How are distributions represented at the level of spikes?

**Spatial:** PPCs

**Temporal:** sampling

HERE: spatio-temporal code

distributed sampling

representation

population code

**Example computation**

\[
\mathbf{s} = \mathbf{A} \cdot \mathbf{x} + \mathbf{\eta}
\]

Inference:

\[
P(x|s) = \mathcal{N}(x; \mu(s), \Sigma)
\]

Langevin sampling

\[
H_s(x) = -\log P(x|s)
\]

\[
\dot{x} = \nabla H_s(x) + \mathbf{d}
\]

**Population code**

\[
\mathbf{y} = \mathbf{\Gamma} \cdot \mathbf{r}
\]

1. fix linear decoder: \( \hat{\mathbf{y}} = \mathbf{\Gamma} \cdot \mathbf{r} \)

2. derive optimal dynamics:

\[
\frac{\partial \mathbf{r}}{\partial t} = -\mathbf{V} - W^{in} \cdot \mathbf{r} + W^{c} \cdot \mathbf{s} + \mathbf{\Gamma} \cdot \mathbf{\epsilon}
\]

Computational advantages of distributed sampling

**Robustness:** redundant representation, the code is tolerant to cell death

**Speed:** \( K \) independent chains running in parallel!

linear tradeoff between time and neural resources

How is distributed sampling reflected in neural responses?

1. **Stimulus tuning**

2. **Single neuron variability**

3. **Voltage correlations**

4. **Synchrony**

5. **Spike-count correlations**

Key idea: distributed sampling by decoupling neural responses from features.

Computational benefits: robustness and increased sampling speed (for a linear increases in no. neurons).

Consistent with a wide range of experimental observations at the level of single neuron and neuron-pairs measures.

Argues for using population level analyses for probing probabilistic representations in experimental data.