Dear referees and chairs,

We would like to thank all referees for their close reading of the manuscript.

Reviewer # 1: We tried to further explain the connection to Neu and Zhivotovskiy in Appendix E, who consider the Online Classification with abstention setting. Specifically in lines 488 to 492 we try to explain the connection between our ideas and the results in the Online Classification with abstention setting. However, we see that it is not clear how the techniques of Neu and Zhivotovskiy relate to Lemma 3 and we will try to clear this up in the final version of the paper. To clarify, a simplified explanation of the connection to Neu and Zhivotovskiy is that they derive the particular learning rate for standard Exponential Weights such that the abstention gap in Lemma 3 is bounded by zero.

Reviewer # 2: There is nothing special in particular about the multiclass setting other than that it becomes more difficult to just guess the correct answer, which is the reason why the number of classes K shows up in the regret bounds of Gaptron. The first reason to include the multiclass setting is to provide a complete picture of what is possible with our new technique. The second reason to include the multiclass setting is the bandit setting. Gaptron does not reduce to a standard algorithm for any K, except when a(W, x) = 0 (see also the discussion in lines 115-120). Note that in the full information setting the regret of Gaptron does not grow as the number of rounds increases, which is a very useful property since in most applications the number of rounds is very large. As for the discussion in lines 175-181, in the worst case, the regret bound of the Perceptron is only better when the total number of rounds T is smaller than K^2 as the regret of the Perceptron is $O(\sqrt{T})$ and the regret of Gaptron is $O(K)$. The simplest setting where Gaptron beats other algorithms is the bandit setting, where the regret of Gaptron is a factor $\sqrt{d}$ smaller than other algorithms, which also have a higher runtime than Gaptron.

For an overview of the different bounds we provided Table 1. The parameters can be found in Section 2. Importantly, Gaptron often is on par with, if not better than slower algorithms such as ONS.

Reviewer # 3: Like Reviewer # 1 states we think that the fact that the regret bound of Gaptron does not explicitly depends on the dimension of the feature vector is a strength rather than a weakness. Indeed, the regret bounds do depend on the norm of the feature vector, which means that the regret bound implicitly depends on the dimension of the feature vector. However, the norm of the feature vector is often the preferred measure of the size of the feature vector, especially when, for example, the elements of the feature vectors are scaled to [-1, 1]. Note that the regret of several other algorithms depends on both the norm of the feature vector and on the dimension of the feature vector, for example the algorithm of Foster and Krishnamurthy (2018).

To clarify, the lower bound of Hazan et al (2014) holds for pure logistic loss regret, which is to say that the learner also suffers logistic loss. This lower bound does not apply to our setting, where the learner suffers the zero-one loss. Unfortunately, we are only aware of lower bounds in the full information and separable setting, for example Theorem 18 by Foster et. al. (2018). This lower bound shows that with logistic loss the regret bound of Gaptron is tight up to logarithmic factors.

We would like to point out that our technique to prove the regret of Gaptron is also novel, not just the algorithm. We hope that this allows future researchers to exploit a similar technique to provide efficient algorithms for other settings.

Reviewer # 4: To clarify, in the full information setting the algorithms with which we compare are deterministic. In the bandit setting all algorithms are randomized. One of the possible future directions is to derive high probability regret bounds to better understand the variance of Gaptron. In the final version, we will clarify that the algorithms with which we compare are deterministic.

Thank you for the suggestion regarding y_t in the bandit setting.