We first thank the reviewers for their insightful comments which we have taken into careful consideration. We address the four main areas of criticism below (reviewers referred to as R1-5).

1. Learning-rule and benchmarking (R1, R2, R5): *A common critique was that a comparison with state-of-the-art (SOTA) SNNs was lacking. It was also pointed out that learning rules for the recurrent and readout weights were missing.*
If our work were to be evaluated using only performance metrics, this criticism would be fair. However, in machine learning and science, an equally important goal (and metric) is how well a contribution elucidates underlying principles of the system under study. We stress that our main objective was to understand the exact nature of computations done by SNNs and how well this matches biology, rather than improved network performance. We think that we successfully attained this objective—a large class of SNNs can now be understood as doing non-linear convex input-output computations within a layer. Moreover, the resulting networks display many biological features such as robustness and E/I-balance. Given these results, our learning section is simply supposed to illustrate, as a proof-of-concept, that the problem of learning can be posed differently, once we understand geometrically what the actual computation is. Learning paradigms for networks of 'convex layers' have been shown to be effective (e.g. Amos & Kolter, 2017), but are highly non-local and therefore unusable for online learning in SNNs.

2. Spike-coding network comparison (R2): *R2 was unsure of the novelty compared to spike coding networks (SCNs), and whether the reported brittleness of SCNs in presence of delays transfers.*
The key advance over standard SCNs is that we show how to perform non-linear computations in these systems. Standard SCNs such as in Boerlin et al (2013) are restricted to linear computations. (Non-linearities in these networks were previously introduced by assuming non-linear computations in subthreshold voltages, see e.g., Thalmeier et al, 2016, or Alemi et al, 2018; however, this leads to implausible assumptions on dendritic trees.) The reviewer's second point, the brittleness of standard SCNs to delays, is indeed well taken and a key criticism of SCNs. Interestingly, our networks do not suffer from this problem, and are instead very robust to delays, as we explain in Fig. 1. We can update the paper figures to include this point.

3. Low-dimensional toy problems (R1, R2, R5): *Several reviewers had doubts about our focus on single-layered networks solving low-dimensional problems.*
We stress that our main goal was to gain an understanding of the computations performed by a single SNN-layer dominated by lateral, inhibitory connectivity. It may seem surprising, but such layers are actually not well understood! We show that the computations done in these layers are far more complex than those of standard LN layers as used in deep networks. The use of low-dimensional examples was mainly to better communicate the intuitions we gained from the geometrical picture, and these intuitions do transfer to higher dimensions: SNNs can then be understood as having a read-out which constantly tries to reach some set-point in the high-dimensional space, while each neuron acts as a hyper-plane off which the read-out will bounce back into the feasible set.

4. Recurrent connectivity (R5): *R5 noted that our networks are missing a crucial aspect of recurrent network computation: working with sequential data.*
Dealing with sequential data is certainly an important application for recurrent connectivity. However, that does not rule out that recurrent connections may have other functions. In our example, recurrent connectivity is useful because it (1) helps networks achieve specific input-output functions and (2) allows networks to be robust to many perturbations (e.g. cell death). We additionally provided a novel way to understand how these two properties and the connectivity are linked.

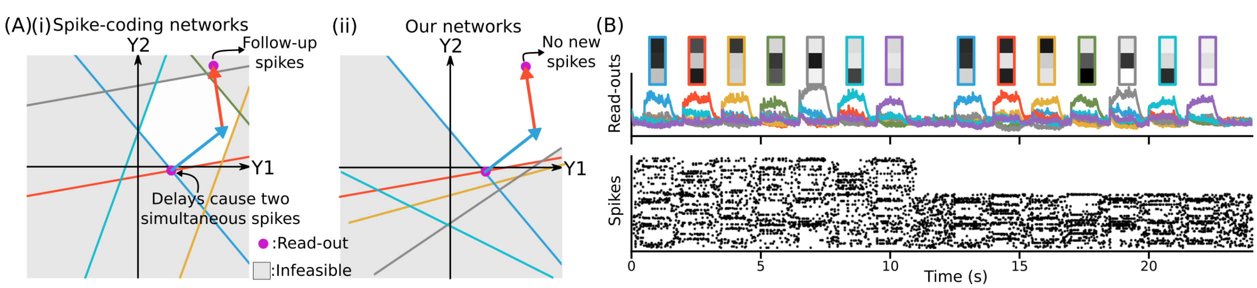

Figure 1: (A) (i) In spike-coding networks the connectivity is set up such that it generates a closed feasible space. Delays can then cause several neurons (red + blue) to spike simultaneously, leading to multiple threshold crossings and causing several follow-up spikes (green + gray neurons), and so on, leading to instability. (ii) In our networks, connectivity is set up to lead to an open feasible space. When multiple neurons fire simultaneously due to delays, the system remains within the feasible set, and hence remains stable. (B) The network from Figure 4 in the paper simulated with a delay of 2msec. Note that neither spiking nor functionality are affected by the delay.