We would like to thank the reviewers for their careful reading and positive assessment of our work (**Rev #1**: "*this technique definitely looks original*", **Rev #2**: "*novelty and importance are both significant*", **Rev #3**: "*this work provides an interesting and useful idea to the field*"). We attempt hereafter to address the main concerns raised by the reviewers.

Related works (Rev #1). Rev #1 points out the lack of references to You et al. [2017] and Gupta et al. [2016] which combine deep neural networks (NN) with monotonic lattice regression in order to learn functions that are monotonic with respect to a subset of their input variables. We thank the reviewer for these references, they are very relevant and will be added to the manuscript. **Rev #1** also suggests to add experiments in order to compare UMNNs with this method. In our opinion, a complete review of monotonic NNs and their use in the context of normalizing flows (NF) are very relevant and certainly worth of many valuable insights, but should be carried out within the scope of an extended or separate paper. Proposing a new parametric monotonic transformation and exploring how to combine it with autoregressive architectures into a NF is already a significant contribution in itself (**Rev #1**: "*This paper contributes a novel technique for modeling monotonic functions [...] that is a significant contribution*" **Rev #1**: "*The technique is applied to autoregressive flows [...] shown to have competitive performance results*" **Rev #3**: "*A new way of parameterizing monotonic networks [...] this is significant, and can inspire more future work*"). Given the page limit constraints, we are afraid that adding experiments in the current manuscript would decrease its clarity and concision.

Universality of UMNNs (Rev #1, #3). We thank **Rev #1** to have pointed out the ambiguity of our statement about the difference in terms of expressiveness between UMNNs and previous monotonic neural networks. We do not want to erroneously claim that UMNNs are the first universal approximator of monotonic transformations. Instead, we argue that other neural architectures for density estimation do so in a way that "leads to a cap on the expressiveness" of the transformations in the non-asymptotic case (finite number of neurons). While Sill [1998] (as well as Huang et al. [2018] and De Cao et al. [2019] for universal *density* approximators) has proven the universality of his approach in the asymptotic case, we believe that the constraints on the positiveness of the weights and on the class of possible activation functions are unnecessarily restraining the hypothesis space in the non-asymptotic case. We will make sure to clarify our statement in the next version of the manuscript. On a similar track, **Rev #3** wonders if UMNN is a uniform density estimator. Yes, for continuous random variables. By relying on the inverse sampling theorem it is enough to prove that UMNNs are universal approximators of continuously derivable (C^1) monotonic functions. Indeed, if UMNNs can represent any C^1 monotonic function, then they can also represent the (inverse) cumulative distribution function of any continuous random variable. Any continuously derivable function $f : \mathcal{D} \to \mathcal{I}$ can be expressed as the following integral: $f(x) = \int_a^x \frac{df}{dx} dx + f(a), \quad \forall x, a \in \mathcal{D}$. The derivative $\frac{df}{dx}$ is a continuous positive function and it is known that this function can be successfully approximated by a NN (such as those used in UMNNs) thanks to the universal approximation theorem of NNs.

Theory: Scalability and complexity analysis (Rev #1, #2, #3). Rev #1 and **#3** show concerns regarding the superior scalability (in terms of memory) of UMNNs in comparison to NAF and B-NAF. UMNNs are particularly well suited for density estimation because the computation of the Jacobian only requires a single forward evaluation of a NN. Together with the Leibniz integral rule, they make the evaluation of the log-likelihood derivative as memory efficient as usual supervised learning, which is equivalent to a single backward pass on the computation graph. By contrast, density estimation with previous monotonic transformations typically requires a backward evaluation of the computation graph of the transformer NN to obtain the Jacobian. Then, this pass must be evaluated backward again in order to obtain the log-likelihood derivative. Both NAF and B-NAF provide a method to make this computation numerically stable, however both fail at not increasing the size of the computation graph of the log-likelihood derivative, hence leading to a memory overhead. **Rev #3** also asked about the speed of Clenshaw-Curtis algorithm. In the case of static Clenshaw-Curtis, the function values at each evaluation point can be computed in parallel using batch of points. Thus, the limitation comes usually from the GPU memory which might not be large enough to store "meta-batches" of size $d \times N \times B$ (with d the dimension of the data, N the number of integration steps and B the batch size). **Rev #3** also asked about the relation between Lipschitzness and number of integration steps. We did not formally assess the impact of the number of integration steps and Lipschitz constant. However, we observed that as long as the Lipschitz constant of the network does not explode (< 1000), a reasonable number of integration steps (< 100) is sufficient to ensure the convergence of the quadrature. **Rev #2** suggests to add a discussion about the design of the NNs. We would like to recall that we provide all the experimental details in the appendix, moreover the code will be publicly released. Finally, **Rev #2** would be in favor of a deeper theoretical discussion. **We will make sure to develop and clarify all these minor elements in the revised version of the manuscript.** We will take advantage of the discussion about Lipschitzness to provide some insights about the design of the different neural networks.

More experiments (Rev #2). More density estimation experiments are suggested by **Rev #2**. We agree with him/her that more experiments are always a plus and can only improve the quality of our work. However we would like to bend the fact that we used the classical benchmarks for NFs (**Rev #3**: "*The experimental evaluation part is also largely satisfying*"). We even did more experiments than most of the competing methods (**Rev #3**: "*this is one of the earliest works (if not the first) that directly inverts an autoregressive flow*").