both consider adding new behaviors to the model, and we find that proposals for fixed-dimension discrete variables are much more likely to be accepted than proposals for high-dimensional continuous parameters. Split-merge proposals with high acceptance rates are essential to the experimental successes of our method, since they allow potentially large changes at each iteration.

At each iteration, we cycle among seven distinct sampler moves:

(1) (Section 6.4) Sample behavior-specific auxiliary variables: $\boldsymbol{\theta}, \boldsymbol{\eta} | \mathbf{F}, \mathbf{z}$.
(2) (Section 6.2) Sample *shared* features, collapsing state sequences: $\mathbf{F} | \boldsymbol{\theta}, \boldsymbol{\eta}$.
(3) (Section 6.3) Sample each state sequence: $\mathbf{z} | \mathbf{F}, \boldsymbol{\theta}, \boldsymbol{\eta}$.
(4) (Section 6.5) Sample BP hyperparameters: $\alpha, c | \mathbf{F}$.
(5) (Section 6.5) Sample HMM transition hyperparameters: $\gamma, \kappa | \mathbf{F}, \boldsymbol{\eta}$.
(6) (Section 6.6) Propose birth/death moves on joint configuration: $\mathbf{F}, \mathbf{z}$.
(7) (Section 7) Propose split/merge move on joint configuration: $\mathbf{F}, \mathbf{z}$.

Note that some moves instantiate $\boldsymbol{\theta}, \boldsymbol{\eta}$ as *auxiliary variables* to make computations tractable and block sampling possible. However, we discard these variables after step 5 and only propagate the core state space $(\mathbf{F}, \mathbf{z}, \alpha, c, \gamma, \kappa)$ across iterations. Note also that steps 2–3 comprise a block sampling of $\mathbf{F}, \mathbf{z}$. Our MCMC steps are detailed in the remainder of this section, except for split-merge moves which are discussed in Section 7. Further information for all moves is also available in the supplemental article [Fox et al. (2014)], including a summary of the overall MCMC procedure in Algorithm D.1.

Computational complexity. The most expensive step of our sampler occurs when sampling the entries of $\mathbf{F}$ (step 2). Sampling each binary entry requires one run of the forward–backward algorithm to compute the likelihood $p(\mathbf{y}_{1:T_i}^{(i)} | \mathbf{f}_i, \boldsymbol{\eta}^{(i)}, \boldsymbol{\theta})$; this dynamic programming routine has complexity $\mathcal{O}(T_i K_i^2)$, where K_i is the number of active behavior states in sequence i and T_i is the number of time steps. Computation may be significantly reduced by caching the results of some previous sampling steps, but this remains the most costly step. Resampling the N state sequences $\mathbf{z}$ (step 3) also requires an $\mathcal{O}(T_i K_i^2)$ forward–backward routine, but harnesses computations made in sampling $\mathbf{F}$ and is only performed N times rather than NK, where K is the total number of instantiated features. The birth/death moves (step 6) basically only involve the computational cost of sampling the state sequences. Split-merge moves (step 7) are slightly more complex, but again primarily result in repeated resampling of state sequences. Note that although each iteration is fairly costly, the sophisticated sampling updates developed in the following sections mean that fewer iterations are needed to achieve reasonable posterior estimates.

Conditioned on the set of instantiated features $\mathbf{F}$ and behaviors $\boldsymbol{\theta}$, the model reduces to a collection of independent, finite AR-HMMs. This structure could be harnessed to distribute computation, and parallelization of our sampling scheme is a promising area for future research.