Statistical models for neural encoding, decoding, information estimation, and optimal on-line stimulus design

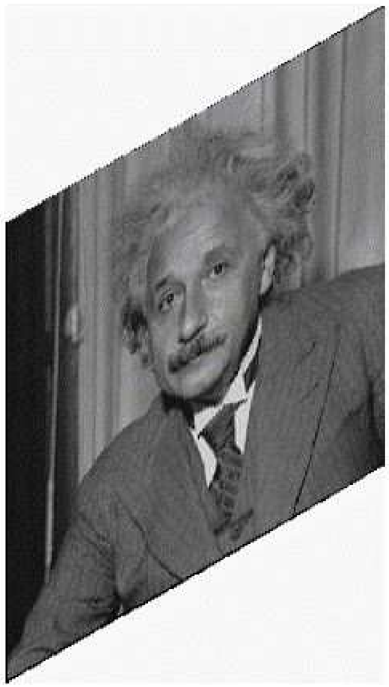
Liam Paninski

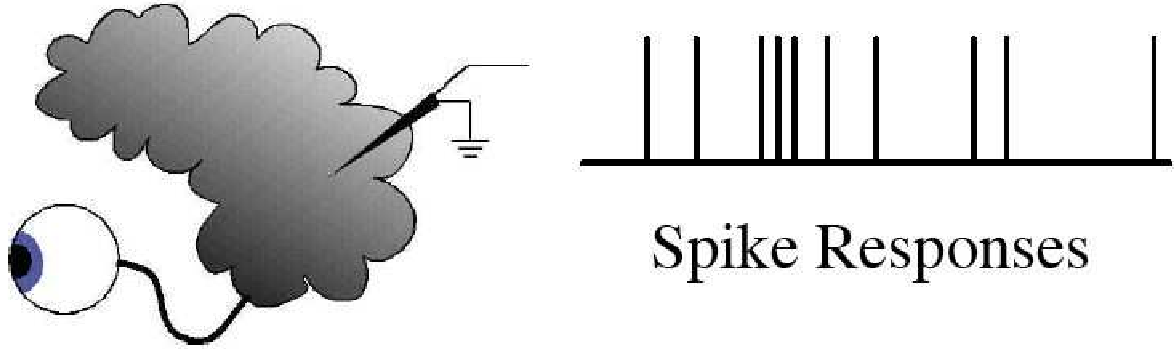
Department of Statistics and Center for Theoretical Neuroscience
Columbia University
http://www.stat.columbia.edu/∼liam
liam@stat.columbia.edu

October 22, 2007

Support: NIH CRCNS award, Sloan Research Fellowship, NSF CAREER award.

The neural code





Input-output relationship between

- External observables x (sensory stimuli, motor responses...)
- Neural variables y (spike trains, population activity...)

Probabilistic formulation: $p(y|x)$

Basic goal

...learning the neural code.

Fundamental question: how to estimate $p(y|x)$ from experimental data?

General problem is too hard — not enough data, too many inputs x and spike trains y

Avoiding the curse of insufficient data

Many approaches to make problem tractable:

1: Estimate some functional $f(p)$ instead

e.g., information-theoretic quantities (Nemenman et al., 2002; Paninski, 2003)

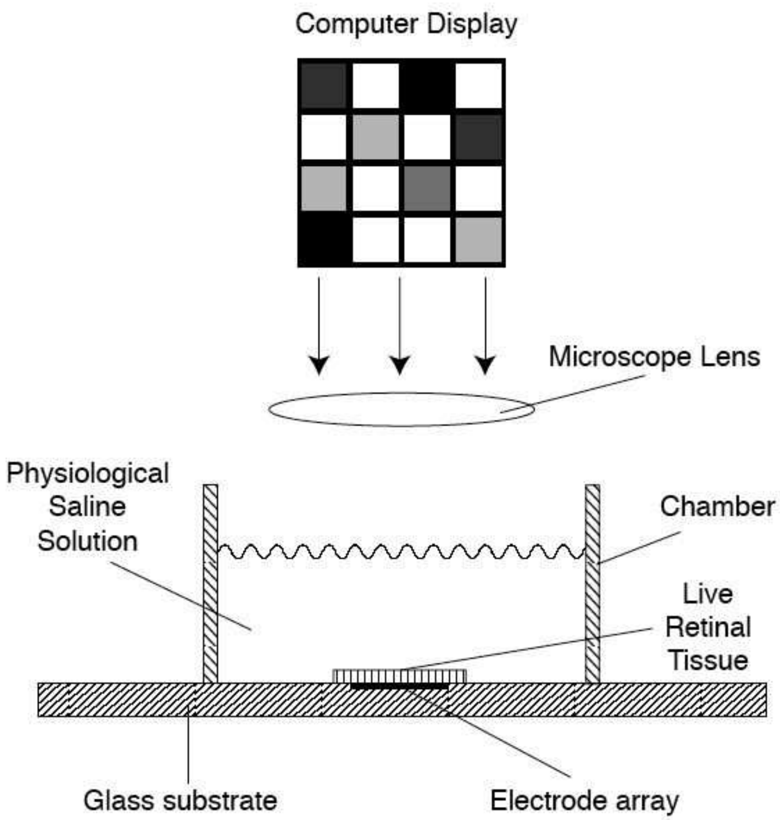
2: Select stimuli as efficiently as possible (Foldiak, 2001; Machens, 2002; Paninski, 2005; Lewi et al., 2006)

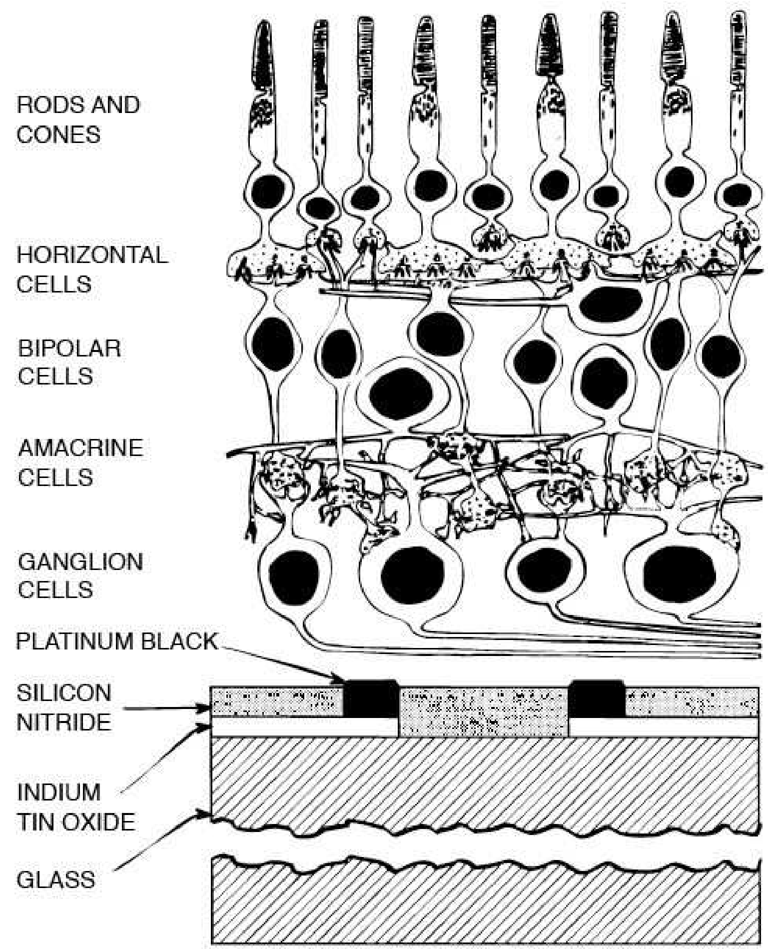
3: Fit a model with small number of parameters

Retinal ganglion neuronal data

Preparation: dissociated macaque retina

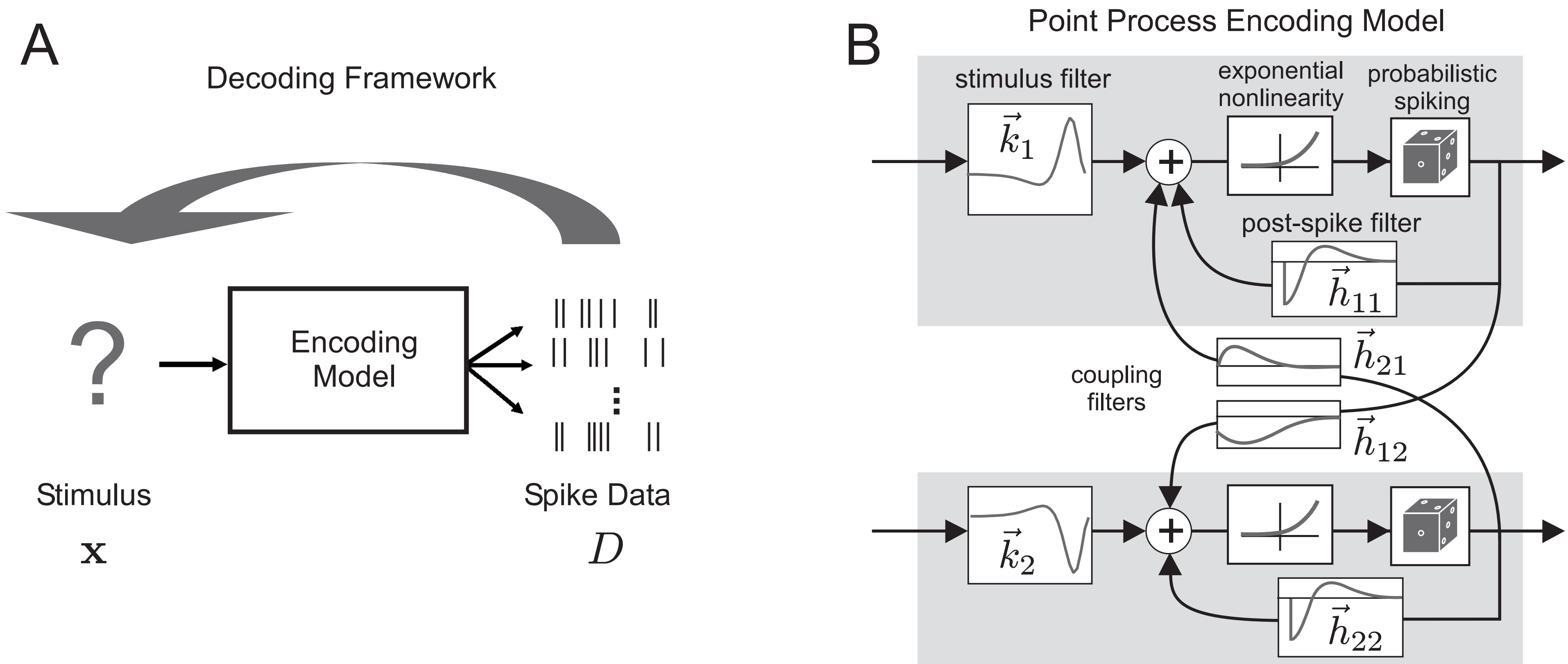
— extracellularly-recorded responses of populations of RGCs





Stimulus: random spatiotemporal visual stimuli (Pillow et al., 2007)

