Dynamic Local Regret for Non-convex Online Forecasting

Sergül Aydöre,

Department of ECE
Stevens Institute of Technology,
NJ, USA

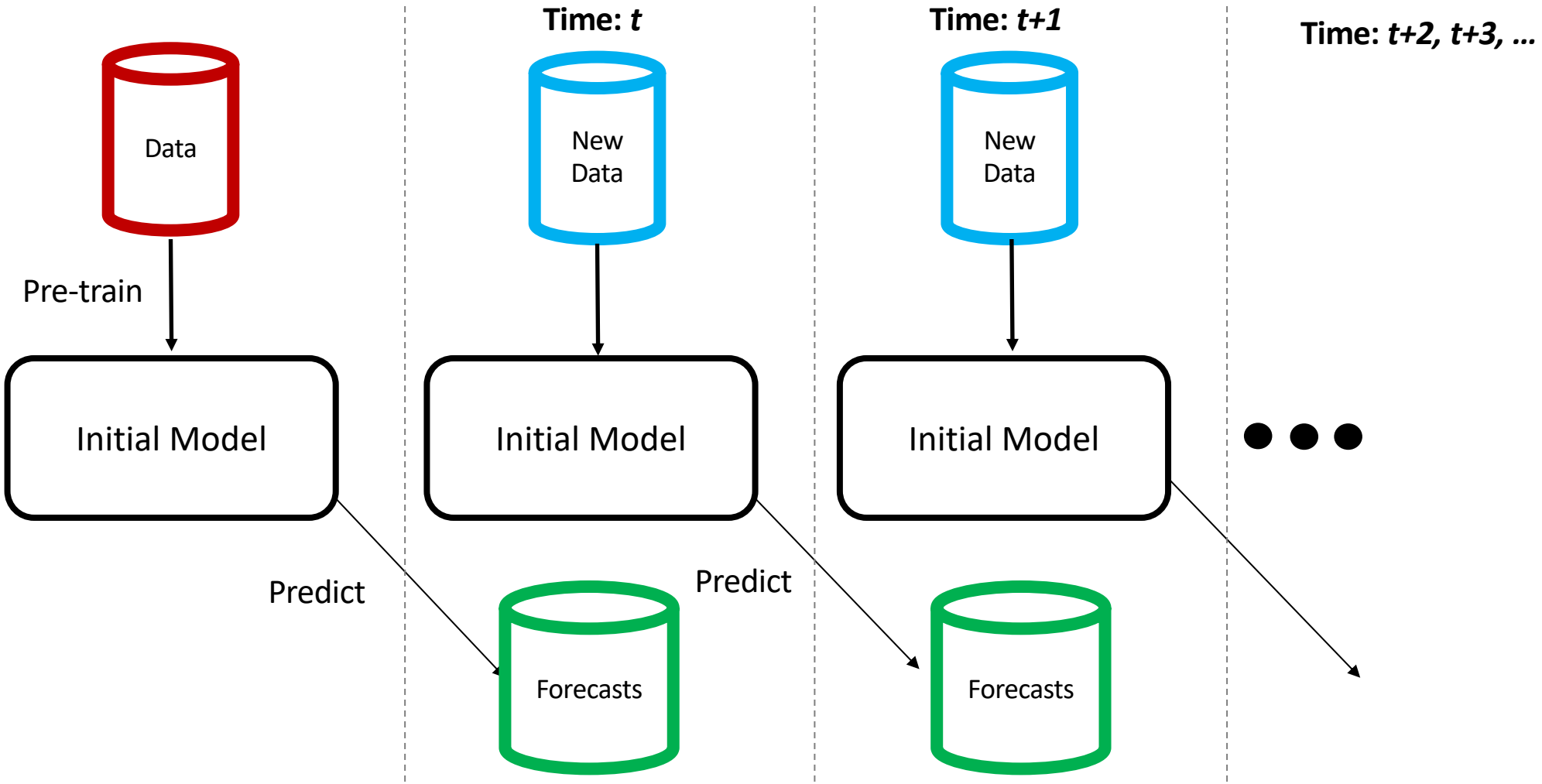
Tianhao Zhu,

Department of ECE
Stevens Institute of Technology,
NJ, USA

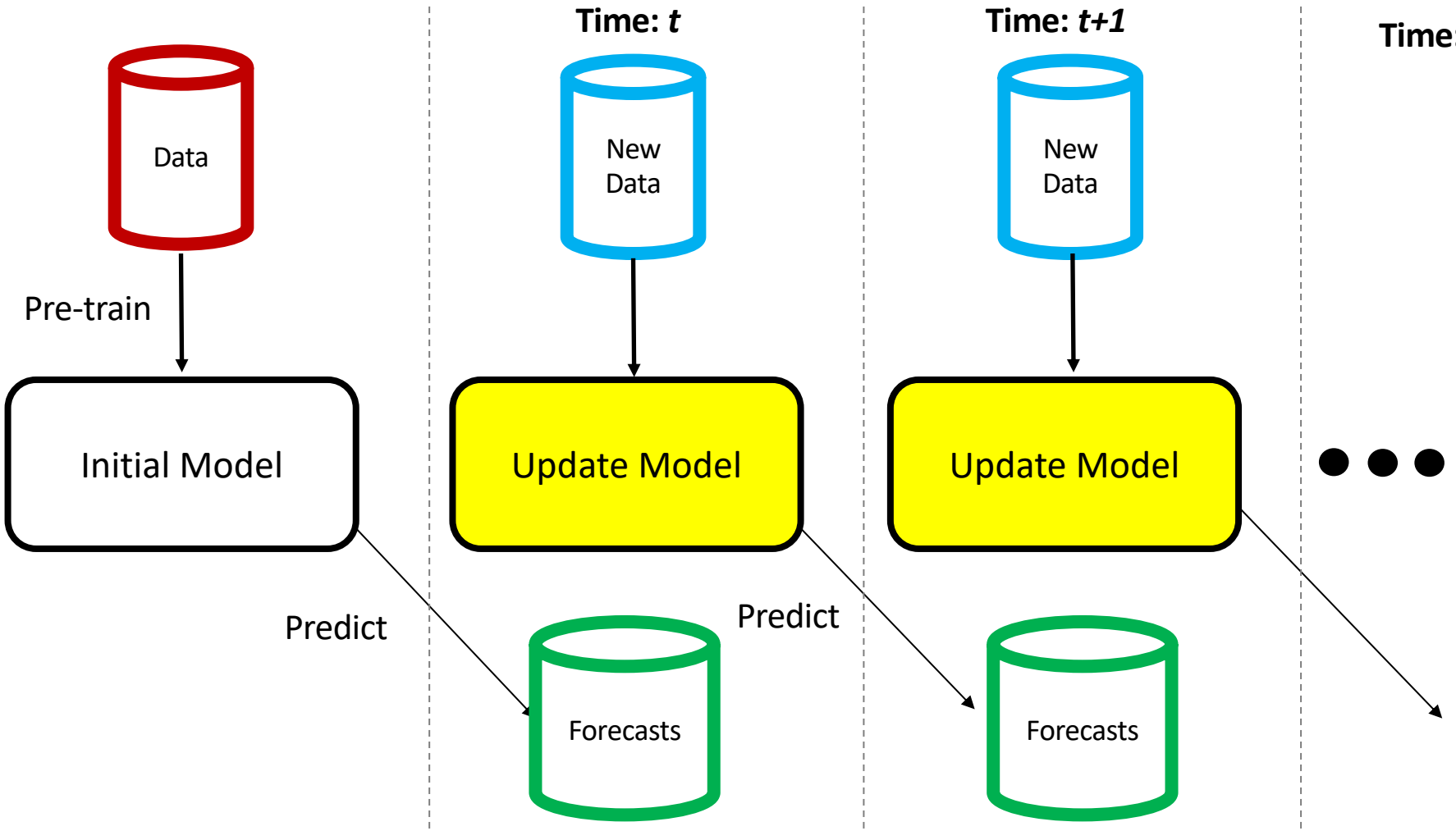
Dean Foster

Amazon
NY, USA

Consider a Forecasting Problem with Frozen Model

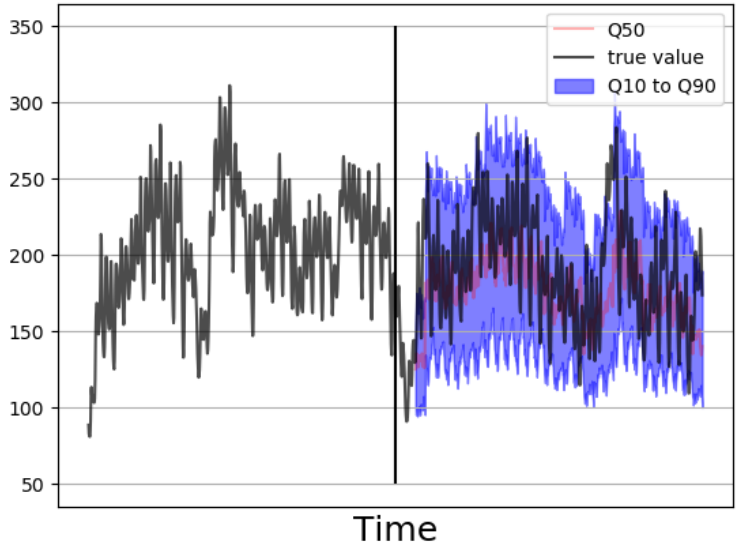


Ideally, we should update our model as fresh data is observed



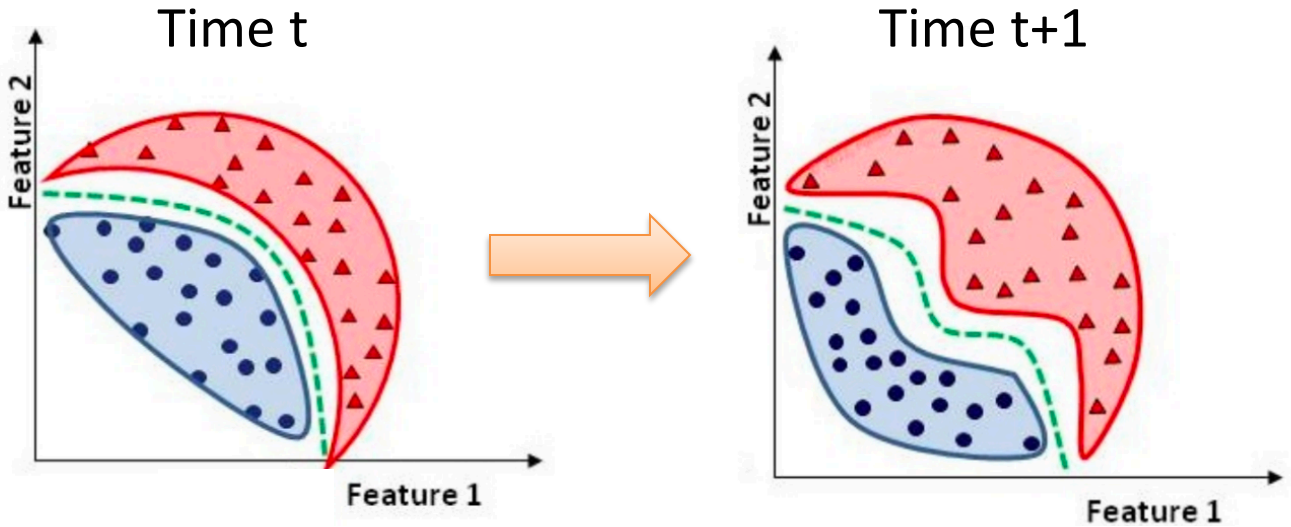
Forecasting Overview

- **Time Series Applications:** Financial market analysis, inventory planning, prediction of weather, earthquake forecasting, etc.
- **Forecasting**: the task of predicting future outcomes based on previous observations



- **Concept Drift:** The underlying relationship between inputs and outputs can change over time.

- **Solution 1**: Re-train model → Computationally Expensive
- **Solution 2:** Update model → Online Learning!



Hoens et al, 2011

Problem Settings: Online Learning

- A collection of T consecutive time points: $t \in \mathcal{T} = \{1, \cdots, T\}$
