LLMTopicTree: Backward-Pass Refinement of LLM-Guided Topic Hierarchies via Split-Classify-Merge with Optimal Transport

Anonymous ACL submission

Abstract

Hierarchical topic modeling requires both coherent topics and well-structured trees, yet existing approaches, from hLDA to neural methods like TraCo and SawETM, build hierarchies in a single forward pass without a backward pass, leaving structural defects uncorrected. We introduce LLMTopicTree, a framework that combines embedding-based clustering with LLM-guided labeling in a forward pass, then refines the resulting hierarchy with *Split-Classify-Merge* (SCM): a bottom-up backward pass that splits impure clusters, reclassifies boundary documents to better-fitting siblings, and merges confirmed duplicates. This refinement step is critical: it corrects the structural defects that single-pass methods leave behind, yielding consistent improvements in cluster purity and reduced redundancy. *OT-Classify*—a Sinkhorn optimal transport replacement for the LLM-based classify phase—improves Normalized Mutual Information (NMI) on all 6 datasets while reducing API costs by up to 23%. We evaluate on 6 benchmarks spanning 4 to 20 ground-truth classes against 6 baselines using a suite of flat and hierarchy-aware metrics. SCM-OT achieves an average NMI of 0.604 across 6 datasets (3 seeds each), outperforming TraCo (0.259), SawETM (0.327), BERTopic-Hier (0.430), HERCULES (0.508), and Forward-only (0.533) – a +19% relative improvement over HERCULES, the strongest LLM-HTM baseline (same backbone), and +40% over the strongest non-LLM baseline (BERTopic-Hier). SCM-OT wins all 6 datasets against the forward-only baseline and goes undefeated (6-0) against every neural hierarchical topic model.

1 Introduction

Topic modeling has evolved from probabilistic methods like LDA (Blei et al., 2003) through neural approaches (Srivastava and Sutton, 2017; Dieng et al., 2020) to recent LLM-based systems (Pham et al., 2024; Stammbach et al., 2023). While most methods produce flat topic sets, hierarchical approaches—hLDA (Blei et al., 2010), BERTopic (Grootendorst, 2022), TraCo (Wu et al., 2024), SawETM (Duan et al., 2021)—organize topics into trees that capture the multi-granularity structure of real-world corpora. Classical agglomerative and divisive hierarchical clustering algorithms iteratively merge or split clusters during tree construction (Murtagh and Contreras, 2012), and methods like BIRCH (Zhang et al., 1996) maintain feature-tree summaries that can be incrementally refined. However, modern hierarchical *topic* models—both probabilistic (hLDA, nCRP) and neural (TraCo, SawETM, BERTopic-Hier)—construct the final topic tree in a single inference pass over fixed document representations, leaving structural defects uncorrected: documents assigned to impure clusters, redundant sibling topics, and boundary documents stranded far from their semantic neighbors. To our knowledge, no prior work performs backward-pass optimization of document-cluster assignments within an LLM-guided topic tree.

We introduce LLMTopicTree, a framework that combines embedding-based clustering with LLM-guided labeling in a forward pass, then refines the resulting hierarchy with *Split-Classify-Merge* (SCM). SCM is a bottom-up refinement pass that (1) *splits* impure clusters via k-means sub-clustering with LLM validation, (2) *classifies* boundary documents by reassigning them to better-fitting siblings, and (3) *merges* confirmed duplicate topics. For the classify phase, we introduce *OT-Classify*, which replaces LLM classification calls with Sinkhorn optimal transport (Cuturi, 2013), reducing API calls by up to 23% while often improving reassignment quality.

