ Aux-Momentum Adapt (ours)	**79.86**	**78.20**	316.7	7433
HRNet-W18 [26] (AuxNet) w/o Adaptation	75.35	75.84	37	2059
DeepLabV3Plus-R50-D8 [3] (MainNet) w/o Adaptation	75.16	77.67	352.7	3961
w/ AuxAdapt [30]	80.03	77.57	445.2	6973
w/ Aux-Momentum Adapt (ours)	**80.22**	**79.30**	482.2	7127

Table 2: Comparison between AuxAdapt and Aux-Momentum Adapt on SceneNet RGB-D and Cityscapes dataset.

Domain Shift Evaluation (from KITTI to Cityscapes): For many datasets, The ground truth labels of every frame are unavailable. In this case, the model is usually trained on a

Method	TC	mIoU
FCN-r101-d8 (AuxNet and MainNet) [15] w/o Adaptation	62.17	55.37
w/ AuxAdapt [30]	67.65	58.86
w/ Momentum Adapt (ours)	**69.64**	**59.47**
HRNetV2-W18 [26] (AuxNet and MainNet) w/o Adaptation	64.42	60.15
w/ AuxAdapt [30]	70.32	62.74
w/ Momentum Adapt (ours)	**71.57**	**64.21**
DeepLabV3Plus-r50-d8 [3] (AuxNet and MainNet) w/o Adaptation	64.03	59.66
w/ AuxAdapt	71.31	63.97
w/ Momentum Adapt (ours)	**72.15**	**65.76**
PSPNet-r101-d8 [32] (Auxnet and MainNet) w/o Adaptation	66.94	61.20
w/ AuxAdapt	72.00	63.46
w/ Momentum Adapt (ours)	**72.78**	**65.97**

Table 3: Comparison between AuxAdapt and Momentum Adapt from Cityscapes to KITTI Dataset.

Method	TC	mIoU
HRNetV2-w18 [26] (auxnet) w/o Adaptation	64.42	60.15
HRNetV2-w48 [26] (mainnet) w/o Adaptation	65.07	58.75
w/ AuxAdapt [30]	**72.06**	60.37
w/ Aux-Momentum Adapt (ours)	71.37	**66.30**
HRNetV2-w18 [26] (auxnet) w/o Adaptation	64.42	60.15
ANN-r50-d8 [33] (mainnet) w/o Adaptation	62.87	58.36
w/ AuxAdapt [30]	71.68	63.34
w/ Aux-Momentum Adapt (ours)	**73.08**	**66.05**
HRNetV2-w18 [26] (auxnet) w/o Adaptation	64.42	60.15
DeepLabV3Plus-r50-d8 [3] (mainnet) w/o Adaptation	64.03	59.66
w/ AuxAdapt [30]	72.88	66.62
w/ Aux-Momentum Adapt (ours)	**73.14**	**67.65**
HRNetV2-w18 [26] (auxnet) w/o Adaptation	64.42	60.15
PSPNet-r101-d8 [32] (mainnet) w/o Adaptation	66.94	61.20
w/ AuxAdapt [30]	72.94	66.42
w/ Aux-Momentum Adapt (ours)	**73.37**	**67.95**

Table 4: Comparison between AuxAdapt and Aux-Momentum Adapt from Cityscapes to KITTI Dataset

similar dataset, and then fine-tuned for the target dataset. By improving the confidence of the model, the model will have more consistent output in the new domain, improving its performance. Therefore, we use our adaptation methods and their counterpart algorithm, AuxAdapt, to show the effectiveness of temporal consistency adaptation for a different domain.

In the following experiments, the models are trained on the Cityscapes dataset and evaluated on KITTI. Figure 3 illustrates the performance of both methods, AuxAdapt and Momentum Adapt. Compared to the base network, both methods significantly improve TC and mIoU. However, Momentum Adapt outperforms AuxAdapt in every experiment.

For Aux-Momentum, as illustrated by Table 4, similar improvements to Momentum Adapt experiments can be observed. However, in one of the models, where HRNetV2-w48 is the MainNet and HRNetV2-w18 is the AuxNet, TC in Aux-Momentum is lower by 0.69% compared to the AuxAdapt model. At the same time, its mIoU is significantly higher than that of AuxAdapt.

