TEST-TIME TRAINING ON VIDEO STREAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

We investigate visual generalization video streams instead of independent images, since the former is closer to the smoothly changing environments where natural agents operate. Traditionally, single-image models are tested on videos as collections of unordered frames. We instead test on each video in temporal order, making a prediction on the current frame before the next arrives, after training at test time on frames from the recent past. To perform test-time training without ground truth labels, we leverage recent advances in masked autoencoders for self-supervision. We improve performance on various real-world applications. We also discover that forgetting can be beneficial for test-time training, in contrast to the common belief in the continual learning community that it is harmful. [1]

1 INTRODUCTION

Computer vision models deployed in the real world are usually expected to handle input data in the form of continuous video streams. However, most such models are trained with large collections of still images, e.g. the COCO dataset (Lin et al., 2014), causing a mismatch between training and testing data. At test time, such models are applied to each video in a frame-by-frame fashion, as if it was a collection of independent images. Thus, model predictions across frames can be inconsistent, even when the frames are visually similar. Simple averaging or temporal smoothing across predictions offer little improvement. The goal of our paper is to improve the performance of models trained on still image collections but deployed on continuous video streams.

For a trained model that is kept fixed at test time, a video is indeed no different from a collection of unordered frames, which can only be treated independently. To take advantage of the temporal nature of our prediction problem, we propose to keep updating the model as it sees more of the video. This is similar to the approach taken in Test-Time Training (Sun et al., 2020; Gandelsman et al., 2022) for the task of generalization under distribution shifts. Indeed, as suggested by Mullapudi et al. (2018), the smooth, non-stationary scene changes within a video stream could be considered as a series of distribution shifts, motivating our approach.

At each point in time, before the model makes a prediction on the current frame, we first fine-tune it on this frame and recent frames in the past. Since there is no ground truth label on the test video, training is performed with self-supervision. After prediction, the model repeats the same process for the next frame, initializing with parameters from the current timestep. In principle, any self-supervised learning technique could be used inside this loop. We use masked autoencoding (He et al., 2021) for self-supervision, by masking out patches in the input frame and learning to predict them.

Our method improves performance on three datasets for four tasks: instance, panoptic and semantic segmentation, and colorization. We collect a new dataset – COCO Videos – with dense annotations for very long videos. Figure 1 visualizes results on COCO Videos for panoptic segmentation, as an example of how our approach improves accuracy and consistency for structural prediction. We also discover new insights on how forgetting can be beneficial, consistent with recent studies in neuroscience (Gravitz, 2019), but contrary to prior ideas in continual learning. Our theoretical analysis characterizes forgetting as finding a sweet spot in a bias-variance trade-off.

[1] Our code and dataset will be made publicly available. Please visit our anonymous project website at `https://video-ttt.github.io/` to watch videos of our results.

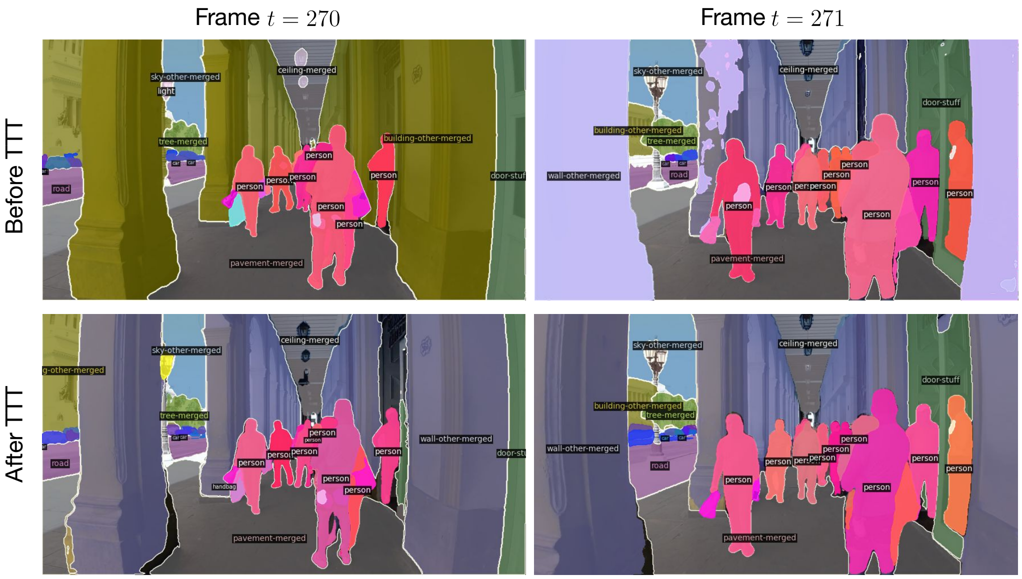


Figure 1: Panoptic segmentation predictions for adjacent frames from a video in our new dataset, COCO Videos. **Top**: Baseline results produced by a state-of-the-art model. Predictions are inconsistent between the two frames. **Bottom**: Results after TTT, by the same model, on the same frames as top. Predictions are now consistent and correct. Please zoom in to see the instance labels.

2 Related Work

2.1 Continual and Online Learning

In the field of continual (a.k.a. lifelong) learning, a model learns a sequence of tasks in temporal order, and is asked to perform well on all of them (Van de Ven & Tolias, 2019; Hadsell et al., 2020). Here is the basic problem setting. Each task is defined by a data distribution P_t, which produces a training set D_t^{tr} and a test set D_t^{te}. At each time t, the model is evaluated on all the test sets $D_1^{\text{te}}, \ldots, D_t^{\text{te}}$ of the past and present, and average performance is reported.