Evaluating hierarchical topic models is itself an open problem. Standard metrics such as NMI and topic coherence (Lau et al., 2014) measure flat cluster quality but ignore hierarchy-specific properties:

whether subtrees are pure, whether siblings are distinct, and whether specificity increases with depth. We propose Hierarchical NMI and Per-Parent Purity, and adapt Hierarchical F1 (Kiritchenko et al., 2005; Silla and Freitas, 2011) from hierarchical text classification, as hierarchy-aware metrics that probe failure modes flat NMI misses. Together with standard clustering and coherence metrics, this yields a comprehensive evaluation framework applied across 6 datasets and 6 baselines (including HERCULES (Petneházi and Aradi, 2025), a recent LLM-HTM that builds hierarchies in a single forward pass without backward refinement).

Our contributions are:

1. LLMTopicTree, a framework for hierarchical topic discovery that uses LLMs for labeling, merging, validation, and backward-pass refinement of topic trees.

2. *Split-Classify-Merge* (SCM), a bottom-up refinement pass—the first automated structural optimization of document-cluster assignments within an LLM-generated topic hierarchy.

3. *OT-Classify*, a Sinkhorn optimal transport replacement for the LLM classify phase that outperforms LLM-based classification on all 6 datasets while reducing API calls by up to 23%.

4. Comprehensive evaluation on 6 datasets against 6 baselines using flat and hierarchy-aware metrics across 3 random seeds (162 runs total). SCM-OT achieves an average NMI of 0.604, outperforming all baselines.

2 Related Work

Classical and neural hierarchical topic models. hLDA (Blei et al., 2010) uses a nested Chinese Restaurant Process to discover tree-structured topic hierarchies but suffers from slow MCMC inference and produces word lists rather than readable labels. BERTopic (Grootendorst, 2022) builds hierarchies through agglomerative merging of HDBSCAN clusters, achieving strong separation but without backward-pass structural optimization. Among neural approaches, SawETM (Duan et al., 2021) uses factorial topic embeddings with a gamma belief network, HyperMiner (Xu et al., 2022) embeds topics in hyperbolic space to capture hierarchical relationships, TSNTM (Isonuma et al., 2020) learns tree structure via a variational autoencoder, and TraCo (Wu et al., 2024) uses optimal transport to align hierarchical topic distributions. All produce hierarchies in a single forward pass without a backward pass to refine document-cluster assignments. **In contrast, LLMTopicTree explicitly introduces a backward pass (Split-Classify-Merge) that corrects structural defects left behind by the forward pass: impure clusters, redundant siblings, and misclassified boundary documents.**

LLM-based topic modeling. TopicGPT (Pham et al., 2024) uses GPT-4 to generate and refine seed topics but operates on flat topic sets without hierarchy. HERCULES (Petneházi and Aradi, 2025) applies recursive clustering with LLM-generated summaries to build hierarchies, and HICode (Zhong et al., 2025) uses LLMs for hierarchical inductive coding, but both construct their trees in a single forward pass; we compare directly against HERCULES in our experiments. LITA (Chang et al., 2024) is the closest prior work: it iteratively reassigns ambiguous documents using LLM feedback, but operates only on flat topics without hierarchical structure. Janssens et al. (2025) augment BERTopic with LLM-based topic merging, again limited to flat representations. The central gap is that LLM-based methods either produce flat topics with iterative refinement or hierarchies without refinement—none perform both. LLMTopicTree fills this gap by using LLMs for both hierarchical construction and backward-pass structural optimization.

Taxonomy refinement. TaxMorph (Golde et al., 2026) uses LLMs to rename and merge taxonomy labels; LORex (Mishra et al., 2025) performs lineage-oriented reasoning for taxonomy expansion; TaxoAdapt (Kargupta et al., 2025) dynamically adapts taxonomy width and depth to evolving corpora. These methods refine the schema—category names, parent-child relationships, and taxonomy breadth—but do not optimize *which documents belong to which clusters*. SCM operates at the content level: it splits impure clusters, moves misplaced documents to better-fitting siblings, and merges confirmed duplicates, directly optimizing document-cluster assignments rather than labels alone.

