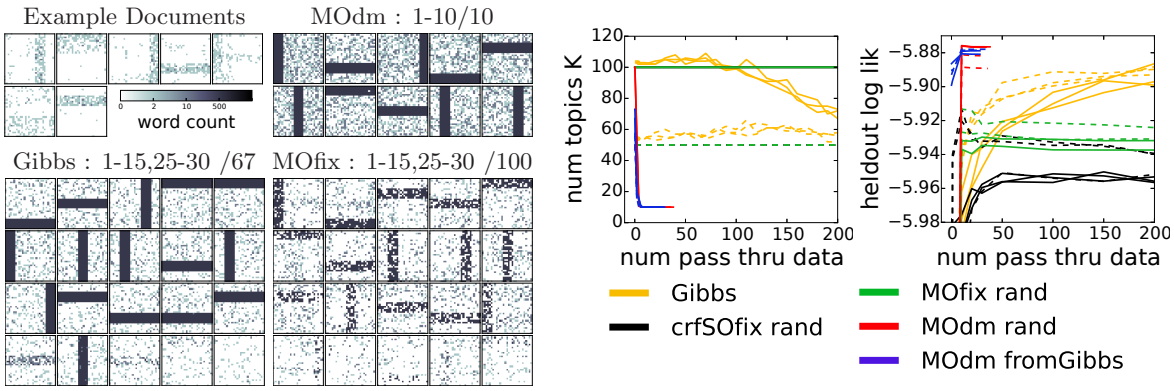


Figure 5: Comparison of inference methods on toy bars dataset from Sec. 6.1. *Top Left:* Word count images for 7 example documents and the final 10 estimated topics from MOdm. Each image shows all 900 vocabulary types arranged in square grid. *Bottom left:* Final estimated topics from Gibbs and MOfix. We rank topics from most to least probable, and show ranks 1-15 and 25-30. *Right:* Trace plots of the number of topics $K$ and heldout likelihood during training. Line style indicates number of initial topics: dashed is $K = 50$, solid is $K = 100$.

Sudderth (2012). Each model is summarized by a point-estimate of the topic-word probabilities $\phi$. For each heldout document $d$ we randomly split its word tokens into two halves: $x'_d, x''_d$. We use the first half to infer a point-estimate of $\pi_d$, then estimate log-likelihood of each token in the second half $x''_d$.

$$\text{heldout-lik}(x|\phi) = \frac{\sum_{d \in \mathcal{D}_{test}} \log p(x''_d|\pi_d, \phi)}{\sum_{d \in \mathcal{D}_{test}} |x''_d|} \quad (22)$$

**Hyperparameters.** In all runs, we set $\gamma = 10$, $\alpha = 0.5$ and topic-word pseudocount $\bar{\tau} = 0.1$. Stochastic runs use the learning rate decay recommended in Bryant and Sudderth (2012): $\kappa = 0.5, \delta = 1$.

### 6.1 Toy bars dataset.

We study a variant of the toy bars dataset of Griffiths and Steyvers (2004), shown in Fig. 5. There are 10 ideal bar topics, 5 horizontal and 5 vertical. The bars are noisier than the original and cover a larger vocabulary (900 words). We generate 1000 documents for training and 100 more for heldout test. Each one has 200 tokens drawn from 1-3 topics.

Fig. 5 shows many runs of all algorithms on this benchmark. Variational methods initialized with 50 or 100 topics get stuck rapidly, while the Gibbs sampler finds a redundant set of the ideal topics and is unable to effectively merge down to the ideal 10.

In contrast, our MOdm method uses merges and deletes to rapidly recover the 10 ideal bars after only a few laps. Without these moves, MOfix runs remain stuck at suboptimal fragments of bars. Furthermore, our MOdm method initialized with the sampler's final topics (fromGibbs) easily recovers the ideal bars.

### 6.2 Academic and news articles.

Next, we apply all methods to papers from the NIPS conference, articles from Wikipedia, and articles from the journal Science (Paisley et al., 2011), with 80%-20% train-test splits. Online methods process each training set in 20 batches. Trace plots in Fig. 6 compare predictive power and model complexity as more data is processed. We summarize conclusions below.

**Anchor topics are good; variational is better.** Using the anchor word method (Arora et al., 2013) for initial topic-word parameters yields better predictions than random initialization (`rand`). However, our methods can still make big, useful changes from this starting point. See Fig. 4 for some examples.

**Deletes and merges make big, useful changes.** Across all 3 datasets in Fig. 6, merges and deletes remove many topics. On Wikipedia, we reduce 200 topics to under 100 while improving predictions. Similar gains occur from the final result of the Gibbs sampler.

**Competitors get stuck or improve slowly.** The Gibbs sampler needs many laps to make quality predictions. The CRF method gets stuck quickly, while our methods (using the direct assignment representation) do better from similar initializations. The stochastic split-merge method (SOsm) grows to a prescribed maximum number of topics but fails to make better predictions. This indicates problems with heuristic acceptance rules, and motivates our moves governed by exact evaluation of a whole-dataset objective.

Next, we analyze the New York Times Annotated Corpus: 1.8 million articles from 1987 to 2007. We withhold 800 documents and divide the remainder into 200 batches (9084 documents per batch). Fig. 6 shows the predictive performance of the more-scalable methods.