

# Online Tensor Learning: Computational and Statistical Trade-offs, Adaptivity and Optimal Regret

Jingyang Li<sup>1</sup>, Jian-Feng Cai<sup>2</sup>, Yang Chen<sup>3</sup>, Dong Xia<sup>2</sup>

<sup>1</sup> Fudan University <sup>2</sup> Hong Kong University of Science and Technology <sup>3</sup> University of Michigan

Selected for the Annals of Statistics journal-to-conference track at ICML 2026

## 1. Motivation

- Tensor data are high-dimensional multi-way arrays: user-item-time ratings, spatial grids evolving over time, image stacks, hyperspectral cubes, and space-weather measurements all have natural tensor structure.
- Low-rank tensor models exploit dependence across modes, but standard offline algorithms store all observations and repeatedly scan the full dataset.
- That offline workflow becomes expensive when the ambient tensor is large, the sample stream is long, or new measurements arrive continuously.
- Online learning is attractive because each new observation can update the predictor immediately, enabling real-time tracking with one-pass computation and limited memory.
- The goal is not only fast updates: the online method should also retain sharp statistical accuracy and cumulative regret guarantees.

## 2. Online Low-Rank Tensor Learning Model

$$D_t = (X_t, Y_t), \quad Y_t \approx \langle X_t, T^* \rangle, \quad T^* \in \mathcal{M}_r.$$

- $\mathcal{M}_r$  is the manifold of tensors with low Tucker rank  $r$ , meaning that the signal can be represented by a compact core and low-dimensional factor spaces along each tensor mode.
- Low Tucker rank reduces the effective number of parameters from the full ambient tensor dimension to a much smaller degree of freedom dof.
- This structure makes learning feasible in high dimensions while preserving interactions across modes.
- The same online framework covers tensor linear regression, tensor GLMs, noisy tensor completion, and binary tensor learning.
- At time  $t$ , the learner predicts with  $T_t$ , observes  $D_t$ , and updates the estimate before the next sample arrives.

## 3. oRGrad Algorithm

$$T_{t+1} = \text{Retr}_r(T_t - \eta P_{T_t} \nabla \ell(T_t, D_t)).$$



- Compute a one-sample gradient using only the current data point  $D_t$ .
- Project the gradient onto the tangent space at  $T_t$  so the update respects the local low-rank geometry.
- Retract the trial point back to  $\mathcal{M}_r$ , typically through a rank- $r$  tensor approximation.
- The observation is used once and then discarded; no replay buffer or repeated pass over the data is required.

### Step Purpose in the online update

Step	Purpose in the online update
Gradient	Uses the newest sample to move toward lower prediction loss.
Projection	Removes directions incompatible with the current low-rank tangent space.
Retraction	Compresses the tentative update back to Tucker rank $r$ .

### One-pass computational profile

oRGrad stores the current low-rank iterate and updates it immediately. The method avoids full-dataset storage, repeated scans, and batch recomputation, while still using the geometry of  $\mathcal{M}_r$  to keep the estimator low-dimensional.

$$\bar{Y}_t = \langle X_t, T_t \rangle \rightarrow \ell(T_t, D_t) \rightarrow T_{t+1}, \quad \text{memory driven by dof.}$$

