Predicting Deeper into the Future of Semantic Segmentation — Supplementary Material —

Pauline Luc[1,2*] Natalia Neverova[1*] Camille Couprie[1] Jakob Verbeek[2] Yann LeCun[1,3]

[1] Facebook AI Research

[2] Inria Grenoble, Laboratoire Jean Kuntzmann, Université Grenoble Alpes

[3] New York University

{paulineluc,nneverova,coupriec,yann}@fb.com jakob.verbeek@inria.fr

For extensive qualitative comparison and examples of predictions of the models, we refer the interested reader to the project page provided at https://thoth.inrialpes.fr/people/pluc/iccv2017.

1. Improved baseline relying on optical flow

We propose an improvement over an initial baseline that has previously been used for next frame prediction in the space of RGB intensities and relying on optical flow employed in [4] and [5]. In both the initial and our improved flow baselines, the approach is based on using the optical flow field $\mathbf{F}_{t\to t-1}$ computed from $\mathbf{X}_t$, the RGB frame at time t, to $\mathbf{X}_{t-1}$, the RGB frame at time $t-1$.

1.1. Initial flow baseline

Let us consider a spatial position p of the frame we wish to predict $\mathbf{X}_{t+1}$. The original baseline sets $\hat{\mathbf{X}}_{t+1}(p)$ by bilinearly interpolating the values of $\mathbf{X}_t$ surrounding spatial position $p+d$, where $d = \mathbf{F}_{t\to t-1}(p)$. The issue is that one should be using the flow vector $d' = \mathbf{F}_{t+1\to t}(p)$ instead, which we cannot access since we are trying to predict $\mathbf{X}_{t+1}$. The displacements d and d' are in general not equal, since the physical point corresponding to d may have been replaced by another, which potentially does not have the same displacement. For instance, this is the case for still points that are about to be occluded by moving objects. In this case, d is zero, whereas d' could correspond to a fast moving point. A qualitative example is shown is Figure 1, where the values for the pixels in front of the moving car are not replaced by those of the car as they should be. Note that this is a systematic failure case of the flow baseline used in [4] and [5], which concerns all pixels which are about to be occluded by a moving object. We call this baseline "$t+1$ - centric", as it can be viewed as looping over the spatial positions of the prediction $\hat{\mathbf{X}}_{t+1}$.

1.2. Improved optical flow baseline

We propose an improved baseline which does not have this shortcoming. Considering a spatial position p in the last input frame $\mathbf{X}_t$, we project its value into the next frame by using the opposite of the flow vector at p as an estimation for the displacement of the corresponding physical point between time steps t and $t+1$, setting

$$\hat{\mathbf{X}}_{t+1}([p+d]) = \mathbf{X}_t(p), \tag{1}$$

where $d = -\mathbf{F}_{t\to t-1}(p)$ and where $[\cdot]$ denotes rounding.

In case of competing values for position $[p+d]$, we prioritize these corresponding to the largest flow to favor displacement of moving and close-by objects, as opposed to still and far objects or stuff. This baseline is called "t - centric"

We apply the same transformation procedure to the flow field $\mathbf{F}_{t\to t-1}$ to get $\hat{\mathbf{F}}_{t+1\to t}$, which we use to predict $\mathbf{X}_{t+2}$ and so on. We solve a Dirichlet boundary value problem to interpolate the missing values that were not determined by the warping in Equation 1. The flow fields themselves are computed using Full Flow [3], a state of the art optical flow estimation method,

*These authors contributed equally

Figure 1: From left to right: two input frames and three predictions, framed in red, obtained from the warping baselines, "$t+1$ - centric" (top) and "t - centric" (bottom). Static pixels that are about to be occluded by the moving objects are systematically mis-predicted to hold their previous values in the original baseline - see the pixels in front of the left car on the top row - leading moving objects to shrink instead of moving. This is corrected in our improved baseline, shown in the bottom row.

using the default parameters given by the authors on the MPI Sintel Flow Dataset [2]. As in our other experiments, we employed a frame interval of 3. Decreasing the frame interval down to 1 leads to worse performance for mid-term prediction (43.5 instead of 44.3 IoU GT for a frame rate of 3) because of error propagation, as more predictions are needed to reach 0.5s.

2. Further exploration for the S2S model

2.1. Adversarial training

As Mathieu *et al*. [4] in the context of raw images, we introduce an adversarial loss term allowing the model to disambiguate between modes corresponding to different turns of events.

