Statistical Machine Learning Methods for Large-scale Neural Data Analysis

Gonzalo E. Mena

March 2nd, 2018

Columbia University

Neural Data Science

- Several large-scale imaging and stimulation technologies. To read and write neural activity.
- Consensus on relevance.

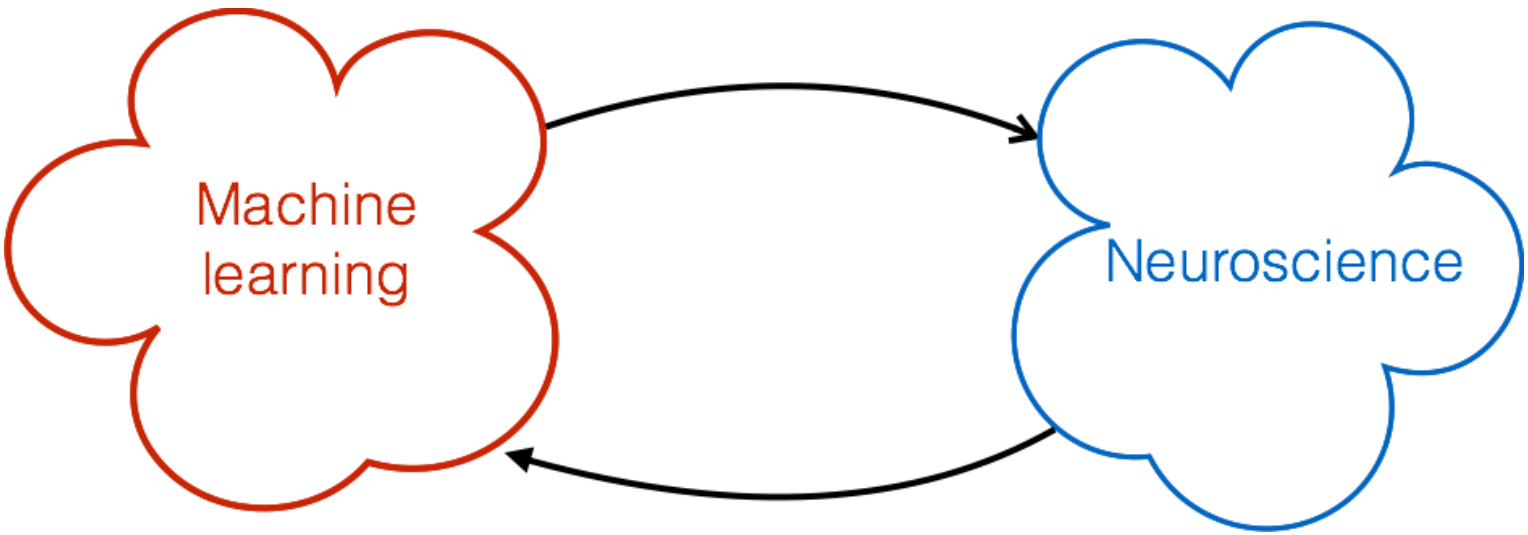
Goal: to develop new experimental tools that will revolutionize our understanding of the brain.

Major bottleneck:

data analysis capabilities are much below high-throughput data collection rates (TB's/hour).

Cannot fully exploit the potential of these technologies.

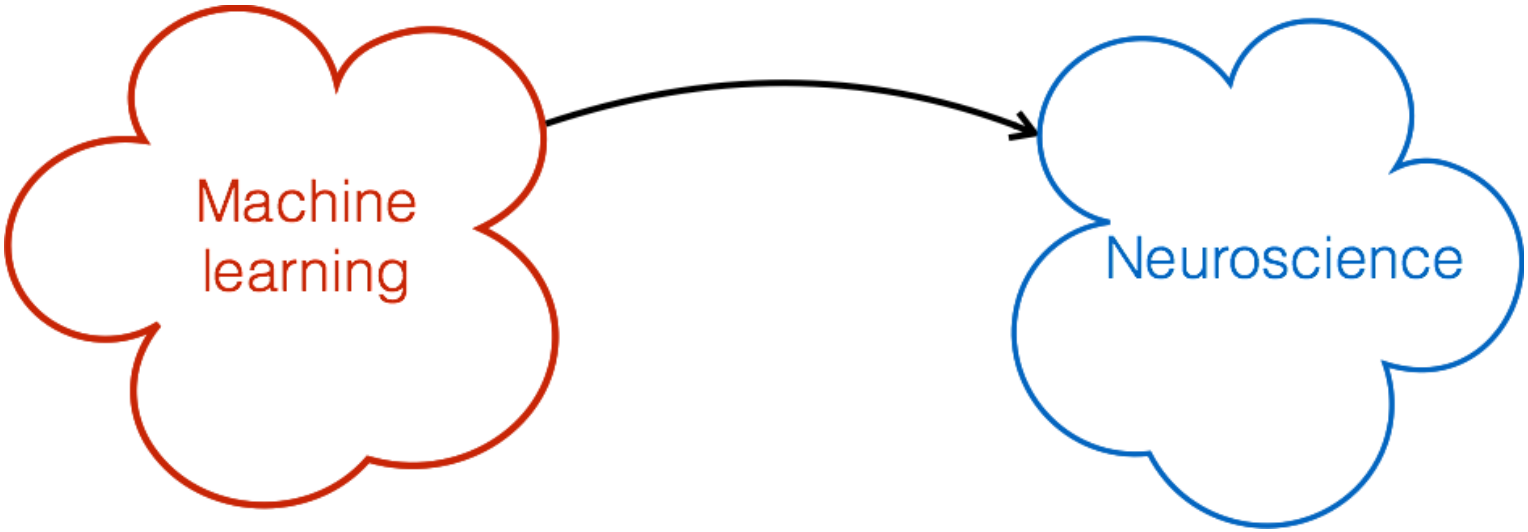
This talk



Claim

Neuroscience and machine learning nourish each other

This talk

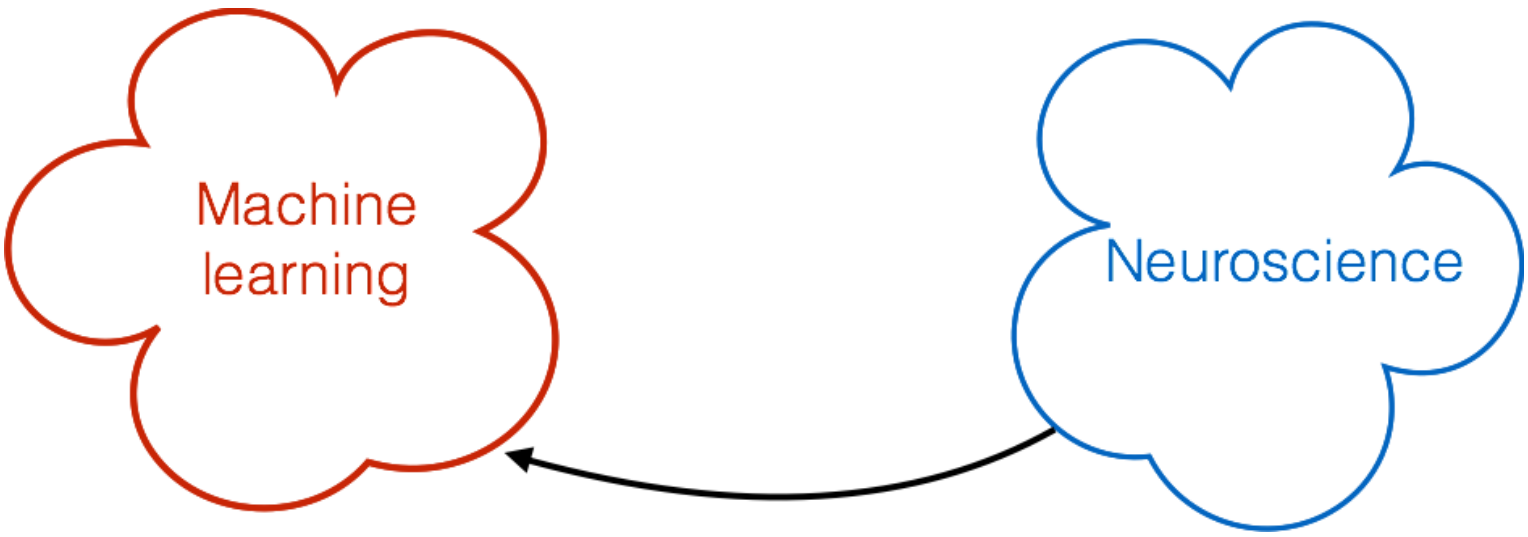


Section 1

Q: How can we use existing machine learning techniques a complex problem in neuroscience?

A: To enhance the capabilities of a neurotechnology for reading and writing of neural activity.

This talk



Section 2

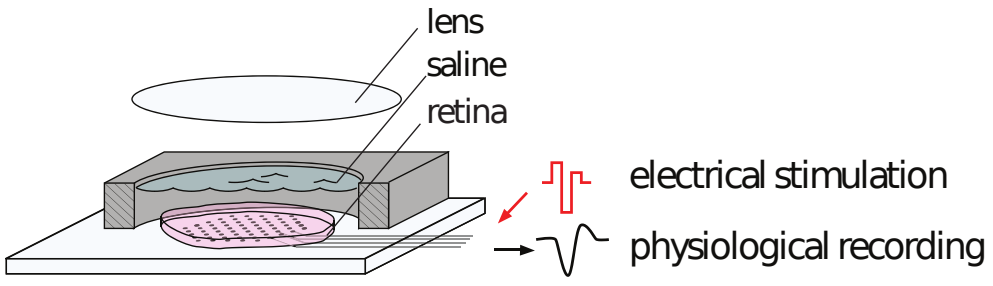
Q: How can complex problems in neuroscience spark original research in machine learning?