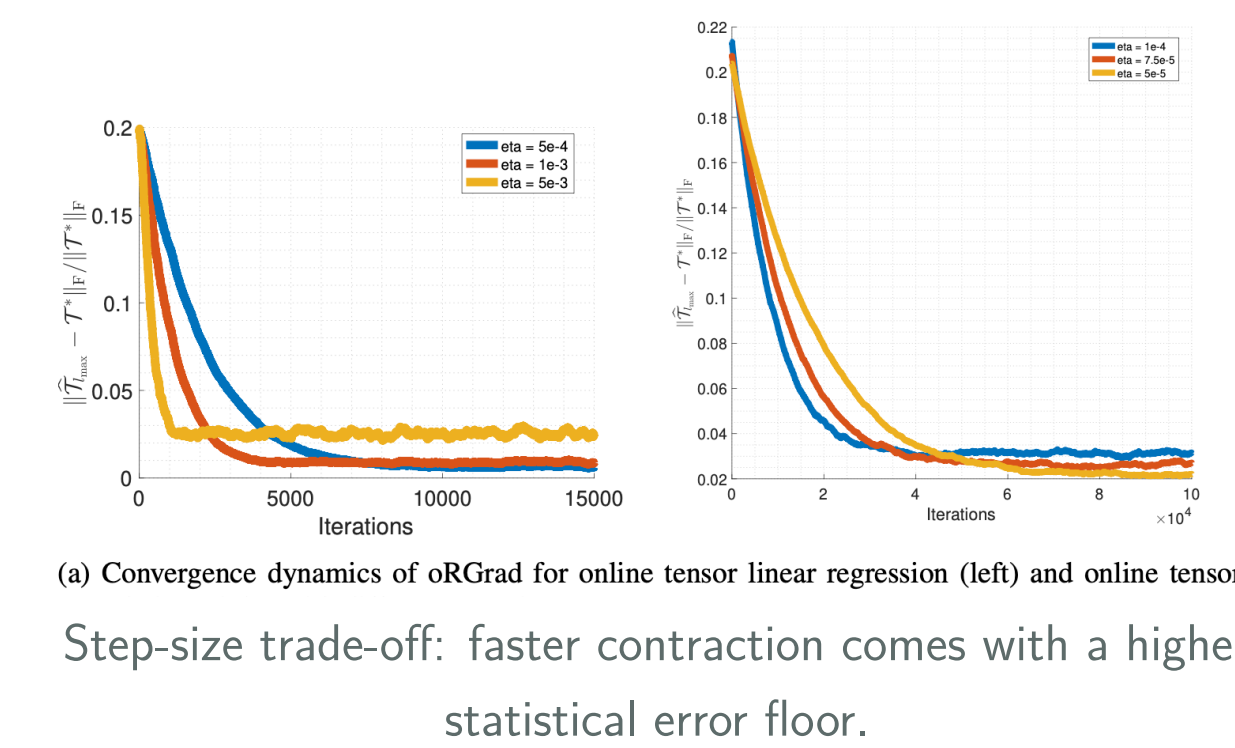
- The update loop produces a fresh predictor after every observation, which is essential when the target process drifts over time.
- The low-rank representation keeps the online state compact even when the ambient tensor dimension is very large.

## 4. Statistical-Computational Trade-off

### Informal tracking bound

$$\|T_t - T^*\|_F^2 \lesssim (1 - \eta)^t \|T_0 - T^*\|_F^2 + \eta \text{dof } \sigma^2.$$

- The first term is the optimization transient: larger  $\eta$  contracts the initial error faster.
- The second term is the stochastic error floor: larger  $\eta$  also accumulates more gradient noise.
- Smaller steps reduce the final statistical error, but they require more iterations to forget the initialization.
- If the horizon  $T$  is known, choosing  $\eta \asymp T^{-1}$  gives the statistically optimal final error up to logarithmic factors.