The best performing solution would be to train the model on all of $D_1^{\text{tr}}, \ldots, D_t^{\text{tr}}$, which collectively have the same distribution as the test sets. However, due to constraints in computation and memory, the model at time t is only allowed to train on D_t^{tr}, or a limited piece of memory (a.k.a. replay buffer) that contains a small fraction of the past training sets. Most solutions, therefore, focus on how to avoid forgetting, since majority of the past data can only be retained by the model parameters (Santoro et al., 2016; Li & Hoiem, 2017; Lopez-Paz & Ranzato, 2017; Shin et al., 2017; Kirkpatrick et al., 2017; Gidaris & Komodakis, 2018).

Among the vast literature on continual learning, many papers extend beyond the basic setting above. Due to space constraints, we point to a few of the most relevant ones. Aljundi et al. (2019) use continuous instead of discrete tasks across time. Purushwalkam et al. (2022) and Fini et al. (2022) perform self-supervised learning on unlabeled training sets, and evaluate the learned features on the test sets. Hoffman et al. (2014), Li & Hospedales (2020) and Panagiotakopoulos et al. (2022) use not only unlabeled training sets $D_1^{\text{tr}}, \ldots, D_t^{\text{tr}}$ (target), but also a labeled training set D_0^{tr} (source), thus sharing some terminology and techniques with the field of unsupervised domain adaptation.

Much of continual learning is motivated by the hope to understand human memory and generalization through artificial intelligence (Hassabis et al., 2017; De Lange et al., 2021). Our work is inspired by the same motivation. However, to the best of our knowledge, all settings in continual learning have distinct splits of training and test sets. This is rarely how humans operate. With our work, we can now use every video frame for both training and testing. Since training and test data coincide, our evaluation focuses on the present instead of the past. In the end, our experiments arrive at a different conclusion from the continual learning community, that forgetting can be beneficial instead of harmful.

The term "online learning" has developed different meanings in different communities. In computer vision, it is sometimes used interchangeably with "continual learning" (Li & Hospedales, 2020; Panagiotakopoulos et al., 2022). Two other relevant papers (Jain & Learned-Miller, 2011; Mullapudi et al., 2018) using this term are discussed in Subsection 2.2. In machine learning, the basic setup of online learning (a.k.a. online optimization) usually repeats the following at each timestep (Shalev-Shwartz et al., 2012; Hazan et al., 2016): receive the input, make a prediction, then receive the label from an adversarial oracle. In contrast, we never reveal the ground truth label, neither is it adversarial.

2.2 Learning at Test Time

The idea of training on test data has been explored in computer vision over the past decade. Jain & Learned-Miller (2011) improve face detection by using the easier faces in a test image to bootstrap the more difficult faces in the same image. Shocher et al. (2018) train neural networks for super-resolution on each test image from scratch. Nitzan et al. (2022) create a personalized generative model, by fine-tuning it on a few images of an individual person's face. These are but a few examples where vision researchers find it natural to continue training during deployment (this idea has also been useful in RL, e.g. in (Pathak et al., 2017; Hansen et al., 2020)).

The most relevant work is test-time training (TTT) (Sun et al., 2020; Gandelsman et al., 2022). The idea is simple: for each test input, before making a prediction on the main task – in their case, object recognition – first train the model on this input. A self-supervised task is used to setup this one-sample learning problem without ground truth label. Recent work (Gandelsman et al., 2022) shows that using masked autoencoding (MAE) (He et al., 2021) as the self-supervised task for TTT produces superior results. TTT has been applied to many other domains (Hansen et al., 2020; Sun et al., 2021; Fu et al., 2021; Banerjee et al., 2021; Karani et al., 2021; Li et al., 2021). Others use a batch (Wang et al., 2020) or dataset (Liu et al., 2021a) of inputs from the same test distribution.

Our paper is also inspired by Mullapudi et al. (2018). To make video segmentation more efficient, their method makes predictions with a student model that performs online learning on each test video from a teacher model. Since the video is temporally smooth, the student only queries the teacher on a few frames, so learning and prediction combined is still faster than the teacher alone. Our method produces learning signal with a self-supervised task instead of a teacher model. This makes it possible for our "student" – the only model considered – to improve performance instead of efficiency.

Recent work (Azimi et al., 2022) also performs TTT on videos, with a few key differences from ours. They treat each video as a dataset of unordered frames instead of a stream. In particular, they always have access to the entire video, including future frames. They also use the same model on each video. In contrast, at each time, we have access to only the current and past frames, and our model keeps learning over time. In addition, all of our results are on real world videos, while Azimi et al. (2022) experiment only on videos with artificial corruptions. From what we understand, their corruptions are also i.i.d. across frames of even the same video.

3 Method

We consider a video as a smoothly changing sequence of frames $x_1, \ldots, x_T$. We want to evaluate an algorithm on the video following its temporal order, as if it was consumed by a human. At each time t, an algorithm should make a prediction on x_t after receiving it from the environment, before seeing any future frame. Naturally, the past frames $x_1, \ldots, x_{t-1}$ are also available at the time of prediction, in addition to x_t. Ground truth labels should never used by the algorithm.

Given a trained model F that takes a single image as input, the obvious baseline is to blithely run F frame-by-frame, predicting $F(x_t)$ at each timestep. Currently, this is indeed the most common mode of deploying a single-image model on videos. Can we do better?

3.1 Architecture

Our goal is to improve F through test-time training (TTT) (Sun et al., 2020; Gandelsman et al., 2022), while watching and making predictions on each test video as a stream. Following prior work, we use a self-supervised task to train F without ground truth labels. In principle, our algorithm can use any self-supervised task. In this paper, we use masked autoencoding (MAE) (He et al., 2021), since recent work (Gandelsman et al., 2022) shows that MAE works well for TTT.

The self-supervised task – in our case, pixel reconstruction from an input image with masked patches – needs its own prediction head. This requires us to modify the network architecture. We first split F into two parts as $F = h \circ f$, where f is the feature extractor, and h is the prediction head for the main task – in our case, segmentation or colorization. For a neural network F composed of a sequence of layers, this split is realized by making f the first few layers, and h the rest.