Hierarchy evaluation. Evaluating hierarchical topic models lacks standardized metrics. Prior work has proposed subtree-purity measures (Kobren et al., 2017) and parent-child structural metrics (Wu et al., 2024), but no prior work provides

a comprehensive evaluation framework spanning both flat and hierarchical quality. We adopt Hierarchical F1 (Kiritchenko et al., 2005; Silla and Freitas, 2011) from hierarchical classification, and propose Hierarchical NMI and Per-Parent Purity to probe content-correction quality at the cluster level alongside standard flat metrics.

3 Method

LLMTopicTree constructs hierarchical topic trees by combining unsupervised clustering with LLM-guided labeling and structural refinement. The system operates in two phases: a *forward pass* that builds an initial hierarchy via recursive clustering, and *Split-Classify-Merge* (SCM), a bottom-up refinement pass that improves cluster purity and eliminates redundancy.

3.1 Pipeline Overview

Given a document corpus $\mathcal{D} = \{d_1, \ldots, d_n\}$, LLMTopicTree first encodes all documents into dense vectors $\mathbf{e}_i = \phi(d_i)$ using a sentence transformer (all-MiniLM-L6-v2, 384 dimensions). The forward pass then performs top-down divisive hierarchical clustering: it recursively partitions documents level by level using k-means++ at each parent node, labels and merges the resulting clusters via the LLM, and validates them, producing an initial tree $\mathcal{T}^{(0)}$. The SCM backward pass refines $\mathcal{T}^{(0)}$ bottom-up, splitting impure clusters, reclassifying borderline documents, and merging redundant siblings to yield the final tree $\mathcal{T}^{(*)}$ (Figure 1). An optional OT-Classify variant (§3.4) replaces LLM-based reclassification with Sinkhorn optimal transport, eliminating all LLM calls in the classify phase.

3.2 Forward Pass

Starting from the root, the forward pass constructs each level $l = 1, \ldots, L$ as follows:

Cluster. Each parent node's documents are partitioned into k children via K-means++ on sentence embeddings.

Label. Clusters are labeled by the LLM in batches of 5 per call. For each batch, the LLM receives sampled documents (up to 30 per cluster) and returns descriptive topic labels. Internal nodes are labeled from their child topics plus representative documents.

Merge. A two-stage merge reduces redundancy: (a) *duplicate removal* eliminates cluster pairs with >90% cosine similarity between centroids, and (b) *high-level abstraction* groups clusters with >60% similarity under a new broader parent, with the LLM generating the abstracted label.

Validate. The LLM evaluates a random sample of each cluster's documents for topic membership, producing a homogeneity score $\text{hom}(c) \in [0, 1]$. Clusters below the acceptance threshold $\tau = 0.70$ are candidates for refinement in the SCM pass.

All labeling and validation calls are parallelized across clusters within each level. Full pseudocode is in Appendix A.

3.3 Split-Classify-Merge (SCM)

SCM is the core contribution of this work. It operates bottom-up from $L_{\max}$ to L_1, applying three phases at each level within a multi-round convergence loop ($R = 3$ rounds, early stopping when no changes occur).

Phase 1: Accuracy-Guided Split. For each cluster c whose homogeneity $\text{hom}(c) < \tau$, we sub-cluster its documents into $k_s = 3$ groups using K-means++ on sentence embeddings, then label and validate each sub-cluster via the LLM. A monotonicity constraint ensures quality: the split is accepted only if the mean homogeneity of sub-clusters exceeds the original by at least $\delta_{\text{acc}} = 0.05$. This prevents fragmentation by rejecting splits that do not materially improve purity.

Phase 2: Outlier-Based Classify. For each cluster, we select the N documents *farthest from the centroid* in embedding space—targeting boundary documents most likely to be misclassified. The LLM classifies each document into the best-fitting sibling topic, where the candidate menu is restricted to the document's current parent's children only—i.e., reassignment is local within a subtree, not global across the level. This local-scope choice preserves the higher-level structure decided in earlier rounds and prevents documents from "jumping" across distant branches of the tree. Documents assigned to a different sibling are moved, with a 15% cap per cluster to prevent catastrophic reassignment.