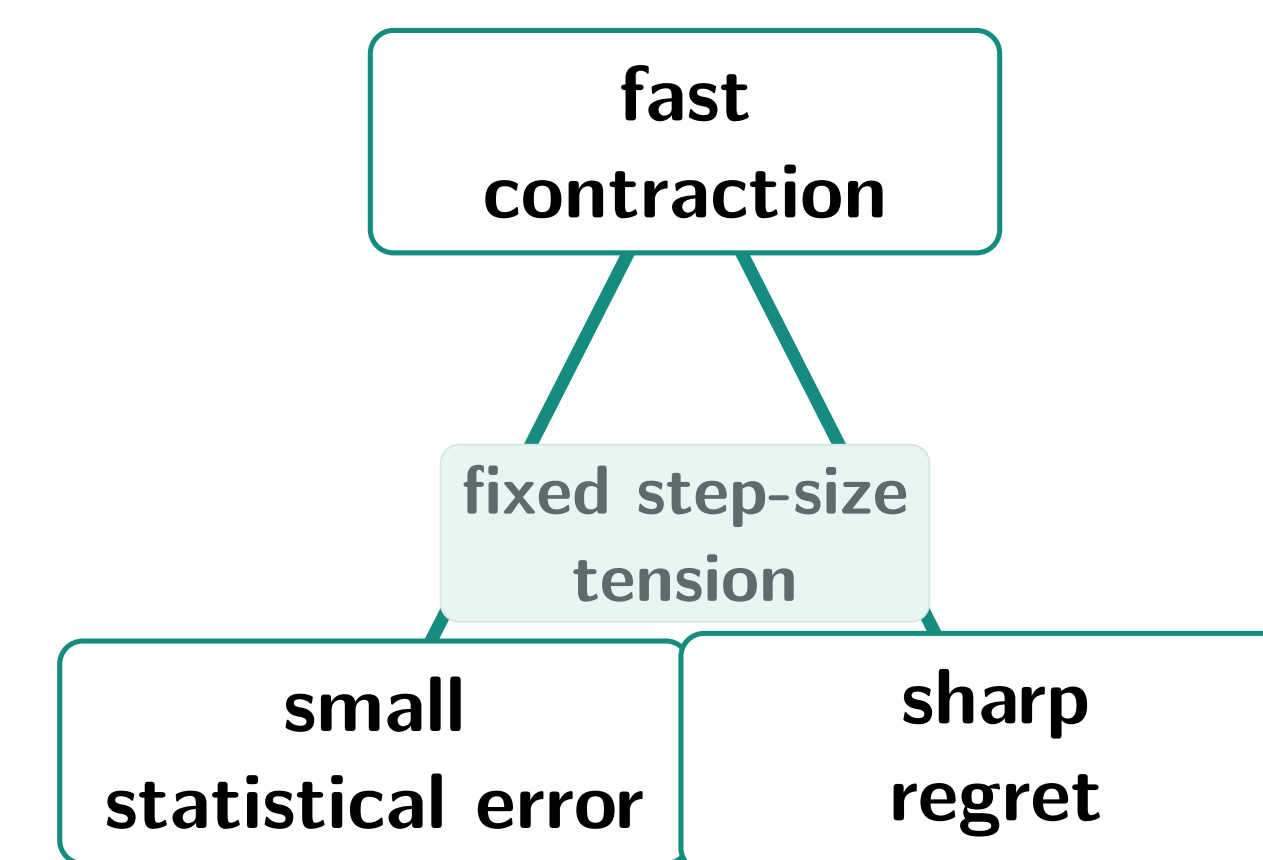


## 5. Regret Trilemma

### Fixed-step regret bound

$$R_T \lesssim \eta^{-1} \|T_0 - T^*\|_F^2 + \eta T \text{dof } \sigma^2.$$

- Statistical optimum:  $\eta_{\text{stat}} \asymp T^{-1}$ .
- Regret optimum:  $\eta_{\text{regret}} \asymp T^{-1/2}$ .
- The statistical choice cools the noise enough to optimize terminal error, but it is too conservative for cumulative online performance.
- The regret choice balances early movement and cumulative noise, but leaves a larger final statistical error.
- This mismatch is the central optimization-statistics-online learning tension.



## 6. Adaptive-oRGrad

**Idea.** Use a phase schedule that starts with a large step size for rapid contraction, then gradually decreases the step size to suppress the statistical error floor. The phase lengths expand so later, more accurate phases receive enough samples.

$$\eta_k = 2^{-k} \eta_0, \quad \text{phase length} \asymp 2^k t_0.$$

### Adaptive guarantee

## Adaptive-oRGrad achieves statistical optimality

$$R_T = O(\log T)$$

without knowing the horizon.

- Early phases behave like an optimization method: move quickly toward the low-rank signal.
- Later phases behave like a statistical estimator: average out noise with smaller updates.
- The horizon-free schedule adapts automatically to streams of unknown length.