- Non-convex loss functions at each time t:

$$f_1, \cdots, f_T : \mathcal{K} \to \mathbb{R}$$

- The loss function using the observed data at time t using the model parameters x_t: $f_t(x_t)$
- The performance of Online Learning algorithms is commonly evaluated by the regret:

$$R(T) \triangleq \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x)$$

- However, this definition is not appropriate for nonconvex problems due to NP-hardness.

Static Local Regret

- It is common to use the magnitude of the gradient to analyze convergence for nonconvex problems.
- Hazan et al. 2017 introduced a local regret measure:

$$SLR_w(T) \triangleq \sum_{t=1}^{T} \|\nabla F_{t,w}(x_t)\|^2 \quad \text{where} \quad \mathcal{K} = \mathbb{R}^d$$

$$F_{t,w} \triangleq \frac{1}{w} \sum_{i=0}^{w-1} f_{t-i}(x_t)$$

- **Problem**: Assumes static best model and effectively evaluates today's forecasts on yesterday's loss functions.

Proposed Dynamic Local Regret

- Static Local Regret definition assumes a static best model.
- Consider the first order Taylor's series expansion of cumulative loss.

$$\sum_{t=1}^{T} f_t(x_t + u) \approx \sum_{t=1}^{T} f_t(x_t) + \sum_{t=1}^{T} \langle u, \nabla f_t(x_t) \rangle$$

- If the updates $\{x_1, \ldots, x_T\}$ are *well-calibrated*, then perturbing x_t by any u cannot substantially reduce the cumulative loss.
- **Lemma**: For all x_s, the following equality holds: $\sup_{\|u\|=1} \sum_{s=t-w+1}^{t} \langle u, \nabla f_s(x_s) \rangle = \left\| \sum_{s=t-w+1}^{t} \nabla f_s(x_s) \right\|$
- **Proposed Dynamic Local Regret**: We address both non-convexity and dynamic environment in a regret framework.

$$DLR_w(T) \triangleq \sum_{t=1}^{T} \|\nabla S_{t,w,\alpha}(x_t)\|^2$$

$$S_{t,w,\alpha}(x_t) \triangleq \frac{1}{W} \sum_{i=0}^{w-1} \alpha^i f_{t-i}(x_{t-i})$$

$$W \triangleq \sum_{i=0}^{w-1} \alpha^i$$

Motivation via a Toy Example

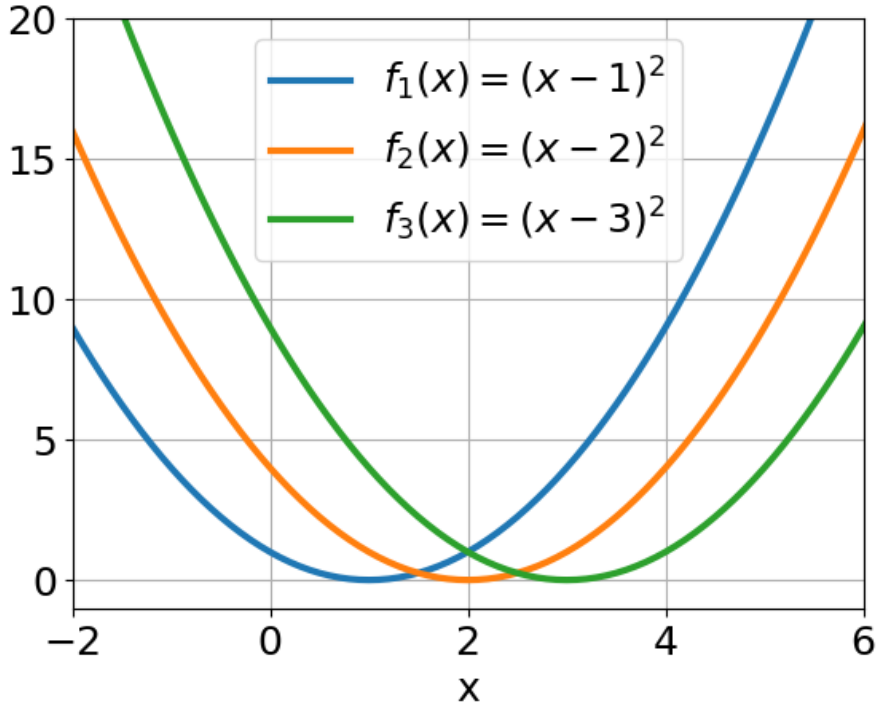
- Consider an online learning problem with concept drift with T=3.
- Oracle Policy: $x_1 = 1, x_2 = 2, x_3 = 3$.
- Stale Policy: $x_1 = 1, x_2 = 1.5, x_3 = 2$.
- Recall the formulation of local regrets