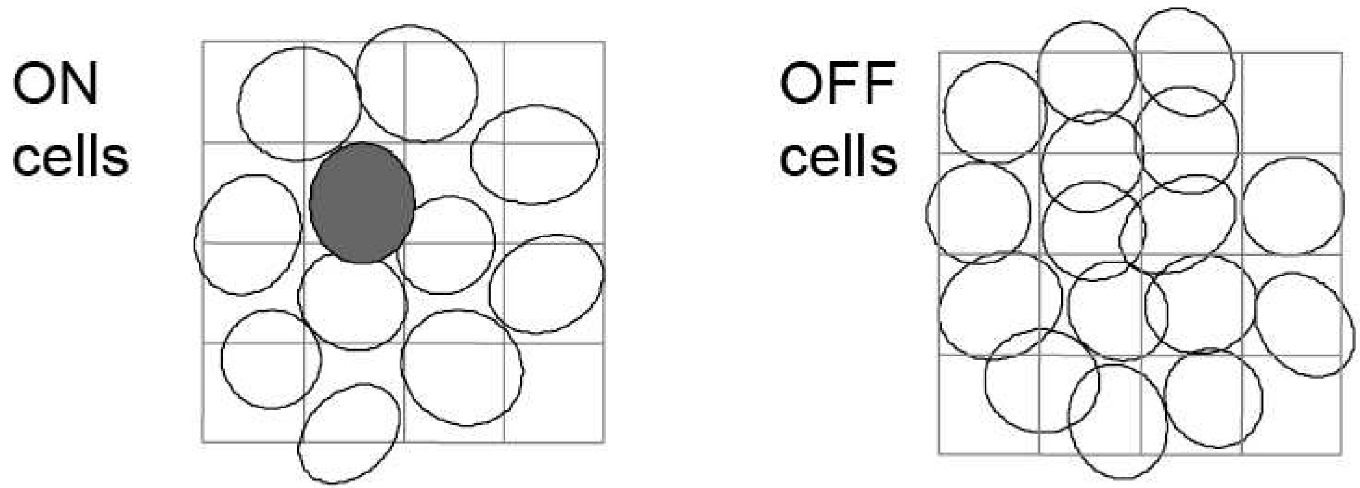
Multineuronal point-process GLM

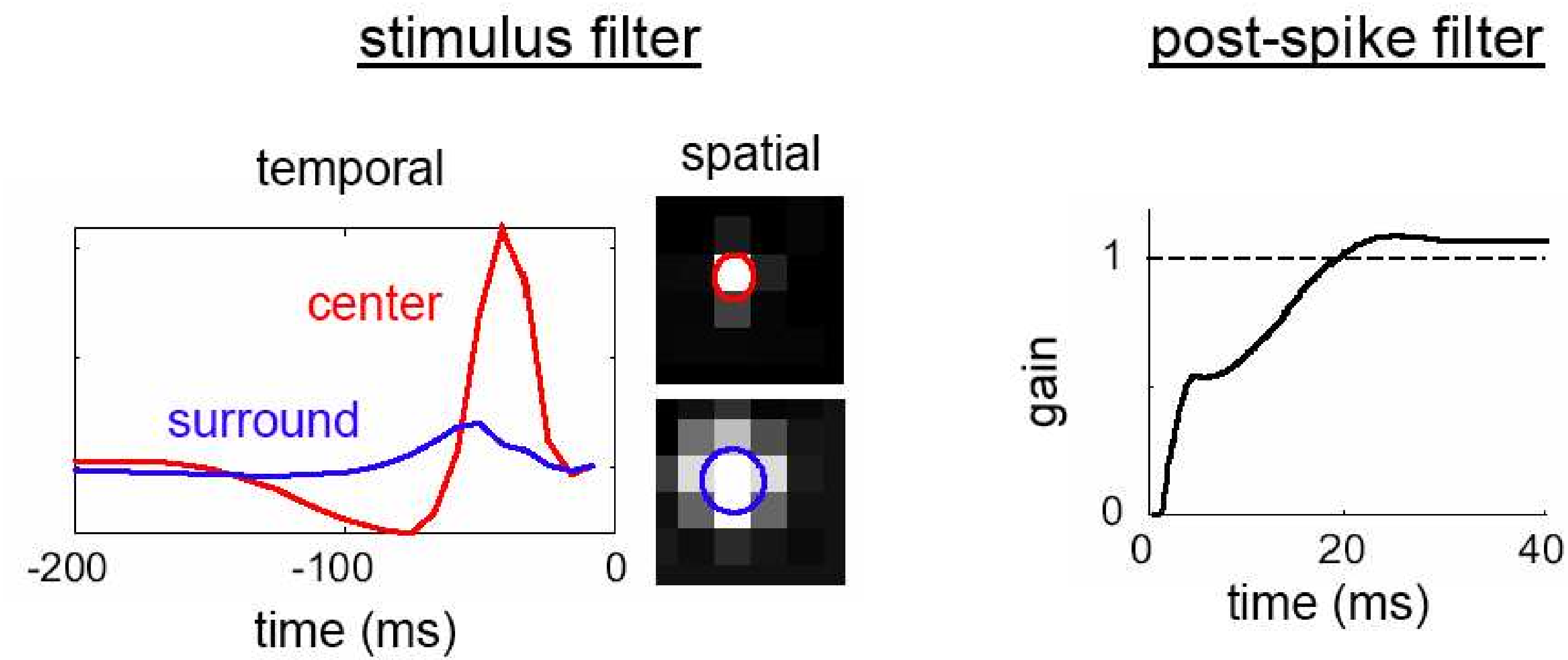


$$\lambda_i(t) = f\left(b + \vec{k}_i \cdot \vec{x}(t) + \sum_{i',j} h_{i',j} n_{i'}(t-j)\right),$$

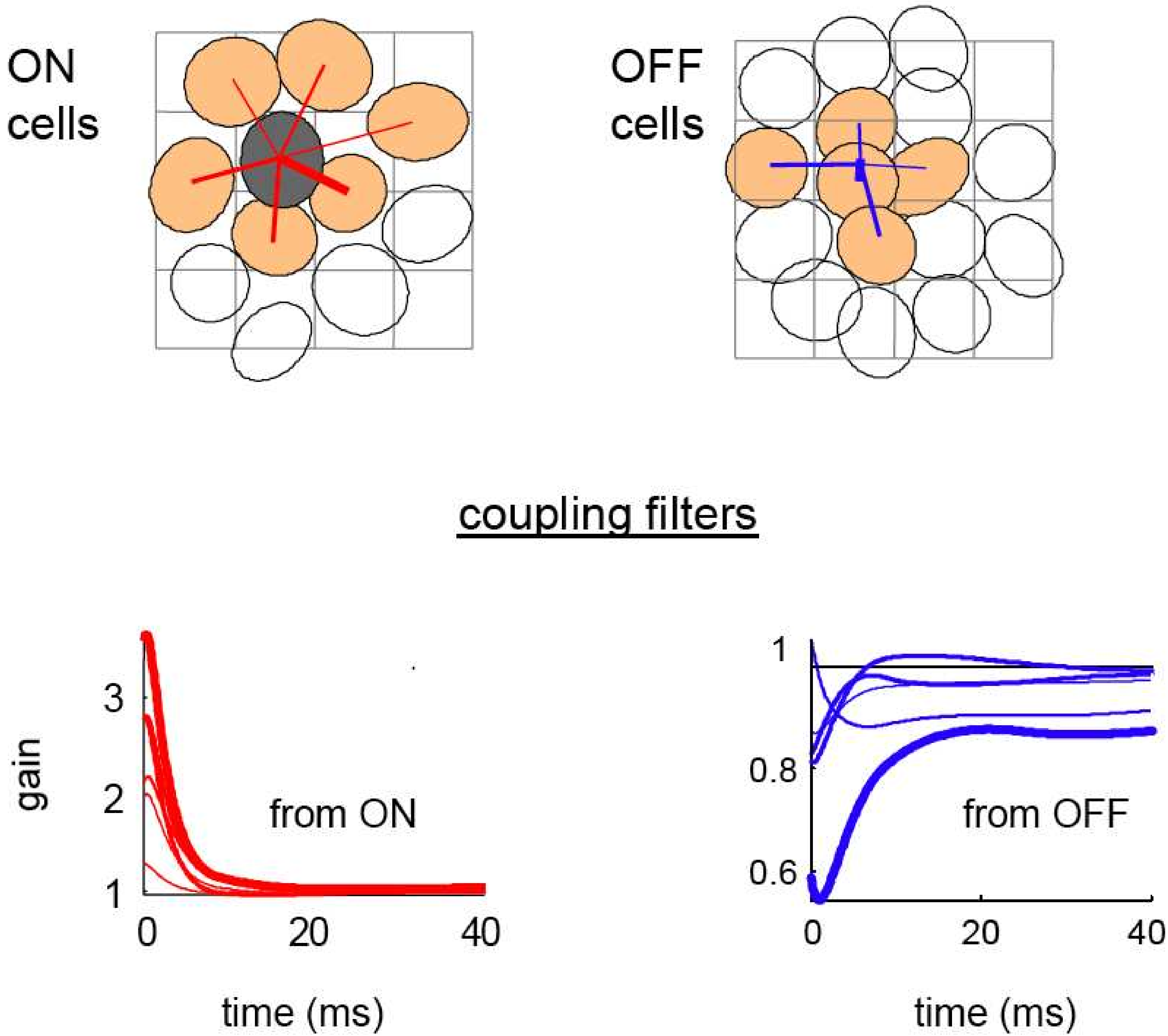
— Fit by L1-penalized max. likelihood (concave optimization) (Paninski, 2004)

— Semiparametric fit of link function: $f(.) \approx \exp(.)$



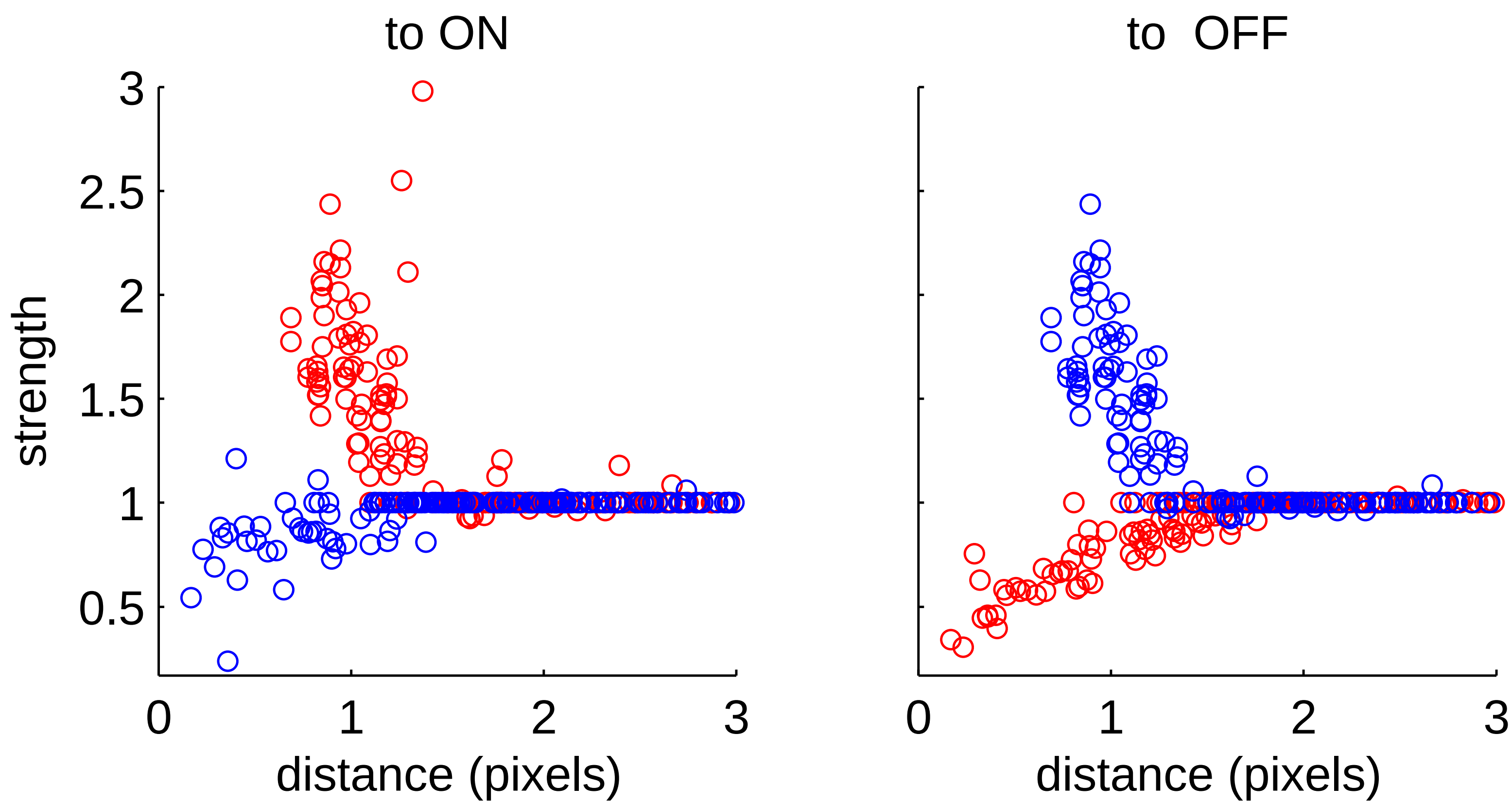


— θ_{stim} is well-approximated by a low-rank matrix (center-surround)

