We would like to thank the reviewers (R1, R2, R3, R4) for the very constructive reviews on our work, pointing out the merits and raising interesting questions to answer. We received very positively the comments on the quality of the presentation and we believe this constructive discussion will greatly improve the quality of the manuscript.

In this work, we propose a link between kernel methods and fixed random weights Recurrent Neural Networks (RNNs), quoting R3, "**an important and novel contribution in helping the community study RNNs further**". We also leverage this theoretical correspondence to accelerate Reservoir Computing using structured transforms, from $O(N^2)$ complexity to $O(N \log N)$ for the main reservoir computation of Eq. (1). Even though there is a gap between theory and practice, we argue these ideas bring significant computational savings, calling for future theoretical studies. We would like now to answer the reviewers' comments thoroughly.

@R2, @R3: **Assumptions of the theoretical study.** Our Theorem requires the assumption of resampling the weights matrices at each time, which is never done in practice. This assumption remains important to obtain a sum of i.i.d. random variables to apply Proposition 1. Nonetheless, actual implementations match this theory well and the conclusions of the theorem still bring interesting insights. For example, the convergence rate of the Mean Square Error matches the $1/N$ scaling of the theory, and we show in Fig. 3 of the appendix there is clearly no difference with and without redrawing the weights.

We recognize this theoretical study is limited and we have tried to present its limits clearly and honestly in the manuscript. To complement it, we show with direct numerical computations and practical applications the convergence of RC towards its Recurrent Kernel limit. To discuss R2's comment about the reason why we propose a theoretical study, we believe it remains essential to explain rigorously the behavior of our ML algorithms, even if conditions sometimes have to be relaxed to obtain informative results, as this paves the way for future studies.

@R1, @R2, @R3: **Broader impact.** To discuss the broader impact of the presented work, we will add to the manuscript: (1) theoretical studies to understand machine learning (ML) are important to avoid relying on black boxes, as more and more applications appear in our daily life; (2) efficient ML is necessary due to the ever-increasing power consumption required for computation.

We deeply think this work is establishing a connection between random RNNs and kernel methods that will open up future studies on this important topic in machine learning. This is the reason we have submitted this work to a conference such as NeurIPS. We will now proceed with the answers to the more technical questions:

@R1: **Kernel function of 1 or 2 variables.** $k(x, y)$ is indeed a function of two variables, and we use the simplifying notation for translation-invariant kernels $k(x - y) \equiv k(x, y)$ and rotation-invariant kernels $k(\langle x, y \rangle) \equiv k(x, y)$. This precision will be added to the manuscript.

@R2: **"How many runs for the time benchmark in Table 2?"** We did not have to average this timing benchmark as the standard deviation of this measurement is negligible on a GPU (less than 1% at $N = 1,000$), both for the matrix multiplication and inversion for each network dimension.

@R2: **"Why does RK [Recurrent Kernels] have the same number for all N?"** RK corresponds to the limit of RC when N tends to infinity, and as such, does not depend on N.

@R2: **Clarification on "forward" and "train" steps:** Since internal weights are not trained in RC, we first compute the network states with Eq. (1) (this is the "forward" step), and training is performed separately with linear regression (no "backward" step is necessary).

@R2: **Absence of conclusion.** We have chosen to summarize our results in the "Main contributions" section of the introduction for clarity. However, we will add a conclusion to discuss future lines of work in the next version.

@R2, @R4: **Additional applications.** We have chosen to focus on chaotic time series prediction, a promising yet challenging application for RC. This choice has been motivated by the substantial amount of prior works and the particular interest shown recently by the RC community as it is well said by R4 "*The focus on only chaotic time series prediction makes sense in light of the original ESN papers, even though it would be nice to see additional applications.*" While additional applications would surely be interesting, it is beyond the scope of this paper to find novel applications of RC, for space reasons (as R4 pointed out, "*The paper makes quite good use of the available space* [...] *there is nothing in the paper that should be replaced*").

@R3: **Stability and Echo-State Property for resampled random weights.** Stability and the ESP can also be described with resampled weights: one can check whether two reservoirs initialized differently converge or not to the same trajectory, provided they share the same weights even with resampling at each time. We would like to thank the reviewer for this very relevant remark and for the reference that will be added to the manuscript. We believe stability is an essential question to investigate further for RK, as this property is central in Reservoir Computing.