This phase exploits a key asymmetry: while backward-pass remapping asks "should this *cluster* move to a different parent?" (a structural judgment correlated with near-identical embeddings), SCM

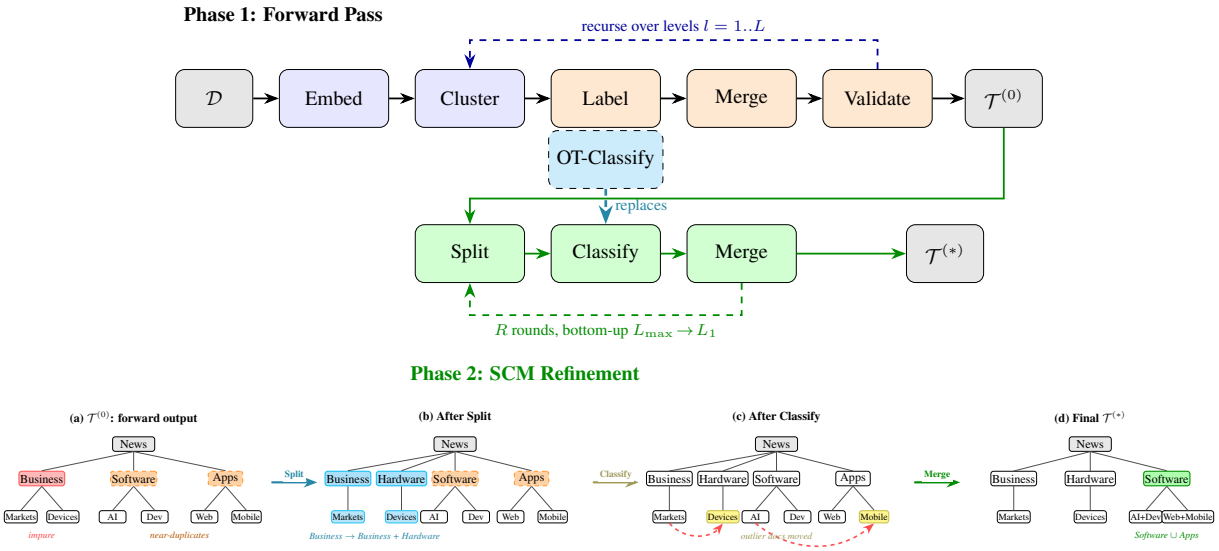

Figure 1: **Top:** LLMTopicTree pipeline. The *forward pass* (blue) recursively produces $\mathcal{T}^{(0)}$ via cluster–label–merge–validate steps at each level. *SCM refinement* (green) then operates bottom-up: **Split** impure clusters, **Classify** outlier documents to siblings, **Merge** confirmed duplicates. The dashed cyan node indicates **OT-Classify**, which replaces LLM classification with Sinkhorn optimal transport. SCM iterates for R rounds with early stopping. **Bottom:** Worked example showing SCM's three phases acting on an initial tree. Red = impure; orange = redundant sibling duplicates; cyan = newly split; yellow = reassigned via classify; green = result of merge.

instead asks "which *topic* does this document belong to?"—a classification task where LLMs excel.

Phase 3: LLM-Confirmed Merge. Sibling cluster pairs whose centroid cosine similarity exceeds 0.95 are presented to the LLM, which judges whether they represent the same topic. On confirmation, the smaller cluster is absorbed into the larger and the merged cluster is relabeled. The high pre-filter threshold ensures only near-duplicate pairs reach the LLM, reducing unnecessary calls.

Level Capping. We apply the full SCM loop (split + classify + merge) to levels L_1 and L_2 only. For L_3 and deeper, only the merge phase is applied. This design reflects the empirical finding that splitting at deep levels increases LLM call count by 3.6× without improving NMI, because fine-grained clusters already have high purity (§5).

Algorithm 1 provides pseudocode for the SCM procedure.