Next we add a prediction head g, for the self-supervised task, after f. In the context of autoencoders, f is also called the encoder, and g the decoder. This creates a Y-shaped architecture, where f is shared, and h and g are the two branches. The main task uses $h \circ f$ and the self-supervised task uses $g \circ f$. For symmetry, we design g to have the same architecture as h, except the last layer mapping to a different output space.

3.2 Joint Training

Since we are given a trained model F, the weights of h and f have already been optimized for performance on the main task, but g is initialized from scratch. This initialization is undesirable for TTT since performance of the self-supervised task would then start from chance level. Therefore, we prepare our model for TTT by training it jointly on both tasks, using the training set of images with labels, following Sun et al. (2020).

Denote the main task loss as ℓ_m, and the self-supervised loss as ℓ_s. We optimize those two losses together to produce a self-supervised head g_0 trained from scratch, as well as a main task head h_0 and feature extractor f_0 further trained from the given pre-trained weights:

$$g_0, h_0, f_0 = \arg\min_{g,h,f} \frac{1}{n} \sum_{i=1}^{n} \Big[\ell_m(h \circ f(x_i), y_i) + \ell_s(g \circ f(\tilde{x}_i), x_i) \Big]. \tag{1}$$

The summation is over the training set with n samples, each consisting of input x_i and label y_i. $\tilde{x}_i$ is x_i transformed as input for the self-supervised task, in our case, by masking 80% of the patches.

3.3 Test-Time Training

Prior work such as Sun et al. (2020) and Gandelsman et al. (2022) perform TTT one image at a time. Given a single input image x (and $\tilde{x}$ transformed for the self-supervised task), they formulate a one-sample learning problem that solves for

$$g', f' = \arg\min_{g,f} \ell_s(g \circ f(\tilde{x}), x), \tag{2}$$

initializing from g_0 and f_0. A prediction for the main task is then made as $h_0 \circ f'(x)$. While we can naively apply this to a video by making the input x_t instead of x, this misses point of using a video. Like the single-image baseline using a fixed model, single-image TTT treats a video as a collection of unordered frames. Neither of the two can improve over time, no matter how long a video explores the same environment.

Improvement over time is only possible through memory, by retaining some information from the past frames $x_1, \ldots, x_{t-1}$ to help prediction on x_t. Because evaluation is performed at each timestep only on the current frame, our memory design should favor past data that are most relevant to the present. Fortunately, with the help of nature, the most recent frames usually happen to be the most relevant due to temporal smoothness – observations close in time tend to be similar. We design memory that favors recent frames in the following two ways.

Explicit memory. The most explicit way of remembering recent frames is to keep them in a sliding window. Let k denote the window size. At each timestep t, our method solves the following optimization problem

$$g_t, f_t = \arg\min_{g,f} \frac{1}{k} \sum_{t'=t-k+1}^{t} \ell_s(g \circ f(\tilde{x}_{t'}), x_{t'}), \tag{3}$$

before predicting $h_0 \circ f_t(x_t)$. Optimization is performed with stochastic gradients: at each iteration, we sample a batch with replacement, uniformly from the same window. Masking is applied independently within and across batches.

Implicit memory. To initialize the optimization problem in Equation 3, we have two choices:

- Use g_0 and f_0. For each timestep, this resets the model parameters before TTT to those at the beginning of the video. Once a prediction is made, the new parameters after TTT are discarded.
- Use g_{t-1} and f_{t-1}. For each timestep, this simply keeps the models parameters after TTT.

Our default is to use g_{t-1} and f_{t-1}, that is, no reset. This creates an implicit memory, since information carries over from the previous model, optimized on previous frames. Our choice takes advantage of temporal smoothness in videos. It also happens to be more biologically plausible: we humans do not constantly reset our minds. In contrast, prior work (Gandelsman et al., 2022) chooses the former, because it does not assume correlation between different inputs. In Sun et al. (2020), the two choices are called the "standard" and "online" version, respectively.

4 Empirical Results

We experiment with four applications on three real-world datasets: 1) semantic segmentation on a public dataset of urban driving videos; 2) instance and panoptic segmentation on a new dataset we collected and annotated, of videos with COCO objects in daily scenes; 3) colorization on black and white films. In all these experiments, our method performs significantly better than the baseline of applying a fixed model frame-by-frame. Please visit our anonymous project website at `https://video-ttt.github.io/` to watch videos of our results.

Architecture. Our default architecture is Mask2Former (Cheng et al., 2021), which has recently achieved state-of-the-art performance on many semantic, instance and panoptic segmentation benchmarks. Our method is generally applicable to modern network architectures, and does not rely on any particular property of Mask2Former. Our Mask2Former uses a Swin-S (Liu et al., 2021b) backbone – in our case, this is also the shared feature extractor (a.k.a. encoder) f. Everything following the backbone in the original architecture is taken as the main task head h, and our self-supervised head (a.k.a. decoder) g copies the architecture of h except the last layer.

Masking. Following He et al. (2021), we split each input into patches, and mask out 80% of them. However, unlike the Vision Transformers (Dosovitskiy et al., 2020) used in He et al. (2021), Swin Transformers use convolutions. Therefore, we must take the entire image as input (with the masked patches in black) instead of only the unmasked patches. Following Pathak et al. (2016), we use a fourth channel of binaries to indicate if the corresponding input pixels are masked. The model parameters for the fourth channel are initialized from scratch before joint training.

4.1 Semantic Segmentation on KITTI-STEP

Dataset. KITTI-STEP (Weber et al., 2021) contains 12 training videos and 9 validation videos of urban driving scenes. [2] At the rate of 10 frames-per-second, these videos are the longest – up to 106 seconds – among public datasets with dense pixel-wise annotations. Since we do not perform regular training on these videos, we select hyper-parameters on the validation set, and use the larger training set as the test set. We report results both on this test set and the validation set.