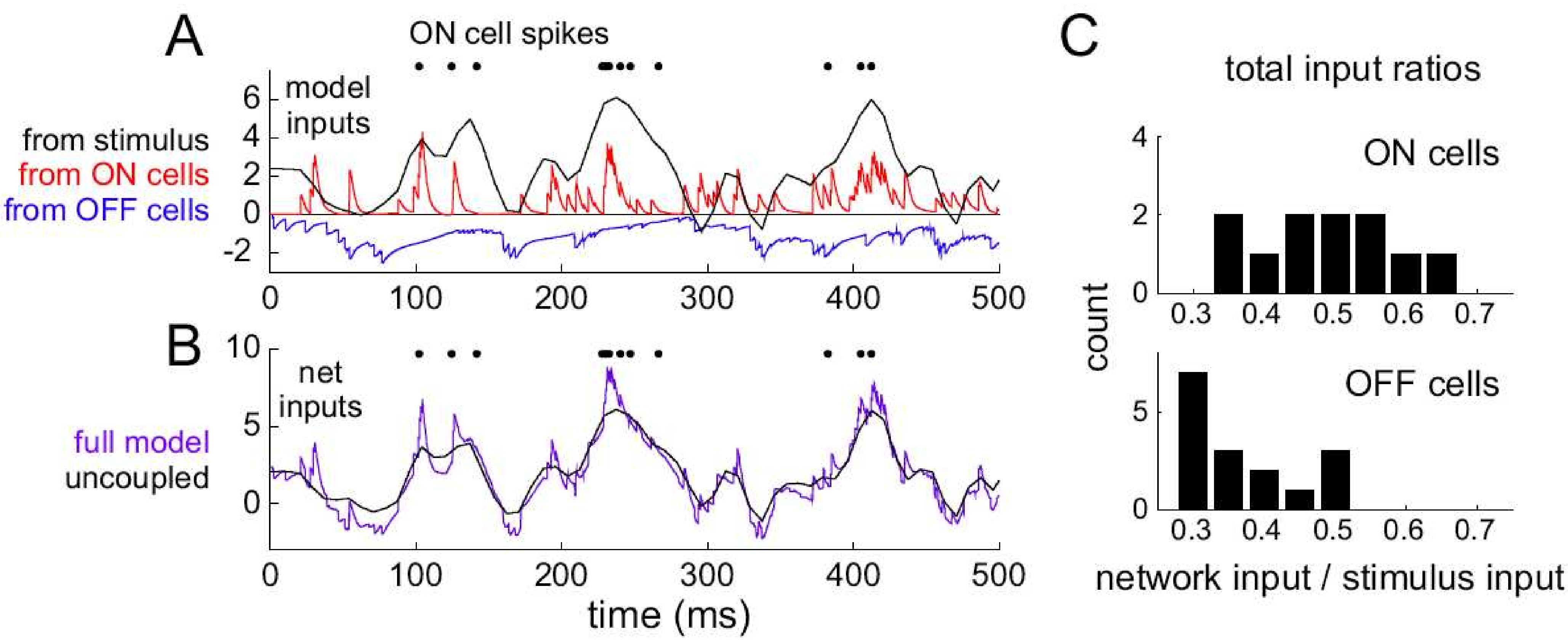


ON
cells
OFF
cells
coupling filters
gain
3
2
1
0
20
40
from ON
time (ms)
1
0.8
0.6
0
20
40
from OFF
time (ms)

Nearest-neighbor connectivity



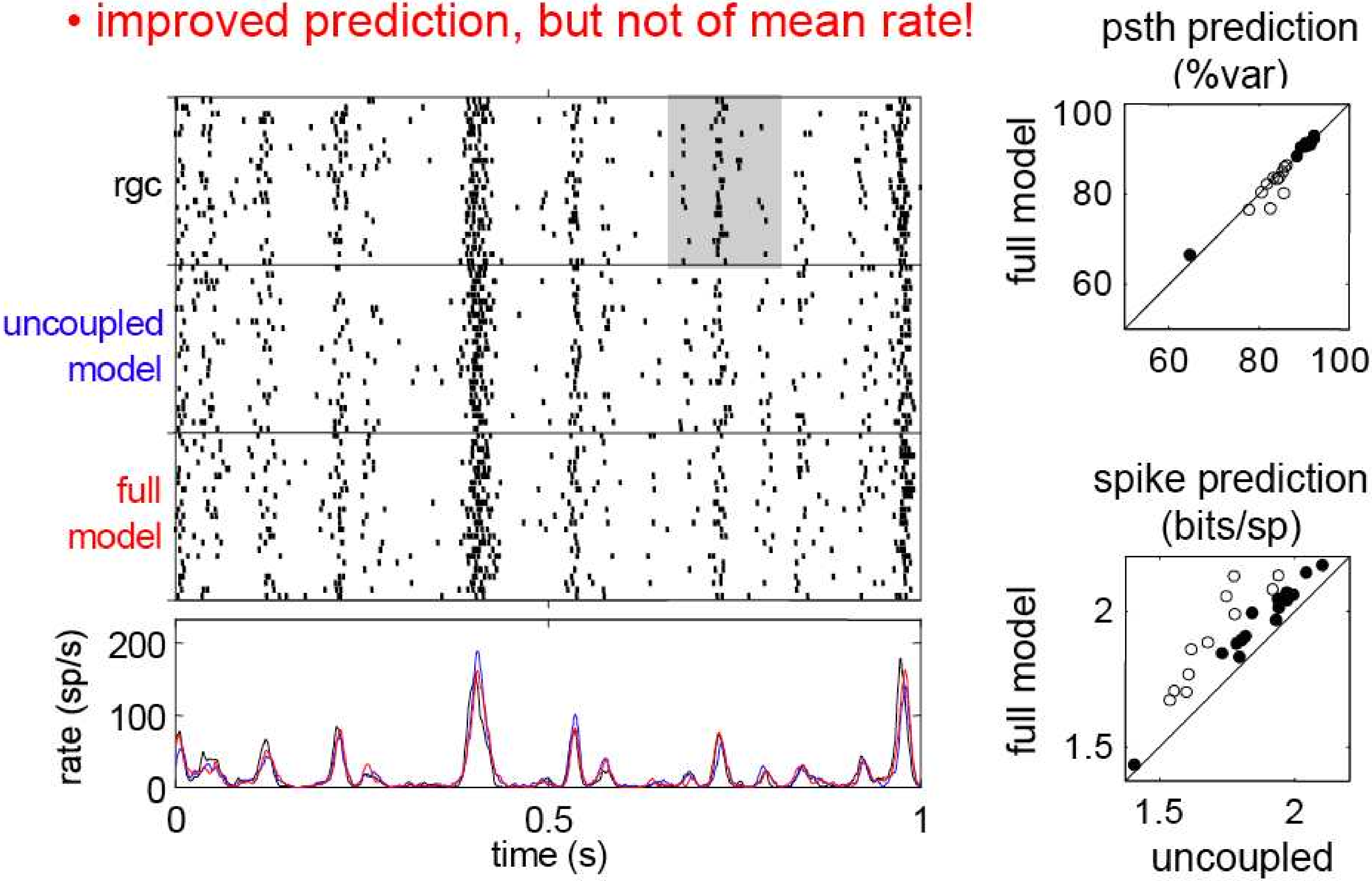
Network vs. stimulus drive



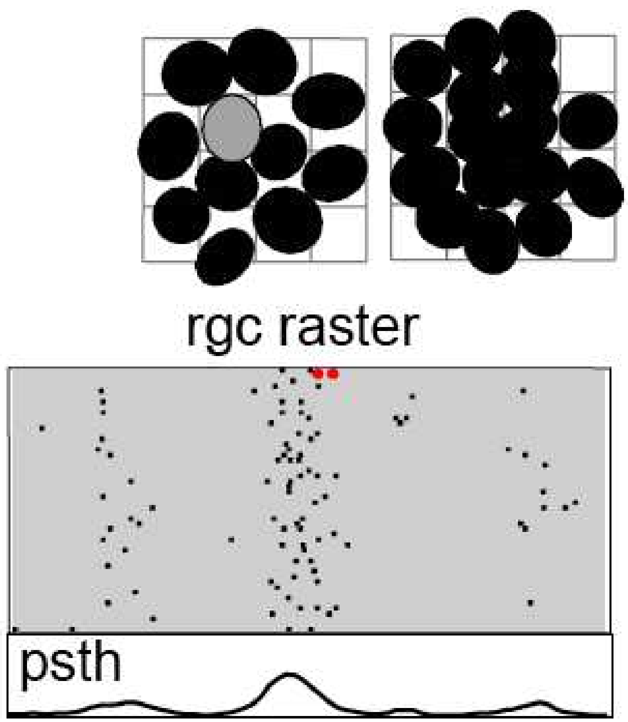
— Network effects are $\approx 50\%$ as strong as stimulus effects

Spike Train Prediction

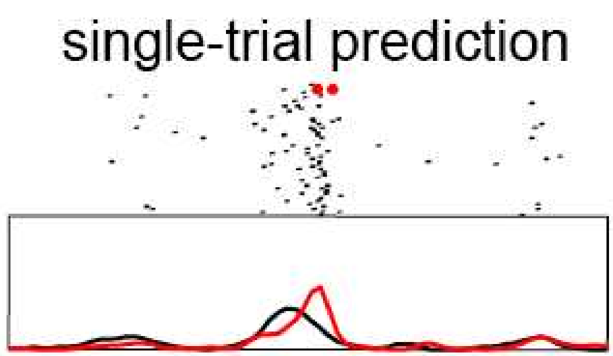
- improved prediction, but not of mean rate!



