# Improving ETM Topic Quality with Hard Top-$k$ Decoder Normalization

**Anonymous ACL submission**

## Abstract

Neural topic models are commonly interpreted through short top-word lists, but in embedding-based decoders those lists depend on how topic–word logits are normalized. In the Embedded Topic Model (ETM), the default full-vocabulary softmax couples all words through normalization, so learning signal can be dominated by low-probability tail words that rarely affect topic summaries. We replace the ETM decoder softmax with hard Top-$k$ normalization: for each topic, only the $k$ highest-logit words participate in normalization, with a tiny uniform mixture for numerical safety. This decoder-only swap leaves the encoder, inference network, and training objective unchanged. Across three corpora, hard Top-$k$ yields higher topic quality (the product of NPMI topic coherence and topic diversity), with gains driven primarily by increased diversity. On WikiText-103 with a 20k vocabulary, topic quality improves from 0.122 to 0.212 under the same tuning protocol. A vocabulary sweep up to 30k shows the best Top-$k$ configuration remains above the best softmax baseline at every vocabulary size. Gradient diagnostics are consistent with Top-$k$ concentrating logit updates on the vocabulary head reflected in top-word displays, while full softmax allocates most gradient mass to the tail.

## 1 Introduction

Topic models remain a practical tool for exploring and summarizing large text collections, especially when interpretability matters more than prediction. In many NLP settings, users rely on topic models to obtain human-readable descriptors of latent themes, to support exploratory analysis, labeling, and qualitative auditing. Neural topic models have made this workflow more scalable by combining probabilistic structure with amortized inference, and embedding-based models in particular offer an appealing inductive bias by tying topic structure to distributional representations. In practice, a common interface between a learned model and a human reader is a small list of top words per topic. Because these lists are often used for interpretation and evaluation, the modeling choices that determine which words appear at the top of each topic can strongly affect how useful the method is in practice. Yet many neural topic models use a standard decoder normalization choice, even when it is not selected specifically for the top-word interpretability interface.

A key choice in embedding-based neural topic models is how topic–word logits are converted into a categorical topic–word distribution. In ETM-style decoders, logits provide a score over the entire vocabulary, but the model is interpreted almost entirely through the small head of the resulting distribution (the top-word list). A common choice is full-vocabulary softmax, which couples all vocabulary items through normalization; as a result, learning signal is not necessarily concentrated on the head words that define interpretability. This creates a potential mismatch: the human-facing summary depends on a few dozen words, while optimization can allocate substantial gradient mass to the long tail of low-probability words that rarely enter topic displays.

We ask whether restricting decoder support can better align learning signal with the head words that define topic interpretability. To test this, we make a controlled decoder-only change to ETM: we replace full-vocabulary softmax with hard Top-$k$ normalization applied independently per topic, so only the $k$ highest-logit words participate in normalization and the rest are masked. We add a tiny uniform mixture term ($\varepsilon = 10^{-8}$) to keep probabilities finite; the encoder, inference network, and training objective remain unchanged under the same tuning protocol and multi-seed evaluation.

**Contributions.** (i) We frame decoder normalization as an interpretability-critical design choice in embedding-based topic models, motivated by a

head-vs-tail mismatch between top-word inspection and full-softmax gradient allocation; (ii) we propose hard Top-$k$ normalization as a minimal decoder swap that caps per-topic support to $k$ without changing the ETM objective or inference network; (iii) we show that Top-$k$ improves topic quality across IMDb, 20 Newsgroups, and WikiText, and that the advantage persists through a WikiText vocabulary sweep up to 30k; (iv) we provide mechanistic analyses (gradient mass and top-word churn) that relate the decoder change to optimization behavior and provide evidence that topics can still evolve despite non-differentiable Top-$k$ selection.

Empirically, hard Top-$k$ improves topic quality relative to a tuned softmax baseline across all three datasets we evaluate, with gains driven primarily by higher topic diversity. Coherence changes are smaller and can trade off with diversity depending on the dataset and hyperparameters. Section 5 reports the full quantitative results, including comparisons to sparsemax and entmax and sensitivity to $k$ and vocabulary size.

Because hard Top-$k$ explicitly masks low-rank words, it raises reasonable concerns about optimization and convergence: if gradients do not flow through cut-off words, do topics become brittle, freeze early, or fail to explore beyond an initial set of top words? We address these concerns with mechanistic analyses that directly measure gradient behavior and the evolution of top-word sets. We quantify how gradient magnitude evolves over training and how much gradient mass flows through the vocabulary tail versus the topic head, and we track the stability and drift of the human-interpretable top words using Top-$k$ churn metrics. Together, these analyses are intended to connect the observed improvements in diversity and topic quality to concrete training dynamics, while clarifying how topics can still change substantially over training despite sparse per-step support.

## 2 Related Work

Topic modeling has advanced in many directions, but we focus on a design choice that directly shapes interpretability in embedding-based neural topic models: how topic–word logits are mapped into a categorical topic–word distribution. This mapping determines the top-word summaries used for interpretation and, by construction, determines which logits participate in normalization and therefore receive nontrivial gradient signal. We situate our work in (i) probabilistic topic models that encourage sparse topic–word usage, (ii) neural and embedding-based topic models that parameterize topic–word logits, and (iii) alternative output mappings that alter the support and gradient behavior of categorical distributions.

