We would like to thank the reviewers for their encouraging and instructive comments. We also thanks the reviewers for pointing out typos and inconsistency in the notations that will be corrected in the paper. We first address generals concerns shared by the reviewers and then give detailed answers to other points raised by each reviewer.

The work is limited to the 1D case. We restricted our attention to the 1D case for two reasons: 1) It is not trivial to propose an analysis formulation for a network even in the 1D case as this requires deriving through the prox - which represents one of our main contribution - and we showed that these development are necessary due to the inefficiency of synthesis based approaches. This is a significant necessary first step to problems in 2D/3D. 2) The extension to larger dimensional signals requires further development that would complexify the paper and make the comparison less clear. For the 2D case, we believe we can use similar tools using equivalence with LASSO using synthesis formulation. However, this would require a projection on the tangent space of the integrable discrete signals, which we feel is out of scope for this paper. The path mentioned in the conclusion and by R#4 of using the dual might also be an alternative but shifting from the dual space to the primal without losing the first order information is again non-trivial. Both these extensions share the same basics as our proposed method but require more tools (duality or manifold analysis) so we consider it is out of scope for this paper.

Reproducibility: We thanks reviewers R#3 & R#4 for highlighting that some details of our experimental setup are unclear in the text. To answer R#4, the weights of the LISTA approximation of the prox are trained jointly. We will clarify this in the paper and also add a section in appendix to further detail the training procedure used.

R#2: **The contribution is mostly a performance improvement.** We would like to point to the reviewer that our contribution is centered on proposing a learning based acceleration for TV-regularized problems by relying on an analysis formulation. To the best of our knowledge, this has not been explored before, and it is not trivial as it requires the derivation through a prox. As a consequence, one can obtain performance improvements, though we envision our contribution impacting several other problems where analysis-based regularization techniques are important. Some of these include applications to functional MRI as mentioned in the paper but also pre-processing of physiological signals for health care and analysis of sensor data in industrial context. Moreover, our work also gives a way to compute the weak derivative of TV-regularized problems, which could be used for hyper-parameter λ tuning or for dictionary learning methods.

R#3: **Theoretical contributions of this work are not clear.** While it is true that many questions of convergence remain open, our theoretical results are important because they depict the central role of optimizing the original analysis formulation. One might think that we can deploy the original synthesis formulation by directly applying LISTA to learn a network. However this is found to be slow (see Fig.3). Our theoretical analysis explains *why* this is the case, and as a result why it is important to develop such network architecture for the analysis formulation instead.

Generalizability of the learned network. This is true for any approach that relies on learned algorithms, and indeed most statistical learning methods rely on training samples to be sampled from the same distribution as those at test time. In our context, the key observation is that most of the time, one does not care about solving this problem for any signal in the entire space but rather only those in a subset of such space. Additionally, using learned algorithms can accelerate the optimization of the loss for a fixed number of iterations. The whole point in learning such a network is that one can provide faster approximate convergence for signals coming from the *same distribution* as your training set with a *fixed number of iterations/layers*.

Uniqueness of the Jacobian This is indeed a weak Jacobian and not the Jacobian. We will clarify accordingly.

Abstract and related work We will improve the abstract to better mention our contributions and also mention ADMM in methods to solve TV regularized problems.

R#4: **Comparison with accelerated PGD.** The computational gain is significant for solutions with some approximation error, i.e. where few layers or iterations are used. If one is after exact solutions, then non-learned solvers are to be employed. However, learned approaches can be very useful in practice as one is often interested in approximated solutions with controlled tolerances. Also, while it is true that accelerated PGD is faster at the end, learned approaches are faster non accelerated PGD. Note that the same acceleration mechanisms could also be used in the network, similarly to LFISTA developed in Moreau & Bruna (2017). However, we chose not to include this as it increases the complexity of the architecture and makes it less clear for comparison purposes.

Computational cost for each layer is the same This is correct. In particular, the iterative version of an algorithm has the exact same computational cost as the one of a learned network based on this method at test time.

Limited computational gain overall. Figure 4 depicts the value of the proximal loss, F(z), in Eq. (3). This shows that our learned approach achieves a slightly lower error on the prox than directly applying ISTA. When combined in the overall algorithm, while the quality of the prox does not change a lot, this leads to a large computational gain, as illustrated in Fig.3.

R#5: **The novelty of the work is not that significant.** The formulation of a learnable network in the analysis instead of the synthesis (LISTA) is not trivial as it requires to differentiate the prox operator, and this has prevented the application of these ideas to this kind of problems. To the best of our knowledge, we are the first to apply deep nets to accelerate methods that contain a non separable prox which are present in many applications.