The adversarial network is a two-scale convolutional network. The coarse-scale discriminator subnetwork has a single convolutional layer $128\times3\times3$, followed by three fully connected layers with 512, 256 and 1 hidden units respectively. The fine-scale subnetwork consists of three convolutional layers ($128\times3\times3$, $128\times3\times3$, $256\times3\times3$) and three fully connected layers with 512, 256, 1 hidden units.

Following [1], we employ clipping of the discriminator weights Θ to the range $[-0.01, 0.01]$ after each gradient update, and set the target coefficient α to 0.9 to prevent saturation. In our setting, every iteration of the discriminator training is followed by a single update of the generator's parameters. We found that $\lambda = 0.1$ provides the optimal balance between the loss terms.

2.2. Deeper models and dilated convolutions

Our multiscale S2S model based on standard convolutions has a field of view of 30 over input resolution 128×256. We perform further architecture exploration to boost the model's performance. First we increase the number of parameters from 1.5M to 8.5M by linearly scaling the number of feature maps per layer. This boosts the IoU GT performance for short-term prediction from 58.3 to 59.0 on the Cityscapes validation set. Next, inspired by the recent developments of semantic segmentation architectures [6], we replace the inner convolutions of each of the two subnetworks of the architecture, with dilated convolutions of dilation respectively 2, 2, 2 and 4. This increases the field of view from 30 to 46 and improves performance from 59.0 to 59.5.

In parallel, we test a simpler but deeper single scale architecture. We design it to have yet a broader field of view of 65. We summarize its architecture below :

$$N_I \times C \quad - \text{conv } k7 \rightarrow \quad 32 * q \quad - \text{conv } k5 \rightarrow \quad 64 * q \quad - \text{conv } k5,d2 \rightarrow \quad 64 * q \quad - \text{conv } k3,d4 \rightarrow \quad 128 * q$$
$$- \text{conv } k5,d8 \rightarrow \quad 64 * q \quad - \text{conv } k5 \rightarrow \quad 32 * q \quad - \text{conv } k3 \rightarrow \quad C,$$

where N_I is the number of input frames, C is the number of classes, $n_1 \quad - \text{conv } ki,dj \rightarrow \quad n_2$ is an adequately padded convolutional layer taking n_1 input feature maps, outputing n_2 feature maps, of kernel size i and dilation j, followed by a ReLU for all the inner layers, and q is a hyperparameter to scale the number of feature maps linearly for simple control over the model capacity. With $q = 4$, this architecture has 8.2M parameters and obtains overall best performance of 60.4. We tried a yet deeper architecture, keeping the number of parameters fixed, but it saturated at 59.4 performance.

Finally, to retain the possibility of fine-tuning this architecture in an autoregressive fashion on a single GPU, we scale the parameters back down to 0.9M, corresponding to a choice of $q = 1.25$. We call this model S2S-dil and record its performance in Tables 1 and 4 of the main paper. We recall here its short-term IoU GT performance of 59.4.

3. Results obtained on the test set

We measure performance on the test set of the Cityscapes dataset for mid-term prediction of our optical flow baseline and of our two models S2S, AR, fine-tune and S2S-dil, AR, fine-tune. We use the same setup as we used on the validation set in the main paper: we take in input frames 2, 5, 8, and 11, and predict outputs for frames 14, 17 and 20 of each sequence. Results for frame 20 are shown in Table 1. For reference, we also show the performance reported by the authors of the Dilation10 architecture [6] on the test set. These "oracle" results give an idea of the maximum performance that could be expected, since this oracle was used to provide the training data.

Model	IoU GT	IoU SEG	IoU-MO GT
Dilation10 oracle	67.1	100	61.5
Warp last input	45.9	49.5	39.1
S2S, AR, fine-tune	47.8	51.8	40.2
S2S-dil, AR, fine-tune	**48.0**	**52.0**	**40.4**

Table 1: Mid-term segmentation prediction for frame 20 using our best S2S model on the Cityscapes test set.

4. Failure cases

Figures 2 and 3 show two failures cases of our S2S model (fine-tuned in autoregressive mode) for mid-term prediction (half-a-second future), where the model respectively underestimates the speed of the camera and fails to predict a future occlusion.

