update exists due to non-conjugacy. Instead, we numerically optimize our surrogate objective, finding the best vectors $\hat{\rho}, \hat{\omega}$ simultaneously. The constrained optimization problem is:

$$\hat{\rho}, \hat{\omega} = \text{argmax}_{\rho,\omega} \mathcal{L}_{HDP}(\rho, \omega, T, \alpha) + \mathcal{L}_u(\rho, \omega, \gamma) \quad (16)$$
$$\text{s.t.} \quad 0 < \rho_k < 1, \omega_k > 0 \text{ for } k \in \{1, 2, \dots K\}$$

where sufficient statistic $T = [T_1 \dots T_K \; T_{K+1}]$ sums the expectation of Eq. (10) across documents:

$$T_k(\hat{\theta}) \triangleq \textstyle\sum_{d=1}^{D} \mathbb{E}[\log \pi_{dk}]. \quad (17)$$

The Supplement provides implementation details, including the exact function and gradients we provide to a modern L-BFGS optimization algorithm.

4.3 Memoized algorithm.

We now provide a memoized coordinate ascent update algorithm. The update cycle comes from Hughes and Sudderth (2013), which was inspired by the incremental EM approach of Neal and Hinton (1998). Data is visited one batch at a time, where the batches are predefined. We call each complete pass through all batches a *lap*. At each batch, we perform a local step update to $q(z_d), q(\pi_d)$ for each document d in the batch, and then a global-step update to $q(u), q(\phi)$.

Affordable batch-by-batch processing is possible by tracking sufficient statistics and exploiting their additivity. For each statistic, we track a batch-specific quantity (denoted N^b) for each batch and an aggregated whole-dataset quantity (N). By definition, $N_k = \sum_{b=1}^{B} N_k^b$. After visiting each batch b, we perform an incremental update to make the aggregate summaries reflect the new batch summaries and remove any previous contribution from batch b.

This algorithm requires storing per-batch summaries N^b, S^b, T^b for every batch during inference. This requirement is modest, remaining size $\mathcal{O}(BK)$ no matter how many tokens or documents occur in each batch.

ELBO computation. Computing the objective $\mathcal{L}$ is possible after each batch visit, so long as we track sufficient statistics as well as a few ELBO-specific quantities. First, we store the entropy H_z from Eq. (5) at each batch, as in Hughes and Sudderth (2013).

Second, consider the computation of $\mathcal{L}_{HDP}$ in Eq. (9). Naively, this computation requires sums over all documents. However, by tracking the following terms we can perform rapid evaluation:

$$G_k^b \triangleq \textstyle\sum_{d \in \mathcal{D}_b} (N_{dk} - \hat{\theta}_{dk}) \mathbb{E}[\log \pi_{dk}], \quad (18)$$
$$Q_0^b = \textstyle\sum_{d \in \mathcal{D}_b} \log \Gamma(\sum_{k=1}^{K+1} \hat{\theta}_{dk}), Q_k^b = \sum_{d \in \mathcal{D}_b} \log \Gamma(\hat{\theta}_{dk}).$$

After aggregating these tracked statistics across all batches, such as $Q_k = \sum_{b=1}^{B} Q_k^b$, Eq. (9) becomes

$$\mathcal{L}_{HDP}(\cdot) = D\mathbb{E}_q[c_D(\alpha\beta)] - Q_0 \quad (19)$$
$$+ \textstyle\sum_{k=1}^{K+1} Q_k + G_k + \alpha \mathbb{E}_q[\beta_k] T_k$$

which given tracked statistics can be evaluated with cost independent of the number of documents D.

4.4 Stochastic algorithm.

Our objective $\mathcal{L}$ can also be optimized with stochastic variational inference (Hoffman et al., 2013). The stochastic global step at iteration t updates the natural parameters of $q(u)$ and $q(\phi)$ with learning rate ξ_t. For example, the new $\hat{\tau}_t$ interpolates between the previous value $\hat{\tau}_{t-1}$ and an amplified estimate from the current batch $\hat{\tau}^b$. When ξ_t decays appropriately, this method guarantees convergence to a local optimum.

4.5 Computational complexity

Our direct assignment representation is more efficient than the CRF approach of Wang et al. (2011). The dominant cost of both algorithms is the local step for each token. We require $\mathcal{O}(N_d K)$ computations to update the free parameters $\hat{r}$ for a single document via Eq. (13). The CRF method requires $\mathcal{O}(N_d K J)$ operations, where $J < K$ is the number of global topics allowed in each document (for more details, see Eq. 18 of Wang et al. (2011)). For any reasonable value of $J > 1$, the CRF approach is more expensive. When $J = \mathcal{O}(K)$, the CRF local step is quadratic in the number of topics, while our approach is always linear.

5 MERGE AND DELETE MOVES.

Here, we develop two moves, merge and delete, which help discover a compact set of interpretable topics. As illustrated in Fig. 4, merges combine redundant topics, while deletes remove unnecessary "junk" topics or empty topics. Both moves enable faster subsequent iterations by making the active set of topics smaller.

5.1 Merge moves.

Each merge move transforms a current variational posterior q of size K into a candidate q' of size $K - 1$ by combining two topics in a single *merged* topic. During each pass we consider several candidate pairs. For each pair $\ell < m$, we imagine simply pooling together all tokens assigned to either topic ℓ or m in the original model to create topic ℓ in q'. All other parameters are copied over unchanged. Formally,

$$\hat{r}'_{dn\ell} = \hat{r}_{dn\ell} + \hat{r}_{dnm}, \forall d, n, \; \hat{\theta}'_{d\ell} = \hat{\theta}_{d\ell} + \hat{\theta}_{dm}, \forall d. \quad (20)$$

A global update to create $\hat{\tau}', \hat{\rho}', \hat{\omega}'$ completes the candidate, and we keep it if the objective $\mathcal{L}$ improves.