3.4 OT-Classify: Sinkhorn Transport for Document Reassignment

The classify phase of SCM accounts for the majority of LLM calls during refinement. We propose *OT-Classify*, which replaces LLM-based classification with Sinkhorn optimal transport (Cuturi, 2013), reducing classify-phase LLM calls to zero.

Cost Matrix. Given m candidate documents and n sibling clusters with centroids $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_n$, we define the transport cost matrix:

$$C_{ij} = 1 - \cos(\mathbf{e}_i, \boldsymbol{\mu}_j) \tag{1}$$

where $\mathbf{e}_i$ is the embedding of document i and $\cos(\cdot, \cdot)$ denotes cosine similarity.

Marginal Constraints. Source marginals are uniform: $\mathbf{a} = \frac{1}{m}\mathbf{1}_m$ (each document is equally likely to be reassigned). Target marginals are proportional to current cluster sizes with a 15% capacity buffer:

$$b_j = \frac{|\mathcal{D}_j| \cdot 1.15}{\sum_{j'} |\mathcal{D}_{j'}| \cdot 1.15} \tag{2}$$

This prevents oversaturating small clusters while allowing moderate growth.

Log-Domain Sinkhorn. We solve for the optimal transport plan $P^* = \mathrm{diag}(\mathbf{u})\, K\, \mathrm{diag}(\mathbf{v})$ where $K_{ij} = \exp(-C_{ij}/\varepsilon)$, using log-domain iterations for numerical stability:

$$\mathbf{u}^{(t+1)} = \log \mathbf{a} - \mathrm{LSE}_j\left(\mathbf{v}^{(t)} - C/\varepsilon\right) \tag{3}$$

$$\mathbf{v}^{(t+1)} = \log \mathbf{b} - \mathrm{LSE}_i\left(\mathbf{u}^{(t+1)} - C/\varepsilon\right) \tag{4}$$

where LSE denotes `logsumexp` and $\varepsilon = 0.05$ controls entropic regularization. Lower ε yields sharper assignments at the cost of slower convergence.

Algorithm 1 Split-Classify-Merge (SCM)

Require: $\mathcal{T}^{(0)}$, $\mathcal{M}$, τ, δ_a, k_s, R, L_c
Ensure: Refined tree $\mathcal{T}^{(*)}$

```
for r = 1, ..., R do
    chg ← FALSE
    for l = L_max down to 1 do
        for each parent v at level l−1 do
            if l ≤ L_c then
                for all c ∈ ch(v) with hom(c) < τ do
                    // Split: sub-cluster and validate
                    {c_i} ← KMEANS++(D_c, k_s)
                    Label c_i; compute hom(c_i)
                    if hom(c_i) > hom(c) + δ_a then
                        Replace c; chg ← TRUE
                    end if
                end for
                for all c ∈ ch(v) do
                    // Classify: reassign outliers
                    S ← N farthest docs from μ_c
                    Reassign S via M or OT (cap 15%)
                end for
            end if
            for all (c_i, c_j) with cos(μ_i, μ_j)>.95 do
                // Merge: confirm and consolidate
                if LLMCONFIRM(c_i, c_j) then
                    Absorb smaller; chg ← TRUE
                end if
            end for
        end for
    end for
    if ¬ chg then break
    end if
end for
return T^(*)
```

Algorithm 2 OT-Classify: Sinkhorn Document Reassignment

Require: Docs $S=\{d_1, \ldots, d_m\}$, clusters $\{c_1, \ldots, c_n\}$, ε, δ
Ensure: Reassignment map $\pi : S \to \{c_1, \ldots, c_n\}$

```
C_ij ← 1 − cos(e_i, μ_j)                    // cost matrix
a ← (1/m) 1_m;  b_j ← |D_j|·1.15 / Σ_j' |D_j'|·1.15
u ← 0;  v ← 0
for t = 1, ..., T do                        // log-domain Sinkhorn
    u ← log a − LSE_j(v − C/ε)
    v ← log b − LSE_i(u − C/ε)
end for
P*_ij ← exp(u_i + v_j − C_ij/ε)
for i = 1, ..., m do
    j* ← arg max_j P*_ij
    if P*_ij* − max_{j≠j*} P*_ij > δ then
        π(d_i) ← c_j*                       // reassign
    else
        π(d_i) ← current(d_i)               // keep
    end if
end for
return π
```