## 7. Why Online Helps Tensor Completion

### Method Samples Entry-wise error?

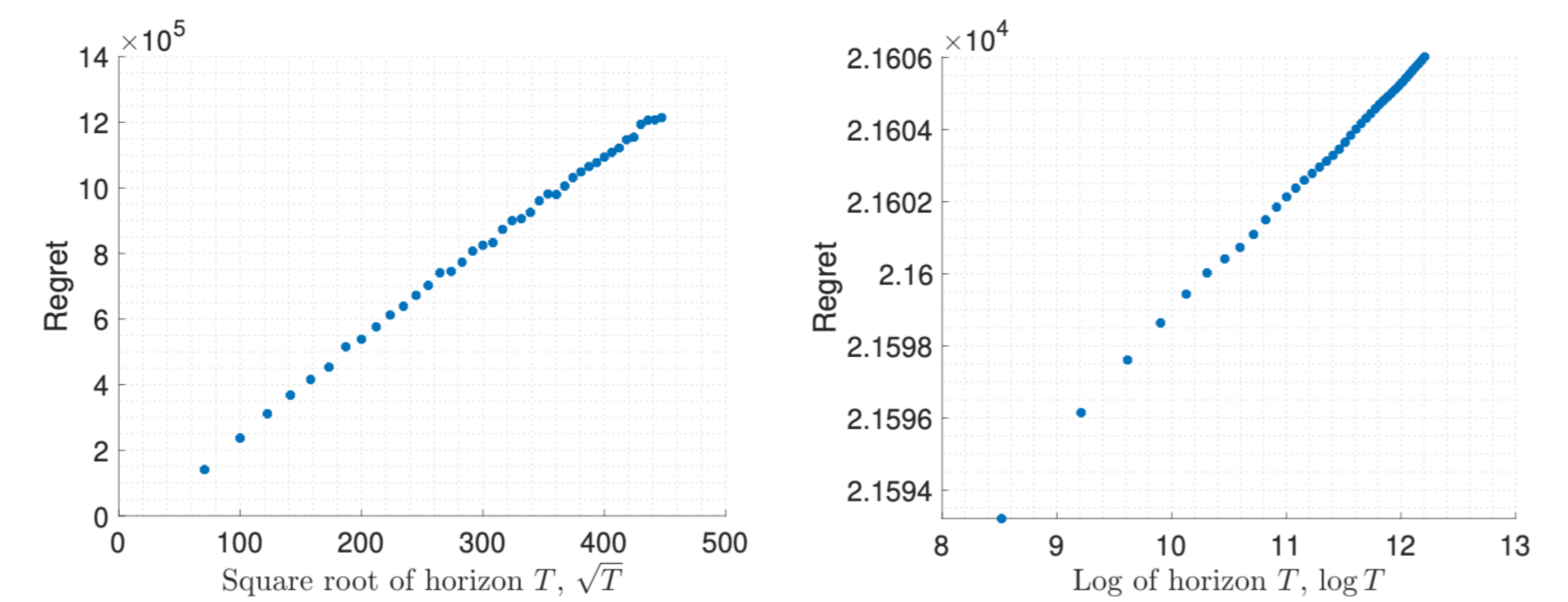
Offline methods	$\bar{O}_m(d^{m/2})$	difficult / often no
oRGrad	$\bar{O}_m(d)$	yes

### Completion message

Online geometry gives near-linear sample scaling and entry-wise control, so the method can quantify errors at individual tensor locations.

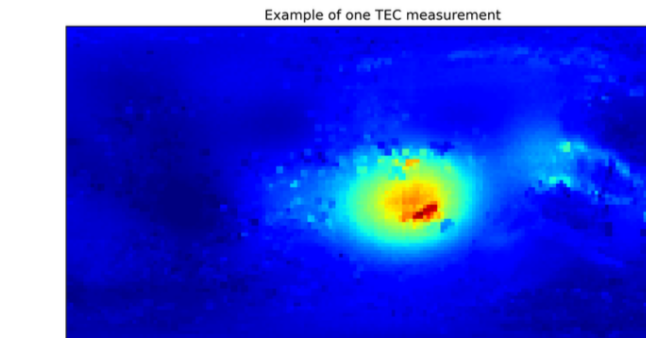
- In tensor completion, incoherence controls whether a low-rank tensor can be recovered from sampled entries.
- Online dynamics help preserve incoherence across updates instead of requiring repeated trimming or projection tricks.
- oRGrad achieves near-linear sample scaling in  $d$  and yields sharp entry-wise error bounds.
- Entry-wise control is useful for inference: it quantifies uncertainty at individual tensor locations, not only global Frobenius error.