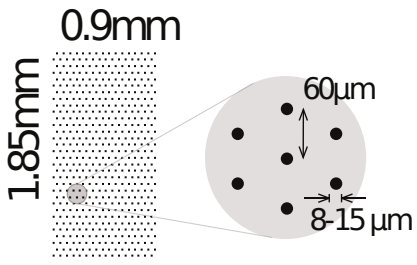
A: To understand how the brain works, learn latent permutations.

Large-scale Spike Sorting with Stimulation Artifacts

Overarching goal

Stimulation and recording in large multi-electrode arrays (MEA) to read and write neural activity in closed-loop.



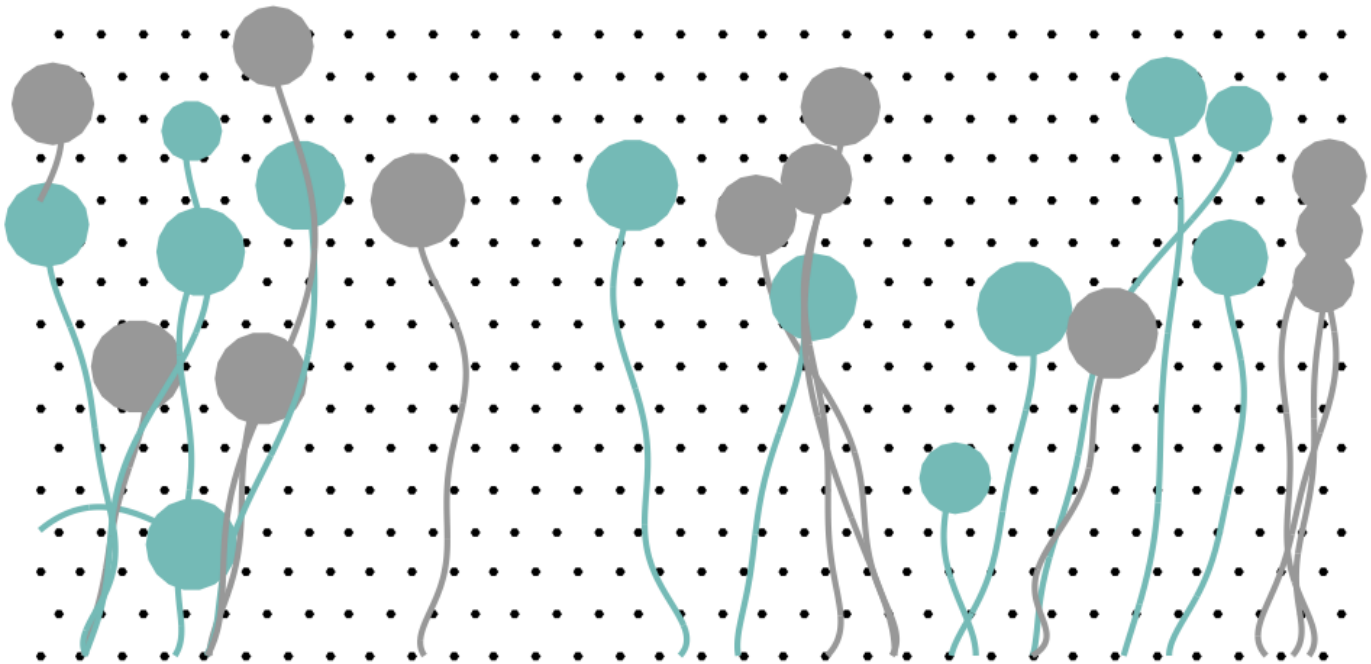


- Closed-loop **control**: stimulate based on neurons output. Need to know the stimulus→ response map.
- Large-scale, online data analysis. 512 electrodes, 20 Khz $\sim$ 50 GB/hour.
- Scientific and **Clinical** significance: development of high-resolution retinal prosthesis.

Tailored activation

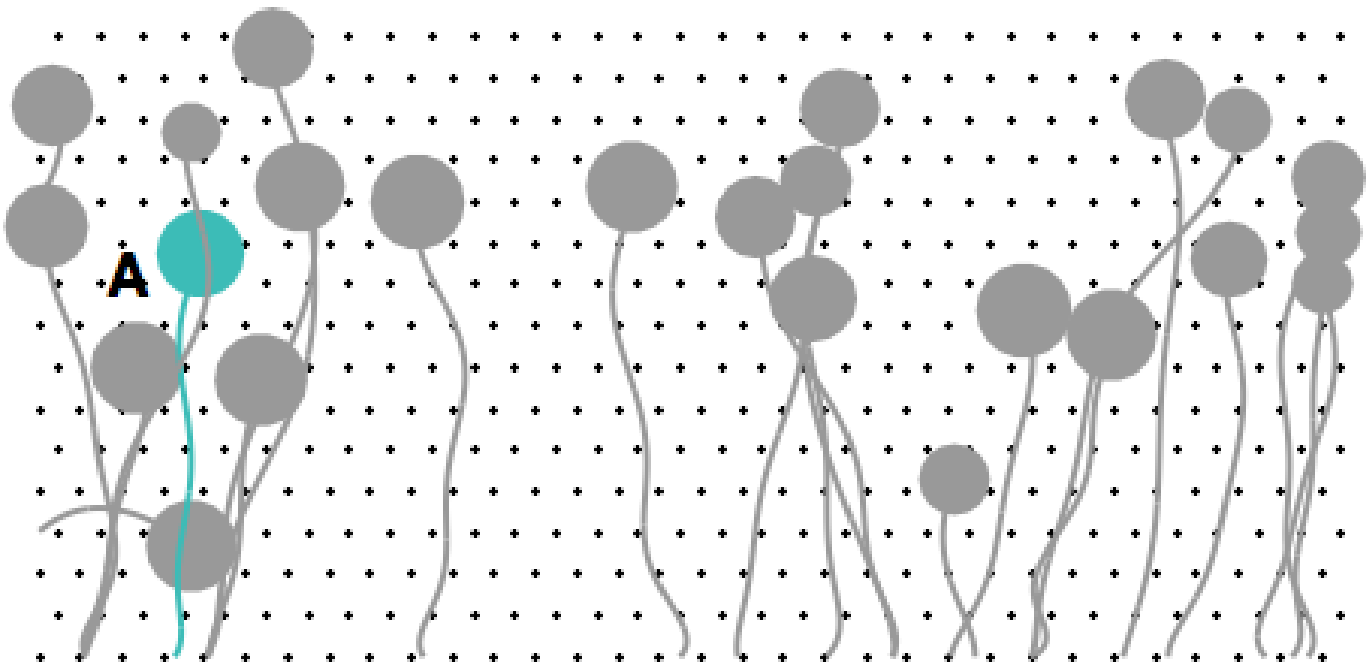
Goal: To generate artificial vision, elicit arbitrary patterns of neural activity with tailored stimuli.

Tailored activation



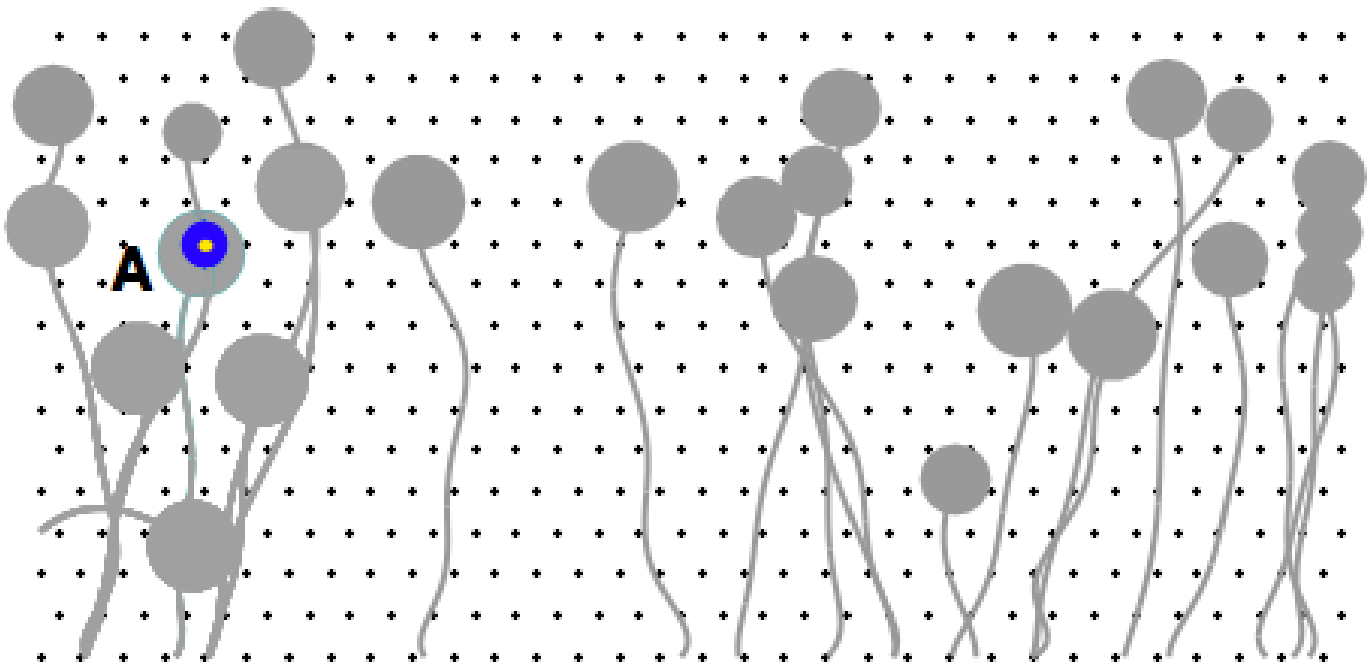
Question: Is it possible to activate *only* the colored neurons?

