Thank you reviewers.

R1: *«Setting Questions»* : The task sequence is arbitrary and is chosen by the *environment*, which may be adversarial (lines 83-86). Thus the environment might present the instance sequence for different tasks in a given order. Alternatively the active task on trial t could be determined by the environment at random, etc. You ask *" ... what is the difference. ... switching constraints ... memory"*: The bound is completely independent of the task sequence.

R2: *"...algorithms require some prior knowledge ...", "parameter-free design"* : See lines 293-295. Automatic parameter tuning has been considered since the advent of online learning, using e.g. "doubling trick", "adaptive parameter rates" and "mixtures over parameters"; there is no reason to suppose such methods would not apply here.

" further survey .. contextual bandit .." : Thank you for the references. We will include them in the manuscript. Note, however, **none of them consider long term memory, the key theoretical concept of this paper.**

"Consider multitask... compare their regret upper bounds..individually" : These comparisons follow directly from the theorem statements. E.g., in Thm. 3, we would pay the learner complexity term $\sum_{h \in m(\boldsymbol{h}^*)} \|h\|_K^2 X_K^2$ on a per-task basis. Finally this gain in the multitask case is not just in the upper-bound but it is also reflected by the lower-bound (see Prop. 4).

R3: Thank you for your comments.

R4: *" In the online multitask expert setting, I would suggest providing a more rigorous comparison with the related work. As indicated above, it seems that the present framework can be cast as a problem already investigated in [1]"* : We provide a rigorous comparison. See lines 167-169 for a discussion that includes [1] where we say about their model "a special case of ours where each task is associated only with a single hypothesis, i.e., no internal switching within a task." Your proposed reduction would not even enable vanilla switching in a single task setting! Unless the environment sent a signal to switch comparators, which trivializes switching altogether.

Further, comparing the framework of [1] to this paper. In [1] they are interested in a model where each task is associated with single comparator, i.e., each task in our Figure 1 would have a *single* color, **not** a sequence of colors. I suspect that your confusion comes from a misinterpretation of [1, Corollary 1]. When the authors of [1] use the word "switch" they do not mean a switch in the "comparator sequence," but a switch in the "task-query sequence" (our vector $\boldsymbol{\ell}$) i.e, if the environment queries task 1 on trial 1 and then queries task 5 on trial 2, then their regret bound pays a $\ln m$ for each such "switch." For our bound such switches of task vector incur NO increase in regret, i.e., we strictly improve.

> *"In the online multitask linear setting - and more generally the RKHS setting (Section 4), there is something wrong about the results. ... In this setting, it is well-known that for linear functions of dimension $d \geq 2$ the Littlestone dimension is infinite (see e.g. [52]). So, we cannot hope to find sublinear regret bounds for linear functions with respect to the zero-one loss!"*

It seems that the source of your confusion is that you did not understand what is an RKHS $\boldsymbol{\mathcal{H}_K}$ or the definition of $\boldsymbol{\mathcal{H}_K^{(x)}}$ (recalling lines 58-61).

> Given a reproducing kernel $K : \mathcal{X} \times \mathcal{X} \to \Re$ we denote the induced norm of the reproducing kernel *Hilbert* space (RKHS) $\boldsymbol{\mathcal{H}_K}$ as $\|\cdot\|_K$ (for details on RKHS see [5]). Given an instance sequence $\boldsymbol{x} := (x_1, \dots, x_T)$, we let $\boldsymbol{\mathcal{H}_K^{(x)}} := \{h \in \mathcal{H}_K : h(x_t) \in \{-1, 1\}, \forall t \in [T]\}$ denote the functions in $\mathcal{H}_K$ that are binary-valued on the sequence.

Here $\boldsymbol{\mathcal{H}_K}$ is real *Hilbert* space i.e., a set of *real-valued* functions endowed with an inner product and thus a norm (which our bounds are in terms of). In the next sentence we define $\boldsymbol{\mathcal{H}_K^{(x)}} \subset \boldsymbol{\mathcal{H}_K}$ as the subset of functions that are interpolants on the instance sequence $\boldsymbol{x}$ (observe that if $h \in \mathcal{H}_K$ in the typical case $\operatorname{sign}(h) \notin \mathcal{H}_K$). Since these are linear interpolants not halfspaces this should alleviate your complexity concerns. We provided the reference [5] for the definition of $\boldsymbol{\mathcal{H}_K}$; we do not write out the full formal definition in the paper because of its length but we note that RKHSs have been a mainstay of statistical learning theory for at least 30 years.

You also misunderstand the implications of infinite *Littlestone* dimension for regret and mistake bounds. Yes for a Gaussian kernel the corresponding spaces have infinite **Ldim**. However, this only rules out a *uniform* bound. One may still have a *non-uniform* bound with respect e.g., to a norm of a given hyp. in the space, see for example Novikoff's Theorem where the inverse margin corresponds to the norm of the classifier. We provide a non-uniform bound (Thm 3).

"Theorem 50 in Appendix C" : Please read theorem statement (esp. line 1071 we recall "for any vector $\boldsymbol{u}$ such that $|\langle \boldsymbol{u}, \boldsymbol{x}_t \rangle| = 1$ for $t = 1, \dots, T$. "). The bound is with respect to *interpolants* not *halfspaces*. Note that [33] has a new version in Arxiv, we refer to Lemmas 19 and 20 in version 3. These are minor algebraic and probabilistic results; the only connection to matrix completion is the notation. We will amend the proof to make this clear.