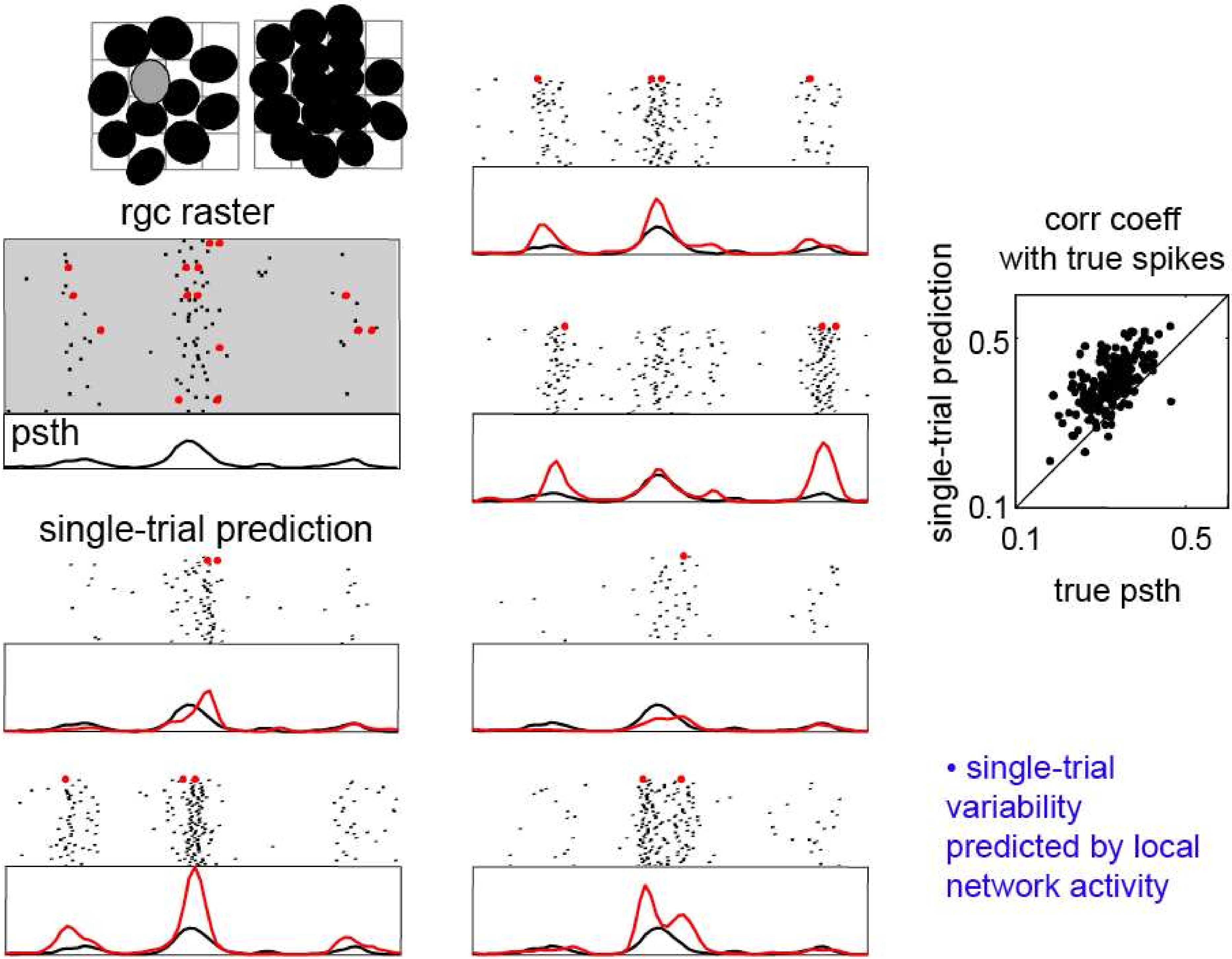
Network predictability analysis



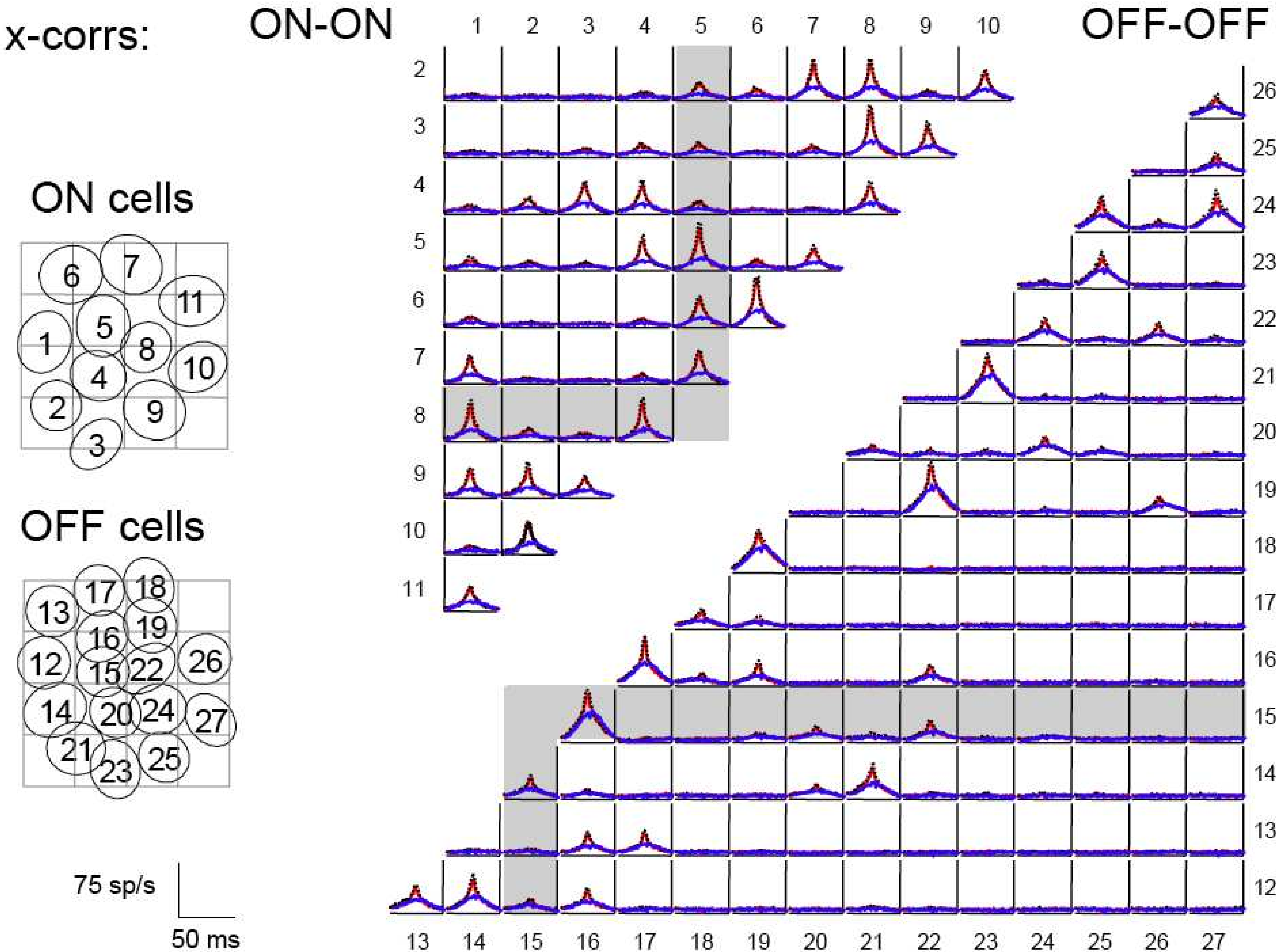
• fix all other neurons for a single trial

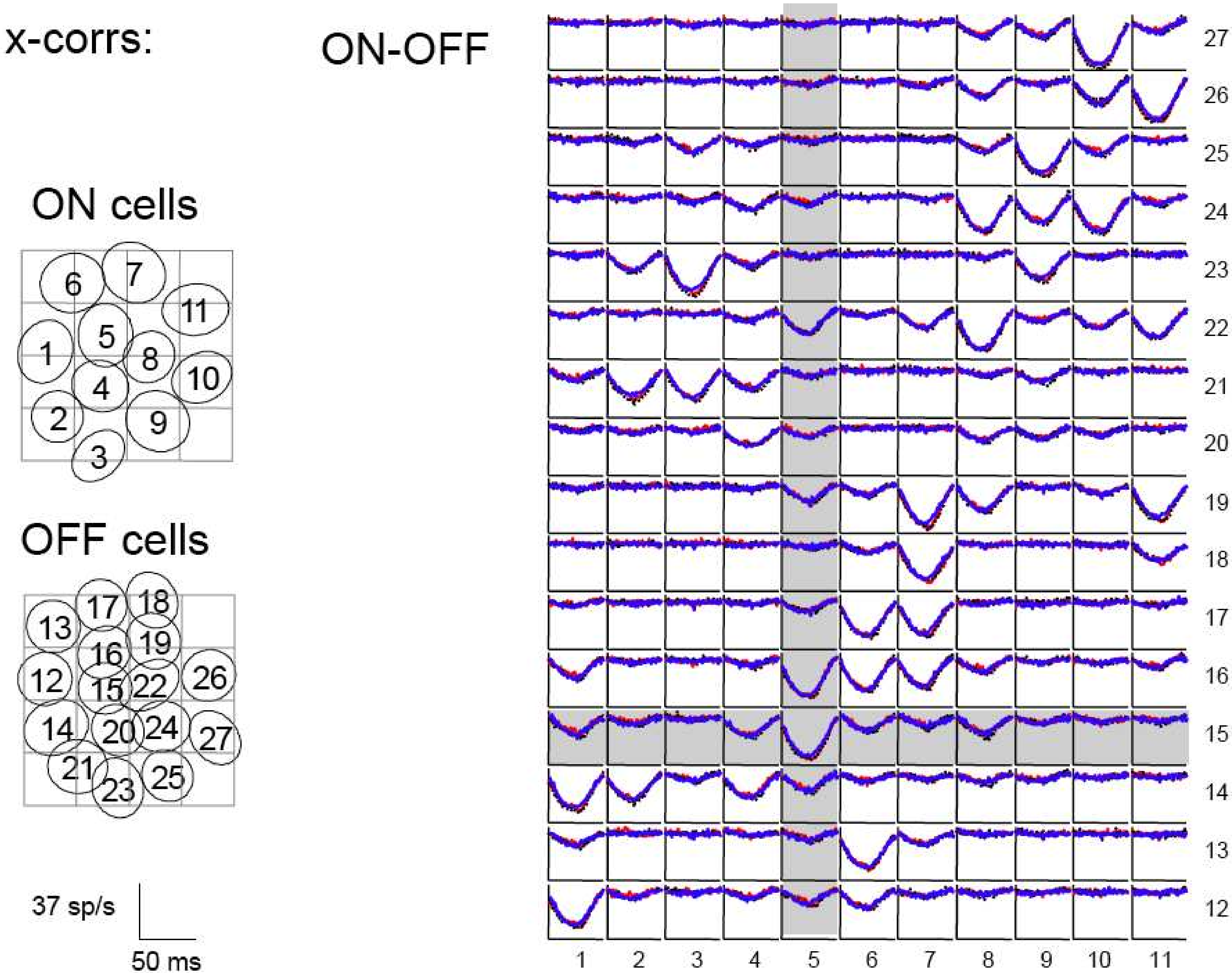


• draw single-trial predictions of this cell's spike train


rgc raster
psth
single-trial prediction
corr coeff
with true spikes
single-trial prediction
0.5
0.1
0.1
0.5
true psth
• single-trial
variability
predicted by local
network activity

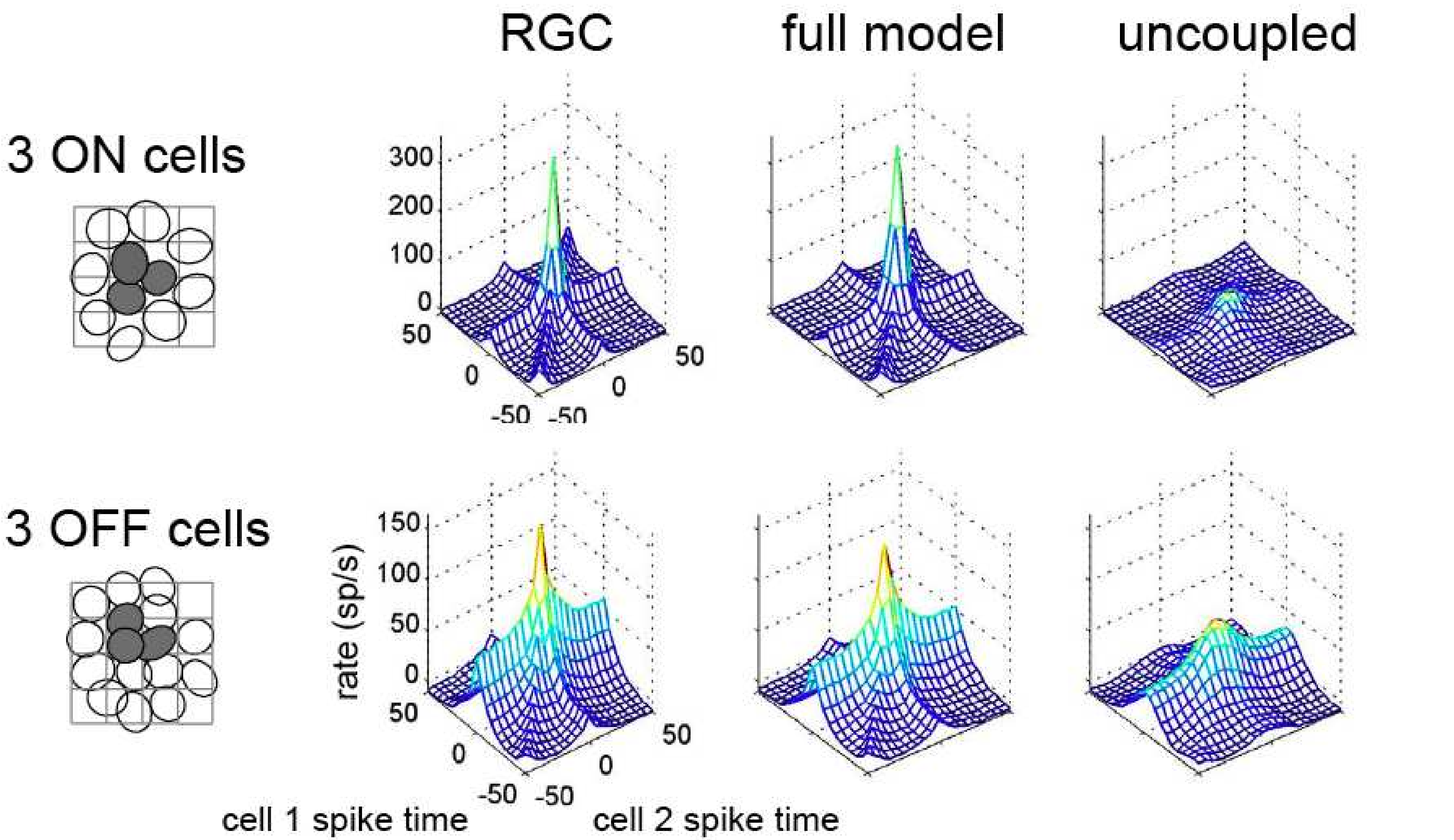
Model captures spatiotemporal cross-corrs