Tailored activation



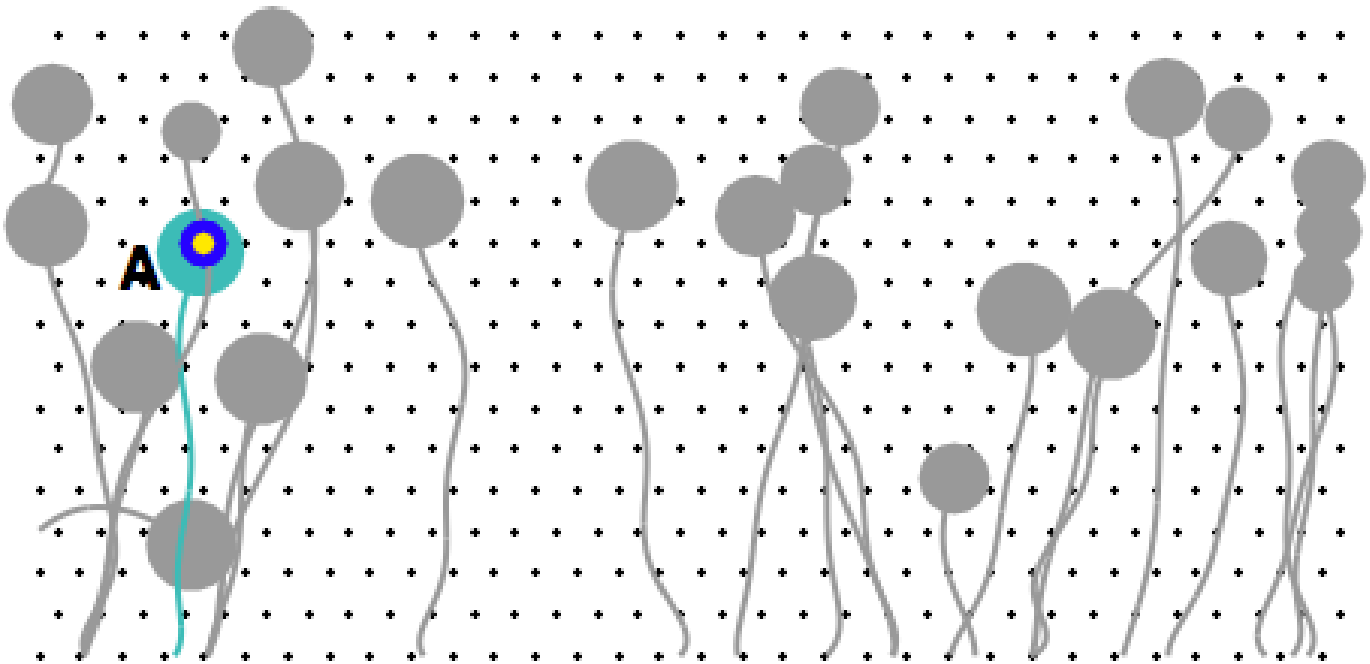
Easier question: is it possible to activate *only* neuron A?

Tailored activation



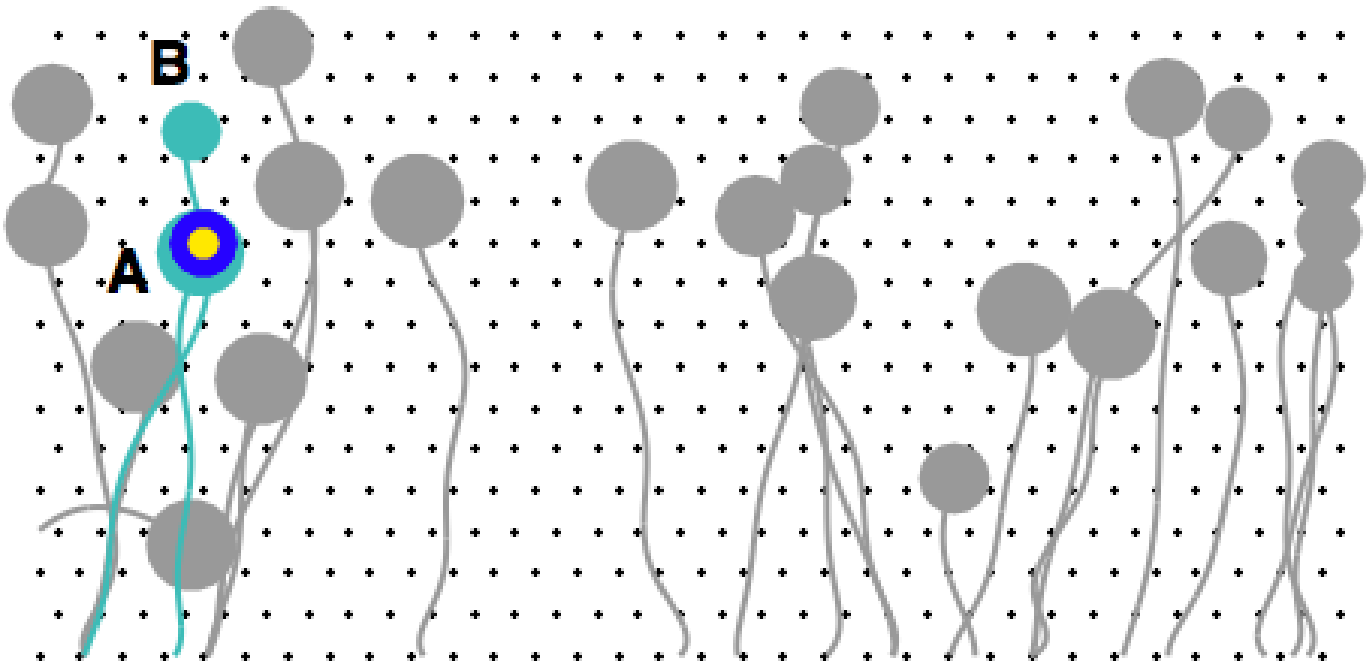
Stimulating with a pulse of $0.5\mu A$ on the electrode around the soma does not activate neuron A.

Tailored activation



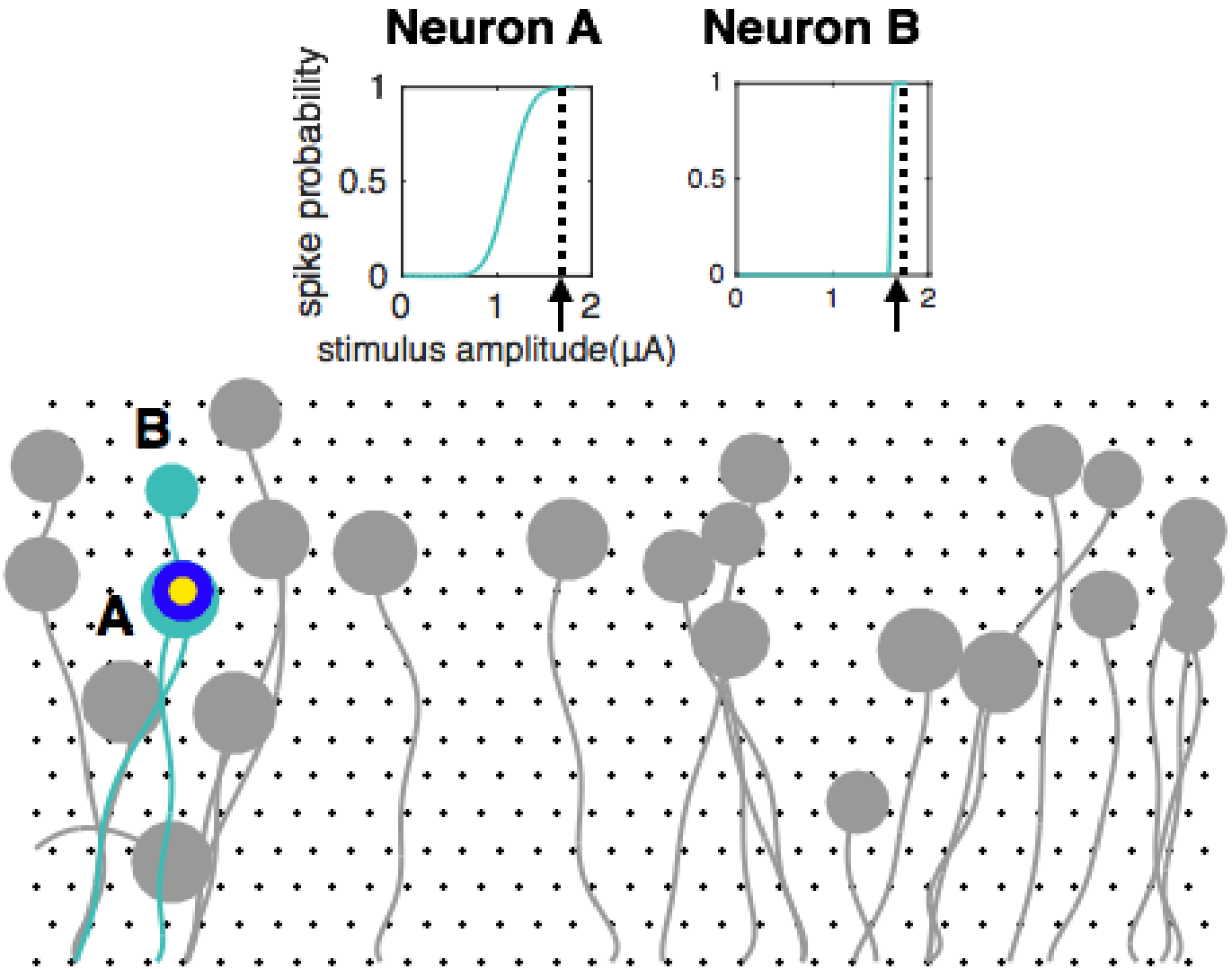
However, stimulating with $1.0\mu A$ does activate the neuron.

