We thank the reviewers for their insightful comments and respond to their concerns and questions below.

Introductory remarks. Our goal was to extend CKNs with motifs gaps. For that, we had three sources of inspiration:
- [4] is the original CKN model for sequences, which introduces links between string kernels and CNNs.
- [19] derives RNNs from the recurrent structure of substring kernels, and show that these RNNs live in a RKHS.
- [35] introduces the local alignment kernel, which appears as a special case of the kernel we define.

Differences with respect to [19] and [4] (R1). RKNs and [19] are undeniably related formulations, as they both exploit the recurrent structure of substring kernels to build a sequence network. Yet, as discussed below, RKNs are grounded on different principles and operate in another functional space, which we should have better explained.

RKNs and RNNs from [19] perform different computations: In [19], a non-linear function is applied on the output of recursive equations, whereas the RKN model involves a non-linear function within the recursion, see the definition of $\mathbf{b}_j$ in (7), and also involves an orthogonalization factor in (6). These constructions lead to different functional spaces and are motivated by different principles: our function space is a subspace of a RKHS defined by (2) and the Nyström approximation, while a function constructed by [19] is an element in some other RKHS. As consequences:

- RKN admits an interpretable feature map, as a weighted mixture of distance functions centered at all k-subsequences of the sequence (Section 3.1). No such interpretation holds for [19] (see Proposition 1 in [19]).

- The orthogonalization factor makes RKNs insensitive to the scale of the starting point, and the interpretation of RKNs filters as anchor points in the Nyström method naturally leads to an efficient unsupervised learning scheme.

- Due to the fact that RKNs provide an approximate embedding for sub-sequences, we were also able to simulate max pooling in RKHSs, bridging an old gap between theory and practice in the literature of string kernels.

Finally, RKNs are related to [4] by their relying on the relaxation of a string kernel and on the Nyström approximation. While [4] starts from the mismatch kernel, RKNs relax the subtring kernel thereby extending [4] to model gaps.

Choice of the SCOP 1.67 benchmark and additional baselines and results (R1,R3). Our motivation for choosing SCOP 1.67 is its extensive use as a benchmark in the kernel literature. As a consequence, most baselines reported in Table 1 are indeed a bit old. Yet, [4] presents also more recent baselines based on CNNs (with models akin to DeepBind) that perform slightly worse than CKN-seq, showing that our approach is competitive.

Following the reviewers' comments, we have benchmarked RKNs on a more recent fold recognition dataset using the protocol of "Jie Hou et al. DeepSF: deep convolutional neural network for mapping protein sequences to folds. 2018", based on SCOP 1.75 and SCOP 2.06 (see this paper for the exact description of task, dataset, and data representation). The accuracy results are shown in the next table, stratified by prediction difficulty (family/superfamily/fold). A striking conclusion is that modeling gaps (as done by RKN) is very important for the fold prediction task—closer homologies are indeed less likely to involve gaps.

Method	♯Params	Accuracy on SCOP2.06			Level-stratified accuracy (top1/top5/top10)		
		top 1	top 5	top 10	family	superfamily	fold
PSI-BLAST	-	84.53	86.48	87.34	82.20/84.50/85.30	86.90/88.40/89.30	18.90/35.10/35.10
DeepSF	920k	73.00	90.25	94.51	75.87/91.77/95.14	72.23/90.08/94.70	51.35/67.57/72.97
CKN (128 filters)	211k	76.30	92.17	95.27	83.30/94.22/96.00	74.03/91.83/95.34	43.78/67.03/77.57
CKN (512 filters)	843k	84.11	94.29	96.36	90.24/95.77/97.21	82.33/94.20/96.35	45.41/69.19/79.73
RKN (128 filters)	211k	77.82	92.89	95.51	76.91/93.13/95.70	78.56/92.98/95.53	60.54/83.78/90.54
RKN (512 filters)	843k	85.29	94.95	96.54	84.31/94.80/96.74	85.99/95.22/96.60	71.35/84.86/89.73

Application to other DNA classification tasks (R2). We chose not to include the DNA experiment from [4] since allowing gaps is unlikely to help predict transcription factor binding sites. Running the RKN model on this dataset shows that it performs similarly to CKN (auROC 0.805 for RKN vs 0.806 for CKN on motif occupancy datasets).

Performance (R3). GPkernel indeed achieves the same auROC as RKN but its auROC50 is much lower. In addition, RKNs is highly scalable, whereas GPkernel suffers from the traditional lack of scalability of kernel methods.

Additional results (R3). The parameters k and q have been chosen as in CKN-seq for fair comparison, but we agree that results for different values would be informative and will include them in the appendix. Regarding the number of layers, we have not encountered a case where a two-layer model performed better than a single-layer model involving the same number of parameters—a similar observation was made in [4]. We have also completed the experiments for GMP with $\lambda \neq 0$. Its auROC/auROC50 for one-hot is (0.848/0.57) and for BLOSUM62 (0.852/0.609).

Interpretability and Minor points (R3). RKN allows to capture gapped motifs. A simple way to observe these gaps would be to align the obtained motifs against a given sequence. We thank the reviewer for these remarks.