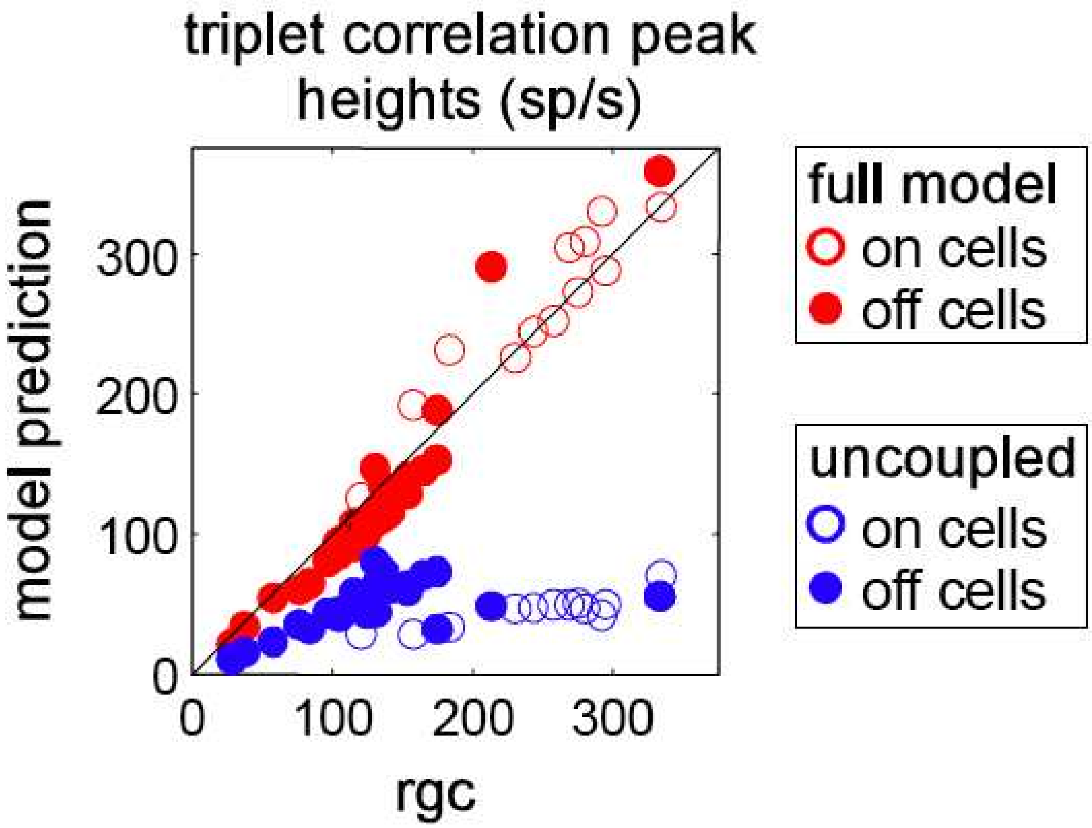


x-corrs:
ON-OFF
ON cells
6
7
11
1
5
8
10
4
2
9
3
OFF cells
17
18
13
16
19
12
15
22
26
14
20
24
27
21
25
23
37 sp/s
50 ms
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
1
2
3
4
5
6
7
8
9
10
11

Triplet correlations



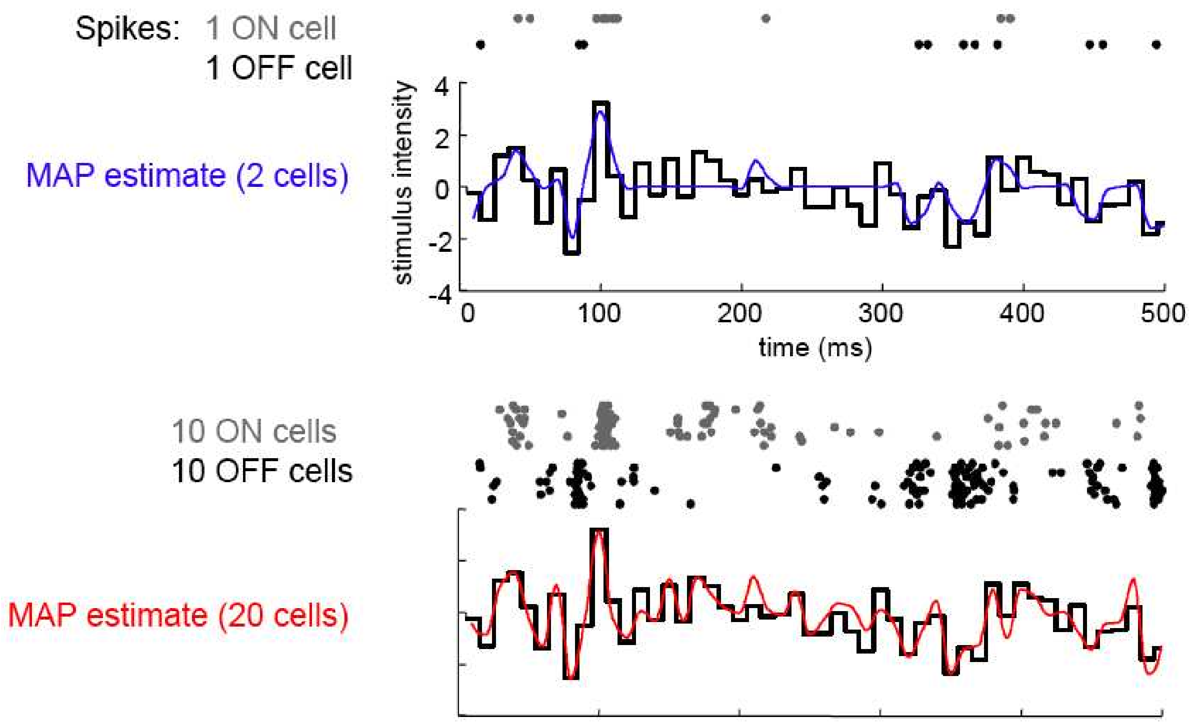
Triplet correlations

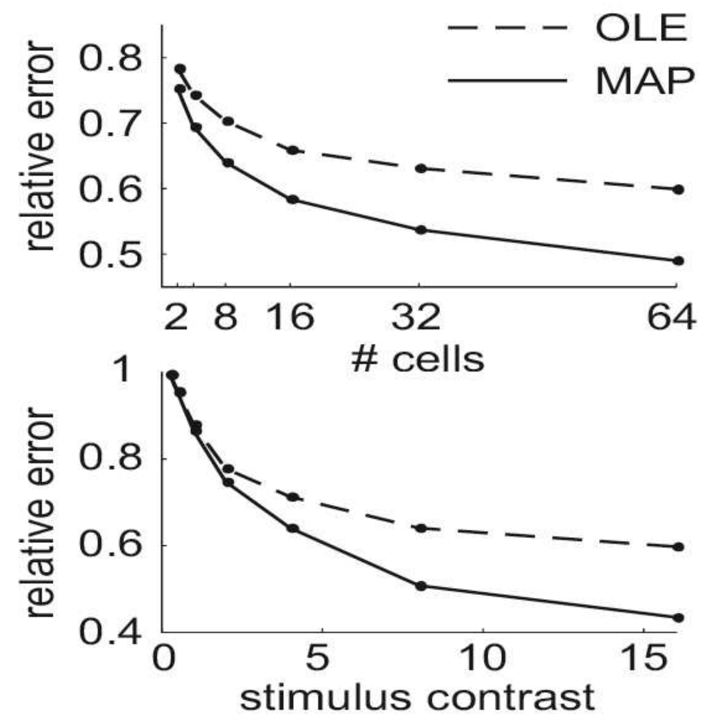


Maximum a posteriori decoding

$\arg\max_{\vec{x}} \log P(\vec{x}|spikes) = \arg\max_{\vec{x}} \log P(spikes|\vec{x}) + \log P(\vec{x})$

— $\log P(spikes|\vec{x})$ is concave in $\vec{x}$: concave optimization again.





Application: Laplace approximation

Key problem: how much information does network activity carry about the stimulus?

$$I(\vec{x}; D) = H(\vec{x}) - H(\vec{x}|D)$$

$H(\vec{x}|D) = \int h(\vec{x}|D)p(D)dD; \quad h(\vec{x}) = -\int p(\vec{x}) \log p(\vec{x}) d\vec{x}$

Laplace approx: $p(\vec{x}|D) \approx$ Gaussian with covariance $J_{x|D}^{-1}$.

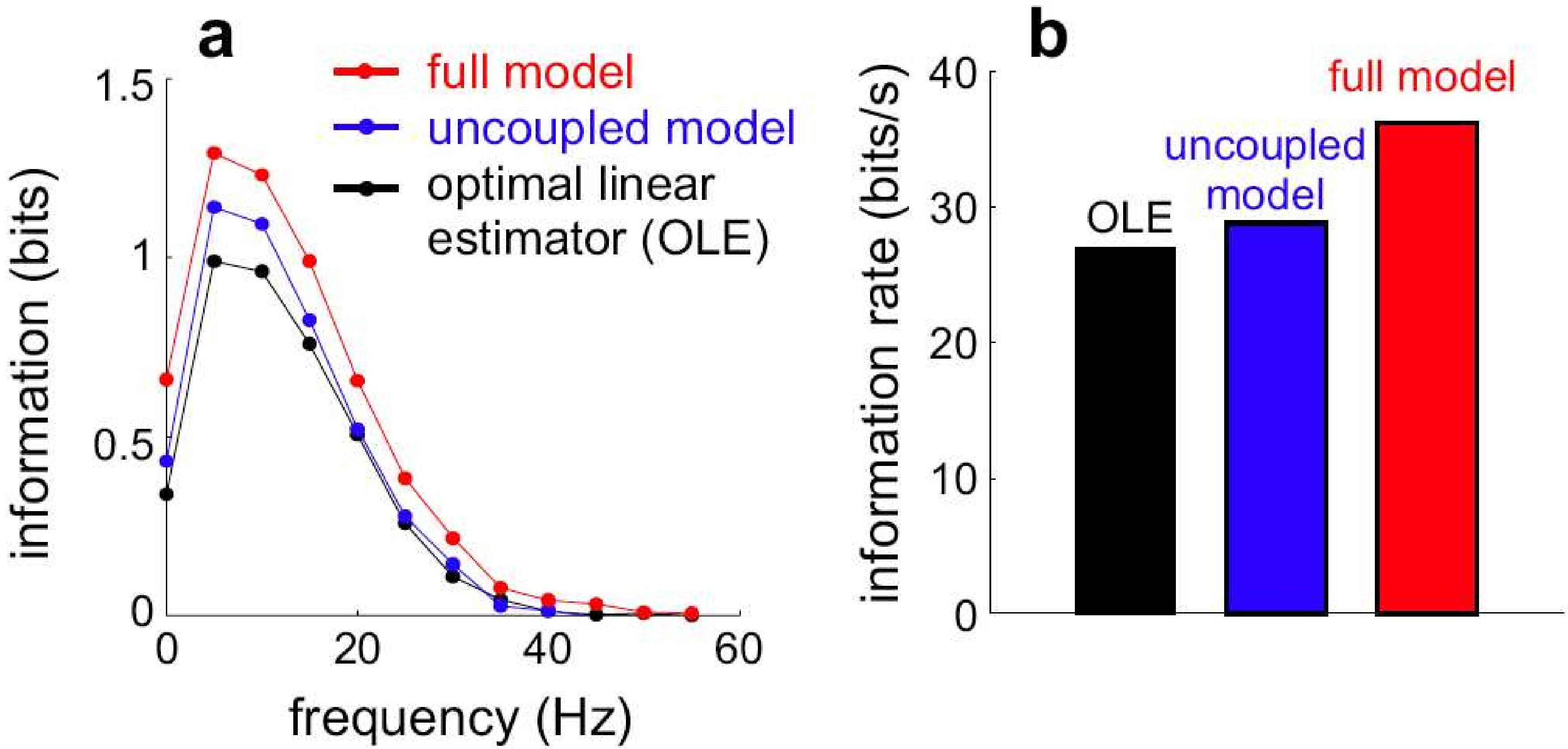
Entropy of this Gaussian: $c - \frac{1}{2} \log |J_{x|D}|$. So:

$$\begin{aligned} H(\vec{x}|D) &= \int h(\vec{x}|D)p(D)dD \\ &\approx c - \frac{1}{2} \int \log |J_{x|D}| p(D)dD \end{aligned}$$

— can sample from $p(D)$ easily, by sampling from $p(\vec{x})$, $p(D|\vec{x})$.