Tailored activation



Further, stimulating with $1.5\mu A$ also activates nearby neuron B, through its axon.

Tailored activation

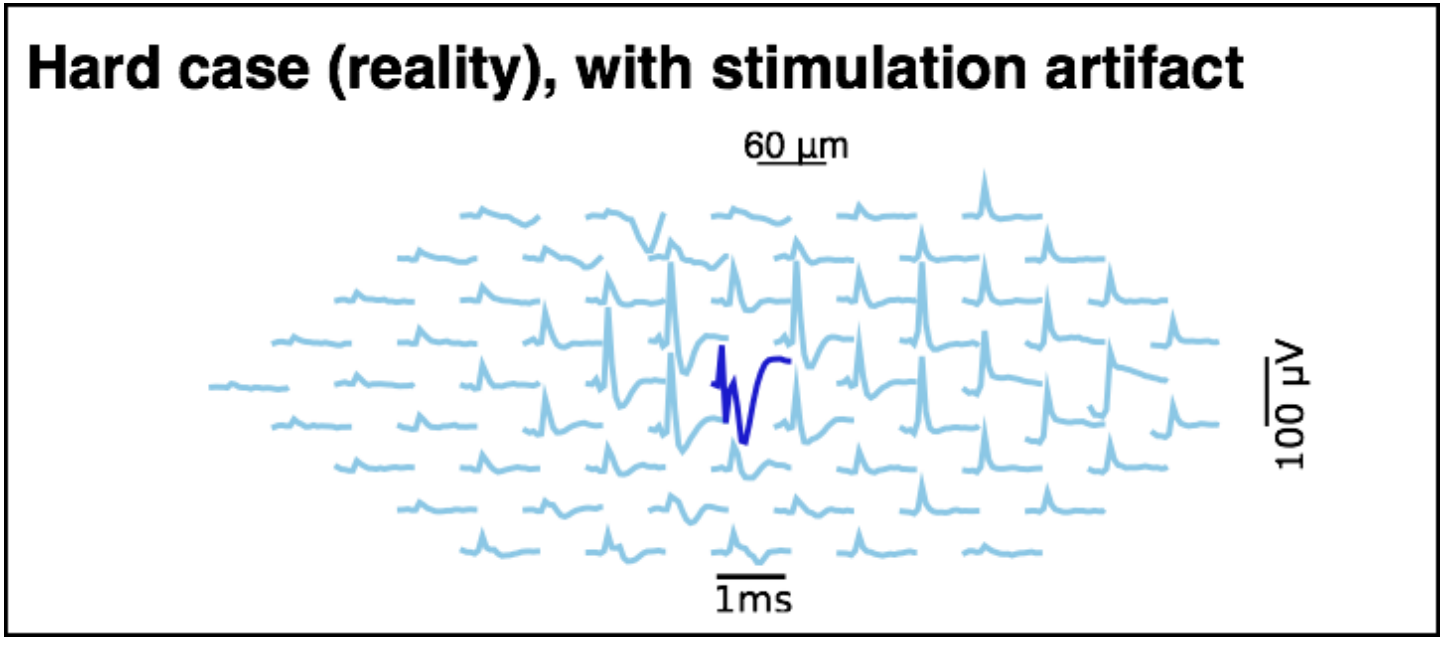


Activation curves summarize responsiveness of neurons. Inferred from many increasing stimuli.

Stimulation artifacts

Major hurdle: electrical stimuli are sensed in electrodes as **artifacts**, stymying identification of neural activity.

- Artifacts are much larger than spikes, overlap temporally with them.



Current solutions break down.

Can take weeks to a human. No closed-loop.

Stimulation Artifacts

Problem

Data contains a *nuisance* parameter A,

$$Y = A + s + \epsilon,$$

Recorded traces Y, artifact A, neural activity s and noise ϵ.
To infer s need to know A. But knowing A can take weeks.

Solution

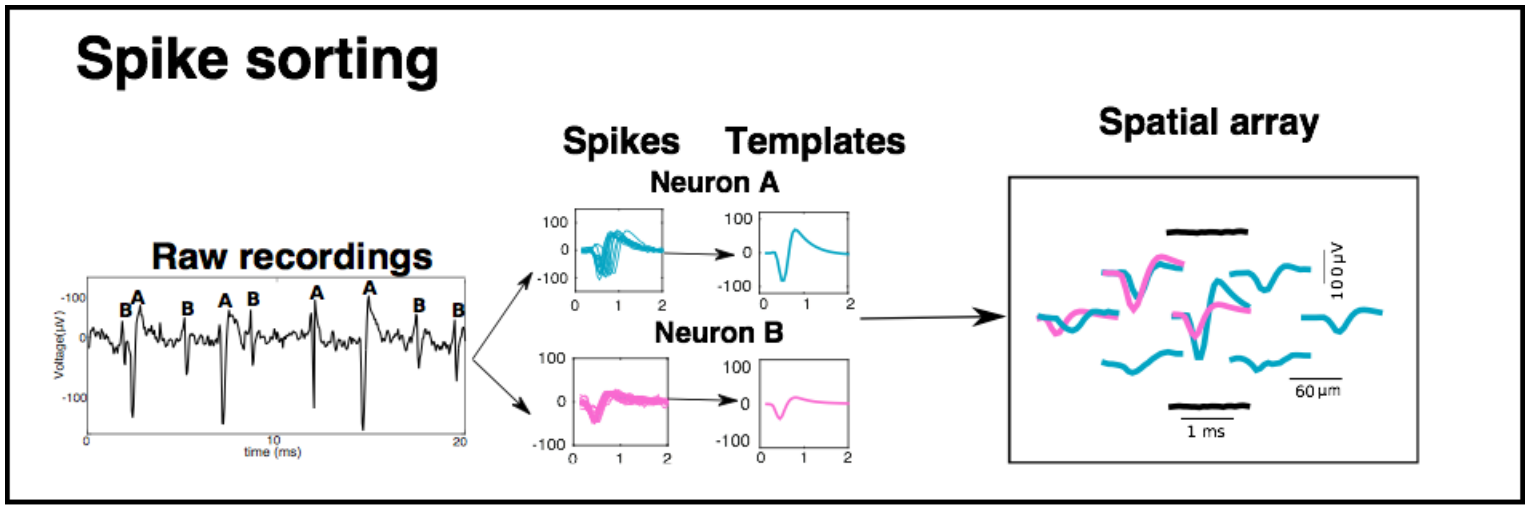
Impose structure and prior knowledge in $A, s,$ and ϵ so $\hat{A}, \hat{s}$ can be resolved.

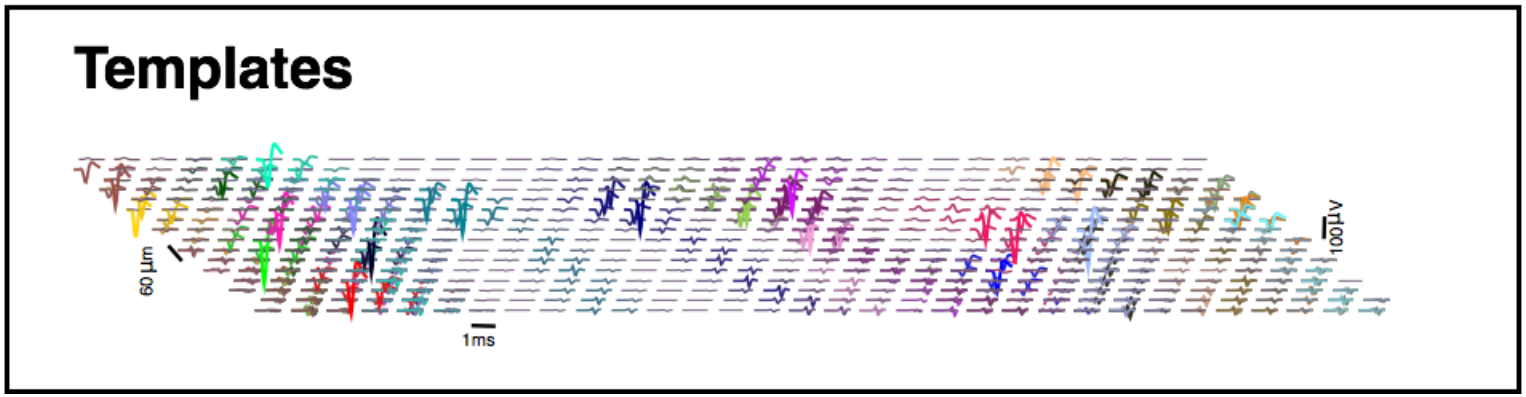
Result

We can find good $\hat{s}$ and $\hat{A}$ fast enough to do closed-loop experiments.