**Sparsity via priors and latent structure.** Classical probabilistic topic models and their extensions often encourage sparse representations by modifying priors or latent structure. Latent Dirichlet Allocation represents each document as a mixture of topics with multinomial topic–word distributions (Blei et al., 2003), and correlated topic models replace the Dirichlet prior with a logistic normal to model topic correlations (Blei and Lafferty, 2005). Other work promotes sparsity by encouraging narrow effective supports for topics or terms (Lin et al., 2014), including Bayesian formulations that explicitly favor sparse topic–word usage (Chien and Chang, 2014). These approaches induce sparsity through probabilistic structure and regularization in the generative model; in contrast, we study sparsity introduced directly at the decoder mapping from logits to probabilities.

**Neural and embedding-based topic models.** Neural topic models use amortized inference and VAE-style optimization to scale bag-of-words topic modeling with explicit document–topic representations (Miao et al., 2016; Srivastava and Sutton, 2017). Embedding-based variants additionally parameterize topics and words in a shared space, producing topic–word logits via inner products between topic and word embeddings (Moody, 2016; Dieng et al., 2020). In these models, logits are an intermediate representation, while interpretability and standard topic evaluation depend on the categorical distribution obtained after normalization. We leverage this separation to isolate a decoder-side change: we keep the encoder, inference network, and training objective fixed and modify only the mapping used to form topic–word distributions from logits.

**Sparse output mappings.** A related line of work changes the softmax transformation itself to produce sparse output distributions with different support and gradient properties. Sparsemax replaces softmax with a projection onto the simplex, yielding exact zeros and restricting gradients to the active set under that projection (Martins and Astudillo, 2016). The entmax family provides a con-

tinuum between softmax and sparsemax and has been used as a sparse alternative for NLP output distributions (Peters et al., 2019); follow-up work targets more efficient computation for large output spaces (Tezekbayev et al., 2022). Within topic modeling, sparse probability mappings have also been incorporated into topic-related objectives to promote sparse structure (Lin et al., 2019). Our approach differs in mechanism and granularity: we impose a fixed-cardinality Top-$k$ restriction separately for each topic's logits, rather than using a continuous projection-based mapping, and we analyze how this per-topic support cap concentrates learning signal and influences the evolution of top-word summaries in embedding-based topic models.

## 3 Method

### 3.1 Embedded Topic Model decoder

We build on the Embedded Topic Model (ETM), which represents each topic as a distribution over the vocabulary obtained by combining topic embeddings with word embeddings. Let $V$ be the vocabulary size, $K$ the number of topics, and $L$ the embedding dimension. Let $\rho \in \mathbb{R}^{V \times L}$ be the word embedding matrix, where row $\rho_v$ is the embedding of word $v$, and let $\alpha_t \in \mathbb{R}^L$ be the embedding of topic $t$.

For each topic $t$, ETM forms topic–word logits $\ell_t \in \mathbb{R}^V$ by a bilinear form between word and topic embeddings:

$$\ell_t(v) = \rho_v^\top \alpha_t, \quad v \in \{1, \dots, V\}. \tag{1}$$

The baseline decoder converts logits to a topic–word distribution $\phi_t \in \Delta^{V-1}$ using a temperature-scaled softmax,

$$\phi_t(v) = \frac{\exp(\ell_t(v)/\tau)}{\sum_{v'=1}^{V} \exp(\ell_t(v')/\tau)}, \tag{2}$$

where $\tau > 0$ controls distribution sharpness. We keep the encoder, variational inference network, and training objective fixed throughout.

### 3.2 Hard Top-$k$ normalization

We propose a decoder-side replacement for the baseline softmax normalization that restricts normalization, and thus learning signal, to a small set of vocabulary items per topic. Fix a cutoff $k$. For each topic $t$, let $\mathcal{S}_t \subseteq \{1, \dots, V\}$ denote the indices of the $k$ largest logits in $\ell_t(v)$. We construct a masked distribution by setting all logits outside $\mathcal{S}_t$ to $-\infty$ and applying softmax:

$$\tilde{\phi}_t(v) = \frac{\exp(\ell_t(v)/\tau)\,\mathbf{1}[v \in \mathcal{S}_t]}{\sum_{v'=1}^{V} \exp(\ell_t(v')/\tau)\,\mathbf{1}[v' \in \mathcal{S}_t]}. \tag{3}$$

Equivalently, $\tilde{\phi}_t$ is a categorical distribution supported only on $\mathcal{S}_t$, with probabilities proportional to the exponentiated logits within that set. We apply this independently per topic.

To avoid zero probabilities and keep log probabilities finite, we mix the masked distribution with a small uniform background:

$$\phi_t(v) = (1 - \varepsilon)\tilde{\phi}_t(v) + \varepsilon \cdot \frac{1}{V}, \tag{4}$$

where $\varepsilon \geq 0$ is a constant. We use $\varepsilon = 10^{-8}$ in all experiments; this term is a numerical safeguard, and in our settings it has a negligible effect on the effective support and top word rankings.

### 3.3 Implementation details

Given a logits tensor of shape $[V, K]$, we apply temperature scaling, select the top $k$ entries per topic $t$ with `topk`, construct a boolean mask, set masked out logits to $-\infty$, and compute a softmax to obtain $\tilde{\phi}$. The Top-$k$ *selection* itself is non-differentiable: in backpropagation the mask is treated as fixed (i.e., no straight-through estimator is used), so gradients flow only through the logits of the currently selected Top-$k$ words. We then apply the uniform mixing (with $\varepsilon = 10^{-8}$ in all experiments) and store $\log \phi$ for numerical stability. This procedure has no effect on the encoder or inference network and can be used as a direct replacement for the decoder's full-vocabulary softmax. We choose $k$ and $\tau$ using the same search protocol as the baseline.