Lower Update Frequency Evaluation: Adapting the model less frequently can save a lot of computation. Therefore, it is important to see how much lower rates of update impact each algorithm. In other words, the model in not updated for some frames, despite being evaluated for those frames. For Momentum Adapt experiments, similar to noise experiments, HRNetV2-w18 is used for both networks in both algorithms. Figure 5 shows lowering the update rate has a similar effect on both algorithms and decreases both measures, TC and mIoU, by a small amount. The widest performance gap is between skipping no frame (updating every frame) and skipping four frames (updating every fifth frame), which is around 1.5% for both metrics.

Method	# Skipped Frames	TC	mIoU
AuxAdapt [30]	0	78.82	75.85
Momentum Adapt (ours)		**79.27**	**76.74**
AuxAdapt [30]	1	78.17	75.56
Momentum Adapt (ours)		**78.86**	**76.97**
AuxAdapt [30]	2	77.85	75.89
Momentum Adapt (ours)		**78.52**	**76.57**
AuxAdapt [30]	4	77.53	75.56
Momentum Adapt (ours)		**78.16**	**76.34**

Table 5: Comparison between AuxAdapt and Momentum Adapt with different Adaptation Frequency, when both AuxNet and MainNet are HRNetV2-W18.

For Aux-Momentum Adapt, similar to noise experiments, HRNetV2-w48 and HRNetV2-w18 are employed as MainNet and AuxNet, respectively, to compare the performance for various update rates. Figure 6 depicts an even better picture for Aux-Momentum Adapt compared to Momentum Adapt. mIoU measure for Aux-Momentum Adapt almost stay constant, despite the lowered adaptation frequency. Temporal Consistency also remains almost unchanged, the widest gap being around 0.6%. Similarly, the performance loss in AuxAdapt is also minimal but more significant than Aux-Momentum Adapt.

Method	# Skipped Frames	TC	mIoU
AuxAdapt [30]	0	79.55	77.01
Aux-Momentum Adapt (ours)		**79.86**	**78.20**
AuxAdapt [30]	1	79.01	76.62
Aux-Momentum Adapt (ours)		**79.58**	**78.27**
AuxAdapt [30]	2	78.89	77.05
Aux-Momentum Adapt (ours)		**79.47**	**78.16**
AuxAdapt [30]	4	78.63	76.74
Aux-Momentum Adapt (ours)		**79.17**	**78.10**

Table 6: Comparison between AuxAdapt and Aux-Momentum Adapt with different Adaptation Frequency, when the MainNet is HRNetV2-w48 and the AuxNet is HRNetV2-w18.

More results, including noise experiments and qualitative measures, are included in supplementary materials.

5 Conclusion

This paper presents a novel unsupervised online adaptation method, Momentum Adapt, for improving the temporal consistency of the output in video semantic segmentation tasks. Comprehensive testing shows that our method, in most cases, significantly outperforms its counterpart algorithm, AuxAdapt. Additionally, we introduce a second algorithm, which combines our approach and AuxAdapt, to take advantage of utilizing a smaller network for training. By conducting more experiments in the presence of noise, domain shift, and lower update frequency, while our methods have more computation, they have superior performance and are more stable.

For Future work, the concepts used in this paper can be applied to other tasks, such as depth estimation and object tracking.

References

[1] URL https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html.

[2] Ali Abbasi, Ramin Toosi, and Mohammad Ali Akhaee. Fast and temporal consistent video style transfer. In *2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pages 1–6, 2021. doi: 10.1109/IPRIA53572.2021.9483531.

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.

[4] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions, 2022. URL https://arxiv.org/abs/2205.08534.

[5] Xu Cheng, Yifeng Zhang, Jinshi Cui, and Lin Zhou. Object tracking via temporal consistency dictionary learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(4):628–638, 2017. doi: 10.1109/TSMC.2016.2618749.

[6] 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, 2016.

[7] Mark Everingham, S. Eslami, Luc Van Gool, Christopher Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111, 01 2014.

[8] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020.

[10] Chenyang Lei, Yazhou Xing, Hao Ouyang, and Qifeng Chen. Deep video prior for video consistency and propagation. *CoRR*, abs/2201.11632, 2022. URL https://arxiv.org/abs/2201.11632.