Neural activity structure

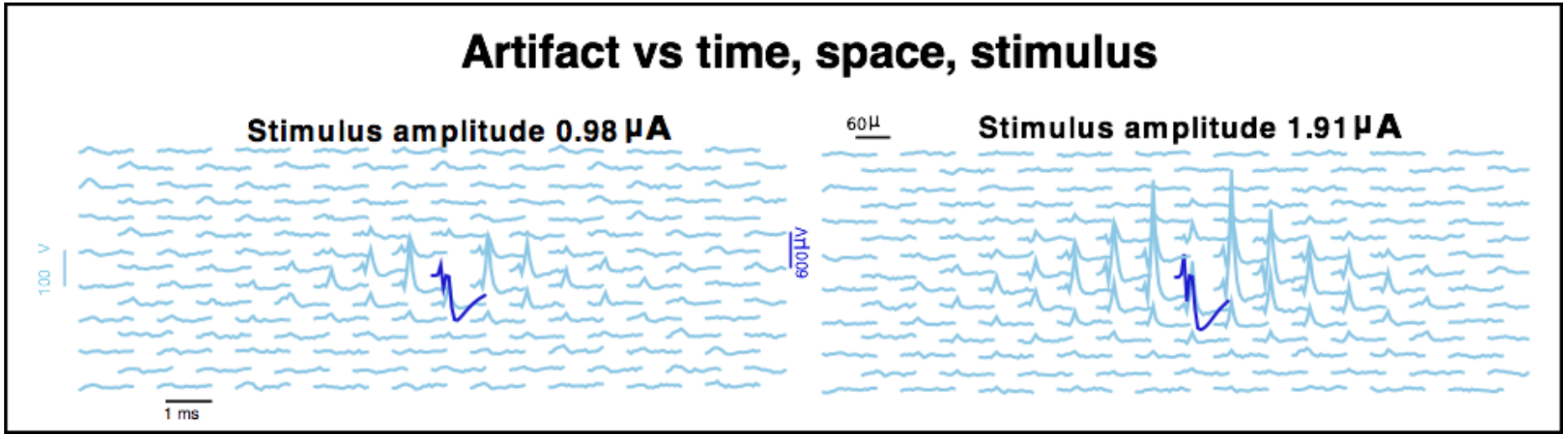
- Spike sorting of spontaneous activity to identify neurons.
- Provide us with templates





The structure of stimulation artifacts

- Properties are revealed by silencing neural activity.
- Decays smoothly with distance from stimulating electrode and has a peak in time. Increases with strength of stimulus. Doesn't change if stimulus is the same.



Non-linear and non-stationary, but structured.

Crafting a principled solution

Consider the model

$$Y = A + s + \epsilon,$$

- Given a single stimulating electrode, data (responses) are four dimensional tensors, $Y = Y_{t,e,j,i}$ over **time** ($t = 1, \ldots T$), **space** (electrode, $e = 1, \ldots E$), **strength** ($j = 1, \ldots J$) and **trial** ($i = 1, \ldots, I$) dimensions.

Imposing structure

- Spikes, if any, are translations of templates. Translations are represented with binary vectors indicating timing.
- Gaussianity provides a reasonable model for noise, $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$.
- Use a Gaussian process (GP) to encode prior knowledge of the artifact $A \sim GP(0, K^\theta)$.

GP artifact model

- Use GP to encode prior knowledge and *borrow strength* in the lack of certainty.
- Problem: $n \approx 10^6$ artifact variables, $O(n^3)$ does not scale.
- Solution: use a Kronecker decomposition (Gilboa et al., 2015) with separate kernels for time, space and strength dimensions.

$$K^{(\theta,\phi^2)} = \rho K_t \otimes K_e \otimes K_j + \phi^2 I.$$

- Each kernel must represent smoothness and non-stationarity.

Algorithm

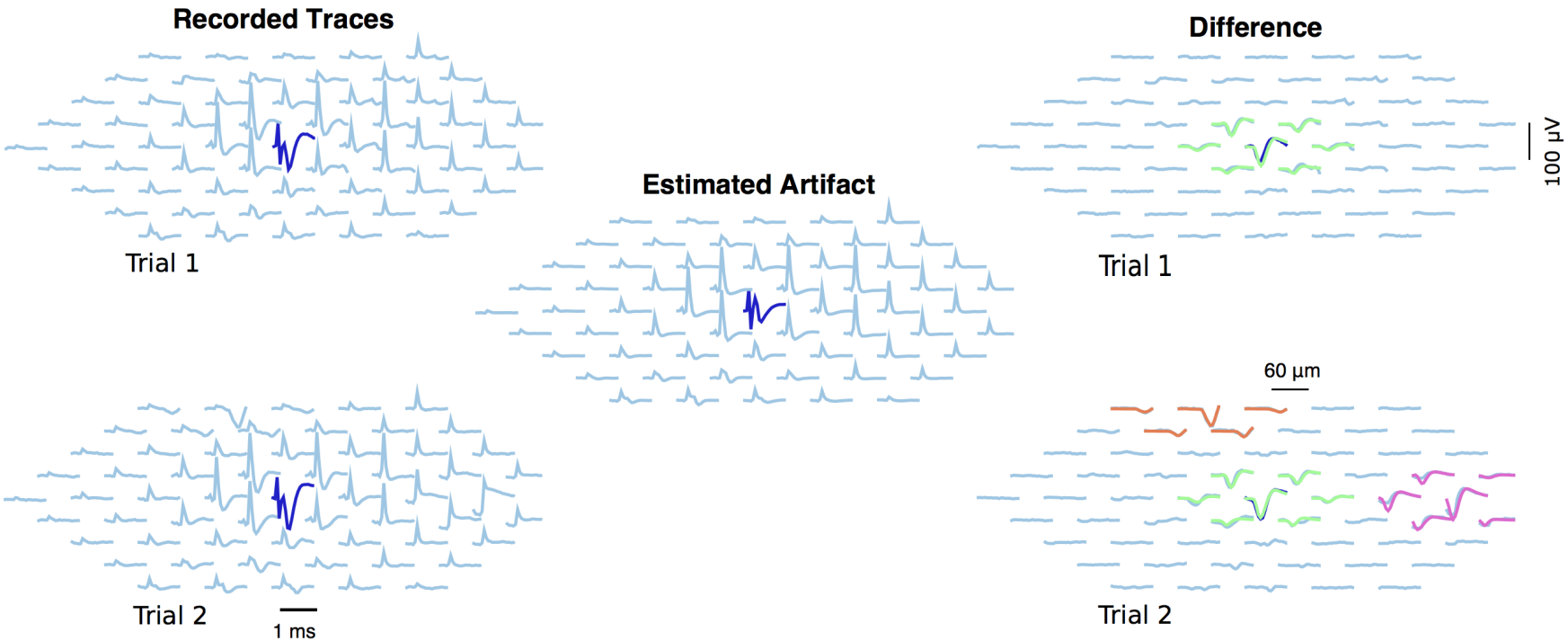
Goal: Obtain $\hat{A}, \hat{s}$ from the model $Y = A + s + \epsilon, A \sim GP(0, K^{\hat{\theta}})$

- Produce estimates increasingly in j (strength).
  - Rationale: at lowest strengths A is better behaved and easier to estimate.
  - Initial guess $\hat{A}^0_{j+1}$ is the extrapolation from $\hat{A}_{[1,j]}$.
- Given j, alternate between maximizing $p(s_j | Y_j, \hat{A}_j, \hat{\theta})$ for $\hat{s}_j$ and maximizing $p(A_j | Y_j, \hat{s}_j, \hat{\theta})$ for $\hat{A}_j$.
  - $\hat{s^n_{j,i}}$ given $\hat{A}_j$: $s^n_{j,i} = T^n b_{j,i}$ are binary vectors; do matching pursuit

