We would like to thank all reviewers for their careful reviews and insightful comments.

Reviewer 2: It seems that the algorithm needs to have a pretty large tree. Will it increase in exponential order of T or other order?

The space grows only linearly in time because the algorithm constructs the tree dynamically, adding a new path of size $\mathcal{O}(D)$ in each round. In particular, the algorithm never allocates the entire tree, but only the paths corresponding to active experts.

The algorithm can only adapt to one of the mentioned local regularities, and it really just depends on which radius tuning function is used.

This is correct, and we will modify the abstract to remove any ambiguity. Still, we believe that being able to adapt to different local regularities just by supplying a specific radius tuning function is interesting. Note also that, even though we are not aware of any online nonparametric algorithms with simultaneous adaptivity (not even to global regularities), it is certainly possible to combine algorithms with different types of adaptivity using online aggregation techniques such as prediction with expert advice.

The paper omitted many details, which can be found in the appendix. In the Algorithm 1, the paper omitted the update for inactive experts after line 10.

This is not an omission: the algorithm does not require any updates of inactive experts, only experts along the path are updated, which makes it computationally attractive. This is one of the beauties of the context tree weighting method, originally described in the information theory literature (see [29]).

We apologize that some important material had to be placed in the appendix, we will certainly bring back the important details you mention to the main body in the revised version of the paper.

The local online predictor for each node is using Follow-the-Leader rule. It kind of makes me confused on how to do that with only one node information.

It would have been more appropriate to call it "follow-the-local-leader". Indeed, each follow-the-leader instance hosted in a ball learns only over the subsequence of examples that fall in that ball.

Improvement in the local dimension case is doubtful — the oblivious adversary can generate the instance sequence that does not have lower local dimension.

Note that our algorithms are deterministic (no distinction between oblivious and nonoblivious adversaries), and our regret bounds hold for any sequence. We do not claim that our bound is always better. In fact, when the local dimension equals the global dimension, we merely do not lose anything in the exponent as our bound becomes of order $\widetilde{O}(T^{\frac{d}{1+d}})$.

The paper assumes that the instance space is the unit ball. What about the case when the instance space is not necessarily a unit ball? Can the algorithm still work in this case? What changes should we make?

We gain a factor of C^d in the regret (note that this is not improvable), where C is the radius of an instance space. At the same time we have to know C since it will appear in the radius tuning function (see, e.g. [13]).

Reviewer 1: It seems that we need to specify the hierarchical levels (e.g Lipschitz constants, metric dimensionality) which sound non-trivial.

The hierarchical levels are hyperparameters determining the reference or comparator class of the nonparametric problem (which in turn defines the inductive bias). Note that the algorithm and its analysis work for *any* choice of these hyperparameters. By choosing specific values, the user controls the trade-off between size of the reference class on one side and the growth of the regret bound and space/time requirements on the other side.

It is not entirely clear whether the additional (algorithm) complexity and input parameters justify the gains, and if so, on what problems.

Note that the overhead in running time is logarithmic in the size of the tree, since at each round we only query and update experts along the path of an instance. As for the space requirements, the tree grows only linearly in time as the algorithm constructs the tree dynamically (see response to R2).

Reviewer 3: The techniques are currently only analyzed for regression with square loss and classification with absolute loss. It would have been nice if the algorithms can be applied beyond these two settings.

Our proofs can be easily extended to any exp-concave losses, as these do not require any tuning of learning rates in local FTL predictors. We also believe that it is also possible to extend our approach to any convex loss by using a parameter-free local learner such as Squint (Koolen and van Erven, COLT 2015).