## 8. Simulation: Regret Scaling

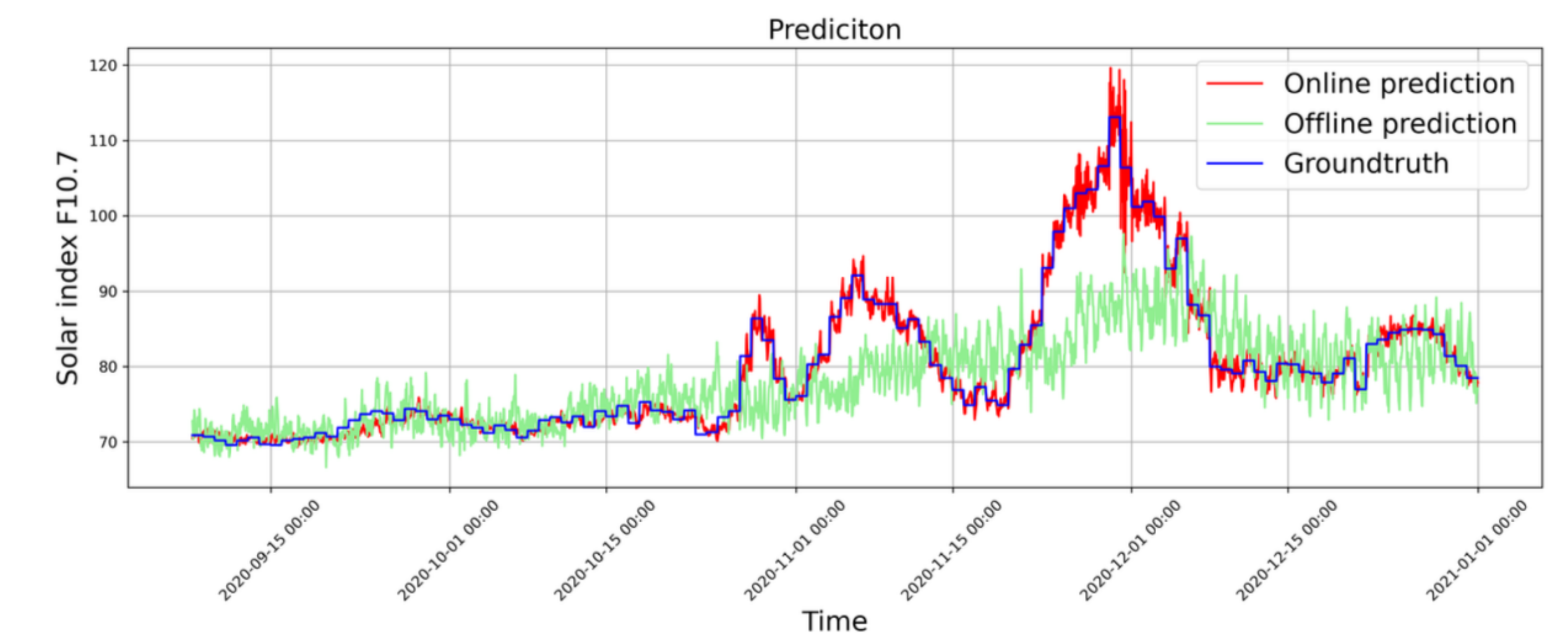


- The left panel follows the square-root horizon, matching the fixed-step regret trade-off.
- The right panel is nearly linear in  $\log T$ , illustrating the adaptive schedule's sharp regret behavior.

## 9. Real Data: Solar F10.7 Prediction



TEC tensor covariate: one spatio-temporal ionospheric measurement used as online side information.



Real-time prediction of solar F10.7: online RGrad tracks rapid changes more accurately than the offline baseline.

- The online curve responds to rapid solar activity changes, while the offline baseline is smoother and delayed.
- This is the practical benefit of updating with the newest tensor covariate as soon as it arrives.

## 10. Quantitative Comparison

### Method Relative error Correlation

Online RGrad	0.018	0.988
Offline RGrad	0.085	0.724

### Observed gain

$$\frac{0.085}{0.018} \approx 4.7 \times \quad 0.988 - 0.724 = 0.264$$

Online updates improve both scale-sensitive error and correlation with the recorded F10.7 index.

### Takeaways

**Why this matters for ML:** This work connects nonconvex low-rank optimization, streaming prediction, and regret guarantees in a unified tensor learning framework.

- oRGrad is a one-pass Riemannian method for online low-rank tensor learning.
- Fixed step sizes reveal a fundamental trade-off between optimization speed, statistical accuracy, and regret.
- Adaptive-oRGrad achieves statistical optimality and optimal  $O(\log T)$  regret without knowing the horizon.
- The method improves real-time tensor prediction in space-weather data.