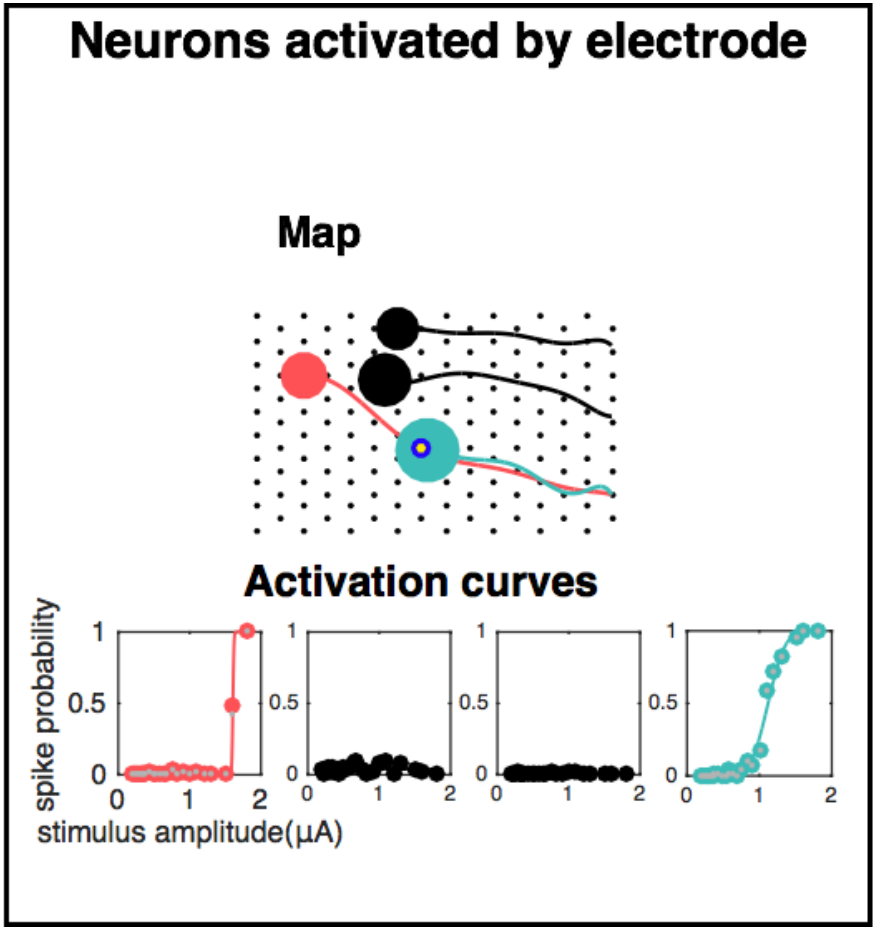
$$\min_{b^n_{j,i}} \left\| (Y_{j,i} - \hat{A}_j) - \sum_n T^n b_{j,i} \right\|^2 .$$

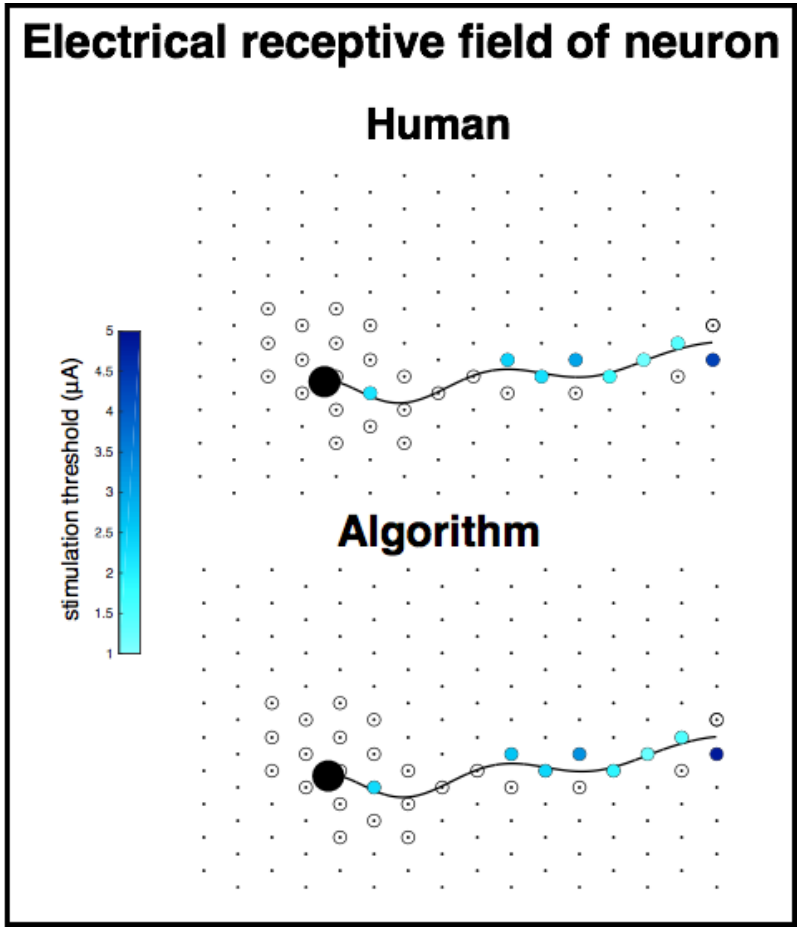
  - $\hat{A}_j$ given $\hat{s}_j$ via filtering (posterior mean) of spike-subtracted traces.

Example of sorting



Large-scale automatic analysis



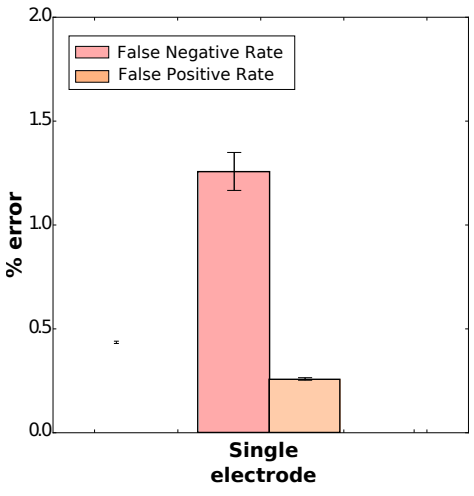


Colors indicate stimulation thresholds.

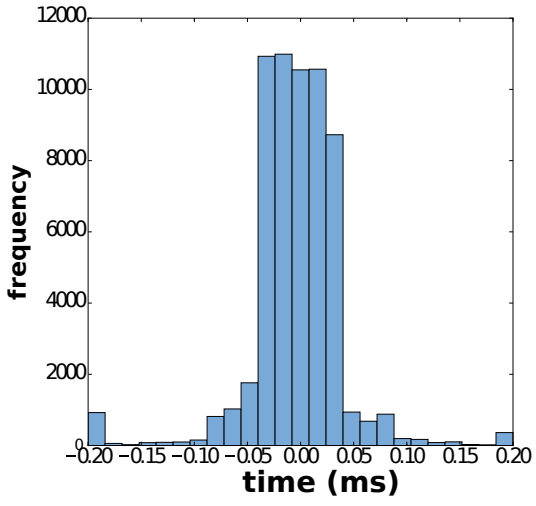
Population results

1,713,233 trials, 4,405 amplitude series.