## 4 Experimental Setup

### 4.1 Datasets and preprocessing

We evaluate on three corpora with different vocabulary sizes: IMDb reviews (10k vocabulary) (Maas et al., 2011), 20 Newsgroups (4034 vocabulary) (Lang, 1995), and WikiText-103 (20k vocabulary by default) (Merity et al., 2017). For each dataset, we construct the vocabulary by selecting the top $V$ most frequent tokens in the corpus, where $V$ is the target vocabulary size for that dataset. All datasets are preprocessed with the same pipeline (details in Appendix A.2). We lowercase text and tokenize with spaCy (Honnibal et al., 2020) using `en_core_web_sm` with the parser and

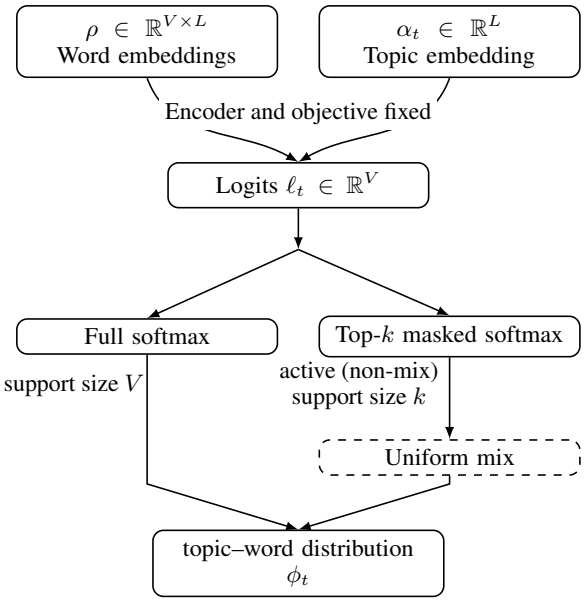



Figure 1: Decoder normalization swap: full softmax vs. hard Top-$k$ masked softmax with uniform mixing to produce $\phi_t$.

NER disabled. We keep alphabetic tokens only and remove spaCy stopwords, then lemmatize tokens; if a lemma has length $\leq 1$, we fall back to the surface form. We additionally remove Gensim stopwords (Řehůřek and Sojka, 2010) and discard tokens with length $< 3$. Documents that become empty after filtering are removed. Vocabularies are constructed from the top $V$ most frequent tokens after preprocessing. Unless otherwise stated, results on WikiText use the 20k vocabulary setting; additional experiments vary WikiText vocabulary size.

### 4.2 Model configuration

Across all datasets and runs, we fix the number of topics to $K = 128$. We compare four decoder mappings for converting topic–word logits into categorical topic–word distributions: full softmax, hard Top-$k$, sparsemax, and entmax. For Top-$k$, we use the uniform mixture term described in Section 3.2 with $\varepsilon = 10^{-8}$ in all experiments. For entmax, $\alpha_{\text{ent}}$ denotes the entmax sparsity parameter, with $\alpha_{\text{ent}} = 1$ recovering softmax and $\alpha_{\text{ent}} = 2$ corresponding to sparsemax.

### 4.3 Evaluation metrics

We report topic diversity (TD), topic coherence (TC), and topic quality (TQ). Each metric is computed from the top $N$ words per topic, and we report the average over $N \in \{5, 10, 15\}$; thus averaged TQ need not equal averaged TC times averaged TD. Topic coherence (TC). We use normalized pointwise mutual information (NPMI) computed from corpus co-occurrence statistics (Bouma, 2009; Lau et al., 2014; Röder et al., 2015). Co-occurrence probabilities are estimated using a sliding window of size 10 over the tokenized corpus, following common practice in automated topic coherence evaluation (Lau et al., 2014; Röder et al., 2015). For a pair of words $(w_i, w_j)$, NPMI is

$$\text{NPMI}(w_i, w_j) = \frac{\log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}}{-\log p(w_i, w_j)}. \quad (5)$$

For a topic represented by its top $N$ words $\{w_{t,1}, \ldots, w_{t,N}\}$, we define the topic-level coherence at $N$ as

$$\text{TC}_t(N) = \frac{1}{\binom{N}{2}} \sum_{1 \leq i < j \leq N} \text{NPMI}(w_{t,i}, w_{t,j}). \quad (6)$$

We define corpus-level coherence at $N$ as the mean over topics,

$$\text{TC}(N) = \frac{1}{K} \sum_{t=1}^{K} \text{TC}_t(N). \quad (7)$$

Topic diversity (TD). For a set of topics, TD is the fraction of unique words among the top $N$ words across all topics:

$$\text{TD}(N) = \frac{\left| \bigcup_{t=1}^{K} \{w_{t,1}, \ldots, w_{t,N}\} \right|}{NK}. \quad (8)$$

Topic quality (TQ). Following prior work that combines coherence and diversity, we define topic quality at $N$ as the product (Dieng et al., 2020):

$$\text{TQ}(N) = \text{TC}(N) \cdot \text{TD}(N). \quad (9)$$

### 4.4 Experimental protocol

For each method and parameter setting, we run five random seeds and report the mean and standard deviation across seeds. When tuning decoder hyperparameters (e.g., temperature $\tau$, the Top-$k$ cutoff $k$, and the entmax parameter $\alpha_{\text{ent}}$), we apply the same multi-seed search procedure for all methods, selecting configurations according to the target metric for that experiment.