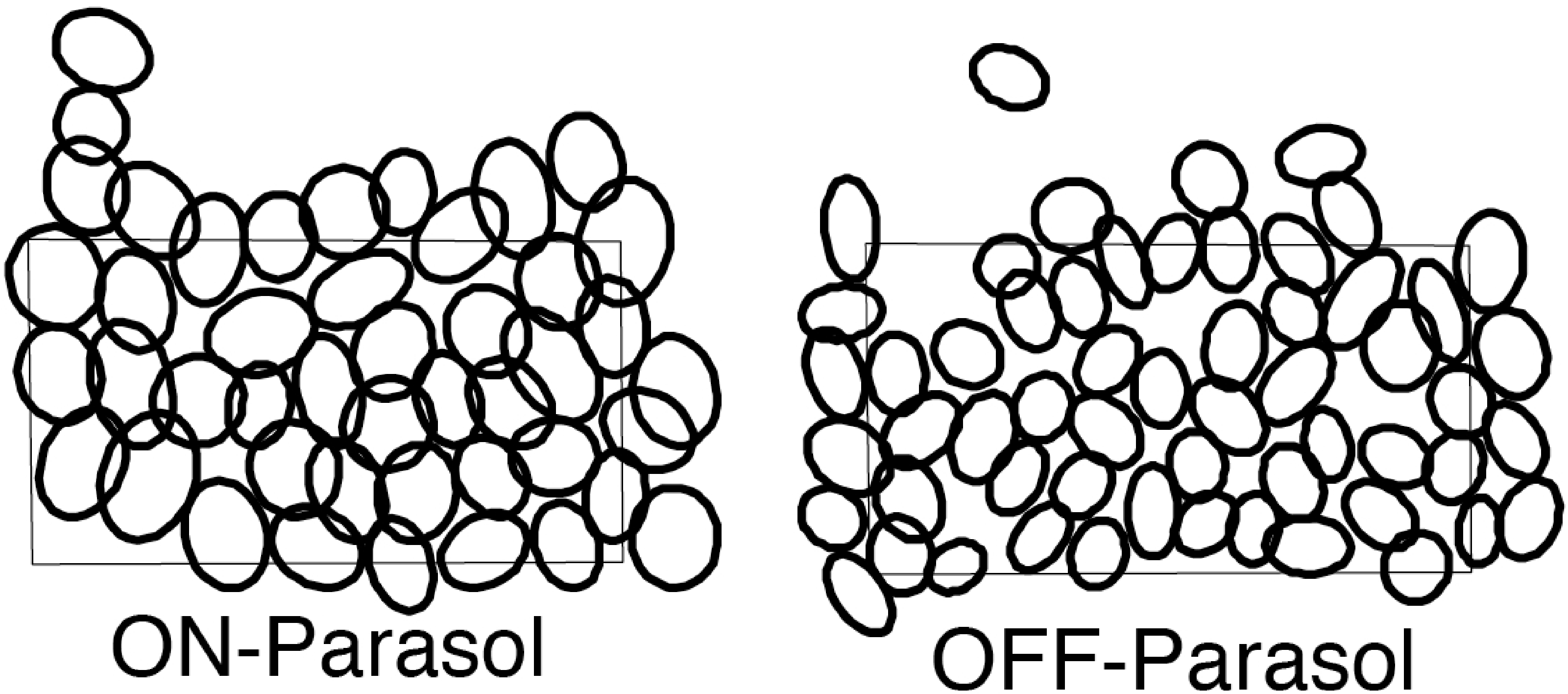
Can check accuracy by Monte Carlo on $p(\vec{x}|D)$ (log-concave, so easy to sample via hit-and-run).

Does including correlations improve decoding?



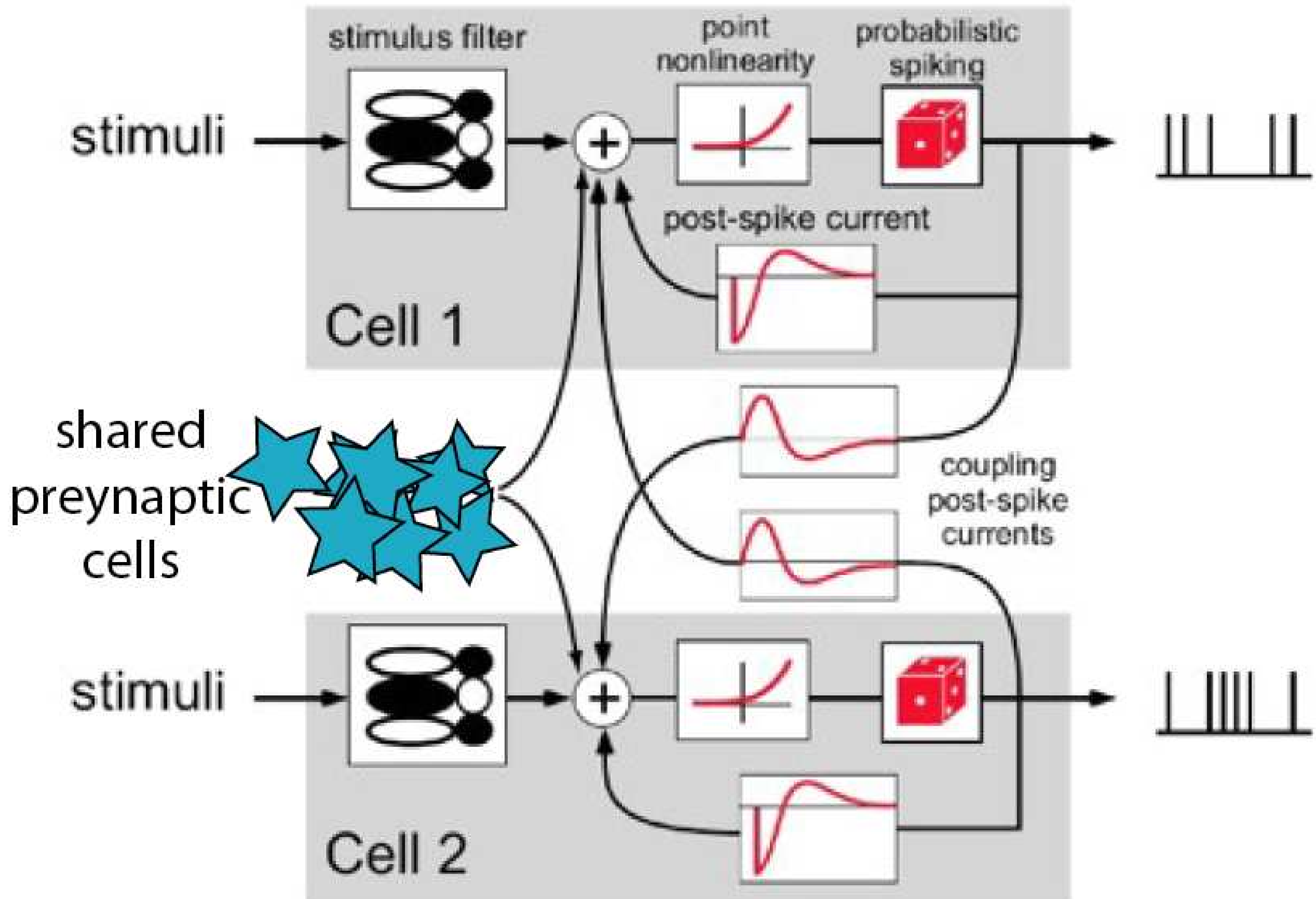
— Including network terms improves decoding accuracy.

Next: Large-scale network modeling



— Do observed local connectivity rules lead to interesting network dynamics? What are the implications for retinal information processing? Can we capture these effects with a reduced dynamical model?

Another extension: latent variable effects



State-space setting (Kulkarni and Paninski, 2007)

Part 2: Adaptive on-line experimental design

Goal: estimate θ from experimental data

Usual approach: draw stimuli i.i.d. from fixed $p(\vec{x})$

Adaptive approach: choose $p(\vec{x})$ on each trial to maximize $I(\theta; r|\vec{x})$ (e.g. "staircase" methods).

OK, now how do we actually do this in neural case?

- Computing $I(\theta; r|\vec{x})$ requires an integration over θ
  — in general, exponentially hard in $\dim(\theta)$
- Maximizing $I(\theta; r|\vec{x})$ in $\vec{x}$ is doubly hard
  — in general, exponentially hard in $\dim(\vec{x})$

Doing all this in real time ($\sim$ 10-100 msec) is a major challenge!

Joint work w/ J. Lewi, R. Butera, Georgia Tech. (Lewi et al., 2006)

Three key steps

1. Choose a tractable, flexible model of neural encoding
2. Choose a tractable, accurate approximation of the posterior $p(\vec{\theta}|\{\vec{x}_i, r_i\}_{i \leq N})$
3. Use approximations and some perturbation theory to reduce optimization problem to a simple 1-d linesearch

Step 1: GLM likelihood

$$\lambda_i \sim Poiss(\lambda_i)$$
$$\lambda_i | \vec{x}_i, \vec{\theta} = f(\vec{k} \cdot \vec{x}_i + \sum_j a_j r_{i-j})$$

$$\log p(r_i | \vec{x}_i, \vec{\theta}) = -f(\vec{k} \cdot \vec{x}_i + \sum_j a_j r_{i-j}) + r_i \log f(\vec{k} \cdot \vec{x}_i + \sum_j a_j r_{i-j})$$

Two key points:

- Likelihood is "rank-1" — only depends on $\vec{\theta}$ along $\vec{z} = (\vec{x}, \vec{r})$.
- f convex and log-concave $\implies$ log-likelihood concave in $\vec{\theta}$

Step 2: representing the posterior

Idea: Laplace approximation

$$p(\vec{\theta}|\{\vec{x}_i, r_i\}_{i\leq N}) \approx \mathcal{N}(\mu_N, C_N)$$

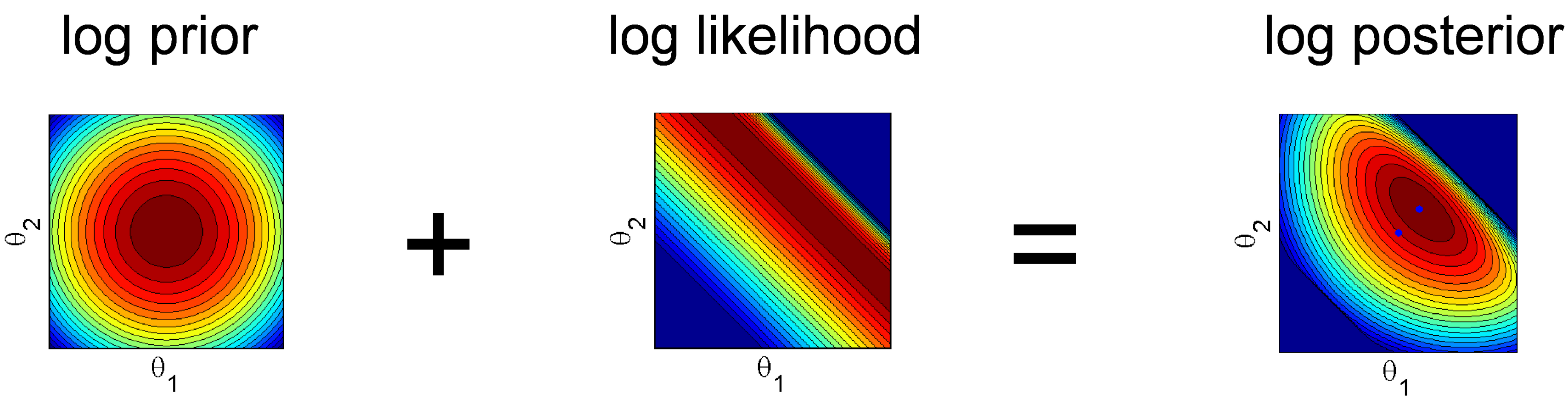
Justification:

- posterior CLT (Paninski, 2005)
- likelihood is log-concave, so posterior is also log-concave:

$$\log p(\vec{\theta}|\{\vec{x}_i, r_i\}_{i\leq N}) \sim \log p(\vec{\theta}|\{\vec{x}_i, r_i\}_{i\leq N-1}) + \log p(r_N|x_N, \vec{\theta})$$

— Equivalent to an extended Kalman filter formulation

Efficient updating



Updating μ_N: one-d search

Updating C_N: rank-one update, $C_N = (C_{N-1}^{-1} + b\vec{z}^t\vec{z})^{-1}$ — use Woodbury lemma

Total time for update of posterior: $O(d^2)$

Step 3: Efficient stimulus optimization

Laplace approximation $\implies I(\theta; r|\vec{x}) \sim E_{r|\vec{x}} \log \frac{|C_{N-1}|}{|C_N|}$

— this is nonlinear and difficult, but we can simplify using perturbation theory: $\log |I + A| \approx \text{trace}(A)$.

Now we can take averages over $p(r|\vec{x}) = \int p(r|\theta, \vec{x}) p_N(\theta) d\theta$: standard Fisher info calculation given Poisson assumption on r.

Further assuming $f(.) = \exp(.)$ allows us to compute expectation exactly, using m.g.f. of Gaussian.

...finally, we want to maximize $F(\vec{x}) = g(\mu_N \cdot \vec{x}) h(\vec{x}^t C_N \vec{x})$.