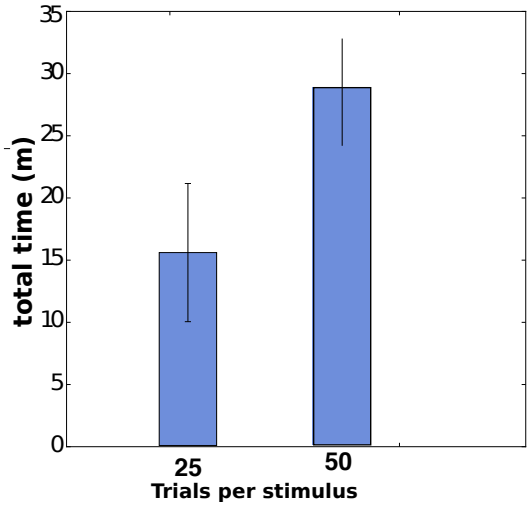
Trial-by-trial analysis



Latency difference histogram



Computational Cost

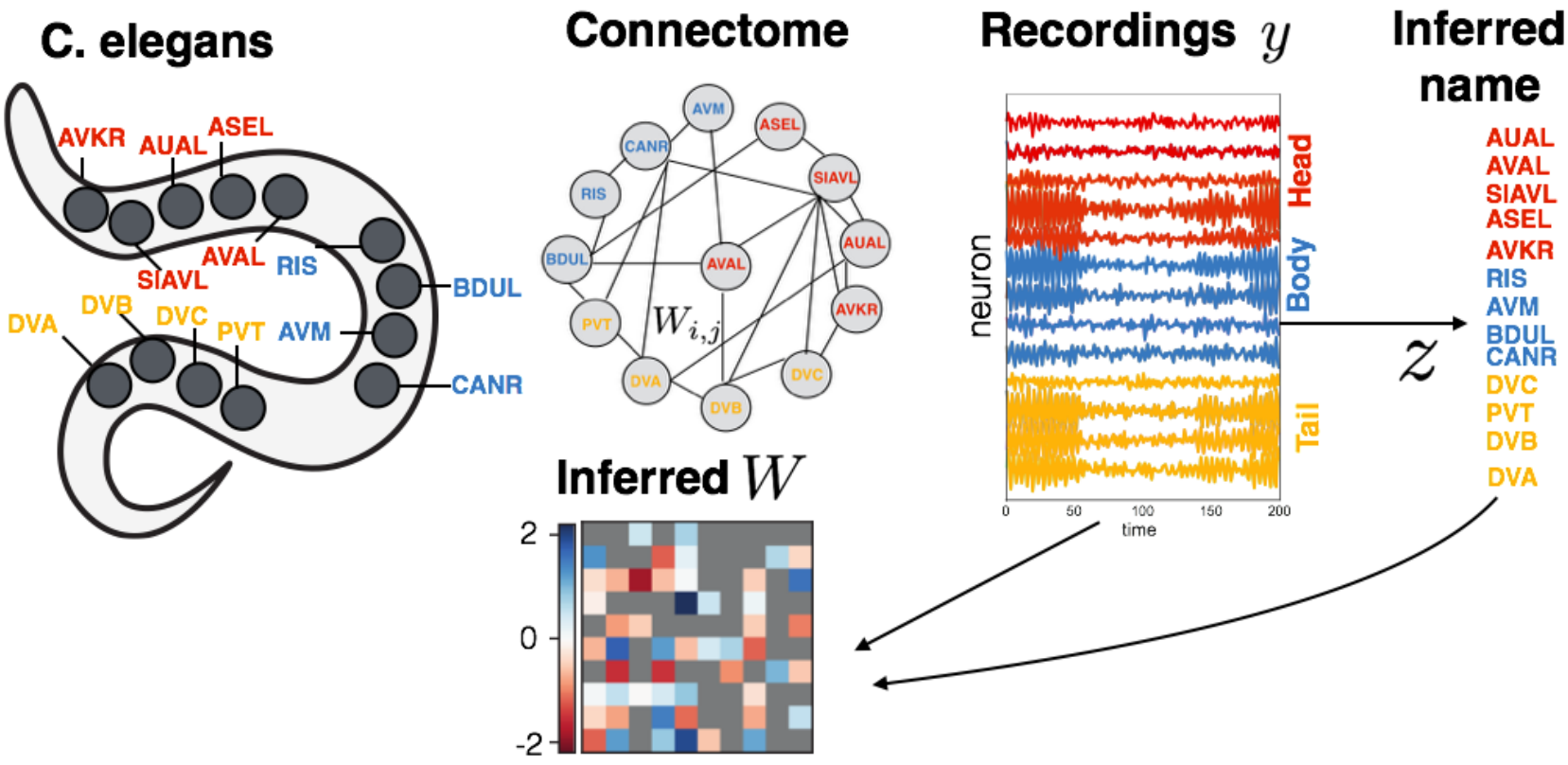


- Accuracy greater than 99.5%, also agreement in latencies.
- Past: weeks$\rightarrow$ Now: $\approx$15 minutes.
- Compatible with closed-loop experiments.
- Established a new technology: artifact-free stimulation and recording in MEAs

Gradient-based Learning of Latent Permutations

Motivation

- Inference of how the brain works in **C.elegans**.
- Always same neurons and connectome.
- Neural dynamics inference = what are the connection **weights** $w_{i,j}$.
- Recordings y should inform $p(W|y)$. But identities are not known.
- Inference of W requires inference of a permutation z by a human.



The challenge with permutations

- How to compute $p(z|y)$ when z is a permutation?
- MCMC is always an option (Diaconis, 2009).
- Alternative: variational inference (Blei et al., 2016).
  Cast the problem as optimization: propose a variational family $\mathcal{F} = \{q_\theta(\cdot), \theta \in \Theta\}$ and minimize

$$q_{\theta^*}(\cdot) = \underset{q_\theta(\cdot)\in\mathcal{F}}{\operatorname{argmin}} KL(q_\theta(z)\|p(z|y)).$$

  using gradient-based optimization.
- Since z is a permutation the problem is non-trivial: there are many ($n!$) and they are discrete objects.
- To deal with discrete data: Concrete or Gumbel-Softmax distribution (Maddison et al., 2017, Jang et al., 2017).

Our contribution

Extending the Gumbel-Softmax estimator permutations by analogy; **Gumbel-Sinkhorn**.

Stochastic optimization & reparameterization trick

- Want to maximize

$$\mathcal{L}(\theta) = E_{q_\theta(z)}\left(f(z,\theta)\right).$$

  For variational inference

$$f(z,\theta) = \log\left(\frac{p(y|z)p(z)}{q_\theta(z)}\right).$$

- If there are unbiased estimators for $\nabla_\theta \mathcal{L}(\theta)$ we can use stochastic optimization (Kushner & Yin, 1997).
- An unbiased estimator is given by **reparameterization trick**: if $z = g(\epsilon,\theta)$ with $\epsilon \sim p(\epsilon)$, then:

$$\begin{aligned}\nabla_\theta \mathcal{L}(\theta) &= E_\epsilon(\nabla_\theta f(g(\epsilon,\theta),\theta)) \\ &\approx \frac{1}{M}\sum_{i=1}^{M} \nabla_\theta f(g(\epsilon_i,\theta),\theta) \quad \epsilon_i \sim p(\epsilon_i).\end{aligned}$$

How do we reparameterize **discrete** variables (and permutations)?

Categorical case: reparameterization as optimization

Let s_i be a category or one-hot vector, $s_i \in ver(S)$, a vertex of the probability simplex $\mathcal{S}$.
Given parameter x (vector), sampling from

$$P(z = s_i) \propto \exp(x_i).$$

Is achieved by solving a Gumbel-perturbed linear program:

$$z = \underset{i}{\operatorname{argmax}}(x_i + \epsilon_i) = \underset{s \in \mathcal{S}}{\operatorname{argmax}} \langle x + \epsilon, s \rangle, \quad \epsilon_i \overset{i.i.d}{\sim} Gumbel(0, 1).$$

Problem: $\theta \to z = \operatorname{argmax}_{s \in \mathcal{S}} \langle x + \epsilon, s \rangle$ is not differentiable.

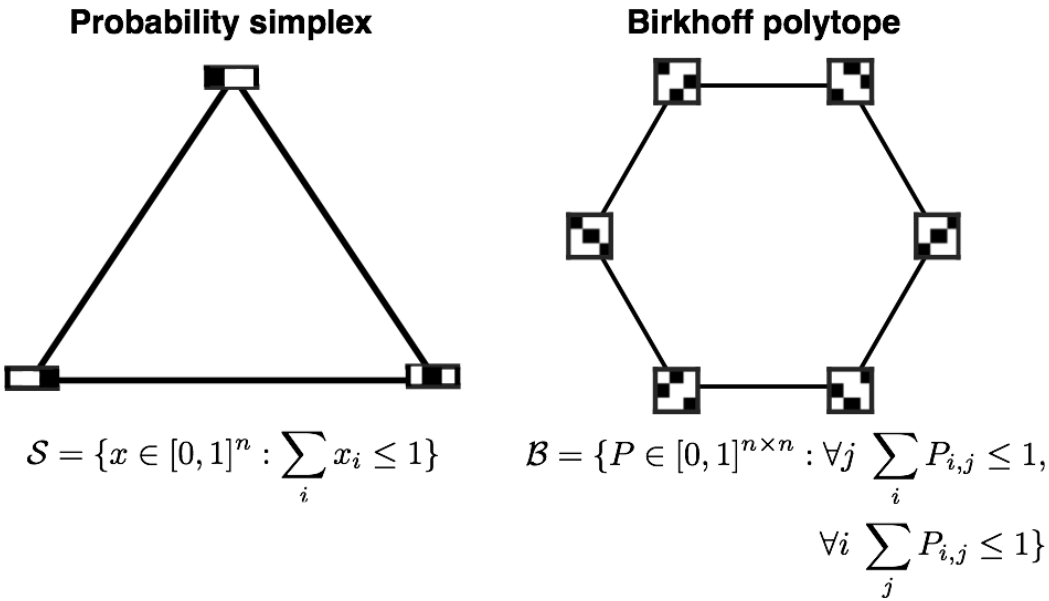
The Concrete distribution gives a differentiable approximate sample.

$$\underbrace{\text{softmax}((x + \epsilon)/\tau)}_{\text{Concrete} \in \mathcal{S}} \underset{\tau \to 0^+}{\Longrightarrow} \underbrace{\underset{s \in \mathcal{S}}{\operatorname{argmax}} \langle x + \epsilon, s \rangle}_{\text{Categorical} \in ver(S)},$$

i.e. $\operatorname{argmax}_i x_i = \lim_{\tau \to 0^+} \text{softmax}(x/\tau)$, where

$$\text{softmax}(x) = \frac{\exp(x)}{\sum_i \exp(x_i)}.$$

Permutation analogs of argmax and softmax



Permutation matrices are vertices of Birkhoff Polytope $\mathcal{B}$ (Birkhoff, 1946).

For categories, $\operatorname{argmax}_i x_i = \operatorname{argmax}_{s \in S} \langle x, s \rangle = \lim_{\tau \to 0^+} \operatorname{softmax}(x/\tau)$

Likewise, in permutations define the *matching operator*

$$M(X) \equiv \operatorname*{argmax}_{P \in \mathcal{B}} \langle P, X \rangle_F .$$

What is a differentiable approximation for $M(X)$?, $M(X) = \lim_{\tau \to 0^+}$?

Sinkhorn operator

Define the Sinkhorn operator $S(\cdot)$ for squared matrices X as

$$S^0(X) = \exp(X),$$

$$S^l(X) = \mathcal{T}_c\left(\mathcal{T}_r(S^{l-1}(X))\right),$$

$$S(X) = \lim_{l\to\infty} S^l(X).$$

With $\mathcal{T}_r, \mathcal{T}_c$ the row and column normalization operators.
Remark: $S(X) \in \mathcal{B}$ (Sinkhorn, 1964).

