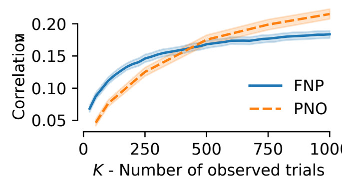
We thank all three reviewers for their valuable comments and positive feedback. All reviewers agree that our approach addresses an important problem: real-time inference of complex predictive neural models for online closed-loop experiments. The reviewers find that our paper "improve[s] the ability to estimate stimulus-response models for neurons in the brain" (R1), that it "is a very interesting paper and set of experiments" and its "use-case [...] compelling" (R2), and that our "approach is well motivated and would address an important problem" (R4). Their main concerns are: 1 our experiments are only on synthetic data (R1,R4), 2 our model is not compared to baseline models (R4), and 3 we need to present results about training and prediction time (R2, R4). We are confident that we can address all concerns and that doing so improves our results. We fit our Factorized Neural Processes (FNP) model to real neural responses from mouse V1. We find comparable predictive accuracy to state-of-the-art models and critically—and by design—predicting the response of unseen neurons is two-five orders of magnitude faster than using an optimization-based methodology. We will include these results in the paper and improve the clarity of the presentation (R1, R2) with additional technical details in supplemental materials. Because of the space limit, we can't respond to all detailed concerns, but we will fix them.

Network	Time
FNP Optimization (once)	6 days on dual V100
FNP Inference for 1000 trials	**250 ms** (1080Ti)
PNO Readout only	$\sim$ 20 s (1080)
PNO CNN+Readout	$\sim$ 5 min (1080)
PNO + Hyperparameters	$\sim$ 12 h (1080)

General motivation Our goal is to rapidly infer a predictive model of newly recorded neurons with minimal latency for online, closed-loop experiments. Our envisioned use case (R4) is active learning of tuning properties where stimuli are selected based on current estimates (and uncertainties, R2) of the tuning function to better constrain the model and learn it more efficiently. This is not feasible online with current models, even if just parts of the model are retrained (see 3). Prior experiments with predictive models fit them to newly acquired neurons overnight and tested them the next day (e.g. Bashivan2019 and Walker2019).



1 **Demonstration on real data (R1, R4)** We trained an FNP on 57,533 mouse V1 neurons responding to static scenes collected across 19 different scans. We tested the K-shot predictive accuracy on 1000 randomly selected neurons from a hold-out scan (i.e. never seen during training) with K up to 1000 natural images to infer the tuning properties and predict responses to stimuli (a rapid network inference with no cell-specific optimization). In agreement with synthetic data, the predictive accuracy improves rapidly with the first several hundred trials and continues to improve with additional observations (see figure) establishing the utility of our method on real neural responses. The FNP also generates realistic receptive field estimates.

2 **Comparison to baseline model (R4)** We now compare the performance of FNP (ours) to a SOTA model in the style of Klindt2017, adapted for mouse V1, which we reference as Per Neuron Optimization (PNO). We trained it on the responses of a single scan with 4,335 neurons to 5k natural scenes. Subsequently, we froze the core CNN and fit a readout (linear+nonlinearity) to 1,000 new neurons on up to 1,000 images and measure the prediction accuracy. Excitingly, FNP generalizes well to new neurons and with 1k images is almost as accurate as PNO (which is optimized for those individual cells), and even *outperforms* it for smaller numbers of observations (see figure). Importantly, this is achieved with massive improvements in run-time (see 3).

3 **Long training time (R2, R4)** We feel that there might have been a misunderstanding and we will make this more clear in the paper: FNP needs to be trained *only once* using all previously recorded data (see 1). During experiments, a predictor is obtained for newly acquired neurons with a nearly instant, single pass through the FNP. This takes only 250ms (for 1k responses) compared to optimization which ranges from $\sim$20s (if only fitting the readout) to $\sim$12h (to optimize hyperparameters as we currently do in experiments). We summarize training times in the above table. Thus getting a predictor with an FNP is two-five orders of magnitude faster, enabling real-time predictions. This is not possible with existing methods (R4). Thus our approach allows inferring updated neural response properties within the time of a single stimulus presentation.

Other (R2) $\mathbf{q}(z|s)$ is a variational approximation to $p(z|s)$; **Two-stage method of ref[6]** is also optimization based; **Eqn 1** integration is approximated by a single sample using the reparameterization trick (c.f. Kingma 2015), MLE is not easy in deep learning; **sub-pixel interpolation** makes the loss differentiable for optimization (c.f. Spatial Transformer Networks); **X-axis** corresponds to real trials (as above). **Section 3.3** $\mathbb{R}^{H\times W\times C}$ is an image with C channels after passing through the convolution and $H \times W$ refers to the spatial dimensions. It is then combined (non-linearly) with the responses to predict the location. Response units can be fluorescence (as above) or spike counts (see simulations) **Uncertainty** is currently only used as a tool for training, we imagine using entropy to drive stimulus selection; **Cell types:** good point, we did not want to emphasize categories, continuum is also possible, will revise terminology; **Fig 2 & LL** see 1,2 for real world example. **Complex cell RFs:** Correct! it looks opposite at times due to the random phase shifts; **Code** was withheld for anonymity, will be released; **Broader impact** will be updated, good suggestions.