## APPENDIX

### A. Reproducibility and detailed experimental Setup

**Implementation Details.** To reproduce results including results of baselines, we present implementation details as follows:

- All methods are implemented in C11, a C standard revision, language with Python2.7 as a wrapper. The experiments are executed in a cluster with 5 nodes. Each node has 28 CPUs and 250Gb memory. For each method, we only use 1 CPU at a time.
- The random seeds for all trials are np.random.seed(17), which makes results of AUC scores and sparse ratios reproducible.
- Critical operations of all baseline methods are scale product $c \cdot \boldsymbol{x}$ and the inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\top \boldsymbol{y}$, which are calculated by cblas_dscal() and cblas_ddot() respectively. These two functions are provided by OpenBLAS [42] [5], an optimized BLAS library.
- For SPAM-$\ell^1$, SPAM-$\ell^2$, and SPAM-$\ell^1/\ell^2$, since they need to estimate $\widehat{p}_T, a(\boldsymbol{w}_t), b(\boldsymbol{w}_t)$, and $\alpha(\boldsymbol{w}_t)$, in our experiments, we estimate them by using $p_t, a_t(\boldsymbol{w}_t), b_t(\boldsymbol{w}_t)$, and $\alpha_t(\boldsymbol{w}_t)$ defined in (8).
- For FSAUC, there exists a projection step onto a $\ell^1$-norm ball. The projection used in the original implementation is the method proposed in [43]. However, there exists a much faster version of $\ell^1$-ball projection [44] as claimed has $\mathcal{O}(d)$ run time in practice [6].

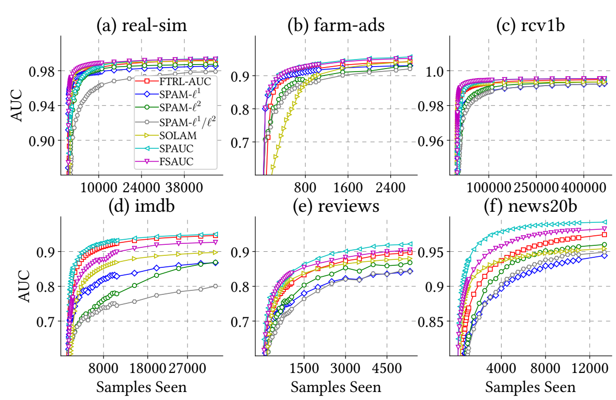


Fig. 8: Convergence rate with respect to the number of training samples seen

**Parameter Tuning.** We list parameter tuning of all methods including the baseline methods as follows:

- FTRL-AUC, it has two parameters. The $\ell^1$-regularization parameter $\lambda$ which is from a sufficient large range $\{10^{-8}, 10^{-7}, \ldots, 10^{-3}, 0.005, 0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 1.0, 3.0, 5.0\}$, and the initial learning rate $\gamma$ is from $\{10^{-5}, 5 \cdot 10^{-5}, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 5.0\}$.

[5]https://github.com/xianyi/OpenBLAS with version 0.3.1 (Accessed in February 2020)

[6]The C version code can be download from https://lcondat.github.io/download/condat_l1ballproj.c(AccessedinFebruary2020).