## 5 Results

### 5.1 Main comparison across datasets

Table 1 reports topic coherence (TC), topic diversity (TD), and topic quality (TQ) for four decoder

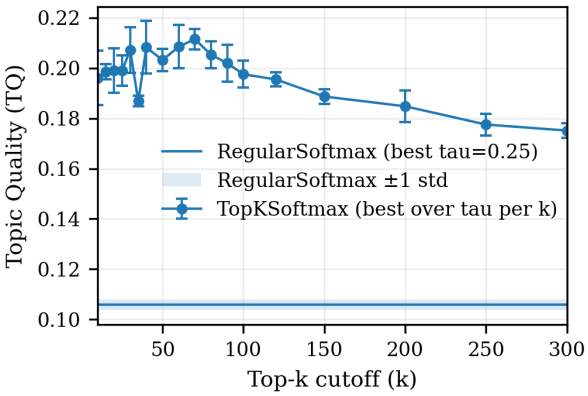



Figure 2: Effect of the Top-$k$ cutoff on WikiText-20k.

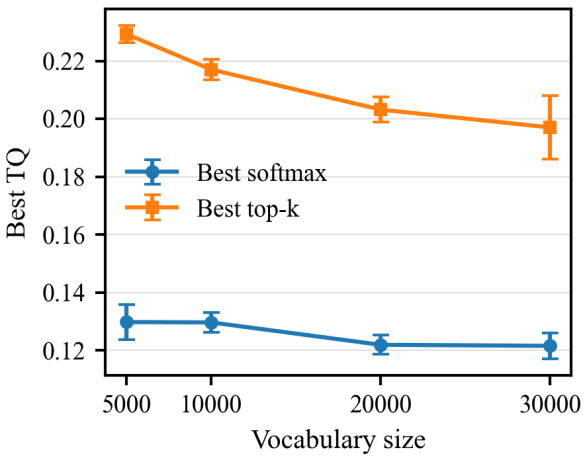



Figure 3: Vocabulary scaling on WikiText. Best topic quality (TQ) for Softmax and Top-$k$ as vocabulary size increases from 5k to 30k.

mappings: full softmax, hard Top-$k$, sparsemax, and entmax. Hard Top-$k$ improves topic quality relative to the best tuned softmax baseline on all three datasets, with gains most consistently driven by substantial increases in topic diversity. On WikiText (20k vocabulary), Top-$k$ increases TD from 0.399 to 0.689 and increases TQ from 0.122 to 0.212. On 20 Newsgroups, Top-$k$ increases TD from 0.382 to 0.680 and improves TQ from 0.083 to 0.121. On IMDb, Top-$k$ increases TD from 0.343 to 0.509 and improves TQ from 0.085 to 0.136. Compared to sparsemax and entmax, Top-$k$ is statistically better (two-sided paired exact randomization test using sign-flip permutations over the five random seeds; $p < 0.05$) in all settings except WikiText-20k, where the Top-$k$ improvement over sparsemax is not significant under the same test ($p \geq 0.05$).

Changes in coherence are smaller and less uniform than changes in diversity, and sparse output mappings can shift the TC–TD balance. For example, on 20 Newsgroups, TC decreases under the Top-$k$ configuration that maximizes TQ, while on IMDb coherence increases slightly. Overall, these results are consistent with Top-$k$ primarily improving the distinctiveness of the top words across topics, with coherence depending more sensitively on the dataset and the chosen hyperparameters.

### 5.2 Effect of the Top-$k$ cutoff on WikiText

To characterize the role of the cutoff $k$, we evaluate Top-$k$ across a range of values on WikiText with 20k vocabulary, selecting the temperature $\tau$ separately for each $k$ to maximize TQ. Figure 2 shows that Top-$k$ consistently outperforms the full softmax baseline across all evaluated cutoffs. Topic Quality increases from 0.196 at $k = 10$ to a peak of 0.212 at $k = 70$, with minor fluctuations at intermediate values, and then gradually declines for larger $k$. The softmax baseline remains at approximately 0.122 and does not approach Top-$k$ performance for any $k$.

These results indicate that restricting normalization to a moderate head set yields the highest topic quality in this setting, while very small or very large cutoffs lead to reduced performance. Importantly, even at the largest cutoffs, Top-$k$ maintains a substantial margin over the softmax baseline, indicating that the gains are robust across a wide range of $k$ values.

### 5.3 Vocabulary scaling on WikiText

We next test whether the benefits of Top-$k$ persist as vocabulary size increases. For WikiText vocabularies of 5k, 10k, 20k, and 30k, we select the best performing configuration for each method by TQ and report the corresponding mean and standard deviation across seeds. Figure 3 shows that Top-$k$ yields higher best TQ than softmax for every vocabulary size considered.

At 5k vocabulary, the best softmax configuration achieves a best TQ of roughly 0.130, while Top-$k$ reaches approximately 0.229 with much smaller variability. At 10k, softmax remains near 0.130, whereas Top-$k$ attains about 0.217. At 20k, softmax drops to roughly 0.122 and Top-$k$ to about 0.212, preserving a large gap. At 30k, softmax remains near 0.122 while Top-$k$ decreases slightly to approximately 0.197; the variance for Top-$k$ is noticeably larger at 30k than at smaller vocabularies.