Hazan et al

$$SLR_3(3) = \left\| \frac{\nabla f_3(x_3) + \nabla f_2(x_3) + \nabla f_1(x_3)}{3} \right\|^2 + \left\| \frac{\nabla f_2(x_2) + \nabla f_1(x_2)}{3} \right\|^2 + \left\| \frac{\nabla f_1(x_1)}{3} \right\|^2$$

Ours

$$DLR_3(3) = \left\| \frac{\nabla f_3(x_3) + \nabla f_2(x_2) + \nabla f_1(x_1)}{3} \right\|^2 + \left\| \frac{\nabla f_2(x_2) + \nabla f_1(x_1)}{3} \right\|^2 + \left\| \frac{\nabla f_1(x_1)}{3} \right\|^2$$



- Let's compute loss and regret for the two policies

Regret	Oracle Policy	Stale Policy	Decision
Cumulative Loss	0	$5/4$	Oracle policy is better
Standard Regret	-2	-3/8	Oracle policy is better
Static Local Regret (Hazan et al.)	$40/9$	$4/9$	Stale policy is better
Dynamic Local Regret (Ours)	0	$10/9$	Oracle policy is better

Time-smoothed SGD Algorithms

Hazan et al, 2017

Algorithm 1 Static Time-Smoothed Stochastic Gradient Descent (STS-SGD)

Require: window size $w \geq 1$, learning rate $\eta > 0$, Set $x_1 \in \mathbb{R}^n$ arbitrarily

1: **for** $t = 1, \cdots, T$ **do**
2: Predict x_t. Observe the cost function $f_t : \mathbb{R}^b \to \mathbb{R}$
3: Update $x_{t+1} = x_t - \frac{\eta}{w} \sum_{i=0}^{w-1} \hat{\nabla} f_{t-i}(x_t)$.
4: **end for**

Guarantees upper bound

$$\frac{T}{w} \times \text{Constant}$$

according to the Static Local Regret

Ours

Algorithm 2 Dynamic Exponentially Time-Smoothed Stochastic Gradient Descent (DTS-SGD)

Require: window size $w \geq 1$, learning rate $\eta > 0$, exponential smoothing parameter $\alpha \to 1^-$ (means that α approaches 1 from the left), normalization parameter $W \triangleq \sum_{i=0}^{w-1} \alpha^i$, Set $x_1 \in \mathbb{R}^n$ arbitrarily

1: **for** $t = 1, \cdots, T$ **do**
2: Predict x_t. Observe the cost function $f_t : \mathbb{R}^b \to \mathbb{R}$
3: Update $x_{t+1} = x_t - \frac{\eta}{W} \sum_{i=0}^{w-1} \alpha^i \hat{\nabla} f_{t-i}(x_{t-i})$.
4: **end for**

Guarantees upper bound

$$\frac{T}{W} \times \text{Constant}$$

according to our dynamic local regret

Quantile Loss

- Standard **mean squared error** summarizes average relationship between inputs and outputs
- The resulting forecast will be a **point forecast** which is the most likely outcome
- Many forecasting applications require richer information that a point forecast
- **Quantile loss** minimizes a sum with asymmetric penalties for overprediction and underprediction
- The quantile loss for a given quantile q between true value y and forecast value $\hat{y}$:

$$L_q(y, \hat{y}) = q \max(y - \hat{y}, 0) + (1 - q) \max(\hat{y} - y, 0)$$

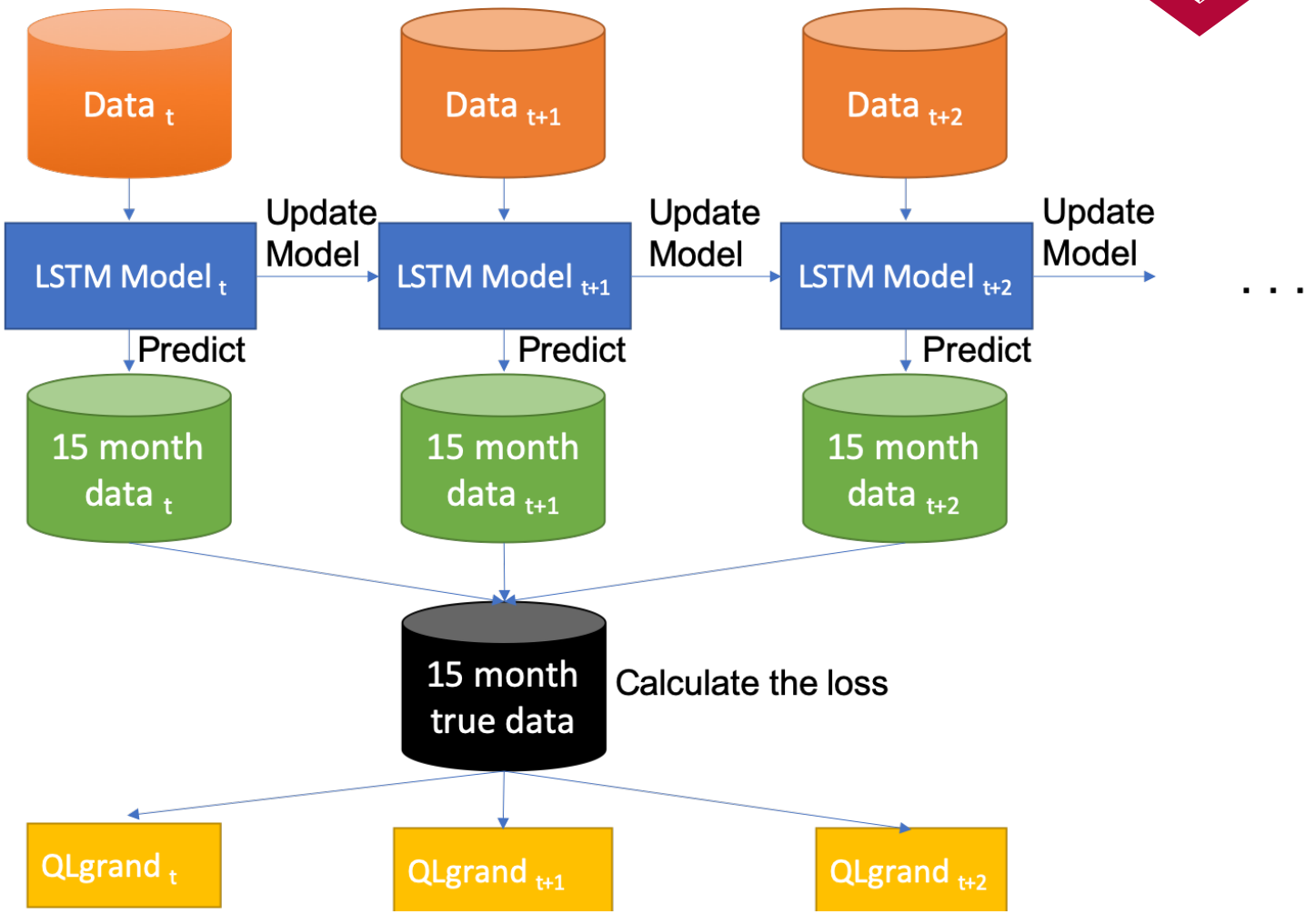
- Typically, in forecasting we may be interested in multiple quantiles and horizons where the total loss function becomes