Training. KITTI-STEP has exactly the same 19 categories as CityScapes (Cordts et al., 2016), another driving dataset of individual images instead of videos. We take the publicly available Mask2Former model pre-trained on CityScapes images. This model is both our baseline (applied frame-by-frame on KITTI-STEP videos), and initialization for joint training, also on CityScapes. Before TTT, our model never sees any image or annotation from KITTI-STEP.

Main results. Quantitative results, measured by mean intersection over union (IoU), are presented in Table 1. Figure 3 in the appendix provides a snapshot of our qualitative results, in comparison with the single-image baseline. Please see our anonymous project website for the complete set of videos with segmentation masks, comparing our method with the baseline. Note that joint training alone does not improve on the baseline, indicating that our improvements come from test-time training.

Additional TTT variant. TENT (Wang et al., 2020) is another method for learning at test time. It minimizes the softmax entropy of the model's outputs by updating normalization parameters. To

[2] Full name: KITTI Segmenting and Tracking Every Pixel. The video frames are part of the KITTI dataset. KITTI-STEP is originally designed to benchmark instance-level tracking, and has a separate test set held-out by the organizers. The official website evaluates only tracking-related metrics on this test set. Therefore, we perform our own evaluation using segmentation labels available on the training and validation set.

	DeepLab	SegForm	Baseline	TENT	Joint	No Mem.	Reset	Ours
Val	42.0	53.1	53.8	53.7	53.5	55.0	54.3	**57.6**
Test	41.6	50.8	52.5	52.6	52.5	54.3	53.4	**56.0**

Table 1: Results for semantic segmentation in mean IoU (%) on KITTI-STEP validation and test set; see Subsection 4.1. Our baseline is Mask2Former, which already outperforms other models of similar size in the first two columns. TENT, another variant of TTT, does not help. Our method with reset, i.e. without implicit memory, improves only modestly. Our complete method improves significantly on the baseline. The three columns before ours are ablations; see Subsection 5.1.

implement TENT on our baseline, we update LayerNorm with the TENT loss, in the same loop as our method. As seen in Table 1, TENT does not help, even after we have searched for its optimal hyper-parameters. One reason could be that TENT is designed for batches of i.i.d. samples from a wide distribution (e.g. the ImageNet test set), but our batches contain highly correlated frames.

Additional baselines. Mask2Former was not evaluated by the authors on KITTI-STEP. To verify that the pre-trained model (69M) is already the state-of-the-art on KITTI-STEP, we compare its performance with two other popular models of comparable size: SegFormer B4 (Xie et al., 2021) (64.1M) and DeepLabV3+/RN101 (Chen et al., 2017) (62.7M); see Table 1. We also experiment with test-time augmentation of the input frame, applying the default recipe in the Mask2Former codebase to the baseline, using the same amount of computation as TTT. This improves mean IoU on the validation set modestly by 1.2%.

Temporal smoothing. As a sanity check, we also implemented temporal smoothing on the baseline, by averaging the predictions across a short window. The window size is selected to optimize performance after smoothing on the validation set. This improves the baseline by only 0.4% mean IoU on average. Applying temporal smoothing to our method also yields an insignificant 0.3% improvement. This indicates that our method does much more than simple smoothing. To keep things conceptually simple, we do not use temporal smoothing anywhere else in this paper.

4.2 COCO VIDEOS

While KITTI-STEP already contains the longest annotated videos among publicly available datasets, they are still far too short for studying long-term phenomenon such as memory and forgetting. KITTI-STEP videos are also limited to driving scenarios, a small subset of the diverse scenarios in our daily lives. These limitations motivate us to annotate our own videos.

We collected 10 videos, each about 5 minutes. They are manually annotated by professionals, in the same format as for COCO instance and panoptic segmentation (Lin et al., 2014). The benchmark metrics are also the same as in COCO: average precision (AP) for instance and panoptic quality (PQ) for panoptic. To put things into perspective, each one of these 10 videos alone contains more frames, at the same rate, than all of the videos combined in the KITTI-STEP validation set. We compare this new dataset with other publicly available datasets in Table 2.

Dataset	Avg. Length (s)	Frames	Rate (fps)	Scene Type
CityScapes-VPS (Kim et al., 2020)	1.8	3000	17	outdoor, driving
DAVIS (Pont-Tuset et al., 2017)	3.5	3455	30	indoor and outdoor
YouTube-VOS (Xu et al., 2018)	4.5	123,467	30	indoor and outdoor
KITTI-STEP (Weber et al., 2021)	40	8,008	10	outdoor, driving
COCO Videos (ours)	309	30,925	10	indoor and outdoor

Table 2: Comparison of video datasets with annotations for segmentation. The videos in our new dataset – COCO Videos – are orders of magnitude longer than other publicly available ones. It is much larger than KITTI-STEP, and almost as large as YouTube-VOS in total duration, taking into account the frame rate. Our dataset contains diverse scenes from daily life activities.

	Baseline	Joint	No Mem.	Reset	Ours
Instance	4.52	4.53	6.59	6.71	**8.12**
Panoptic	9.34	9.21	13.22	13.25	**15.43**

Table 3: Results for instance and panoptic segmentation on COCO Videos; see Subsection 4.2. Our baseline Cheng et al. (2021) is already the state-of-the-art on COCO. Our method improves significantly on the baseline for both tasks. The metrics for instance and panoptic segmentation are, respectively, average precision (AP) and panoptic quality (PQ). The three columns before ours are ablations; see Subsection 5.1.

All videos are egocentric, similar to the visual experience of a human walking around. In particular, they do not follow any tracked object like in Oxford Long-Term Tracking (Valmadre et al., 2018) or ImageNet-Vid (Shankar et al., 2021). Objects leave and enter the camera's view all the time. The scenes in our videos are both indoors and outdoors, taken from diverse locations such as sidewalks, markets, schools, offices, restaurants, parks and households. Please view these videos on our anonymous project website.

