We thank the reviewers for their time, and for their valuable feedback, which will improve the quality and clarity of the manuscript. We feel that we didn't sufficiently highlight the contributions of our methodology with respect to PSnet. PSnet provides a frequency estimate that can be used to perform frequency estimation only if the right number of components is known. Then it is competitive with traditional methods at high noise levels. In contrast, DeepFreq provides an improved frequency representation learned with a novel multilinear architecture, which is significantly superior to PSnet's representation. In addition, it incorporates a separate counting module. Combining these contributions results in a fully automatic method that *outperforms* the state of the art across a wide range of noise levels. We address the reviewers' comments in detail below.

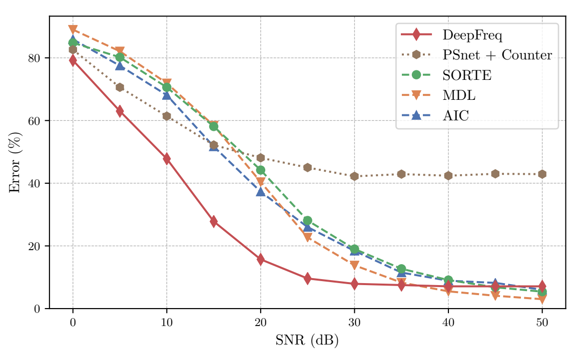
Reviewer 1: *Yet, when using neural network one important aspect . . . authors should mention training time and energy consumption (or CPU number/GPU type).* This is a good point. We will report training time (11 hours on a NVIDIA 1080Ti), and the corresponding energy consumption. We will highlight that learning-based methodology introduces a new trade-off in signal-processing applications: it requires prior training, but provides fast test-time performance.

Reviewer 2: *One of the novelty of the approach is the component counting module : I would have liked to see the influence of this module in terms of performances : what if this module is removed ? Are the improvement due to this module or to the change in the encoder architecture ?* We agree with the reviewer that this is an important point that should be clarified. The proposed frequency-representation and frequency-counting modules are trained separately. Section 3.2 provides an evaluation of the frequency-representation module (and a comparison to other frequency representations). To decouple the evaluation from the performance of the frequency-counting module we assume knowledge of the true number of frequencies. The results show that the novel elements in the architecture of the frequency-representation module produce a significant boost in performance. We will edit the paper to make this clearer.

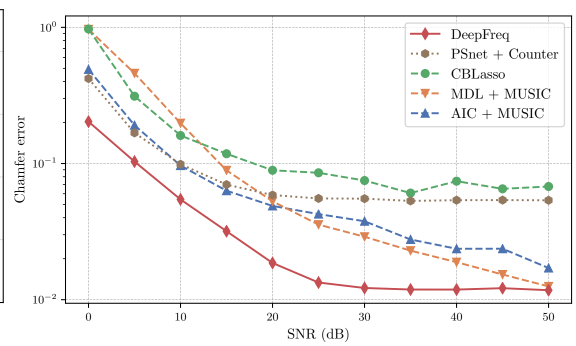
What if the number of components is misevaluated ? Does it degrade the results ? This is a very interesting point. Our framework is robust to mistakes in frequency counting because the frequency-representation module is trained separately from the counting module. This follows from the results in Section 3.2 which show that the m highest local maxima of the learned representation are accurate estimates of the true frequencies (m is the true number of frequencies). As a result, if the counting module wrongly estimates that there are $m-1$ frequencies (for example), those $m-1$ frequencies will be estimated accurately. We will edit the paper to highlight this point.

Same questions about the noise level : what if this parameter is misevaluated ? Section 3.4 gives some insights about this, but I think that the fact that the method is fully automatic is an important point that would have needed a more detailed discussion. We will add a detailed discussion about this. In our framework the deep neural net is trained simultaneously over a wide range of noise levels (up to the point where the signal and the noise have the same energy), so there is no need for an explicit estimate of the noise level. This is the case for the results reported in Sections 3.3 and 3.4, which therefore showcase the robustness of the method to variations in the noise.

How would the standard PSnet behave with an additional component counting module? Would it achieve the same performances? This is a very good point. The performance of the counting module is highly dependent on the quality of the frequency representation provided as input. The figures below– which are modified versions of Figures 6 and 7 that will be added to the paper– illustrate the performance of PSnet combined with a counting module. The counting module has the same architecture as in DeepFreq, but is trained from scratch using the representation produced by PSnet. Using the PSnet representation results in a significant degradation of performance in both estimation error and Chamfer error.



(a) Figure 6 adding PSnet and a counter module.



(b) Figure 7 adding PSnet and a counter module.

Reviewer 3: *For the sinusoid counting task, the error metric is not described . . .* We completely agree that this was not clear. The metrics used to train and evaluate the counting module are different. In Section 3.3 the error is computed by counting the fraction of signals in the test set for which the number of components is not estimated correctly. Training is indeed performed by regression, using the squared difference between the true number of frequencies and the non-rounded output of the counting module (as stated briefly at the end of Section 2.3[1]).

[1]There is a small typo, which we will fix: the cost function is the squared ℓ_2 norm of the difference not the ℓ_2 norm.