$$\sum_t \sum_k \sum_q L_q(y_{t+k}, \hat{y}^q_{t+k})$$

t: Forecast creation date; k: horizon; q: quantiles

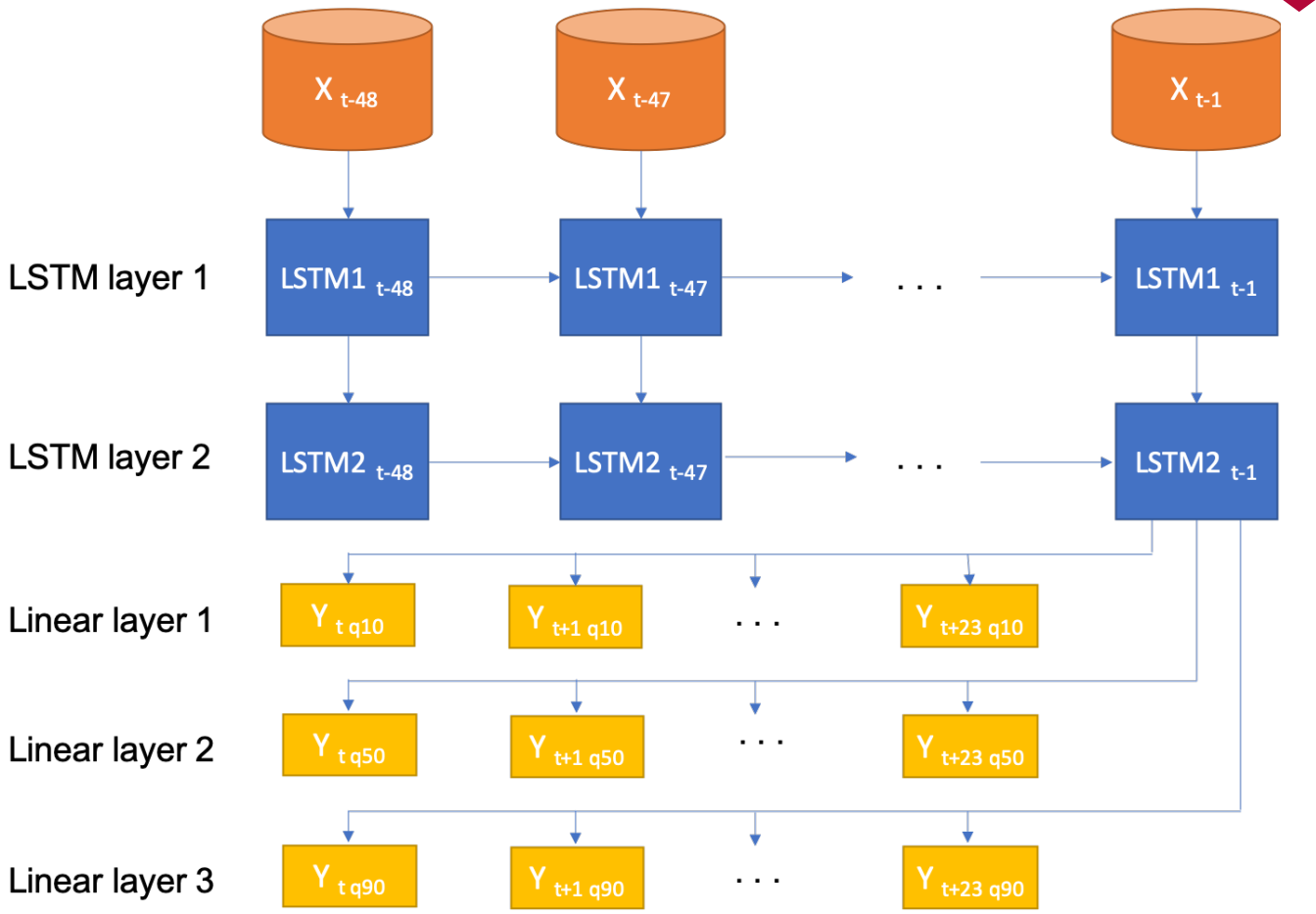
Flowchart of Our Experiments

- Public GEFCom2014 data [Barta et al. 2017]
- Hourly Electric Load values:
  - **Training**: January 2005 – September 2010
  - **Test**: October 2010 – December 2012
- **Metric**: Average quantile loss of 15 months in test data



LSTM Model

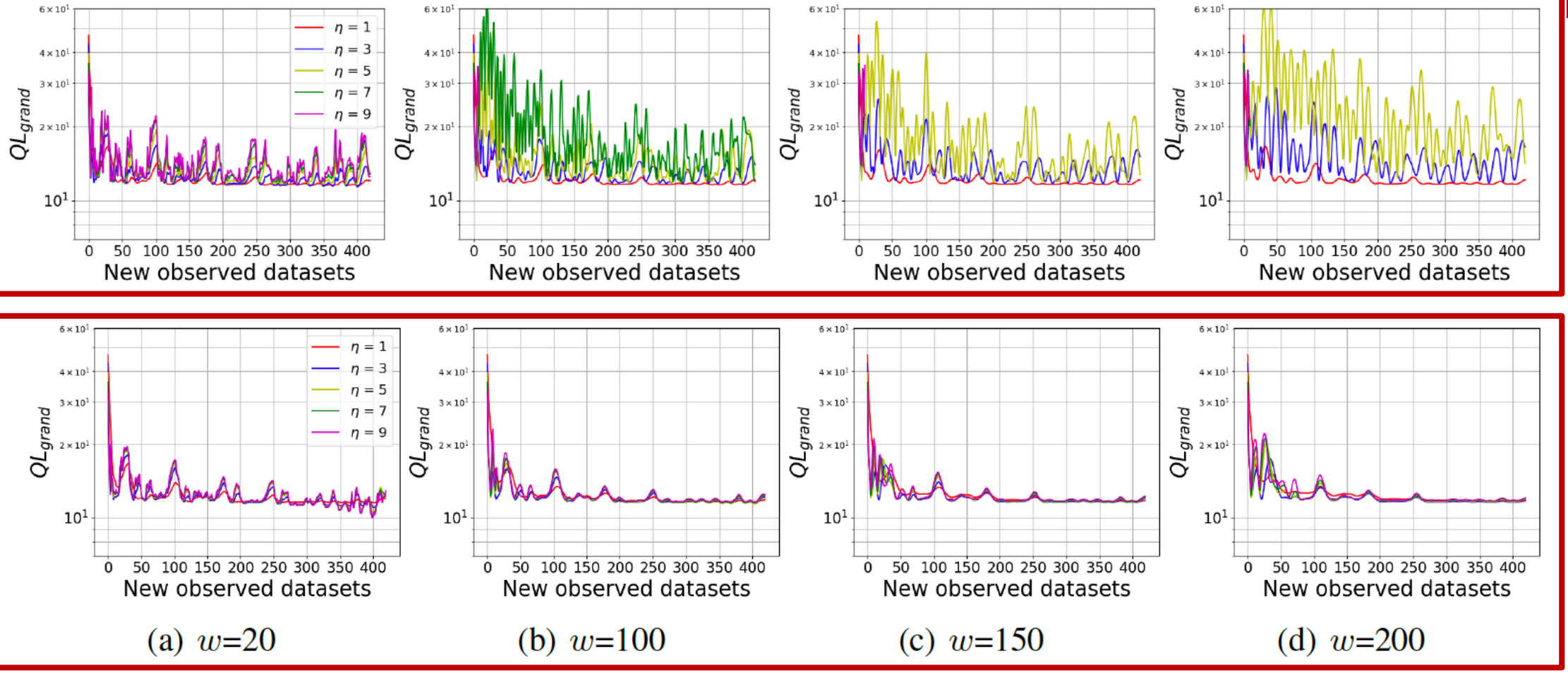
- LSTMs are special kind of RNN [Hochreiter and Schmidhuber, 1997]
- Multi-step LSTM is used to forecast multiple quantiles
- **Metric**: Average quantile loss of 15 months in test data