Theorem (Mena et al., 2018)

$$M(X) = \lim_{\tau\to 0^+} S(X/\tau).$$

Remark: related to entropy-regularized optimal transport (Cuturi 2012).

Completing the analogy

	Categories	Permutations
Polytope	Probability simplex $\mathcal{S}$	Birkhoff polytope $\mathcal{B}$
Linear program	$\operatorname{argmax} x_i = \operatorname{argmax}_{s \in \mathcal{S}} \langle x, s \rangle$	$M(X) = \operatorname{argmax}_{P \in \mathcal{B}} \langle P, X \rangle_F$
Approximation	$\operatorname{argmax}_i x_i = \lim_{\tau \to 0^+} \operatorname{softmax}(x/\tau)$	$M(X) = \lim_{\tau \to 0^+} S(X/\tau)$
Entropy	$h(s) = \sum_i -s_i \log s_i$	$h(P) = \sum_{i,j} -P_{i,j} \log (P_{i,j})$
Entropy regularized linear program	$\operatorname{softmax}(x/\tau) = \operatorname{argmax}_{s \in \mathcal{S}} \langle x, s \rangle + \tau h(s)$	$S(X/\tau) = \operatorname{argmax}_{P \in \mathcal{B}} \langle P, X \rangle_F + \tau h(P)$
Reparameterizable distribution	**Gumbel-max trick** $\operatorname{argmax}_i(x_i + \epsilon_i)$	**Gumbel-Matching** $\mathcal{G}M(X)$ $M(X + \epsilon)$
Continuous reparameterization	**Concrete** $\operatorname{softmax}((x + \epsilon)/\tau)$	**Gumbel-Sinkhorn** $\mathcal{G}S(X, \tau)$ $S((X + \epsilon)/\tau)$

Results

Sensible alternative to (naive) MCMC sampler for identification of neurons in C.elegans

Mean number of candidates	10		45		60	
Difficulty	1 worm	4 worms	1 worm	4 worms	1 worms	4 worms
MCMC	.34	.65	.14	.17	.13	.16
Linderman, Mena et al. (2018)	.77	.93	.18	.48	.17	.37
Gumbel-Sinkhorn	**.79**	**.94**	**.25**	**.51**	**.21**	**.44**
Gumbel-Sinkhorn (no regularization)	.77	.92	.25	.44	.21	.39

Sinkhorn Networks

Problem Statement: learning to solve jigsaw puzzles; i.e. decode scrambled objects $\tilde{X}$ into non-scrambled X.

- Data are pairs $(X_i, \tilde{X}_i)$ where $\tilde{X}_i$ are constructed by permuting pieces of X_i.
- Formally, a permutation valued regression,

$$X_i = P^{-1}_{\theta,\tilde{X}_i} \tilde{X}_i + \varepsilon_i,$$

with

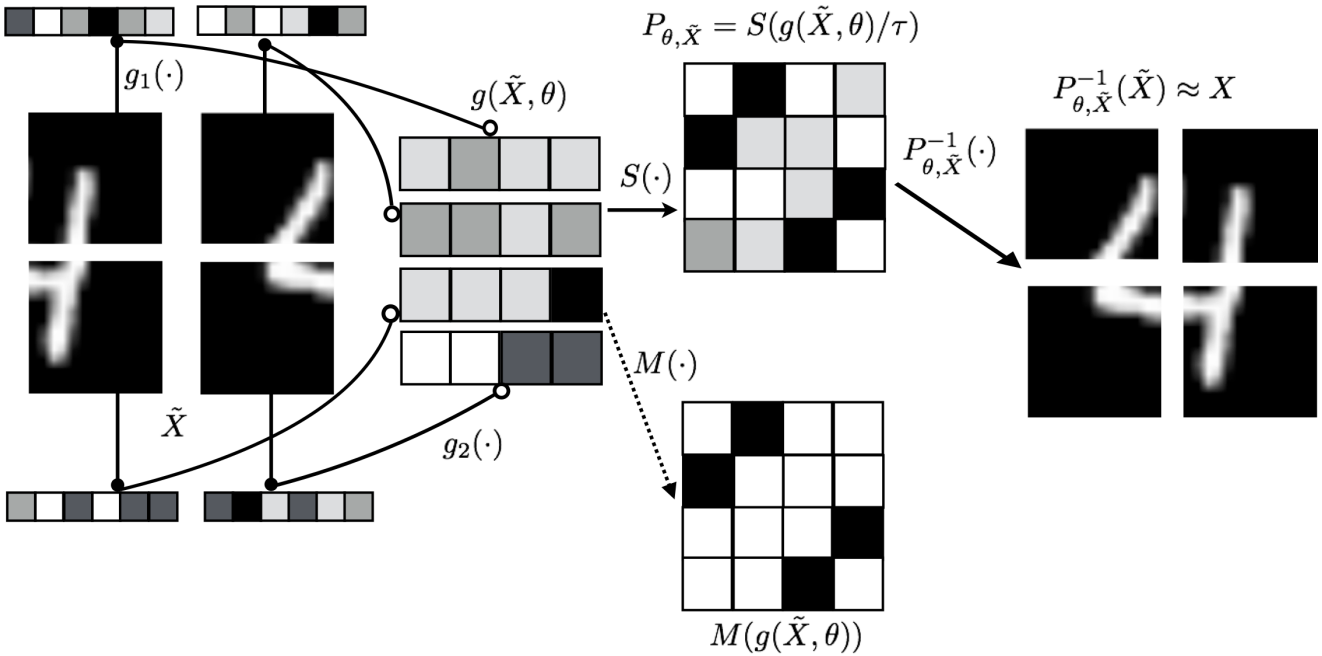
$$P_{\theta,\tilde{X}} = M(g(\tilde{X}, \theta)),$$

and g an artificial neural network.

- To train, replace $S(g(\tilde{X}, \theta)/\tau) \approx M(g(\tilde{X}, \theta))$.

Sinkhorn Networks

Sinkhorn Network for solving jigsaw puzzles.



- We impose permutation equivariance.
- Each piece is processed by the same network, saving parameters. Finals outputs of each pieces are concatenated and the Sinkhorn operator is applied.

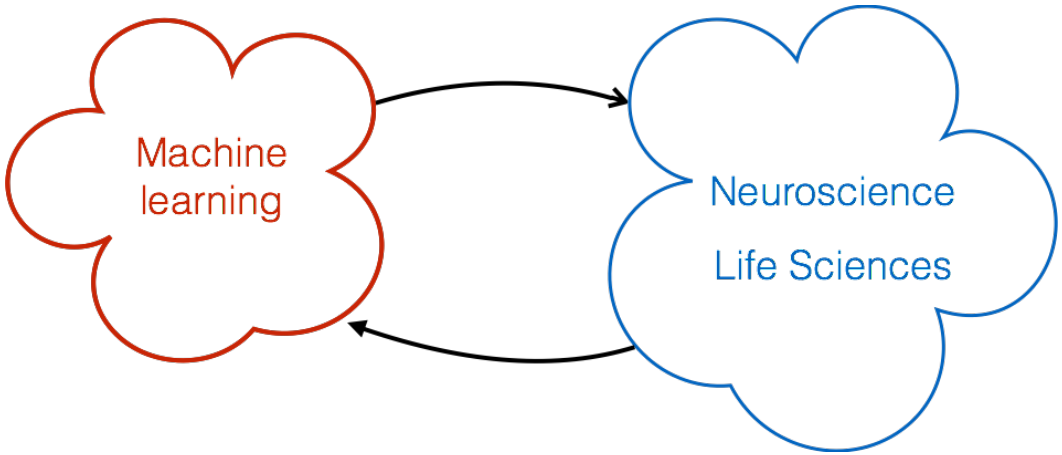
Sinkhorn Networks

- Tie with state-of-the-art on jigsaw puzzles (Cruz et al., 2017), but on a much simpler architecture.
- Achieve state-of-the-art on sorting numbers using neural networks. (up to 120 numbers). Previous state-of-the-art, 15 numbers (Vinyals et al., 2015.)
- Original application: generate digits from atomic pieces.

Scrambled Hard reconstructions Soft Reconstructions

Future Work

Future Work



- Section 1: closed-loop: turn images into stimulation patterns. Artifacts in optogenetics.
- Section 2: a new technology for identification of neurons.
- Discrete structure:
  - Theory: how to make AD work? why the Concrete distribution works?
  - In permutations: learning to generate objects from pieces.
  - Other combinatorial objects (graphs, trees) via entropy regularization in polytopes.
  - Applications in the sciences: generating DNA (Killoran et al., 2017), Tumor phylogeny inference (Deshwar et al., 2015), molecular design (Duvenaud et al., 2015).

Thanks!

Main references

Mena, G.E., Grosberg, L.E., ..., J., Chichilnisky, E.J. and Paninski, L., 2017. Electrical Stimulus Artifact Cancellation and Neural Spike Detection on Large Multi-electrode Arrays. PLoS computational biology.

Linderman, S.W.,* **Mena, G.E.***, ..., Paninski, L. and Cunningham, J.P., 2018. Reparameterizing the Birkhoff Polytope for Variational Permutation Inference. AISTATS.

Mena, G.E., Belanger, D., Linderman, S, Snoek, J., 2018. Learning Latent Permutations with Gumbel-Sinkhorn Networks. ICLR.

Collaborators