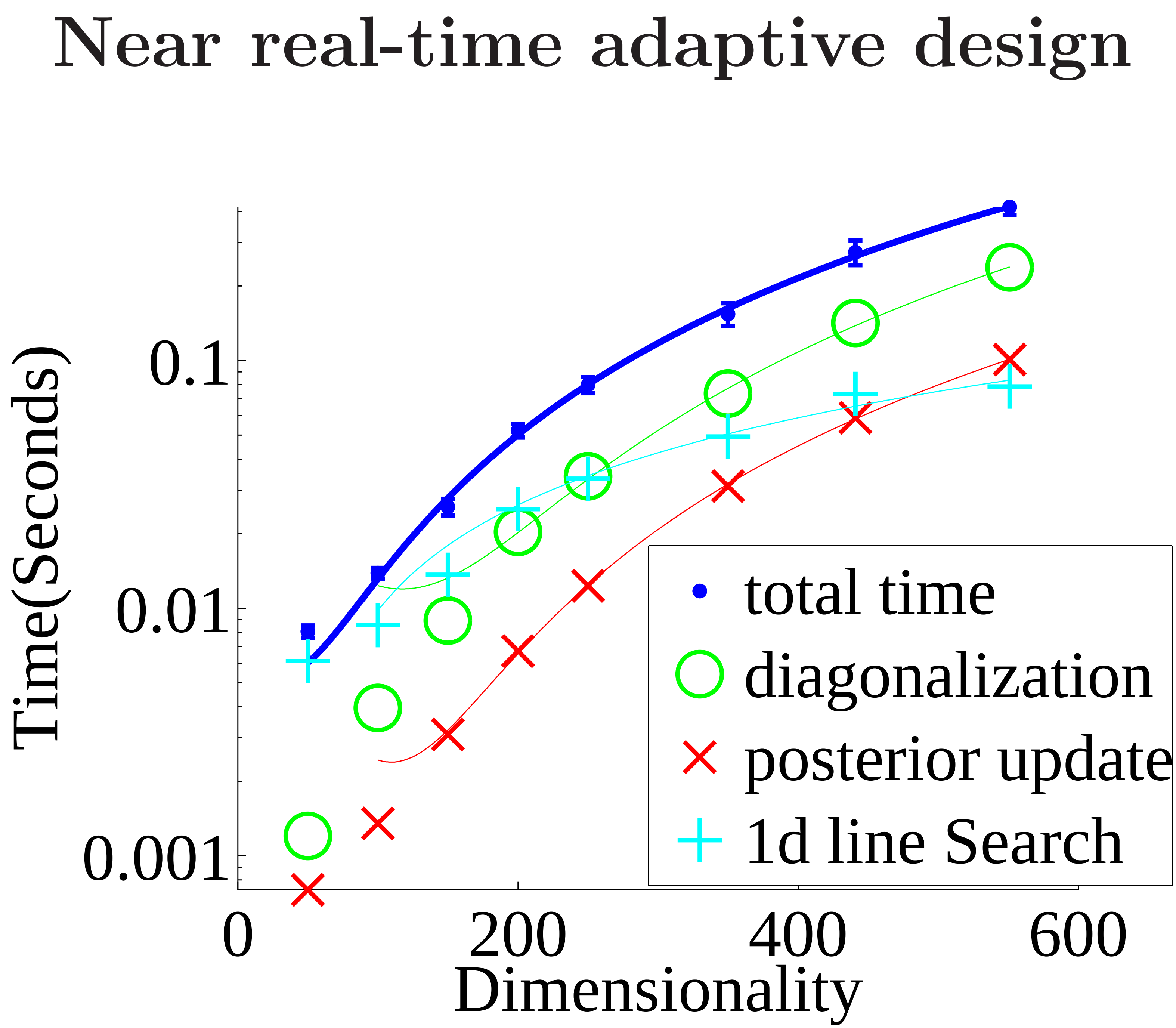
Computing the optimal $\vec{x}$

$\max_{\vec{x}} g(\mu_N \cdot \vec{x}) h(\vec{x}^t C_N \vec{x})$ increases with $||\vec{x}||_2$: constraining $||\vec{x}||_2$ reduces problem to nonlinear eigenvalue problem.

Lagrange multiplier approach (Berkes and Wiskott, 2006) reduces problem to 1-d linesearch, once eigendecomposition is computed — much easier than full d-dimensional optimization!

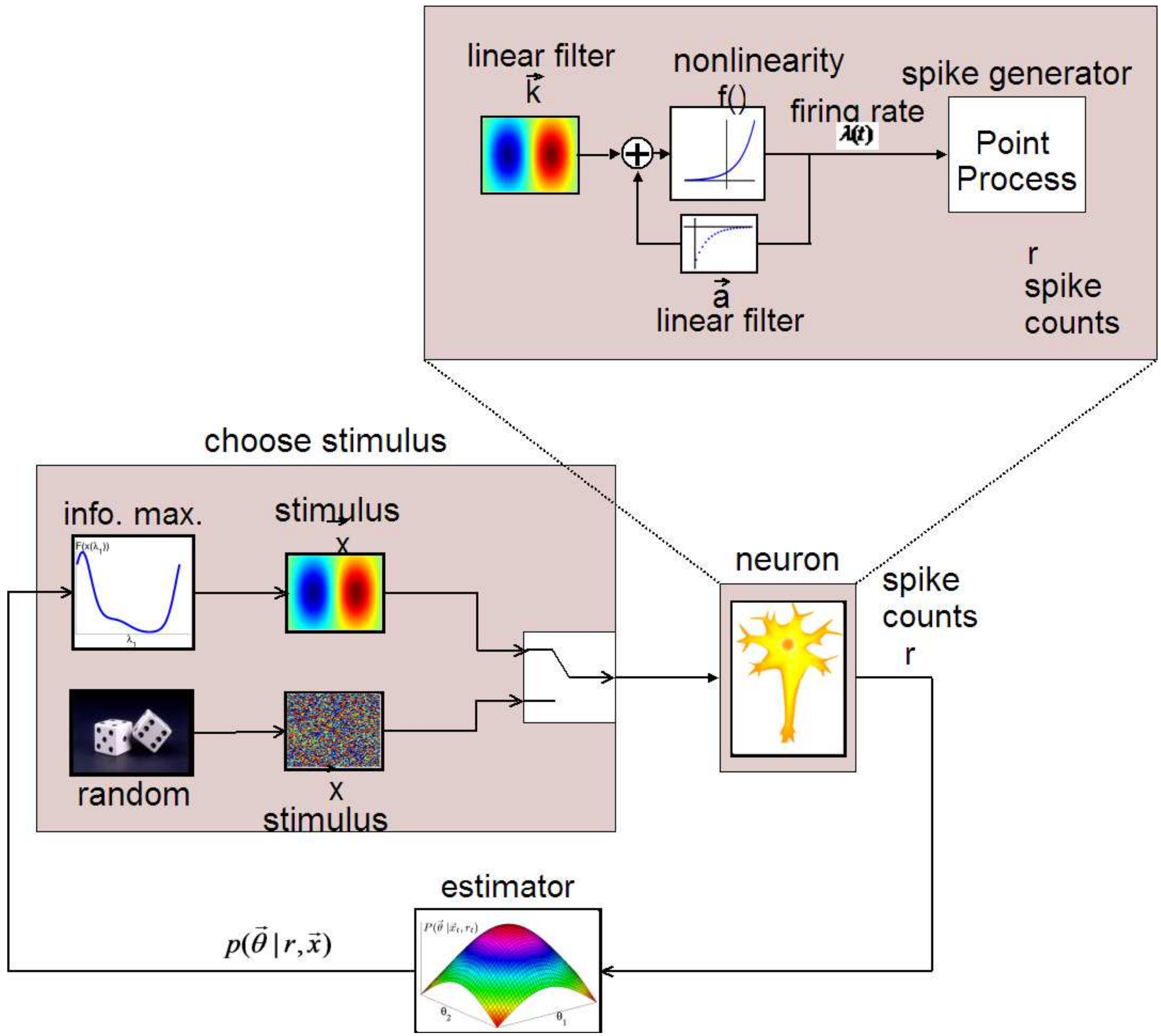
Rank-one update of eigendecomposition may be performed in $O(d^2)$ time (Gu and Eisenstat, 1994).

$\implies$ Computing optimal stimulus takes $O(d^2)$ time.

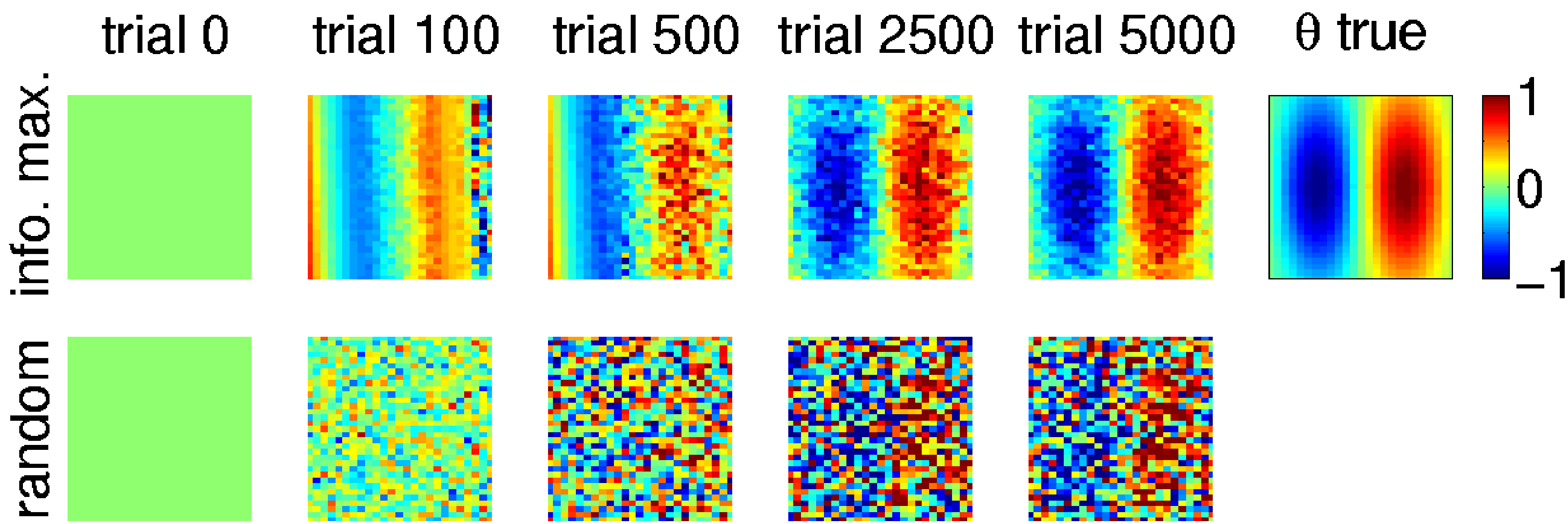


Near real-time adaptive design
Time(Seconds)
0.1
0.01
0.001
0
200
400
600
Dimensionality
total time
diagonalization
posterior update
1d line Search

Simulation overview



Gabor example



— infomax approach is an order of magnitude more efficient.

Conclusions

- GLM provides flexible, powerful methods for answering key questions in neuroscience
- Close relationships between encoding, decoding, and experimental design (Paninski et al., 2008)
- Log-concavity makes computations very tractable
- Many opportunities for machine learning techniques in neuroscience

Collaborators

Theory and numerical methods

- Y. Ahmadian, S. Escola, G. Fudenberg, Q. Huys, J. Kulkarni, M. Nikitchenko, K. Rahnama, G. Szirtes, T. Toyoizumi, Columbia
- E. Simoncelli, NYU
- A. Haith, C. Williams, Edinburgh
- M. Ahrens, J. Pillow, M. Sahani, Gatsby
- J. Lewi, Georgia Tech
- J. Vogelstein, Johns Hopkins

Retinal physiology

- E.J. Chichilnisky, J. Shlens, V. Uzzell, Salk

Cortical *in vitro* physiology

- B. Lau and A. Reyes, NYU

References

Berkes, P. and Wiskott, L. (2006). On the analysis and interpretation of inhomogeneous quadratic forms as receptive fields. *Neural Computation*, 18:1868–1895.

Foldiak, P. (2001). Stimulus optimisation in primary visual cortex. *Neurocomputing*, 38–40:1217–1222.

Gu, M. and Eisenstat, S. (1994). A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem. *SIAM J. Matrix Anal. Appl.*, 15(4):1266–1276.

Kulkarni, J. and Paninski, L. (2007). Common-input models for multiple neural spike-train data. *Network: Computation in Neural Systems*, In press.

Lewi, J., Butera, R., and Paninski, L. (2006). Real-time adaptive information-theoretic optimization of neurophysiological experiments. *NIPS*.

Machens, C. (2002). Adaptive sampling by information maximization. *Physical Review Letters*, 88:228104–228107.

Nemenman, I., Shafee, F., and Bialek, W. (2002). Entropy and inference, revisited. *NIPS*, 14.

Paninski, L. (2003). Estimation of entropy and mutual information. *Neural Computation*, 15:1191–1253.