Experimental Results – Stability against window size

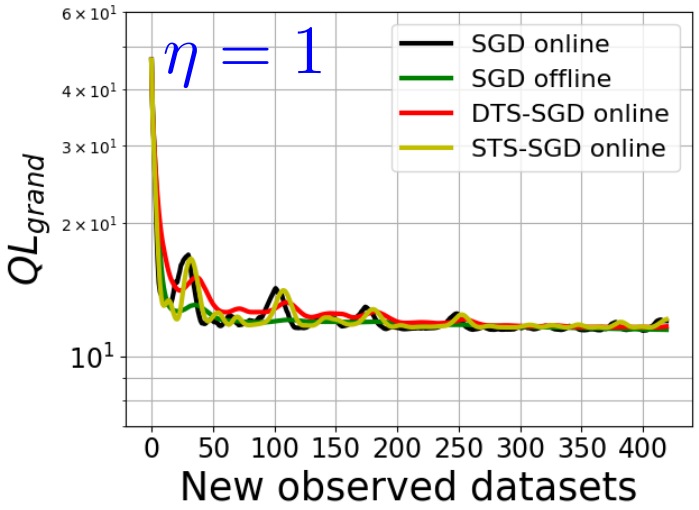
STS-SGD (Hazan et al. 2017)

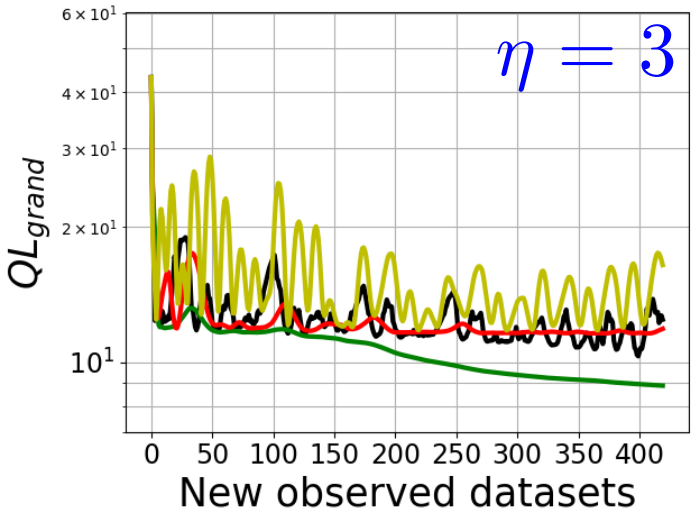
DTS-SGD (ours)

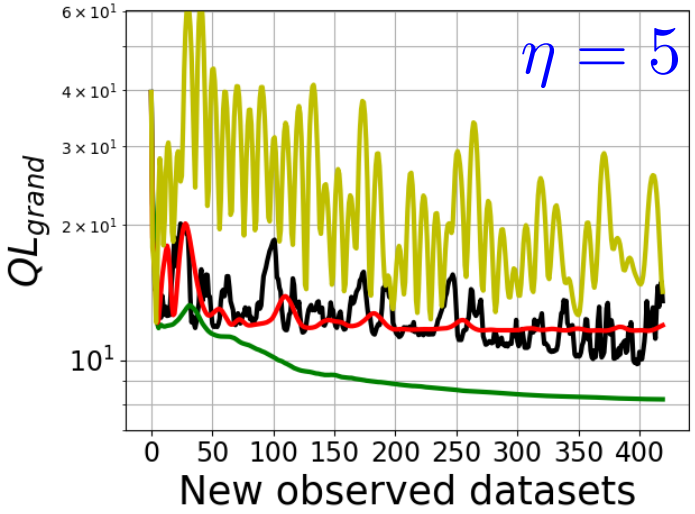


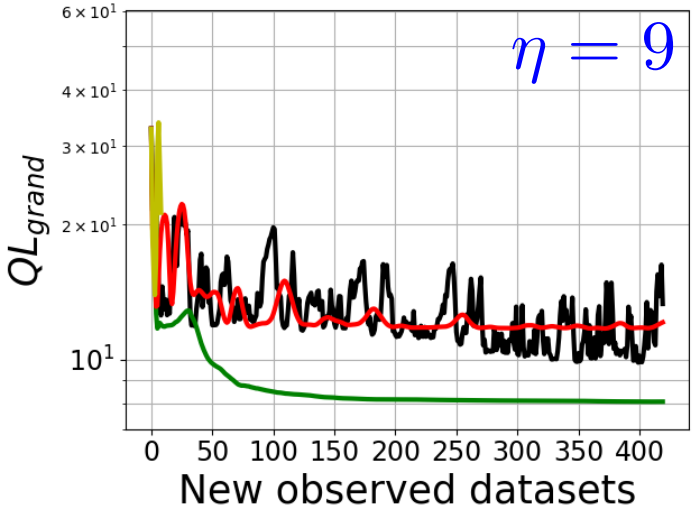
- The learning curves for DTS-SGD stay more stable than STS-SGD against different window sizes and learning rates.