Training. We take the publicly available Mask2Former model pre-trained on images in the COCO training set, for instance and panoptic segmentation respectively, as both our baseline and initialization for joint training. Analogous to our procedure for KITTI-STEP, joint training is also on COCO images, and our 10 videos are only used for evaluation. We use exactly the same hyper-parameters as tuned on the KITTI-STEP validation set, for both joint training and TTT. That is, all of our results for COCO Videos were completed in one run.

Main results. We outperform the baseline by a large margin on both tasks, as seen in Table 3. Figure 1 provides a snapshot of our qualitative results, in comparison with the single-image baseline. Please see our anonymous project website for the complete set of videos with segmentation masks and instance labels, comparing our method with the baseline. For context, our baseline is the state-of-the-art on the COCO validation set, with 44.9 AP for instance and 53.6 PQ for panoptic segmentation. The fact that baseline performance drops to single digits speaks about the challenging nature of our videos, and the fragility of single-image models when evaluated on videos.

4.3 Video Colorization

The goal of colorization is to add realistic RGB colors to a gray-scale input image. Since many old films were taken in black and white, single-image models are often applied to colorize videos (Lei & Chen, 2019; Zhang et al., 2019). The temporal inconsistency of colors often results in visually apparent flickering for video colorization. Our goal in this section is to demonstrate the generality of our method, not to achieve the state-of-the-art in colorization.

Dataset. We colorize the 10 original black-and-white Lumiere Brothers films from 1895, roughly 40 seconds enough. To make the hyper-parameters compatible with those for segmentation, we again use 10 frames per second. Please see Section B in the appendix for a complete list of the films and their lengths.

Training. Following Zhang et al. (2016), we simply treat colorization as a supervised learning problem. We use the same architecture as for segmentation – Swin Transformer with two heads, trained on ImageNet (Deng et al., 2009) to predict the colors given input images processed as grayscale. We try to make only the changes required to map to a different output space for the main task. We do not use modern techniques that improve domain-specific performance, e.g., perceptual losses, adversarial learning, and diffusion models. Our bare-minimal baseline already achieves results comparable, if not superior, to those in Zhang et al. (2016). To apply our method to this baseline, we use exactly the same hyper-parameters as for segmentation.

Main results. It is a field consensus (Zhang et al., 2016; 2017) that quantitative metrics often misrepresent performance, because colorization is inherently multi-modal. Figure 2 provides a snapshot of our qualitative results, in comparison with the single-image baseline. Please see our anonymous project website for the complete set of original and colorized videos, comparing our method with the baseline. Our method visually improves the quality of colorization in all the videos that we experimented with, especially in terms of consistency across frames.

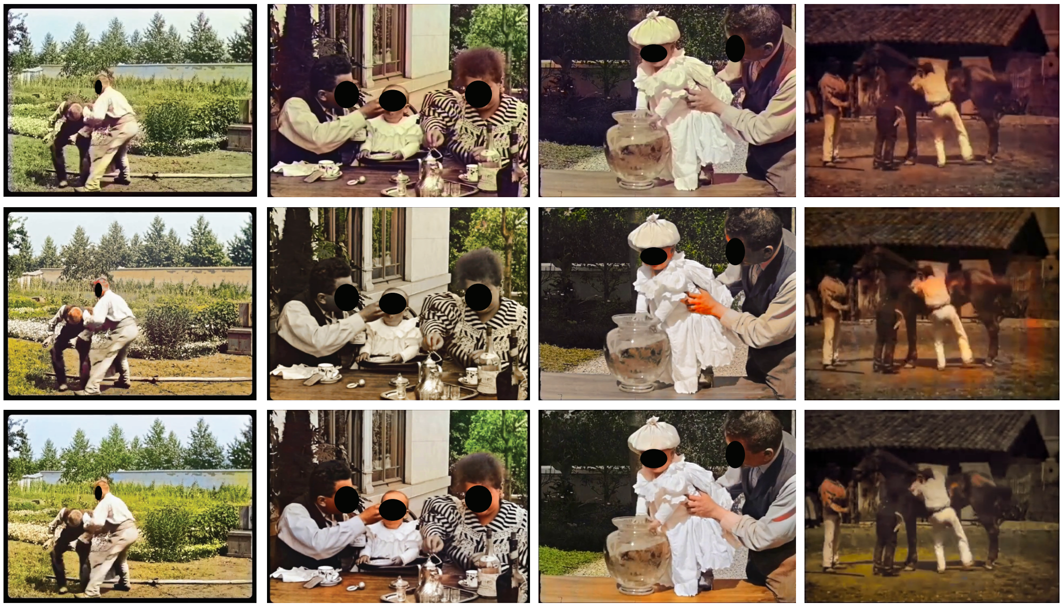


Figure 2: Samples results for video colorization on the Lumiere Brothers films. **Top**: Results using the released model of Zhang et al. (2016). **Middle**: Results using our own baseline, which are already comparable, if not superior to those of Zhang et al. (2016). **Bottom**: Results after applying our method on the baseline. Our colors are more vibrant (e.g. the second column) and more consistent within regions (e.g. human body parts in the third column). The complete set of results are available as videos on our anonymous project website.

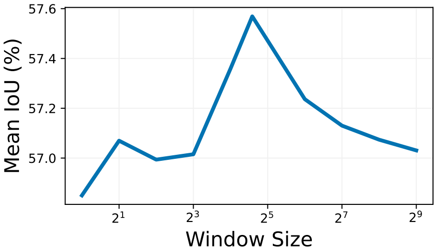
5 Analysis on Memory and Forgetting

In Section 3, we discussed two ways to create memory: implicitly through parameter reuse, and explicitly through a sliding window. In this section, we first analyze the effect of both through ablations, then characterize the trade-off between memory and forgetting as equivalent to that between bias and variance.