Overall, both methods show decreasing best TQ as vocabulary grows, but Top-$k$ remains consis-

Table 1: Results on WikiText-20k, 20 Newsgroups, and IMDb. Metrics are mean (std) across five seeds.

| Dataset | Method | $k$ | $\tau$ | $\alpha_{\text{ent}}$ | TC | TD | TQ |
|---|---|---|---|---|---|---|---|
| WikiText-20k | Softmax | – | 0.250 | – | $0.274 \pm 0.003$ | $0.399 \pm 0.010$ | $0.122 \pm 0.003$ |
| | Top-$k$ | 70 | 0.250 | – | $0.307 \pm 0.010$ | $0.689 \pm 0.016$ | $0.212 \pm 0.004$ |
| | Sparsemax | – | 2 | – | $0.254 \pm 0.017$ | $0.816 \pm 0.013$ | $0.208 \pm 0.012$ |
| | Entmax | – | 1 | 1.7 | $0.235 \pm 0.014$ | $0.833 \pm 0.011$ | $0.196 \pm 0.011$ |
| 20 Newsgroups | Softmax | – | 0.250 | – | $0.243 \pm 0.011$ | $0.382 \pm 0.024$ | $0.083 \pm 0.005$ |
| | Top-$k$ | 25 | 0.250 | – | $0.203 \pm 0.014$ | $0.680 \pm 0.004$ | $0.121 \pm 0.005$ |
| | Sparsemax | – | 10 | – | $0.104 \pm 0.006$ | $0.588 \pm 0.013$ | $0.061 \pm 0.004$ |
| | Entmax | – | 0.250 | 1.3 | $0.136 \pm 0.004$ | $0.554 \pm 0.026$ | $0.075 \pm 0.004$ |
| IMDb | Softmax | – | 0.250 | – | $0.247 \pm 0.006$ | $0.343 \pm 0.005$ | $0.085 \pm 0.002$ |
| | Top-$k$ | 25 | 0.250 | – | $0.268 \pm 0.010$ | $0.509 \pm 0.019$ | $0.136 \pm 0.002$ |
| | Sparsemax | – | 4 | – | $0.024 \pm 0.012$ | $0.674 \pm 0.011$ | $0.016 \pm 0.008$ |
| | Entmax | – | 0.250 | 1.3 | $0.196 \pm 0.005$ | $0.418 \pm 0.007$ | $0.082 \pm 0.002$ |

tently superior through 30k. The larger variance at 30k suggests that sensitivity to hyperparameters, initialization, or both increases at the largest vocabulary size, reinforcing the practical importance of selecting sensible $k$ values and tuning protocols when scaling vocabularies.

## 6 Mechanistic analysis

Hard Top-$k$ normalization changes the decoder's support by construction: only the $k$ highest-logit words per topic participate in normalization (Section 3.2). This raises a direct optimization concern: if gradients only flow through the active set, topics might lock into early top-word choices and fail to revise their interpretable summaries over training. We address this concern with two complementary analyses on WikiText with $V = 30$k and $k = 50$: (i) direct measurements of gradient magnitude and where gradients flow in the vocabulary, and (ii) stability and drift measurements of the human-interpretable top word sets over training.

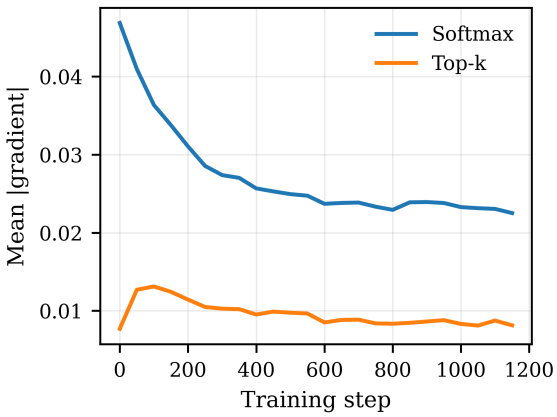
### 6.1 Gradient magnitude and tail gradient flow

We analyze gradient behavior under full softmax and Top-$k$ decoding using two complementary views, shown in Figure 4. The top panel reports the overall scale of gradients during training, while the bottom panel shows how gradient mass is distributed across the vocabulary.

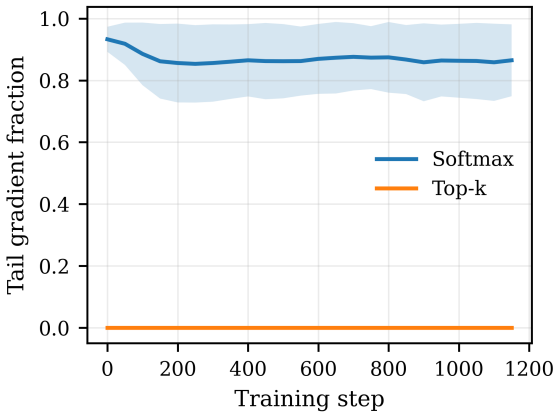
The top plot shows the mean absolute gradient magnitude over training. Full softmax produces substantially larger gradients throughout optimization. The mean $|\nabla|$ under softmax starts at approximately 0.047 and gradually decays to about 0.023 by step 1150. In contrast, Top-$k$ yields gradients that are smaller by roughly a factor of three to six: the mean $|\nabla|$ begins near 0.0077, briefly increases to an early peak of about 0.013 around step 100, and then stabilizes near 0.0081 by step 1150. This reduction is consistent with Top-$k$ restricting normalization to a limited subset of logits, thereby reducing the number of parameters receiving substantial updates at each step.

The bottom plot examines where these gradients flow by measuring the fraction of gradient mass assigned to the tail of the vocabulary, defined as all words outside the top 50 ranks within each topic at a given training step. Under full softmax, the tail dominates gradient flow: the tail fraction begins around 0.93, decreases to roughly 0.85 early in training, and then stabilizes between 0.86 and 0.87. Under Top-$k$ with $k$=50, the tail fraction is zero for gradients with respect to the topic–word logits, by construction: masked-out logits do not participate in the forward pass and we do not backpropagate through the Top-$k$ selection (Section 3.2). We include this plot to make the contrast with softmax explicit and as a check that the measured tail fraction matches the masking behavior implied by the decoder.