Dataset	N	K	Domain
20 Newsgroups	5,000	20	News / discussion
AG News	5,000	4	News
BBC News	2,225	5	News
DBpedia	5,000	14	Ontology
Tweet Topic	4,400	6	Social media
StackExchange	5,000	20	Q&A

Table 1: Evaluation datasets. N = number of documents, K = number of ground-truth classes.

Margin-Thresholded Assignment. From the transport plan P^*, document i is reassigned to cluster $j^* = \arg\max_j P^*_{ij}$ only if the probability margin exceeds threshold δ:

$$P^*_{ij^*} - \max_{j \neq j^*} P^*_{ij} > \delta \tag{5}$$

with $\delta = 0.25$. Documents below this margin remain in their current cluster, preventing marginal reassignments that introduce noise. This threshold acts as a confidence gate: only documents with a clear best-fit destination are moved.

Algorithm 2 summarizes the OT-Classify procedure. It eliminates all classify-phase LLM calls, reducing total API calls by up to 23% (§5), while often producing sharper reassignments than LLM-based classification.

4 Experimental Setup

4.1 Datasets

We evaluate on 6 text classification datasets spanning diverse domains and granularities (Table 1). Ground-truth labels serve as the reference clustering for NMI and ARI computation; they are never exposed to any method during topic tree construction. For 20 Newsgroups we evaluate against the 20-class fine-grained ground truth, matching the granularity used by neural baselines (TraCo, SawETM, HyperMiner).

4.2 Baselines and Ablation Setup

Our methods. We evaluate three configurations: **Forward** (forward pass only, no refinement), **SCM-LLM** (full SCM with LLM-based classify), and **SCM-OT** (SCM with Sinkhorn OT classify; §3.4). The Forward variant ablates the entire SCM refinement pass; the SCM-LLM vs SCM-OT contrast isolates the contribution of OT-based reassignment.

Neural HTM baselines. We compare against three neural hierarchical topic models run via the TopMost toolkit (Wu et al., 2023): **TraCo** (Wu et al., 2024), **SawETM** (Duan et al., 2021), and **HyperMiner** (Xu et al., 2022).

Other baselines. BERTopic-Hier (Grootendorst, 2022): agglomerative merging via `.hierarchical_topics()`. **hLDA** (Blei et al.,

2010): Bayesian nonparametric hierarchy via tomotopy (γ=1.0).

4.3 Configuration

Forward pass. 3-level hierarchy, K=10 clusters per level, `all-MiniLM-L6-v2` sentence embeddings (384d) (Reimers and Gurevych, 2019), k-means++ clustering (Arthur and Vassilvitskii, 2007).

SCM refinement. Accuracy threshold τ=0.70, split sub-clusters k=3, merge pre-filter cosine similarity > 0.95, maximum 3 SCM rounds with early stopping, applied to levels 1–2 only.

OT-Classify. Sinkhorn regularization ε=0.05, move margin δ=0.25, capacity buffer 15% (Cuturi, 2013).

LLM. Claude 4.5 Haiku via AWS Bedrock, temperature 0.1, max tokens 128.

4.4 Evaluation Metrics

Flat quality. Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and Purity, all computed at the first hierarchy level (L1) against ground-truth labels.