5.1 Empirical Analysis

In Table 1 and 3, the third to last column, abbreviated for no memory, reports performance of our method without either implicit or explicit memory. Here TTT is performed on each image independently, and each video is simply treated as a test set of frames. In this case, our method still improves modestly on the baseline, for both datasets and all three tasks. This corroborates the findings in Gandelsman et al. (2022) and Sun et al. (2020), that even single-image TTT (so-called "standard version") helps under distribution shifts.

The second to last column of Table 1 and 3 reports performance with only explicit memory, that is, using a sliding window of the same size as for our default method. To eliminate implicit memory, after each timestep we reset the model parameters to those at the beginning, i.e. g_0 and f_0. Results are, again, only modestly better than the baseline and worse than our default method, indicating that adding implicit memory in this case could help.



However, it is not always the case that more memory is better. The figure in the left plots semantic segmentation performance on the KITTI-STEP validation set, for various window sizes k on the x-axis in log-scale, from $k = 1$ to $k = 512$. Too little memory hurts, but too much hurts too. The sweet spot in the middle corresponds to our result of 57.6% in Table 1. This observation makes intuitive sense: frames in the distant past become less relevant for the current.

In the end, not surprisingly, if we shuffle all frames of the videos, results for the last two columns (our method and with reset) of Table 1 on the KITTI-STEP validation set become much worse than the baseline, while results for no memory does not change at all. As discussed in Section 3, both our implicit and explicit memory are designed to favor recent frames, which are assumed to be more relevant. Without temporal smoothness, this assumption is broken, and memory hurts.

5.2 Theoretical Analysis

To complement our empirical observation that forgetting can be beneficial, we now rigorously analyze the effect of our window size k through simple mathematics.

Setting. To simplify notations, define the following functions of the vector of model parameters θ:

$$\nabla \ell_m^{\,t}(\theta) := \nabla_\theta \ell_m(x_t, y_t; \theta), \tag{4}$$

$$\nabla \ell_s^{\,t}(\theta) := \nabla_\theta \ell_s(x_t; \theta). \tag{5}$$

Taking gradient steps with $\nabla \ell_m^{\,t}$ would directly optimize the test loss, since (x_t, y_t) is the test input and its ground truth. However, y_t is not available, so TTT optimizes the self-supervised loss ℓ_s instead. Among the available gradients, $\nabla \ell_s^{\,t}$ is obviously the most relevant. But we also have all the past test inputs $x_1, \ldots, x_{t-1}$. Should we use some, or even all of them?

Theorem. For every timestep t, consider TTT with gradient-based optimization using:

$$\frac{1}{k} \sum_{t'=t-k+1}^{t} \nabla \ell_s^{\,t'}, \tag{6}$$

where k is the window size. Let θ_0 denote the initial condition, and $\tilde{\theta}$ where optimization converges for TTT. Let θ^* denote the optimal solution of $\ell_m^{\,t}$ in the local neighborhood of θ_0. Then we have

$$\mathbb{E}\left[\ell_m(x_t, y_t; \tilde{\theta}) - \ell_m(x_t, y_t; \theta^*)\right] \leq \frac{1}{2\alpha}\left(k^2\beta^2\eta^2 + \frac{1}{k}\sigma^2\right), \tag{7}$$

under the following three assumptions:

1. In a local neighborhood of θ^*, $\ell_m^{\,t}$ is α-strongly convex in θ, and β-smooth in x.
2. $\|x_{t+1} - x_t\| \leq \eta$.
3. $\nabla \ell_m^{\,t} = \nabla \ell_s^{\,t} + \delta_t$, where δ_t is a random variable with mean zero and variance σ^2.

The proof is in Section A of the appendix.

Bias-variance trade-off. Disregarding the constant factor of $1/\alpha$, the upper bound in Equation 7 is the sum of two terms: $k^2\beta^2\eta^2$ and $1/k \cdot \sigma^2$. The former is the bias term, growing with η. The latter is the variance term, growing with σ^2. More memory, i.e., larger k, reduces variance, but increases bias. This is consistent with our intuition: a larger sliding window reduces variance by simply adding more data, but the additional data is less relevant, i.e. more biased. Optimizing this upper bound w.r.t. k shows the sweet spot to occur at

$$k = \left(\frac{\sigma^2}{\beta^2\eta^2}\right)^{1/3}.$$

Remark on assumptions. The assumption that neural networks are locally convex around minima is widely accepted (Allen-Zhu et al., 2019; Zhong et al., 2017; Wang et al., 2021). The assumption that the main task and self-supervised task have correlated gradients comes from Sun et al. (2020).

6 Limitations

We perform more computation than direct inference. In theory, TTT is 8 (`batch_size`) $\times 100$ (`number_of_iterations`) $\times 2 = 1600$ times slower than direct inference. In practice it is only about 200 times slower because GPUs are optimized for processing batches.

Reproducibility Statement

Our code is submitted as a zip file in the supplementary materials. We also include implementation details of our algorithm in the main text. Our new dataset – COCO Videos – is available through our anonymous project website. The other two datasets used are publicly available. We include details of the 10 Lumiere Brothers films in the appendix.

References

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11254–11263, 2019.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 242–252. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/allen-zhu19a.html`.

Fatemeh Azimi, Sebastian Palacio, Federico Raue, Jörn Hees, Luca Bertinetto, and Andreas Dengel. Self-supervised test-time adaptation on video data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3439–3448, 2022.

Pratyay Banerjee, Tejas Gokhale, and Chitta Baral. Self-supervised test-time learning for reading comprehension. *arXiv preprint arXiv:2103.11263*, 2021.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *arXiv preprint arXiv:2112.01527*, 2021.

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.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Enrico Fini, Victor G Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2022.