Taken together, these measurements quantify the gradient redistribution induced by masking: at the logit level, Top-$k$ updates only the currently active set, while full softmax distributes gradient mass broadly across the vocabulary, including the long tail. This offers a plausible account of why Top-$k$ tends to increase topic diversity: by emphasizing optimization of the same head words that dominate




(a) Mean absolute gradient magnitude over training




(b) Fraction of gradient mass in the vocabulary tail

Figure 4: Gradient behavior under full softmax and Top-$k$: overall magnitude (top) and tail gradient fraction (bottom).

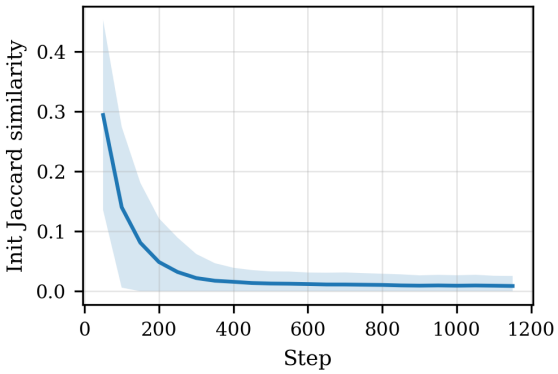



Figure 5: Init Jaccard similarity of top-50 word sets over training for Top-$k$ on WikiText-30k ($k$=50). Shaded band shows $\pm 1$ std across topics.

each topic's displayed top list, Top-$k$ can encourage topic specialization rather than repeatedly adjusting many low-probability tail words that rarely affect the top-word summaries.

### 6.2 Stability versus drift of top word sets

A second concern is that restricting gradient flow to the current top $k$ words might lock topics into their initial top word sets and prevent the model from discovering better topic signatures. We test this using Top-$k$ churn metrics that track how top word membership evolves over training. We focus on the Jaccard similarity between the current top 50 set and an "initial" snapshot recorded early in training.

Figure 5 shows that the initial set similarity collapses rapidly, indicating that the final top word sets are not simply those that happen to be highest ranked at initialization. The init Jaccard similarity starts at 0.29 at step 50 and drops sharply over the next few hundred steps (0.14 at step 100, 0.081 at step 150, 0.049 at step 200), reaching 0.016 by step 400. After this early phase, the curve remains near zero but continues to decay slowly, ending at 0.0087 by step 1150. This implies that by the end of training, the overlap between the final top 50 sets and the initial top 50 sets is only a small fraction of what it was early in training.

These dynamics reconcile two seemingly conflicting properties of hard Top-$k$. On one hand, masking implies that only a small portion of the vocabulary is updated at each step. On the other hand, the interpretability-relevant top word sets can still change substantially over the course of training. The observed pattern is consistent with an accumulation process in which changes occur primarily near the cutoff boundary: words close to rank $k$ occasionally exchange membership with nearby candidates, and once a word enters the top $k$ it begins receiving learning signal. Although each individual update alters the set only slightly, these rare boundary swaps compound over many steps, yielding large long-range drift even when step-to-step membership changes are small.

### 6.3 Implications for the Top-$k$ mechanism

Overall, the gradient and churn analyses are consistent with a simple account of the Top-$k$ effects. Full softmax distributes gradient across nearly the entire vocabulary, with most gradient mass flowing through words outside the top list that defines topic interpretability. Hard Top-$k$ restricts gradient flow to the active set at the logit level by construction, so updates focus on the same head region reflected in the displayed topic–words. The churn results provide evidence that topics nonetheless remain

dynamic: top word sets quickly diverge from early snapshots and continue to drift over training via accumulated rank crossings near the cutoff. Together, these observations connect the decoder change to the gains in topic diversity and topic quality observed in Section 5.

## 7 Discussion

Hard Top-$k$ normalization restricts each topic–word distribution to high-logit words, changing which items participate in normalization and receive learning signal. Because topic inspection uses top-word lists, this mapping directly affects interpretability. Across datasets, topic quality gains are driven mainly by higher topic diversity, yielding more distinct top-word summaries with fewer repeated high-frequency terms. Coherence behaves differently: under the configurations that maximize topic quality, coherence can decrease (as observed on 20 Newsgroups) or increase (as observed on IMDb), while on WikiText it remains close. This suggests Top-$k$ shifts the balance toward distinct topic summaries, and coherence depends on the dataset and on the cutoff and temperature. The mechanistic evidence in Section 6 suggests an association between the diversity gains and decoder updates shifting toward the same high-rank words used for topic inspection. Under full softmax, much of the logit gradient magnitude is allocated to low-probability tail words, whereas hard Top-$k$ concentrates updates on a topic-specific active set; despite the non-differentiable selection, top-word sets continue to evolve through occasional rank crossings near the cutoff that accumulate over training. These dynamics also clarify where Top-$k$ should help and where it may not. When topics are consumed primarily as short top-word lists and interpretability is the main objective, emphasizing the vocabulary head can produce more distinct, human-readable topic labels. Conversely, if an application depends on probability mass beyond the displayed top list (e.g., rare-word coverage or likelihood-sensitive downstream use), overly small cutoffs can exclude informative mid-rank words and reduce coherence. Vocabulary scaling results further suggest that hard Top-$k$ normalization remains effective up to 30k vocabulary, but they also highlight practical tradeoffs. The advantage in best topic quality persists across 5k to 30k, while the best cutoff shifts with vocabulary size and with whether one optimizes topic quality or coherence. The increased variance observed at 30k is consistent with greater sensitivity to hyperparameters and noisier boundary movement around the cutoff. Mechanistically, this approach differs from sparsity induced by priors in probabilistic topic models (Blei et al., 2003; Blei and Lafferty, 2005) and from projection-based sparse output mappings such as sparsemax and entmax (Martins and Astudillo, 2016; Peters et al., 2019): hard Top-$k$ normalization enforces a fixed-cardinality support cap per topic, producing an explicit separation between active and inactive words at each step. For practitioners using ETM-style models, these findings suggest treating the decoder mapping as a tunable design choice: tune $k$ and temperature jointly, monitor the diversity–coherence balance, and pay attention to variability at larger vocabularies. When interpretability is the priority, hard Top-$k$ normalization offers a targeted way to concentrate learning on the words that define topic summaries while still allowing topics to evolve over training. In conclusion, swapping ETM's full-vocabulary softmax for hard Top-$k$ improves topic quality mainly via diversity, while coherence varies with data and tuning. Gradient and churn analyses suggest that Top-$k$ concentrates learning on the vocabulary head reflected in displayed topic summaries while still allowing topics to evolve.