Hierarchy quality. We report three hierarchy-aware metrics: two we propose and one canonical. **Hierarchical NMI (H-NMI; ours):** mean of standard NMI computed independently at each tree level, H-NMI $= \frac{1}{L}\sum_{\ell=1}^{L} \text{NMI}_\ell$, capturing whether refinement preserves quality across all levels rather than only L1. **Per-Parent Purity (PPP; ours):** for each L1 cluster c, purity$(c) = \max_y |D_c \cap y|/|D_c|$ where y ranges over ground-truth classes; PPP is the macro-average across L1 clusters, PPP $= \frac{1}{|C_1|}\sum_{c \in C_1}$ purity(c). The macro-average controls for K, rewarding methods that produce fewer but purer clusters. **Hierarchical F1 (hF1; Kiritchenko et al. 2005; Silla and Freitas 2011):** for each document, augment the predicted leaf with all tree ancestors, mapped to the majority ground-truth class of each cluster. Let P and T be the augmented predicted and true ancestor sets. Then $hP = \sum_i |P_i \cap T_i| / \sum_i |P_i|$, $hR = \sum_i |P_i \cap T_i| / \sum_i |T_i|$, and hF1 is their harmonic mean. Following Silla and Freitas (Silla and Freitas, 2011), this generalizes flat F1 to tree-structured predictions.

Efficiency. Total LLM API calls and wall-clock time.

5 Results

5.1 Main Results

Table 2 reports NMI across 6 benchmark datasets. We focus on datasets with ground-truth class count $K_{\text{GT}} \leq 20$ to control for the well-studied K-mismatch issue inherent to fixed-K hierarchical methods.

SCM-OT achieves the best average NMI (0.604). SCM-OT wins on all 6 datasets, with the largest gains over Forward on Tweet Topic (+0.147), StackExchange (+0.127), and 20NG (+0.078). The OT-based classify phase provides sharper document reassignment than LLM-based classification, yielding a +0.067 mean NMI advantage over SCM-LLM. Across 18 paired runs (6 datasets $\times$ 3 seeds), the SCM-OT vs. Forward improvement is statistically significant (Wilcoxon signed-rank: W=18, $p = 0.0019$). SCM-OT also outperforms HERCULES (Petneházi and Aradi, 2025), the strongest LLM-HTM baseline, by +0.096 mean NMI on a same-stack comparison (same LLM backbone and embeddings).

SCM-OT dominates all hierarchical baselines. SCM-OT wins 6-0 against every neural HTM baseline (TraCo, SawETM, HyperMiner), against BERTopic-Hier and hLDA, and against HERCULES, an LLM-HTM baseline using the same backbone. The strongest LLM-HTM baseline, HERCULES (0.508 mean NMI), trails SCM-OT by 0.096 – a +19% relative improvement; the strongest non-LLM baseline, BERTopic-Hier (0.430 mean NMI), trails by 0.174 – a +40% relative improvement. TraCo collapses on AG News (NMI = 0.064) and StackExchange (0.001), likely because its transport-cost objective is sensitive to topic granularity mismatches.

5.2 Hierarchy Quality

Table 3 compares hierarchy-aware metrics, aggregated across all 6 datasets and 3 seeds (18 runs per method).

SCM-OT leads on hierarchy-aware metrics. SCM-OT achieves the best mean H-NMI (0.530) and hF1 (0.775); on PPP, HERCULES leads marginally (0.713 vs. 0.729), but the difference is not statistically significant (paired Wilcoxon W=59, $p = 0.26$). This indicates that backward-pass refinement improves quality consistently across all hierarchy levels, not just at L1. The

Method	20NG	AG	BBC	DBp	Tweet	StackEx	Mean
SCM-OT (ours)	**.528**±.10	**.579**±.06	**.796**±.01	**.728**±.03	**.548**±.09	**.443**±.03	**.604**
SCM-LLM (ours)	.433±.03	.575±.03	.738±.02	.694±.07	.453±.01	.333±.07	.537
Forward (ours)	.450±.01	.567±.02	.776±.03	.691±.05	.401±.01	.316±.02	.533
HERCULES	.397±.00	.449±.03	.717±.04	.693±.01	.381±.04	.412±.04	.508
BERTopic-Hier	.381±.00	.378±.00	.623±.01	.613±.00	.298±.00	.284±.00	.430
hLDA	.428±.02	.219±.04	.595±.06	.586±.05	.078±.03	.240±.03	.358
SawETM	.318±.02	.345±.02	.613±.04	.426±.01	.088±.02	.172±.04	.327
TraCo	.354±.01	.064±.08	.575±.03	.434±.31	.127±.02	.001±.00	.259
HyperMiner	.307±.03	.373±.00	.162±.07	.455±.02	.062±.01	.027±.01	.231

Table 2: NMI at L1 across 6 datasets ($K_{\text{GT}} \in [4, 20]$), reported as 3-seed mean±std. Best per column in **bold**. Methods ordered by mean NMI within each block.

