We thank all reviewers for the constructive comments. Below, we label each comment by Reviewer#.Comment#.

R1.1 [ascent when p exceeds the range in Thm 2, descent-region] As p becomes larger, eventually the model error will increase again. The reason is that there will be some columns of $\mathbf{X}_{\text{train}}$ very similar to those of true features. Then, BP will pick those very similar but wrong features, and the error will approach null risk. As a result, the whole double-descent behavior should indeed be <descent-ascent(p<n)-descent(p>n)-ascent> (as shown in Fig. 1, the same is also true for ℓ_2-minimization). By the "descent region", we mean the second "descent" part. Thus, Fig. 1 shows that model error is insensitive to $\|\beta\|_2$ before it ascends again. We will make these clear in the final version.

R1.2 [constant term on the RHS of (9)] Currently we can reduce, but cannot completely get rid of, the constant term in (9). It does not appear in Prop. 4 due to the different path of proof (see (88) of Supplemental Material).

R1.3 [upper-bound of BP is not affected by $\|\beta\|_2$] This insensitivity to $\|\beta\|_2$ is a special feature of BP. This can be seen by considering the following (perhaps over-simplified) example. Given one training datum $(x \in \mathbb{R}^2, y \in \mathbb{R})$ generated by the true model $y = x^T\beta + 0.1$ (here 0.1 is the noise), where $\beta = [\beta_1,\ \beta_2]^T$, and $x = [1,\ 0.5]^T$. Assume $\beta_2 = 0$ (for sparsity). If we minimize $\|\hat{\beta}\|_1$ subject to $y = x^T\hat{\beta}$, we get $\hat{\beta} = [\beta_1 + 0.1,\ 0]^T$ (and thus the model error is always $\|\beta - \hat{\beta}\|_2 = 0.1$) for any β_1. In a higher dimensional space, due to a similar reason (under certain conditions), the generalization error of BP could be insensitive to $\|\beta\|_2$.

R2.1 [upper bound may be larger than the null risk] We agree. On the other hand, one important goal is to understand in what settings min-ℓ_1 is better than min-ℓ_2. Eq. (9) can already provide new and useful answers, e.g., when p is exponentially large and when noise is low, the model error of min-ℓ_1 is much lower than that of min-ℓ_2 (null risk).

R2.2 [bounds are loose, range of p does not include p near n, squared loss and isotropic Gaussian data] We acknowledge these limitations. We believe the results can be extended to independent non-Gaussian data, while other generalization may require more work. However, we also would like to point out that, even under our model, the analysis for BP is significantly more challenging than min-ℓ_2 solutions because, unlike min-ℓ_2 solutions that can be written as pseudo-inverses, min-ℓ_1 solutions have no closed form. Thus, we have to resort to large-p asymptotes to capture the model error of BP, which is why the characterization of $p \approx n$ remains unknown. We believe that this difficulty is the reason why there are so few results on overfitting BP solutions. Despite these limitations, our results still successfully reveal key insights on the difference between min-ℓ_1 and min-ℓ_2 solutions, which we believe are of significant interest to the community (as the other reviewer commented).

R2.3 [required sparsity relates to n] Such requirements are not uncommon, e.g., Thm. 3.1 of [16] also requires the sparsity to be below some function of M (incoherence of $\mathbf{X}_{\text{train}}$), which is related to n according to Prop. 9 in our paper.

R2.4 [establish double descent by the upper-bound] Our result already implies double descent. The reason is that, when $p \leq n$, the MSE solution is independent of ℓ_1 vs. ℓ_2. From known results for min-ℓ_2, the model error has a peak when $p \to n$ from below. Thus, our upper bound implies that the model error of BP has to descend from that peak.

R2.5 [related work: arXiv preprint arXiv:2002.01586 (2020)] Thank you for the pointer and we will cite this reference. However, the reference studies classification, while we study regression. The notion of "fitting the data" is quite different between the two models. Hence, we feel that the conclusions are not directly comparable.

R2.6 [(minor) Fig. 2 starts from different positions (x axis)] That is because we focus on the interpolating regime $p > n$ and n varies. We will add back the regime $p < n$ for the final version. Thank you for the suggestion!

R2.7 / R3.1 [connection to DNN, motivation of studying ℓ_1 minimization] We agree that the connection between min-ℓ_1 and DNN is not clear yet. Similar to other related work, our approach is to use linear models as a starting point to understand which types of overfitting solutions approximate the generalization power of DNN better. Our hope is that such analysis would eventually lead to better training methods than SGD (which is closely related to ℓ_2 minimization).

R3.2 [Gaussian features assumption, limited range of p compared with ℓ_2, constants in Theorem 2 are not sharp] We acknowledge these limitations. Please also refer to our response to R2.2.

R4.1 [n and p are large for theory] One reason for large p and n is that we aim for with-high-probability results, which leads to stricter conditions than average-case analysis. Another reason is that we have not optimized the constants. Nonetheless, the numerical results suggest that the main predicted trends hold for much smaller n and p. We will clarify this in the final version. Thank you for the suggestion!

R4.2 [model error independent of the signal strength] Thanks for sharing your code! There seems to be a small yet consequential bug on Line 39, which should be `print(np.linalg.norm(x[:p] - beta_exact))` instead of `print(np.linalg.norm(x[:p] - beta_exact[:, 0]))`. (The original code returns the norm of a $p \times p$ matrix instead of a $p \times 1$ vector.) After correcting this line, the result becomes: 0.019, 0.020, 0.021, and 0.021 for $\|\beta\|_2$ equals to 0.1, 1, 10, and 100, respectively, which verifies our conclusion "model error can be independent of the signal strength". Please also refer to our response to R1.3 for the intuition. We will post our code for the final version.