Yang Fu, Sifei Liu, Umar Iqbal, Shalini De Mello, Humphrey Shi, and Jan Kautz. Learning to track instances without video annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8680–8689, 2021.

Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei A. Efros. Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, 2022.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, 2018.

Lauren Gravitz. The importance of forgetting. *Nature*, 571(July):S12–S14, 2019.

Raia Hadsell, Dushyant Rao, Andrei A Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in cognitive sciences*, 24(12):1028–1040, 2020.

Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. *arXiv preprint arXiv:2007.04309*, 2020.

Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *CoRR*, abs/2111.06377, 2021. URL `https://arxiv.org/abs/2111.06377`.

Judy Hoffman, Trevor Darrell, and Kate Saenko. Continuous manifold based adaptation for evolving visual domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 867–874, 2014.

Vidit Jain and Erik Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *CVPR 2011*, pp. 577–584. IEEE, 2011.

Neerav Karani, Ertunc Erdil, Krishna Chaitanya, and Ender Konukoglu. Test-time adaptable neural networks for robust medical image segmentation. *Medical Image Analysis*, 68:101907, 2021.

Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9859–9868, 2020.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Chenyang Lei and Qifeng Chen. Fully automatic video colorization with self-regularization and diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3753–3761, 2019.

Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. In *European Conference on Computer Vision*, pp. 382–403. Springer, 2020.

Yizhuo Li, Miao Hao, Zonglin Di, Nitesh Bharadwaj Gundavarapu, and Xiaolong Wang. Test-time personalization with a transformer for human pose estimation. *Advances in Neural Information Processing Systems*, 34:2583–2597, 2021.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34, 2021a.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021b.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pp. 6467–6476, 2017.

Ravi Teja Mullapudi, Steven Chen, Keyi Zhang, Deva Ramanan, and Kayvon Fatahalian. Online model distillation for efficient video inference. *arXiv preprint arXiv:1812.02699*, 2018.

Yotam Nitzan, Kfir Aberman, Qiurui He, Orly Liba, Michal Yarom, Yossi Gandelsman, Inbar Mosseri, Yael Pritch, and Daniel Cohen-Or. Mystyle: A personalized generative prior. *arXiv preprint arXiv:2203.17272*, 2022.

Theodoros Panagiotakopoulos, Pier Luigi Dovesi, Linus Härenstam-Nielsen, and Matteo Poggi. Online domain adaptation for semantic segmentation in ever-changing conditions. *arXiv preprint arXiv:2207.10667*, 2022.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

Senthil Purushwalkam, Pedro Morgado, and Abhinav Gupta. The challenges of continuous self-supervised learning. *arXiv preprint arXiv:2203.12710*, 2022.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pp. 1842–1850, 2016.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Vaishaal Shankar, Achal Dave, Rebecca Roelofs, Deva Ramanan, Benjamin Recht, and Ludwig Schmidt. Do image classifiers generalize across time? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9661–9669, 2021.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3118–3126, 2018.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pp. 9229–9248. PMLR, 2020.

Yu Sun, Wyatt L Ubellacker, Wen-Loong Ma, Xiang Zhang, Changhao Wang, Noel V Csomay-Shanklin, Masayoshi Tomizuka, Koushil Sreenath, and Aaron D Ames. Online learning of unknown dynamics for model-based controllers in legged locomotion. *IEEE Robotics and Automation Letters*, 6(4):8442–8449, 2021.

Jack Valmadre, Luca Bertinetto, Joao F Henriques, Ran Tao, Andrea Vedaldi, Arnold WM Smeulders, Philip HS Torr, and Efstratios Gavves. Long-term tracking in the wild: A benchmark. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 670–685, 2018.

Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Yifei Wang, Jonathan Lacotte, and Mert Pilanci. The hidden convex optimization landscape of regularized two-layer relu networks: an exact characterization of optimal solutions. In *International Conference on Learning Representations*, 2021.

Mark Weber, Jun Xie, Maxwell Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, et al. Step: Segmenting and tracking every pixel. *arXiv preprint arXiv:2102.11859*, 2021.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 585–601, 2018.

Bo Zhang, Mingming He, Jing Liao, Pedro V Sander, Lu Yuan, Amine Bermak, and Dong Chen. Deep exemplar-based video colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8052–8061, 2019.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016.

Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017.

Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *International conference on machine learning*, pp. 4140–4149. PMLR, 2017.

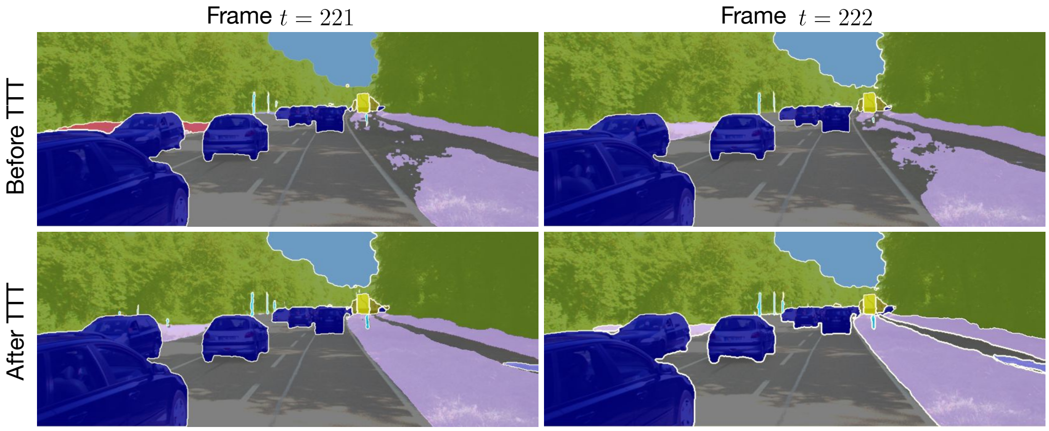


Figure 3: Semantic segmentation predictions for adjacent frames from a video in KITTI-STEP. **Top**: Baseline results produced by a state-of-the-art model. Predictions are inconsistent between the two frames. The terrain on the right side of the road is incompletely segmented in both frames, and the terrain on the left is incorrectly classified as a wall on the first frame. **Bottom**: Results after TTT, by the same model, on the same frames as top. Predictions are now consistent and correct.