- SPAM-$\ell^1$ has two parameters. The initial learning rate $\xi$ is from $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$. The $\ell^1$-regularization parameter is the same as FTRL-AUC's.
- SPAM-$\ell^2$ has two parameters. The initial learning rate $\xi$ is from $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$. The $\ell^2$-regularization parameter is the same as $\lambda$.
- SPAM-$\ell^1/\ell^2$ has 3 main parameters. To avoid large cross-validation time, the parameter $\xi$ and $\lambda^2$ is used by the parameter tuned from SPAM-$\ell^2$. We only tune the $\ell^1$ parameter $\lambda_1$ which is the same $\lambda$.
- FSAUC has 2 parameters. The $\ell^1$-norm ball which is from $\{10^{-1}, 10^0, \ldots, 10^5\}$. The corresponding initial learning rate is from $\{2^{-10}, 2^{-9}, 2^{-8}, \ldots, 2^8, 2^9, 2^{10}\}$ as suggested in [18].
- SOLAM has two parameters. The $\ell^2$-norm ball diameter which is from $\{10^{-1}, 10^0, \ldots, 10^5\}$ and the initial learning rate $\xi \in \{1.0, 10.0, 19.0, 28.0, \ldots, 100.0\}$ as suggested in [16].
- SPAUC has two parameters. The initial learning rate parameter is from $\{10^{-7.0}, 10^{-6.5}, 10^{-6.0}, \ldots, 10^{-2.5}\}$. Since we use the $\ell^1$-regularization and it is the same $\lambda$.
- FTRL-PRO has the same parameter tuning strategy as FTRL-AUC.
- RDA-$\ell^1$ has three parameters. It has an initial learning rate from the range $\{10.0, 50.0, 100.0, 500.0, 1000.0, 5000.0\}$. The sparsity-enhancing parameter is from $\{0.0, 0.005\}$, where 0.0 corresponding to non-enhancing sparsity. The $\lambda$ is the same as FTRL-AUC's.
- ADAGRAD has three parameters. The $\epsilon$ is fixed to $10^{-8}$ to avoid the divided by zero error. The learning rate parameter is from $\{0.001, 0.01, 0.1, 1.0, 10.0, 50.0, 100.0, 500.0, 1000.0, 5000.0\}$ while $\lambda$ is the same as others.

### B. More Results

We first provide the experimental details for Figure 2. We fix the initial learning rate $\gamma = 1.0$ and the sparsity parameter $\lambda = 0.5$. The convergence curves illustrate in Figure 2 and 9 are the AUC scores averaged on 10 trials.

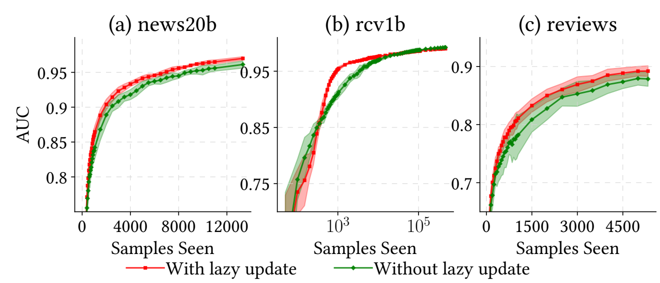


Fig. 9: Comparison of with and without "lazy update"

The convergence curve with respect to the number training samples seen are illustrated in Figure 8. In general, the performance of FTRL-AUC on the convergence is better than SPAM-based. Figure 10 illustrates the convergence curve as a function the number of training samples seen for the datasets of imbalance ratio $T_+/T = 0.05$. Figure 11 illustrate the sparse ratio and corresponding AUC scores as a function of the parameter $\lambda$ for the datasets of imbalance ratio $T_+/T = 0.05$.

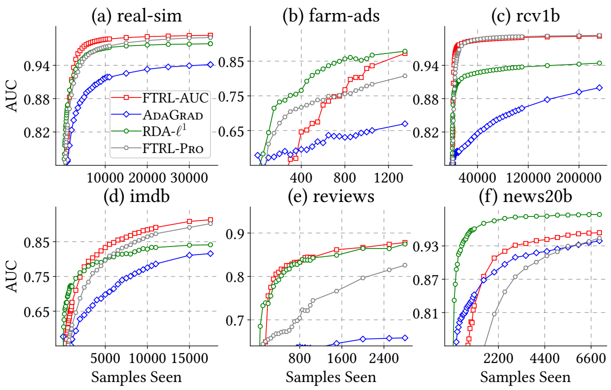


Fig. 10: Convergence curve with respect to the number of training samples seen

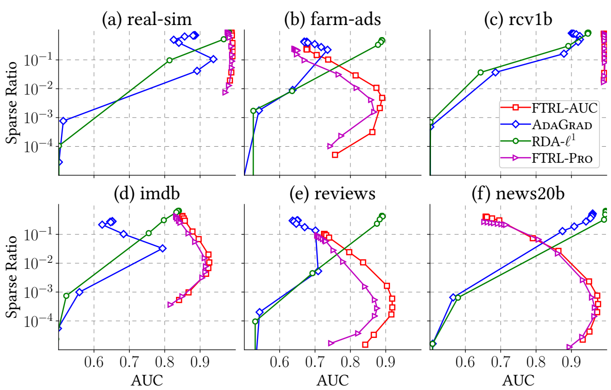


Fig. 11: Sparse Ratio as a function of the AUC score.