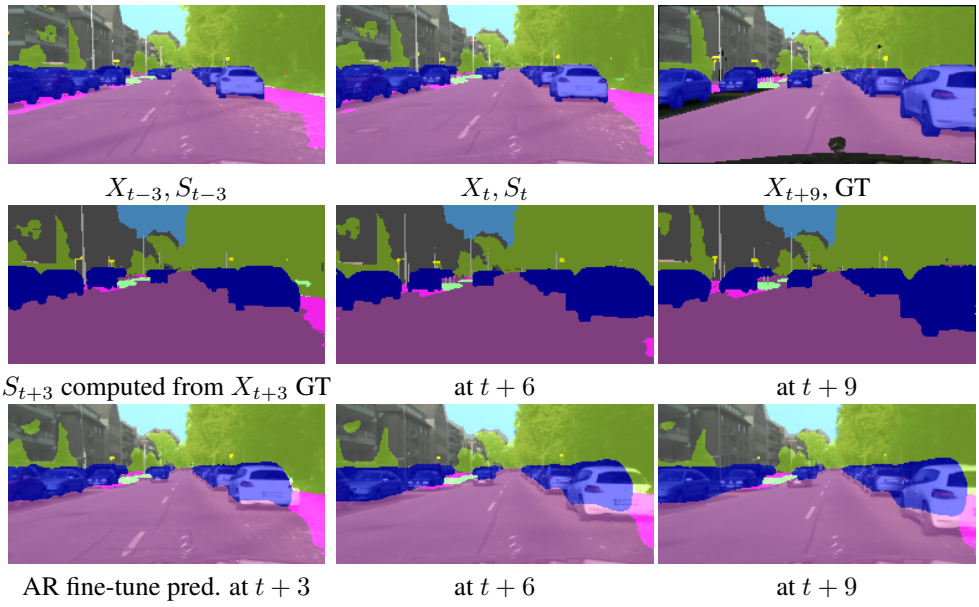

Figure 2: Failure case of the autoregressive fine-tuned S2S model. First row: last inputs and ground truth. Second row: future segmentations obtained using the Dilation10 network computed using the future RGB frames. Third row: S2S, AR, fine-tune predictions overlaid with the true future frames. In this example, the speed of the camera is underestimated by our model, resulting in large errors in the segmentation of the closest car.

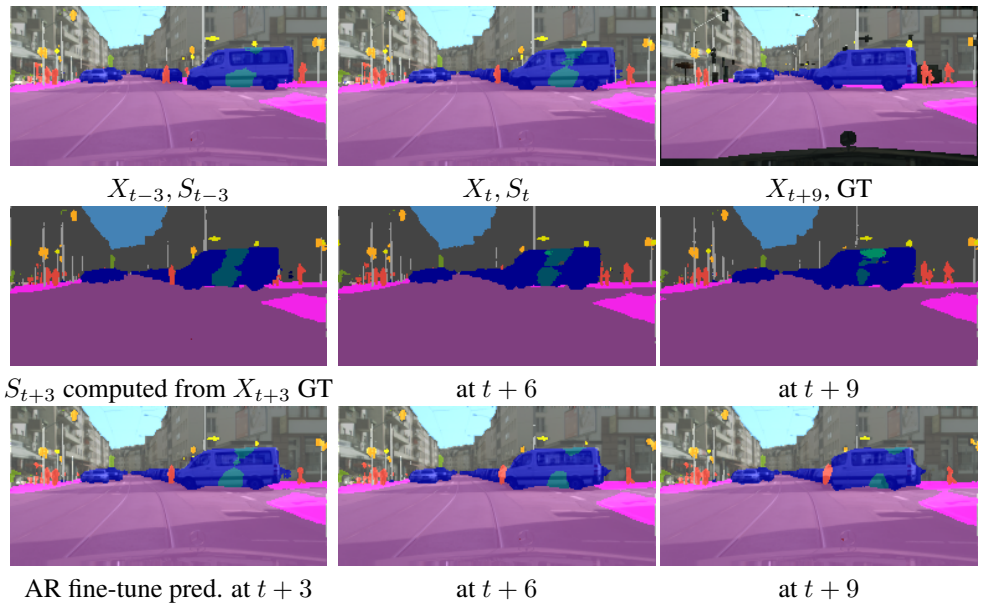

Figure 3: Failure case of the autoregressive fine-tuned S2S model. First row: last inputs and ground truth. Second row: future segmentations obtained using the Dilation10 network computed using the future RGB frames. Third row: S2S, AR, fine-tune predictions overlaid with the true future frames. In this example, the occlusion of the pedestrian by the vehicle coming from the right is not predicted by our system. The green blobs that appear in the vehicle correspond to the "bus" category. This second mistake is hard to avoid because it also appears in the Dilation10 input segmentations.