[11] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation, 2018. URL https://arxiv.org/abs/1804.00389.

[12] Zhaoshuo Li, Wei Ye, Dilin Wang, Francis X. Creighton, Russell H. Taylor, Ganesh Venkatesh, and Mathias Unberath. Temporally consistent online depth estimation in dynamic scenes. *CoRR*, abs/2111.09337, 2021. URL https://arxiv.org/abs/2111.09337.

[13] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. *CoRR*, abs/2002.11433, 2020. URL https://arxiv.org/abs/2002.11433.

[14] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer V2: scaling up capacity and resolution. *CoRR*, abs/2111.09883, 2021. URL https://arxiv.org/abs/2111.09883.

[15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015.

[16] Vasile Lup and Sergiu Nedevschi. Video semantic segmentation leveraging dense optical flow. In *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 369–376, 2020. doi: 10.1109/ICCP51029.2020.9266150.

[17] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J.Davison. Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth. 2016.

[18] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. *CoRR*, abs/1612.08871, 2016. URL http://arxiv.org/abs/1612.08871.

[19] Andreas Pfeuffer, Karina Schulz, and Klaus Dietmayer. Semantic segmentation of video sequences with convolutional lstms. *CoRR*, abs/1905.01058, 2019. URL http://arxiv.org/abs/1905.01058.

[20] Manuel Rebol and Patrick Knöbelreiter. Frame-to-frame consistent semantic segmentation. *CoRR*, abs/2008.00948, 2020. URL https://arxiv.org/abs/2008.00948.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

[22] Serin Varghese, Yasin Bayzidi, Andreas Bär, Nikhil Kapoor, Sounak Lahiri, Jan Schneider, Nico Schmidt, Peter Schlicht, Fabian Hüger, and Tim Fingscheidt. Unsupervised temporal consistency metric for video segmentation in highly-automated driving. pages 1369–1378, 06 2020. doi: 10.1109/CVPRW50498.2020.00176.

[23] Serin Varghese, Sharat Gujamagadi, Marvin Klingner, Nikhil Kapoor, Andreas Bär, Jan David Schneider, Kira Maag, Peter Schlicht, Fabian Hüger, and Tim Fingscheidt. An unsupervised temporal consistency (tc) loss to improve the performance of semantic segmentation networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 12–20, 2021. doi: 10.1109/CVPRW53098.2021.00010.

[24] Serin Varghese, Sharat Gujamagadi, Marvin Klingner, Nikhil Kapoor, Andreas Bär, Jan David Schneider, Kira Maag, Peter Schlicht, Fabian Hüger, and Tim Fingscheidt. An unsupervised temporal consistency (tc) loss to improve the performance of semantic segmentation networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 12–20, 2021. doi: 10.1109/CVPRW53098.2021.00010.

[25] Hao Wang, Weining Wang, and Jing Liu. Temporal memory attention for video semantic segmentation, 2021. URL https://arxiv.org/abs/2102.08643.

[26] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition, 2020.

[27] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *CoRR*, abs/2105.15203, 2021. URL https://arxiv.org/abs/2105.15203.

[28] Jinze Yu, Jiaming Liu, Xiaobao Wei, Haoyi Zhou, Yohei Nakata, Denis Gudovskiy, Tomoyuki Okuno, Jianxin Li, Kurt Keutzer, and Shanghang Zhang. Mttrans: Cross-domain object detection with mean-teacher transformer, 2022.

[29] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation. *CoRR*, abs/1908.03706, 2019. URL http://arxiv.org/abs/1908.03706.

[30] Yizhe Zhang, Shubhankar Borse, Hong Cai, and Fatih Porikli. Auxadapt: Stable and efficient test-time adaptation for temporally consistent video semantic segmentation, 2021. URL https://arxiv.org/abs/2110.12369.

[31] Yizhe Zhang, Shubhankar Borse, Hong Cai, Ying Wang, Ning Bi, Xiaoyun Jiang, and Fatih Porikli. Perceptual consistency in video segmentation. *CoRR*, abs/2110.12385, 2021. URL https://arxiv.org/abs/2110.12385.

[32] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016. URL http://arxiv.org/abs/1612.01105.

[33] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation, 2019.