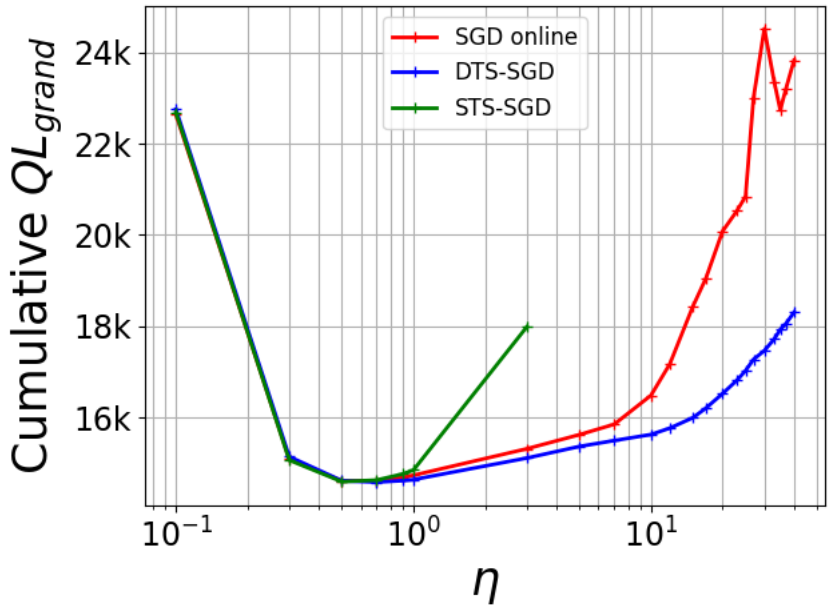
Experimental Results – Stability against learning rate







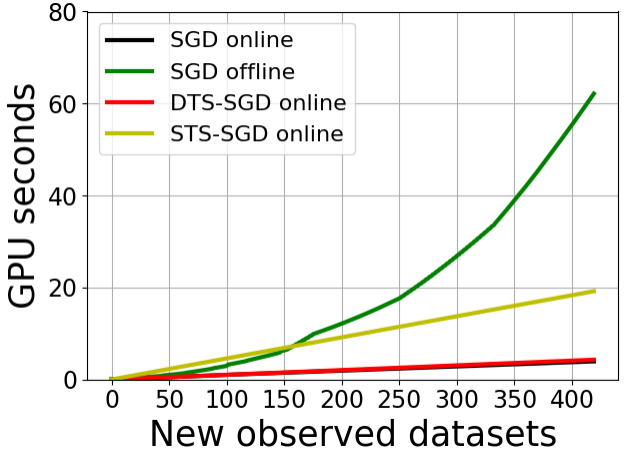




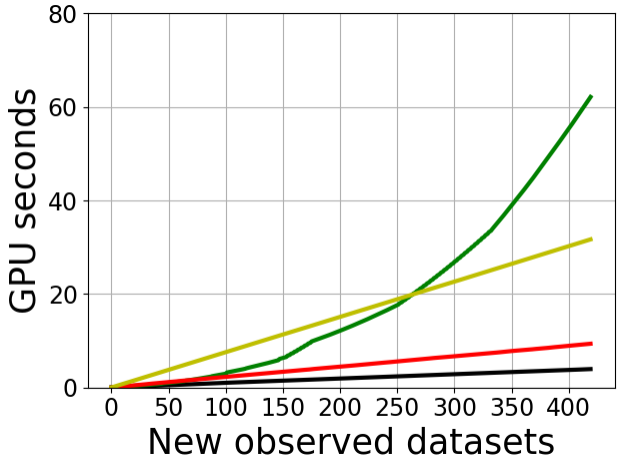
- Our DTS-SGD is less sensitive to η than SGD online and STS-SGD and performs well for a wider range of η.

Experimental Results – Computation Time

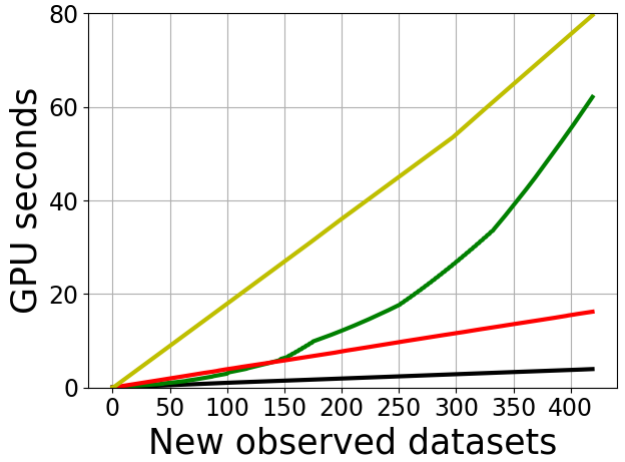
$w = 10$



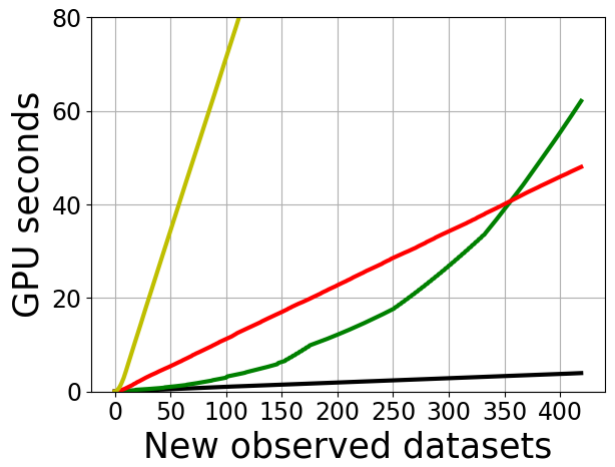
$w = 20$



$w = 50$



$w = 200$



- Computation time for STS-SGD and DTS-SGD increases as w increases.
- Our DTS-SGD is more efficient than the SGD offline even for large w.

Conclusion

- We introduced a novel local regret for forecasting problems with non-convex models
- We introduced an update rule for our regret: dynamic exponentially time-smoothed SGD update
- Our update rule is sublinear according our proposed regret
- Our experimental results show that our PTS-SGD is
  - not sensitive to the learning rate
  - computationally efficient

Thank You!

www.sergulaydore.com