## Limitations

This study evaluates hard Top-$k$ normalization only within ETM-style decoders and only on three bag-of-words datasets (IMDb reviews, 20 Newsgroups, and WikiText) with fixed topic count ($K = 128$). This matters because the conclusions may not transfer to other topic models, inference setups, or text regimes beyond those tested. As a result, the findings should be interpreted as evidence about this specific decoder change rather than a general statement about sparsifying output distributions. Readers should validate the effect in their own ETM variant and data setting before relying on it.

All datasets use a shared preprocessing pipeline (Gensim stopword removal and dropping tokens shorter than three characters) and vocabularies defined by top-frequency tokens. This matters because both coherence and diversity are sensitive to tokenization and vocabulary construction, and different preprocessing could change the balance between repeated high-frequency words and rare words. The reported improvements could therefore

depend partly on these choices. A practical check is to re-run with the same model and tuning but alternative, task-appropriate preprocessing.

The evaluation focuses on TC (NPMI), TD, and their product TQ, averaged over top ($N \in \{5, 10, 15\}$). This matters because these metrics emphasize top-word summaries and may not capture other aspects of topic usefulness, and TQ can mask tradeoffs by conflating coherence and diversity. Consequently, improvements in TQ driven by TD do not necessarily imply better semantic coherence or downstream utility. Readers should inspect qualitative topic lists and consider whether the observed diversity aligns with their interpretability needs.

The mechanistic evidence is limited to gradient summaries and churn statistics from specific configurations (e.g., WikiText 30k with ($k = 50$)). This matters because these measurements support a consistent narrative but do not establish causality or exclude alternative explanations for diversity gains. As a result, claims about mechanism should be taken as suggestive rather than definitive. Readers should confirm that similar gradient concentration and stability–drift patterns hold in their settings of interest.

## Acknowledgments

Acknowledgments are omitted in the review version.

## References

David M. Blei and John D. Lafferty. 2005. Correlated topic models. In *Advances in Neural Information Processing Systems 18*, pages 147–154.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. In *From Form to Meaning: Processing Texts Automatically, Proceedings of the Biennial GSCL Conference 2009*, pages 31–40, Tübingen.

Jen-Tzung Chien and Ying-Lan Chang. 2014. Bayesian sparse topic model. *Journal of Signal Processing Systems*, 74:375–389.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML '95)*, pages 331–339.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539, Gothenburg, Sweden. Association for Computational Linguistics.

Tianyi Lin, Zhiyue Hu, and Xin Guo. 2019. Sparsemax and relaxed wasserstein for topic sparsity. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*, pages 141–149, Melbourne, VIC, Australia. ACM.

Tianyi Lin, Wentao Tian, Qiaozhu Mei, and Hong Cheng. 2014. The dual-sparse topic model: Mining focused topics and focused terms in short text. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14)*, pages 539–550, Seoul, Republic of Korea. ACM.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

André F. T. Martins and Ramón Fernandez Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 1614–1623.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 1727–1736.

Christopher Moody. 2016. Mixing Dirichlet topic models and word embeddings to make lda2vec. *Preprint*, arXiv:1605.02019.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ben Peters, Vlad Niculae, and André F. T. Martins. 2019. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519, Florence, Italy. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM '15)*, pages 399–408.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. *Preprint*, arXiv:1703.01488.

Maxat Tezekbayev, Vassilina Nikoulina, Matthias Gallé, and Zhenisbek Assylbekov. 2022. Speeding up entmax. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1142–1158, Seattle, United States. Association for Computational Linguistics.

# A Additional Implementation and Reproducibility Details

This appendix provides implementation and training details for reproducing the experiments in the main paper.

## A.1 Datasets and splits

We use the following public datasets: (i) IMDb Large Movie Review dataset (Maas et al., 2011), (ii) 20 Newsgroups via the HuggingFace `SetFit/20_newsgroups` dataset loader, and (iii) WikiText-103 v1 via the HuggingFace `Salesforce/wikitext` dataset loader (Merity et al., 2017). For each corpus, we train on the full set of available documents (that is, we concatenate all official splits when train, validation, and test partitions are provided), since our objective is unsupervised topic discovery rather than held-out prediction. For topic coherence (NPMI), co-occurrence statistics are computed from the same full-corpus texts after preprocessing.

