To Reviewer #1

Novelty of our approach, compared with the previous approach (i.e. Liu 2017). Our approach is NOT 'an application of the previous model (Liu 2017)'. Our proposed approach (i.e. DyEnsemble) consists of three main parts: state-space modeling, model candidate construction (a part of model-solving), and dynamic ensemble (a part of model-solving). 1) For the state-space modeling, we proposed a NOVEL dynamic observation formula (eq (3)), which described the nonstationary changes in neural signals. Liu's approach only work with multiple noise models, and could not describe changes of observation functions, thus it is unusable for nonstationary neural decoding. 2) For the model candidate construction, we proposed two new operations, namely Neuron dropout and Weight perturbation, to construct proper model candidates from neural signals. This stage is very critical for the effectiveness of our approach. 3) For dynamic ensemble, we mainly employed the framework of Liu's robust particle filter approach.

Effectiveness of Neuron dropout and Weight perturbation. Indeed, we had evaluated the two new operations, and did not put in the paper for space limit. Part of them is shown in the table below. The results demonstrated that both dropout and perturbation significantly improve the correlation coefficient (CC) .The table includes the particle filter (PF) baseline (without neuron dropout and perturbation), the PF with perturbation (p=0.1) alone, and the PF with both perturbation (p=0.1) and dropout (drop 5 neurons). Compared with the PF baseline, weight perturbation improves the performance by about 10% in noisy situations. Neuron dropout operation leads to a further 20% performance improvement with 4 noisy neurons. Thanks.

Table 1: Evaluation of dropout and perturbation in terms of correlation coefficient (CC)

Method	Rat 1			Rat 2		
	Original	**Noisy (#2)**	**Noisy (#4)**	**Original**	**Noisy (#2)**	**Noisy (#4)**
PF Baseline	0.776±0.002	0.684±0.014	0.558±0.009	0.798±0.002	0.579±0.066	0.377±0.155
PF+Perturbation(0.1)	0.780±0.008	0.711±0.004	0.557±0.035	0.780±0.006	0.665±0.024	0.472±0.080
PF+Perturbation(0.1)+Dropout(5)	0.775±0.015	0.739±0.021	0.671±0.039	0.803±0.009	0.584±0.035	0.596±0.035

Response to the questions. (1) Under what instances is low α useful? - Low α can be useful when the adjacent time windows are not strongly correlated, e.g. with small time windows. (2) What is w_{k-1}^i and how is it computed? - w_{k-1}^i is the weight of the i-th particle at time $k-1$. We initialize w_0^i at time 0, and update it iteratively as described in Section 2.3. (3) What is the function form of the observation function, and how are the models trained. - The observation function takes form of $\mathbf{y} = \mathbf{A}\mathbf{x}$, and $\mathbf{A}$ is estimated by the least square algorithm. (4) For the other suggestions/issues, we will revise the paper accordingly. Many thanks for your valuable comments.

To Reviewer #2

Description to techniques dealing with the same problem. Most existing neural decoders dealing with nonstationary problem can be classified into two groups. The first group is recalibration-based, which uses a static model, and periodically recalibrates it (with offline paradigms) or adaptively updates the parameters online (usually require true intention/trajectory). Most approaches belong to this group (Gilja and Henderson 2015) (Shanechi and Carmena 2016). The second group uses dynamic models to track nonstationary changes in signals (Eden and Donoghue 2004) (Wang and Principe 2016). The dynamic model-based approaches can avoid the expense of recalibration, which are potentially more suitable for long-term decoding tasks. However, there is very few study in this group for the challenge to model nonstationary neural signals.

Comparison with state-of-the-art. The proposed DyEnsemble approach belongs to the second group. Given strict time limit, we implemented dual decoder (Wang and Principe, 2016) with a Kalman filter, which tracks the gradual changes of individual neurons. The comparison with dual decoder is shown in the table below. Our approach demonstrates the superiority especially with noisy situations.

Table 2: Performance comparison in terms of correlation coefficient (CC)

Method	Rat 1			Rat 2		
	Original	**Noisy (#2)**	**Noisy (#4)**	**Original**	**Noisy (#2)**	**Noisy (#4)**
Dual decoder	0.779±0.000	0.694±0.010	0.575±0.013	**0.803±0.000**	0.585±0.025	0.387±0.030
DyEnsemble(18)	**0.799±0.012**	**0.735±0.006**	**0.583±0.090**	0.788±0.009	**0.633±0.064**	**0.516±0.092**

To Reviewer #3

About analysis of noises. We injected noise into real data because it could provide an intuitive ground truth to investigate the dynamic ensemble process of candidate models. Indeed, analysis of real-world noises in neural signals would demonstrate stronger results. We are collecting some long-term neural signals to analyze real-world noises. For the baseline approach you mentioned, we will add discussions to compare with it. Thanks for the thoughtful review and constructive suggestions.