We thank the reviewers for the helpful feedback and the positive assessment of our submission. We plan to update the text and bibliography following their suggestions.

*Reviewer #1, "It is interesting to see if further increase the width of the network (from linear in d to polynomial in d and even exponential in d), does the discontinuous approximation requires a smaller depth."*

In the setting of our paper (minimization of the total network size) a large depth is in some sense unavoidable (as e.g. Theorem 3.2 shows). However, in general there is of course some trade-off between width and depth. Depth is more important for expressiveness (since a parallel computation can be serialized, but not vice versa) – for example, even for 3-layer nets, Eldan & Shamir [1] show existence of small 3-layer nets that require exponential width if expressed as 2-layer nets. As for the dependence on the input dimension $d$, the key point is how the approximated functional family depends on $d$. Assuming a sufficiently constrained family (e.g. a ball in the Barron space[2] or functions with a compositional structure[3]), one can in a sense avoid the curse of dimensionality and find low-depth approximations with the width determined mostly by the required accuracy rather than $d$. We remark also that the weight discontinuity is present to a certain extent in all optimized nonlinear models (not necessarily deep or involving coding constructions)[4].

*Reviewer #3, "Concentrating information in a small proportion of high-precision weights in the network according to a discontinuous assignment function seems unstable in the face of noise or imperfect optimization... Can the authors comment on how their results might interact with these other sources of error?"*

There is indeed a significant instability, especially for approximations with periodic functions, when the information concentration is highest. In our construction for ReLU networks, the issue is ameliorated by the fact that the information is divided into independent chunks: first on the level of weights corresponding to different patches, and then also on the level of weight digits corresponding to different positions in a patch. This suggests that the errors can be localized and, borrowing again from the coding theory, one can hypothesize that we can improve stability by allowing some redundancy and, for example, using something like error-correcting codes. On the other hand, in the construction with periodic activations, the classifier output is a chain of interdependent computations, so any error will have a stronger negative effect.

*Reviewer #4, "Theorem 5.1 extends the approximation results to all piece-wise linear activation functions and not just ReLUs. So in theory, this should also apply to max-outs and other variants of ReLUs such as Leaky ReLUs?"*

That's right, all these functions are easily expressible one via another using just linear operations ($\mathrm{ReLU}(x) = \max(0, x), \mathrm{LeakyReLU}_a(x) = \mathrm{ReLU}(x) - a\mathrm{ReLU}(-x), \max(x, y) = \frac{x+y}{2} + \frac{1}{2(1-a)}(\mathrm{LeakyReLU}_a(x-y) + \mathrm{LeakyReLU}_a(y-x))$), so any network of one type can be converted into another, exactly equivalent one, at the cost of merely increasing the number of neurons by a constant factor.

*Reviewer #4, "I fail to see some intuitions regarding the typical values of r, d, and H for the networks used in practice. While the approximation rate depends on r and d, the power law relationship uses a term $W^{-r/d}$ and d is the dimension of the input and W is the size of the network parameters. In practice, d can be of the order of millions in imaging-related applications, since we have million pixels in the image. In such cases, the approximation guarantees may be very weak and may not provide any insights."*

In imaging, an important difference from our setting is that reasonable images form only a small and complex subset of the whole ambient million-dimensional space, whereas in our setting the approximation is defined for each input vector from $[0, 1]^d$. Accordingly, the number of pixels is not the right value of $d$ here, a more appropriate value would be something like the intrinsic dimension of the image manifold. For example, for MNIST, suitable feature extraction and dimension reduction allows to reparameterize the data set by 9 parameters[5] while retaining classification accuracy above $98\%$, suggesting $d \lesssim 10$. As for smoothness $r$, usual classification problems such as MNIST do not quite fit our setting as the predicted output (the image label) is piecewise constant. One can assume some low value of smoothness (say $r \sim 1$, assuming a Lipschitz continuation), or refer to results on approximation of piecewise smooth functions[6]. Of course, all these estimates are very crude, and there are various other considerations for imaging-related problems that must be taken into account (e.g., our results assume $W \to \infty$ at a fixed $d$, but in a practical problem larger networks will have a higher "effective input dimension" in the earlier mentioned sense).

[1] R. Eldan, O. Shamir, The power of depth for feedforward neural networks, COLT 2016

[2] A. R. Barron, Universal Approximation Bounds for Superpositions of a Sigmoidal Function, 1993, DOI: 10.1109/18.256500

[3] T. Poggio et al., Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. Int. J. Autom. Comput. 14, 503–519 (2017)

[4] P. Kainen et al., Approximation by neural networks is not continuous, Neurocomputing 29(1), 47–56 (1999)

[5] A. Das et al, Dimensionality Reduction for Handwritten Digit Recognition, 2018, DOI: 10.4108/eai.12-2-2019.156590

[6] Ph. Peterson, F. Voigtlaender, Optimal approximation of piecewise smooth functions using deep ReLU neural networks, 2018, DOI: 10.1016/j.neunet.2018.08.019