We thank the reviewers for their detailed reports, and we are particularly happy that they appreciated one of our key innovation to use complex weights to cancel higher order moments. We view this as the most original part of our work.

Reviewer 1: *1.* We agree with you that it is important yet non-obvious that our set of weights can be found by SGD /GD. We do believe that with more work one could reproduce our results directly with SGD/GD (see line 118). The paper being already quite long, we decided to postpone these technicalities to future extensions of our work.
2. We meant this informally, in the sense of combining iteratively weak learners (namely, our procedures add either a single or a couple of neurons at each iteration).
3. From the point of view of the proofs, the multiplicative error seemed more natural. For the Baum network the difference is immaterial (since one achieves exact fit), and for the NTK network it is only a logarithmic difference. For the Harmonic network however the issue is more severe, and it comes back to the difficulty we mention on line 77. It is a great open problem in our opinion to fix this issue for the Harmonic network.
4. We apologize for the brevity of Lemma 1's proof, which we viewed as a mere technicality. We did not expand on it because the argument is standard in the convex optimization literature, but we will add further details in the final version. Also note that the proof of Theorem 6 contains essentially the same argument, in a detailed form.
5. Thank you!
6. The key difference with Ji and Telgarsky 2020 is that the latter work makes more assumptions on the data, and in turn it allows them to construct smaller networks. We believe the two papers are complementary.
7. The non-polynomial assumption is necessary to invoke universality theorems (line 160).
8. We agree with you that more precise definitions of the NTK regime would be useful, we will add them in the revision.
9. Basically we have to assume that n is not exponentially large in the dimension (to ensure that the data is well-spread), and the parameter m is connected to the exponent relating n and d.
10. It is not optimal only in the sense that the dependency on epsilon can be improved. It is the same open problem as the one mentioned in point 3 above.

Reviewer 2: Thank you for the positive feedback! Following your suggestion we will add further comments on Daniely's work on and [Bresler and Nagaraj 2020] (see also response to Reviewer 3, in particular BN2020 requires much larger networks than ours).

Reviewer 3: Thank you for thoroughly reading the paper and for mentioning [Bresler and Nagaraj 2020]; we plan to add a detailed discussion comparing to this work, which we summarize here: One notable similarity is in fact that both works rely on an iterative procedure. However, a crucial difference between the two memorization results is that the result of [BN20] applies for the *random features* regime whereas in our work the iterative procedure actually updates the w_j's, and a much smaller number of neurons is needed as a result. The similarity in the use of complex weights seems to be superficial, and is done in a completely different context.

We will make a careful pass on the paper to fix the issues raised by the reviewer, as detailed below:
Line 415: We apologize for this mix-up, which is a result of a last-minute change of notation: α should be replaced by b, hence $\langle c, u\rangle = \alpha$. Other than that, we believe that the argument is correct.
Line 540: The constant C_m comes from analysis of polynomial functions in Gaussian space. Since the parameter m relates between γ and n there is an indirect dependence of C_m on the those quantities, but thanks to Assumption (22) everything can be expressed in terms of m. The constant is actually super-exponential in m and we will make it explicit in the final version.
Line 175: By adding an additional neuron we can assume that the label 1 is the minority. Hence $\lceil \frac{n}{2d} \rceil$ iterations suffice to cover all relevant labels.

Reviewer 4: We thank you for your detailed reading of the paper! We will take into consideration your suggestion for the organization of the paper, and focus the rebuttal mostly on correctness. First, regarding the weakness, note that the results in Section 3 do not require any assumption on the data (besides full rank, which is essentially always the case). We agree that in Section 4 and 5 the "wide-spread" assumption exclude certain real datasets, and it is a great open problem on how to go beyond this assumption. We felt however that it was a natural first step to analyze iterative methods such as the ones proposed in Sections 4,5. Secondly, regarding your memorization comment: Our constructions are actual learning algorithms, in the sense that they take as input the training data, and output a set of weights. It is not however a classical GD/SGD type learning algorithm; it would be a very nice extension to our work to prove similar results in this case. Now regarding your clarificaiton points:

1. Note the normalization $1/n$. The point is that an error of any fixed (but small) value of ε can be attained.
2. We will provide more details in the final version. In the meantime, the proof of Theorem 6 contains essentially the same iterative procedure (in fact, a generalization of Lemma 1).
3. It should be $f_{u,v,b}$ in (3). Thank you for catching this! Also note the line just above (3), which agrees with $\delta \to 0$.
4. Since ReLU is piece-wise linear, we can take δ small enough to ensure that (3) holds true even without the limit.
5. Note that albeit the log function, the expression diverges at $\gamma = 1$ so this dependence is actually polynomial.