A Proof

We first prove the following lemma.

Lemma. Let $f : \mathbb{R}^n \to \mathbb{R}$ be α-strongly convex and continuously differentiable, and denote its optimal solution as x^*. Let

$$\tilde{f}(x) = f(x) + v^T x, \tag{8}$$

and denote its optimal solution as $\tilde{x}^*$. Then

$$f(\tilde{x}^*) - f(x^*) \leq \frac{1}{2\alpha}\|v\|^2. \tag{9}$$

Proof of lemma. It is a well known fact in convex optimization (Bubeck et al., 2015) that for f α-strongly convex and continuously differentiable,

$$\alpha(f(x) - f(x^*)) \leq \frac{1}{2}\|\nabla f(x)\|^2, \tag{10}$$

for all x. Since $\tilde{x}^*$ is the optimal solution of $\tilde{f}$ and $\tilde{f}$ is also convex, we have $\nabla \tilde{f}(\tilde{x}^*) = 0$. But

$$\nabla \tilde{f}(x) = \nabla f(x) + v, \tag{11}$$

so we then have

$$\nabla f(\tilde{x}^*) = \nabla \tilde{f}(\tilde{x}^*) - v = -v. \tag{12}$$

Make $x = \tilde{x}^*$ in Equation 10, we finish the proof.

Proof of theorem. By Assumptions 1 and 2, we have

$$\|\nabla \ell_m^t(\theta) - \nabla \ell_m^{t-1}(\theta)\| \leq \beta\eta. \tag{13}$$

$$\frac{1}{k}\sum_{t'=t-k+1}^{t} \nabla \ell_s^{t'} = \frac{1}{k}\sum_{t'=t-k+1}^{t} \nabla \ell_m^{t'} + \frac{1}{k}\sum_{t'=t-k+1}^{t} \delta_{t'} \tag{14}$$

$$= \frac{1}{k}\sum_{t'=t-k+1}^{t} \left[\nabla \ell_m^t + \sum_{t''=t'}^{t-1}\left(\nabla \ell_m^{t''} - \nabla \ell_m^{t''+1}\right)\right] + \frac{1}{k}\sum_{t'=t-k+1}^{t} \delta_{t'} \tag{15}$$

$$= \nabla \ell_m^t + \frac{1}{k}\left[\sum_{t'=t-k+1}^{t}\sum_{t''=t'}^{t-1}\left(\nabla \ell_m^{t''} - \nabla \ell_m^{t''+1}\right) + \sum_{t'=t-k+1}^{t} \delta_{t'}\right] \tag{16}$$

To simplify notations, define

$$A = \sum_{t'=t-k+1}^{t} \sum_{t''=t'}^{t-1} \left(\nabla \ell_m^{t''} - \nabla \ell_m^{t''+1} \right), \tag{17}$$

$$B = \sum_{t'=t-k+1}^{t} \delta_{t'}. \tag{18}$$

So

$$\frac{1}{k} \sum_{t'=t-k+1}^{t} \nabla \ell_s^{t'} - \nabla \ell_m^{t} = (A+B)/k. \tag{19}$$

Because ℓ_m^t is convex in θ, we know that taking gradients with $\nabla \ell_m^t$ reaches the local optima of ℓ_m^t. Because $\frac{1}{k}\sum_{t'=t-k+1}^{t} \nabla \ell_s^{t'}$ differs from $\nabla \ell_m^t$ by $(A+B)/k$, we know that taking gradients with the former reaches the local optima of $\ell_m^t + (A+B)\theta/2$. Now we can invoke our lemma. To do so, we first calculate

$$\mathbb{E} \left\| \frac{1}{k} \sum_{t'=t-k+1}^{t} \nabla \ell_s^{t'} - \nabla \ell_m^{t} \right\|^2 = \frac{1}{k^2} \mathbb{E} \|A+B\|^2 \tag{20}$$

$$= \frac{1}{k^2} \left(\|A\|^2 + \mathbb{E} \|B\|^2 + \mathbb{E} A^T B \right) \tag{21}$$

$$\le \frac{1}{k^2} \left(k^4 \beta^2 \eta^2 + k \sigma^2 \right) \tag{22}$$

$$= k^2 \beta^2 \eta^2 + \frac{1}{k} \sigma^2. \tag{23}$$

Then by our lemma, we have

$$\mathbb{E} \left[\ell_m(x_t, y_t; \tilde{\theta}) - \ell_m^* \right] \le \frac{1}{2\alpha} \mathbb{E} \left\| \frac{1}{k} \sum_{t'=t-k+1}^{t} \nabla \ell_s^{t'} - \nabla \ell_m^{t} \right\|^2 \le \frac{1}{2\alpha} \left(k^2 \beta^2 \eta^2 + \frac{1}{k} \sigma^2 \right). \tag{24}$$

This finishes the proof.

B Lumiere Brothers Films

We provide results on the following 10 Lumiere Brothers films, all in the public domain:

1. Workers Leaving the Lumiere Factory (46 s)
2. The Gardener (49 s)
3. The Disembarkment of the Congress of Photographers in Lyon (48 s)
4. Horse Trick Riders (46 s)
5. Fishing for Goldfish (42 s)
6. Blacksmiths (49 s)
7. Baby's Meal (41 s)
8. Jumping Onto the Blanket (41 s)
9. Cordeliers Square in Lyon (44 s)
10. The Sea (38 s)