5. Comparison of batch and autoregressive methods

We present visualizations that extend the ones given in Figure 5 of the main paper. Figures 4 and 5 compare S2S models for mid-term prediction (half-a-second future) with the different approaches presented in the paper: batch, autoregressive, autoregressive using adversarial training, and autoregressive fine-tuned. Results are also compared with our optical flow baseline. All segmentations are overlaid with the true video sequence to facilitate assessment of the predictions.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. In *ICML*, 2017. 2

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 2

[3] Q. Chen and V. Koltun. Full flow: Optical flow estimation by global optimization over regular grids. In *CVPR*, 2016. 1

[4] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. 1, 2

[5] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv:1412.6604*, 2014. 1

[6] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2, 3

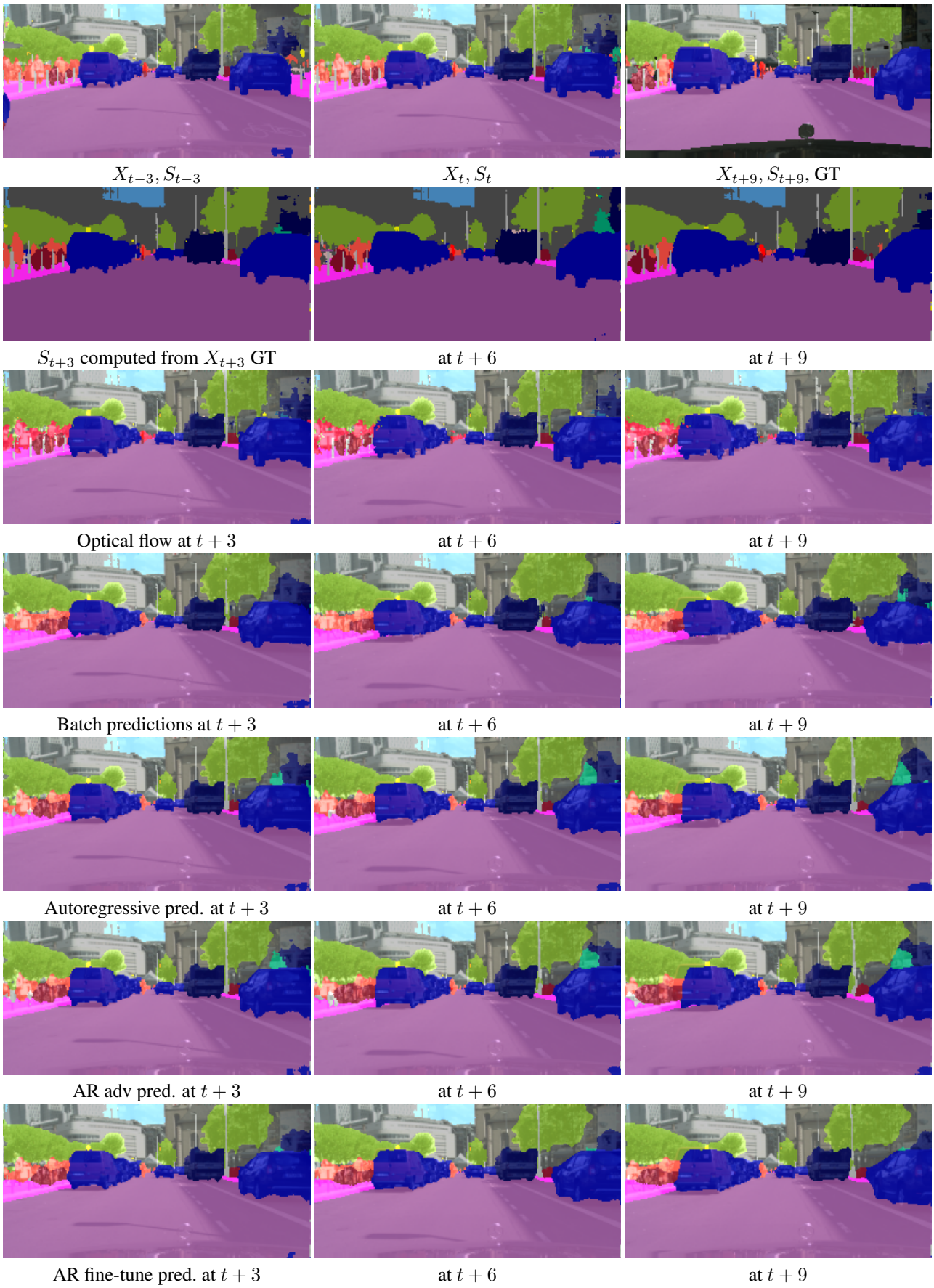

Figure 4: Comparison between optical flow baseline, batch, autoregressive, adversarially fine-tuned autoregressive, and autoregressive fine-tuned S2S model predictions. First row: last inputs and ground truth segmentation. Second row: target segmentations obtained using the Dilation10 network. Other rows show predictions overlaid with the true future frames.

