General comments. We thank the Reviewers for their detailed reviews and the feedback regarding the **theoretical study** of rational neural networks (NNs) and their **promising applications**. Reviewer 4 noted that the introduction of rational NNs is motivated by theory, while most exotic activation functions are only empirically supported. We address the referees' remarks on the theoretical results and numerical experiments here. The paper will be revised accordingly.

Theoretical results. Reviewer 3 highlights the comparison between this paper and Telgarsky's work. We wish to emphasize that our key contribution is to employ a **composition of low-degree rationals** $r(x) = r_{\#\text{layers}} \cdots r_2(r_1(x))$, naturally realized by a NN, to approximate functions efficiently, while Telgarsky's work approximates a ReLU NN by a high-degree rational function not in the form of a NN. The Newman polynomials used by Telgarsky do not preserve (minimax) optimality under composition and result in an exponentially larger number of trainable parameters (see Fig. 1 of the paper for a comparison). Finally, we present **optimal lower and upper approximation bounds**. The notion of size and Zolotarev functions will be explicitly defined in the revised version, as requested by Reviewers 3 and 4.

In response to Reviewer 2, concerning the choice of the degree $(3, 2)$, we emphasize that **this degree appears naturally in the technical analysis** due to the composition property of the Zolotarev functions (see Section 3.1): the degree of the overall rational function r is a whopping $3^{\#\text{layers}}$, while the **number of trainable parameters only grows linearly**. A superdiagonal degree $(3, 2)$ allows r to behave like a **nonconstant function at** $\pm\infty$, unlike a diagonal degree e.g. $(2, 2)$. The theory ensures that low-degree rational activation functions minimize the number of trainable parameters given a fixed overall degree. The choice is also motivated empirically, and we do not claim that the degree $(3, 2)$ is the best choice for all situations as the best configuration may well depend on the application; see below Fig. 1 (right).

Reviewers 1 and 3 discussed the apparently marginal difference between $\log(1/\epsilon)$ and $\log\log(1/\epsilon)$ in Thm. 4. To clarify, the bound for rational NNs is **close to optimal**, given by $\epsilon^{-d/n}$ (DeVore et al., 1989). Most importantly, a rational NN can achieve this approximation power with a depth of only $\log\log(1/\epsilon)$, which is **exponentially smaller** than the $\log(1/\epsilon)$ layers needed by a ReLU network to approximate a smooth function to within ϵ. This improvement is obtained from the composition of low-degree rational functions and is not hidden in multiplicative constants, which do not depend on ϵ. This improved approximation power has **practical consequences** for larger NNs given that a deep NN is computationally expensive to train due to expensive gradient evaluation and slower convergence. The constants inside the $\mathcal{O}$ notations are computed whenever possible (see Lem. 1 and Cor. 3 of the Supplementary Material) but the main theorems of the paper treat a general setting, for which few (if any) papers in the literature give explicit constants.

Experimental results. We thank the Reviewers for their comments on the experiment in Section 3.1. While Fig. 2 (right) of the paper showed that rational NNs are easier to train than ReLU and Sinusoid NNs, it did not show the accuracy loss. We have performed **new experiments with a validation set** independent of the training set and display the validation loss throughout training (see Fig. 1 (left)). We find that **rational NNs outperform the other NNs** during the training phase and on the testing set. Reviewer 3 was concerned that this result was due to the difference in number of trainable parameters but the ReLU NN had 8000 and the rational had $7 \times \#\text{layers} + 8000 = 8035$. **Rational NN are never much bigger in terms of trainable parameters** than ReLU NNs since the increase is only linear with respect to the number of layers. Reviewer 2 suggested a **comparison with polynomial approximation**, which we have performed (see Fig. 1 (left)). Here, we train a NN with degree 3 polynomial activation functions. We observe that this NN is **harder to train than rational NNs** as shown by the non-smooth validation loss. Polynomials perform poorly on non-smooth functions such as ReLU, with an algebraic convergence of $\mathcal{O}(1/\text{degree})$ (Trefethen, 2013) rather than the (root-)exponential convergence with rationals. Finally, following Reviewer 2's suggestion **we compare rational NNs with different degree activation functions** and find that they **all perform better than ReLU NNs** (see Fig. 1 (right)).

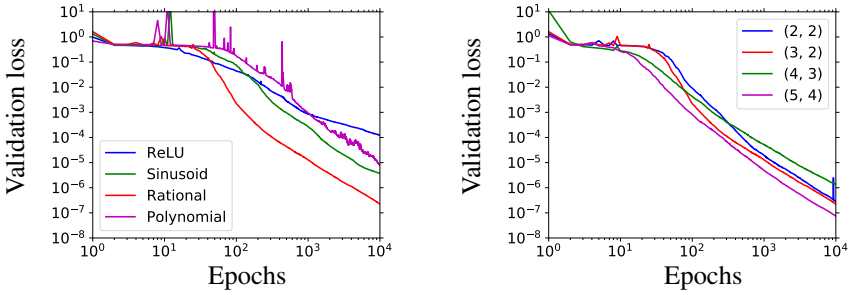

Figure 1: Left: Validation loss of a ReLU (blue), sine (green), polynomial (purple) of degree 3, and rational (red) NNs of type $(3, 2)$ with respect to the number of optimization steps. Right: Comparison between the validation losses of rational neural networks of types $(2, 2)$, $(3, 2)$, $(4, 3)$, and $(5, 4)$.