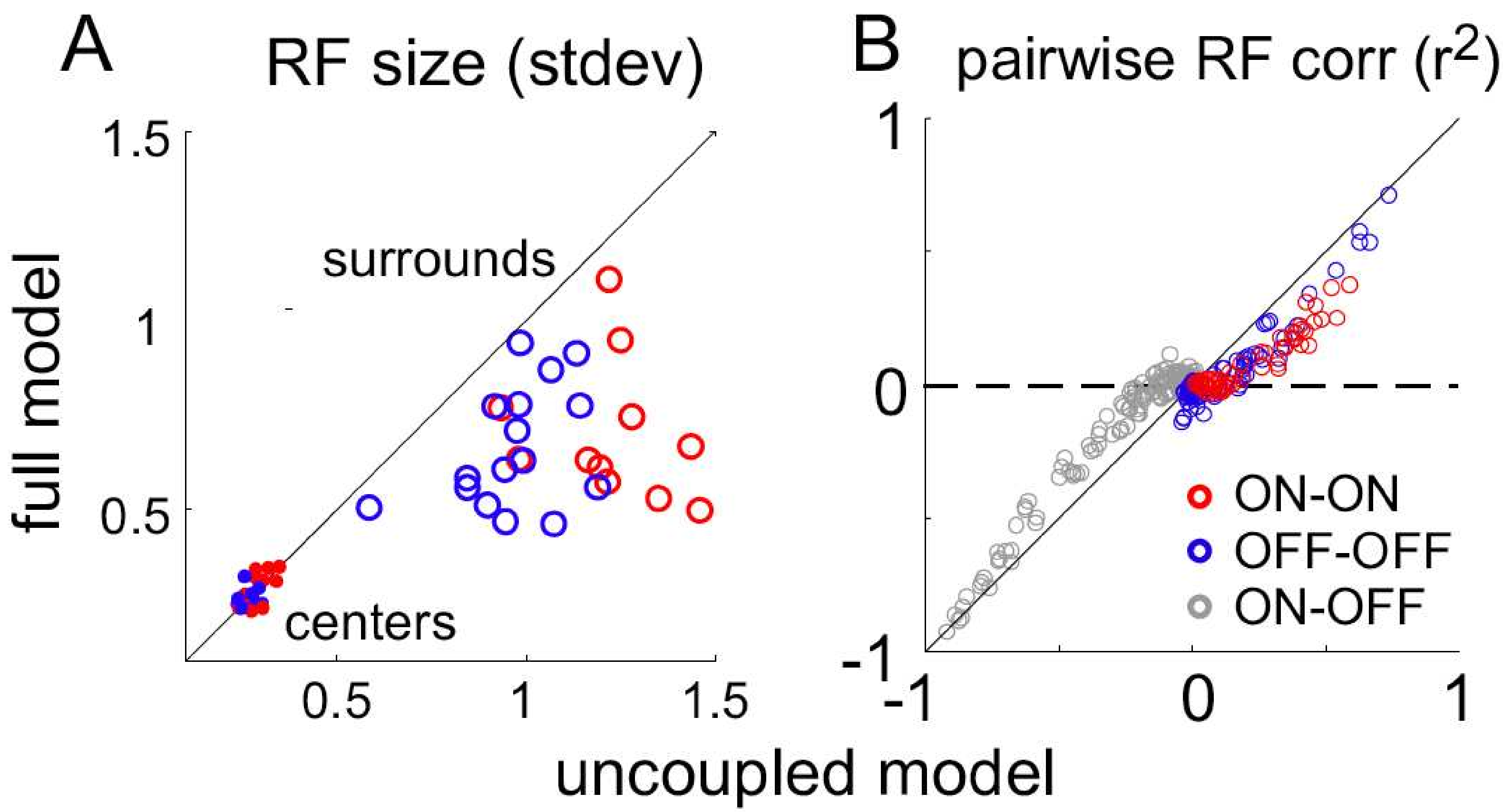
Paninski, L. (2004). Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15:243–262.

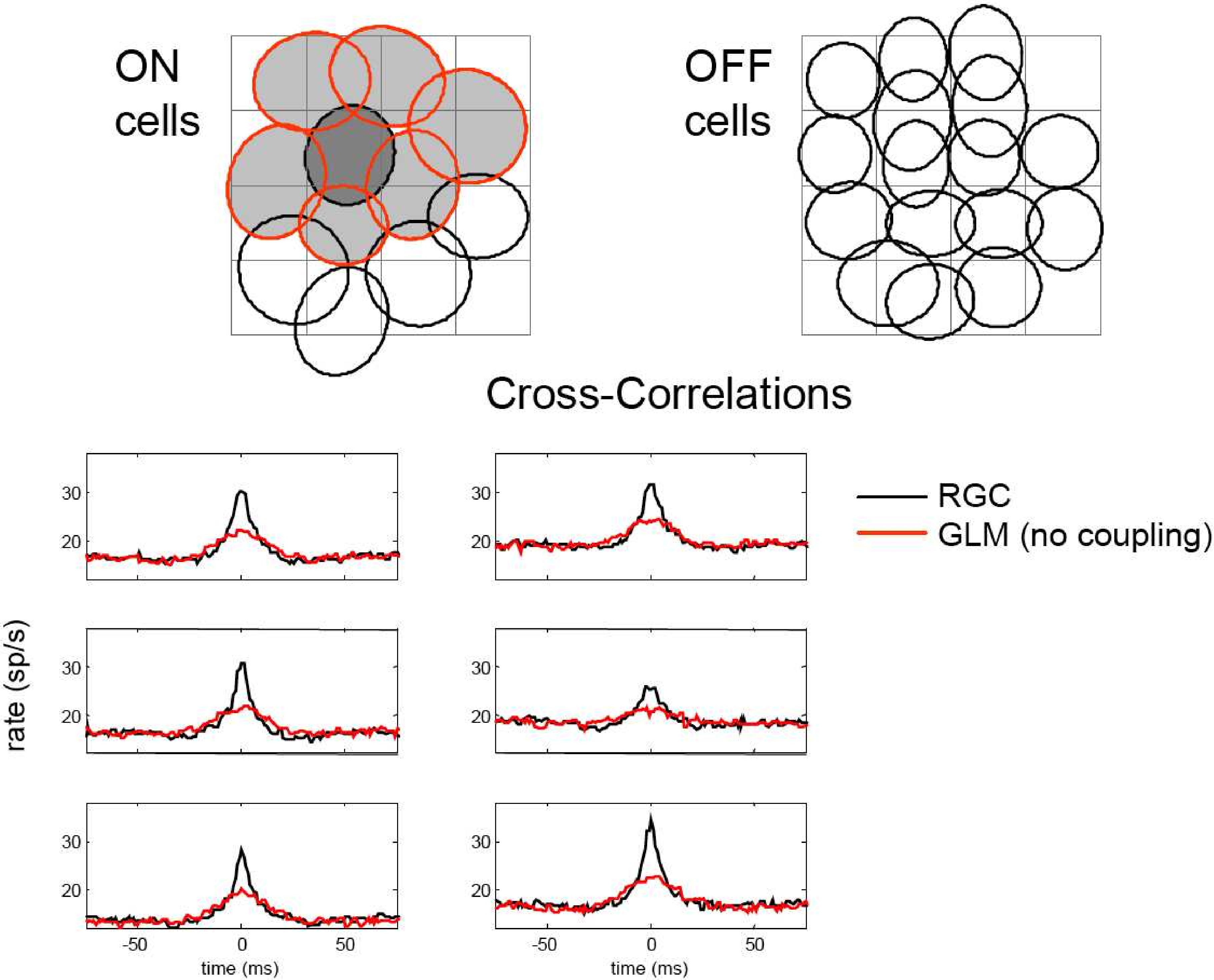
Paninski, L. (2005). Asymptotic theory of information-theoretic experimental design. *Neural Computation*, 17:1480–1507.

Paninski, L., Pillow, J., and Lewi, J. (2008). Statistical models for neural encoding, decoding, and optimal stimulus design. In Cisek, P., Drew, T., and Kalaska, J., editors, *Computational Neuroscience: Progress in Brain Research*. Elsevier.

Pillow, J., Shlens, J., Paninski, L., Simoncelli, E., and Chichilnisky, E. (2007). Visual information coding in multi-neuronal spike trains. *Under review, Nature*.

Fitting coupling terms exposes smaller receptive fields





ON
cells
OFF
cells
Cross-Correlations
RGC
GLM (no coupling)
rate (sp/s)
30
20
-50
0
50
time (ms)

Handling nonstationary parameters

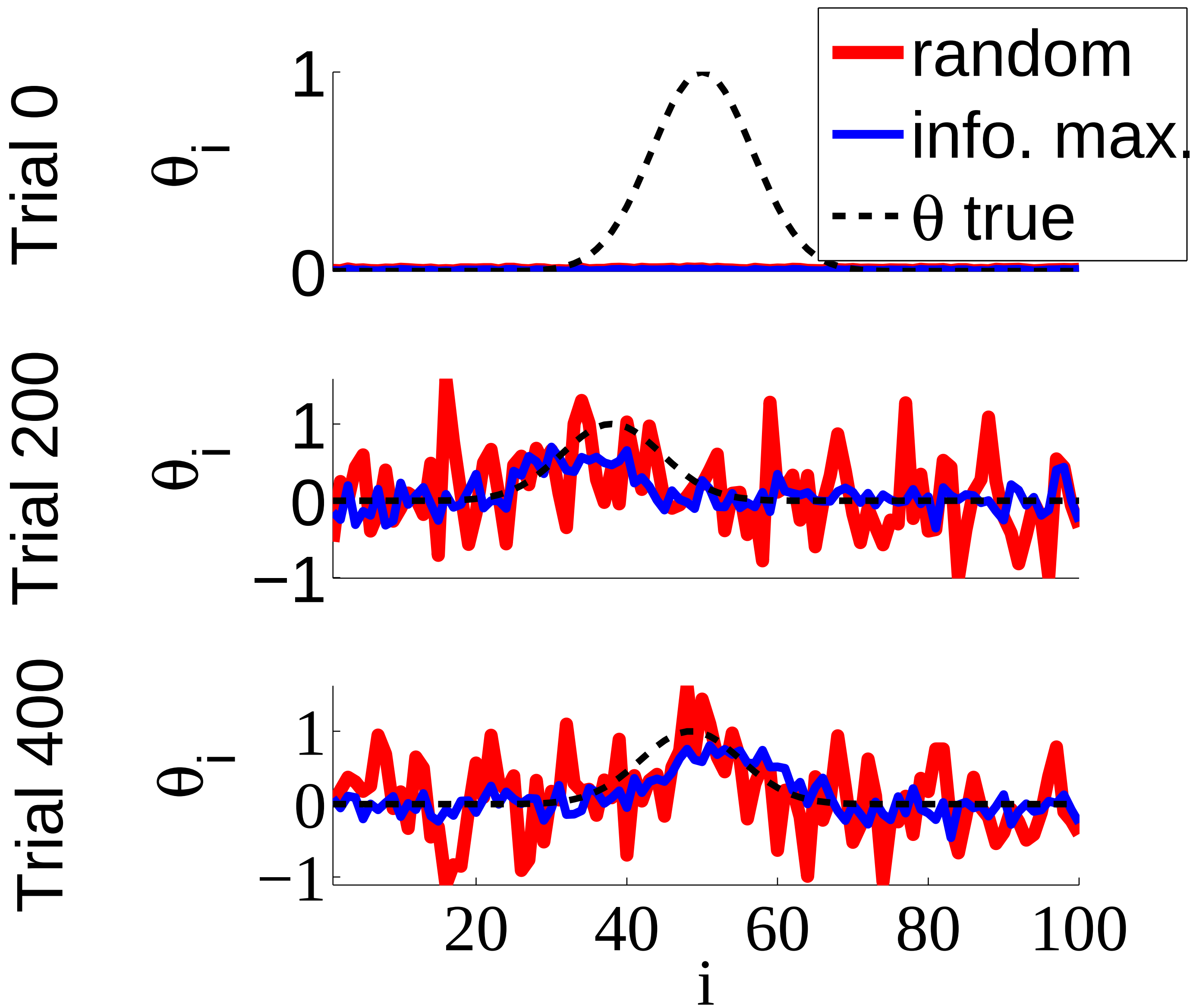
Various sources of nonsystematic nonstationarity:

- Eye position drift
- Changes in arousal / attentive state
- Changes in health / excitability of preparation

Solution: allow diffusion in extended Kalman filter:

$$\vec{\theta}_{N+1} = \vec{\theta}_N + \epsilon; \quad \epsilon \sim \mathcal{N}(0, Q)$$

Nonstationary example



Nonstationary example