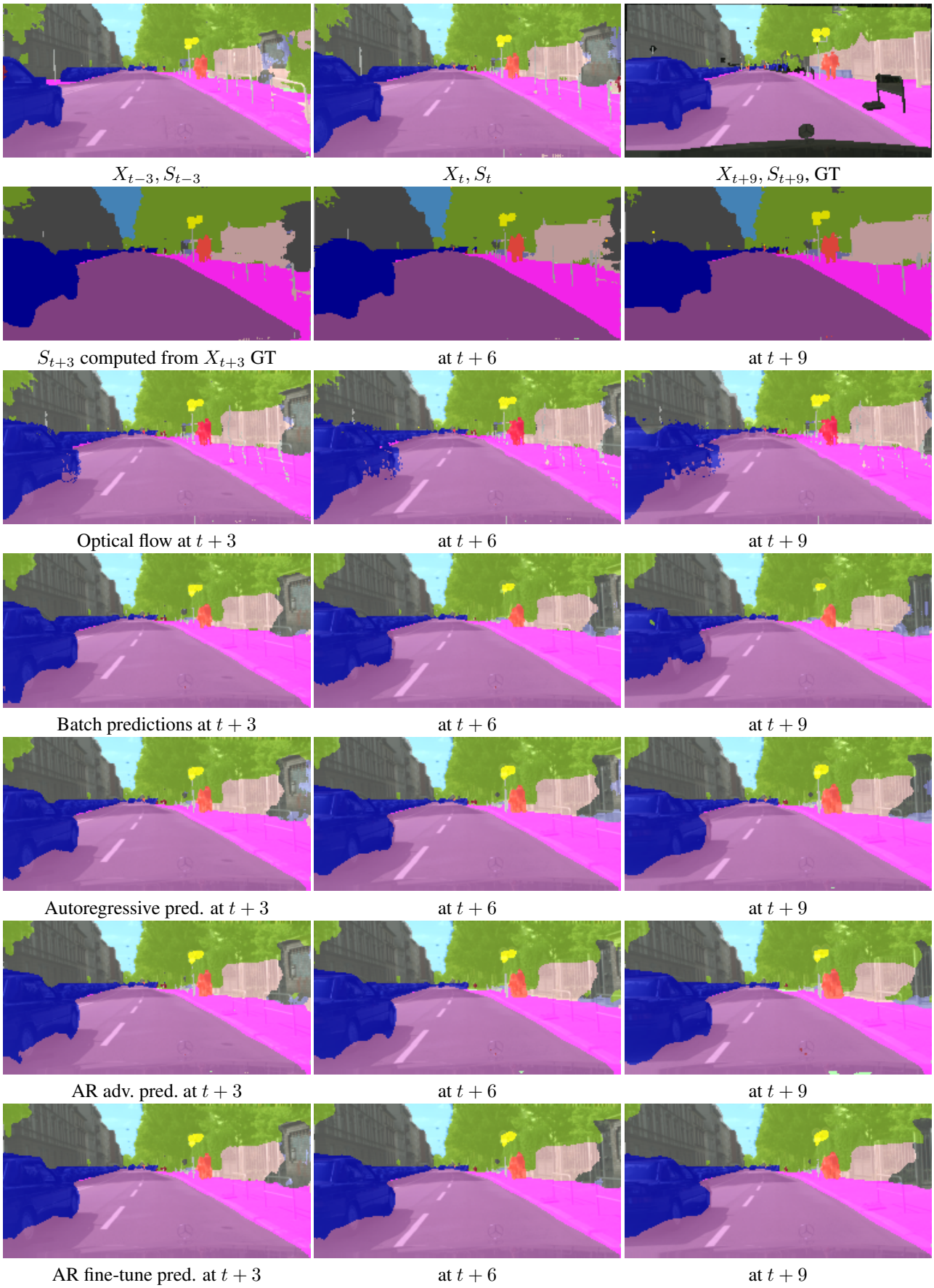

Figure 5: Comparison between optical flow baseline, batch, autoregressive, adversarially fine-tuned autoregressive, and autoregressive fine-tuned S2S model predictions. First row: last inputs and ground truth segmentation. Second row: target segmentations obtained using the Dilation10 network. Other rows show predictions overlaid with the true future frames.