## A.2 Text preprocessing and tokenization

All datasets share the same preprocessing pipeline.

**Tokenizer and lemmatizer.** We tokenize with spaCy (Honnibal et al., 2020) using the English model `en_core_web_sm`, with the parser and NER components disabled for speed. Text is lowercased before tokenization. For each token, we keep only alphabetic tokens (`tok.is_alpha`) and drop spaCy stopwords (`not tok.is_stop`). Tokens are lemmatized using `tok.lemma_`; if the lemma length is $\leq 1$, we fall back to `tok.text`.

**Stopwords and short tokens.** After spaCy filtering, we apply Gensim stopword removal (Řehůřek and Sojka, 2010) (`remove_stopword_tokens`) and drop short tokens using `remove_short_tokens(minsize=3)`. Documents that become empty after filtering are removed.

**Vocabulary construction and BoW.** For each dataset, we build the vocabulary by selecting the top $V$ most frequent tokens after preprocessing (as described in the main paper). Bag-of-words vectors use raw token counts (not binarized, not TF-IDF). In our experiments, the selected vocabularies are fully covered by the pretrained embeddings used to initialize $\rho$, so no additional tokens are removed due to missing vectors.

## A.3 Model configuration

Across experiments we set the number of topics to $K = 128$ and the embedding dimension to $L = 50$.

**Encoder (BoW inference network).** The encoder maps a dense BoW vector $\mathbf{x}_d \in \mathbb{R}^V$ to a diagonal Gaussian posterior over latent topic logits:

$$q(\mathbf{z}_d \mid \mathbf{x}_d) = \mathcal{N}(\boldsymbol{\mu}_d, \mathrm{diag}(\boldsymbol{\sigma}_d^2)).$$

The encoder is a 2-layer MLP with hidden size 512, SiLU activations, and dropout 0.1 applied before linear layers. It outputs $\boldsymbol{\mu}_d$ and $\log \boldsymbol{\sigma}_d^2$ via two linear heads. During training we sample using the standard reparameterization trick; at evaluation we use $\mathbf{z}_d = \boldsymbol{\mu}_d$. Topic proportions are computed as

$$\boldsymbol{\theta}_d = softmax(\mathbf{z}_d),$$

**Word embeddings.** We initialize $\rho$ using 50-dimensional pretrained GloVe vectors trained on Wikipedia 2024 plus Gigaword 5 (11.9B tokens, 1.2M vocabulary, uncased) (Pennington et al., 2014). The word embedding matrix is frozen during training.

## A.4 Training objective and reconstruction loss

We optimize the standard VAE objective with reconstruction plus KL:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \mathrm{KL}(q(\mathbf{z} \mid \mathbf{x}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})) \,.$$

The KL term is computed in closed form against a standard normal prior and averaged over the minibatch.

**BoW reconstruction loss.** Given $\log p(w \mid d)$ for all vocabulary items and the raw-count BoW vector, the reconstruction loss is a length-normalized negative log-likelihood per document:

$$\mathcal{L}_{\text{recon}} = \frac{1}{B} \sum_{d=1}^{B} \left( - \frac{\sum_{v=1}^{V} x_{d,v} \, \log p(w{=}v \mid d)}{\max(1, \sum_{v=1}^{V} x_{d,v})} \right).$$

We use mean reduction over the minibatch.

### A.5 Optimization and stopping

We train with Adam (learning rate $10^{-3}$), batch size 256, dropout 0.1, and a maximum of 50 epochs. Early stopping uses patience 5 based on the total loss (reconstruction plus KL).

### A.6 Hyperparameter tuning protocol

For the softmax baseline we tune the decoder temperature:

$$\tau \in \{0.10,\ 0.25,\ 0.50,\ 0.75,\ 1.00,\ 1.25,\ 1.50,\ 2.00,\ 3.00,\ 4.00,\ 5.00,\ 7.00,\ 10.00\}.$$

For Top-$k$ we tune the cutoff $k$ and temperature $\tau$. For the main experiments on IMDb, 20 Newsgroups, and WikiText-20k, we use:

$$k \in \{10,\ 20,\ 30,\ 40,\ 50,\ 70,\ 100\},$$
$$\tau \in \{0.10,\ 0.25,\ 0.50,\ 0.75,\ 1.00\}.$$

For the WikiText-30k $k$ scaling experiment only, we sweep:

$$k \in \{10,\ 20,\ 30,\ 40,\ 50,\ 60,\ 70,\ 80,\ 90,\ 100,\ 120,\ 150,\ 200,\ 250,\ 300\},$$
$$\tau \in \{0.10,\ 0.25,\ 0.50,\ 0.75,\ 1.00\}.$$

For sparsemax we tune the decoder temperature:

$$\tau \in \{0.10,\ 0.25,\ 0.50,\ 1.00,\ 2.00,\ 3.00,\ 4.00,\ 5.00,\ 6.00,\ 8.00,\ 10.00\}.$$

For entmax we tune temperature and the entmax parameter $\alpha_{\text{ent}}$:

$$\tau \in \{0.10,\ 0.25,\ 0.50,\ 1.00,\ 1.25,\ 1.50,\ 2.00,\ 3.00,\ 4.00\},$$
$$\alpha_{\text{ent}} \in \{1.3,\ 1.5,\ 1.7\}.$$

For each configuration we run five random seeds and report mean and standard deviation. When selecting a configuration, we choose the setting that maximizes the target metric for that experiment (e.g., TQ for the main results). We use the same multi-seed tuning procedure for both methods.