Method	H-NMI↑	PPP↑	hF1↑
SCM-OT (ours)	**.530**±.11	.713±.16	**.775**±.13
SCM-LLM (ours)	.506±.12	.656±.18	.773±.13
Forward (ours)	.505±.12	.609±.22	.751±.15
HERCULES	.496±.13	**.729**±.14	.752±.14
BERTopic-Hier	.430±.14	.649±.18	.665±.21
hLDA	.282±.16	.537±.16	.592±.16
SawETM	.378±.18	.464±.18	.570±.20
HyperMiner	.248±.17	.402±.20	.481±.19
TraCo	.288±.22	.460±.24	.455±.27

Table 3: Hierarchy quality metrics, mean±std across 6 datasets × 3 seeds (18 runs). **H-NMI** (ours) = NMI averaged across L1, L2, L3. **PPP** (ours) = macro-average L1 majority-class purity. **hF1** (Kiritchenko et al., 2005; Silla and Freitas, 2011) = hierarchical F1 with ancestor augmentation. Best per column in **bold**.

Variant	S	C	M	NMI	ΔNMI
Forward (baseline)				.381±.05	—
Split only	✓			.459±.11	+.078
Classify only		✓		.411±.01	+.030
Merge only			✓	.399±.03	+.018
Split+Classify	✓	✓		.381±.06	+.000
Split+Merge	✓		✓	.477±.06	+.096
Classify+Merge		✓	✓	.388±.02	+.007
Full SCM	✓	✓	✓	**.546**±.18	**+.165**

Table 4: SCM ablation on Tweet Topic (K_{GT}=6, N=4,400). 3-seed mean±std NMI; ΔNMI is the gain over forward-only. S = Split, C = Classify, M = Merge.

PPP gap to BERTopic-Hier (+0.064) is especially notable: although BERTopic-Hier produces 4–7× more L1 clusters (Appendix B), PPP's macro-average normalization correctly demotes this brute over-segmentation; we discuss the cluster-count tradeoff in §6.

5.3 SCM Ablation

Table 4 shows the contribution of each SCM-OT component on Tweet Topic (K_{GT}=6), the dataset where SCM's +0.147 headline gain (Table 2) provides the cleanest signal for component-level analysis.

All three phases contribute; Split dominates singletons. Every component yields a positive individual gain, with Split largest (+0.078). Split-containing pairs dominate (Split+Merge +0.096 vs. Classify+Merge +0.007), and Full SCM (+0.165) is strongest, indicating the three phases address complementary failure modes – impure parents (Split), misplaced boundary documents (Classify), redundant siblings (Merge) – rather than substitutes.

5.4 Efficiency

All neural baselines (TraCo, SawETM, HyperMiner) require zero API calls but are purely statistical. Among our methods, SCM-OT uses fewer LLM API calls than SCM-LLM on most datasets by replacing the classify phase with Sinkhorn OT (e.g., 692 vs. 901 calls on 20NG, a 23% reduction). SCM-OT wall time is approximately 1.5–2× that of forward-only, an acceptable cost given the consistent NMI gains.

6 Discussion

Top-down refinement vs. bottom-up agglomeration. SCM-OT's top-down divisive construction with content-level refinement leads decisively on flat clustering (mean NMI 0.604 vs. 0.430 for BERTopic-Hier, a +40% relative improvement) and on the multi-level hierarchy metrics (H-NMI 0.530, hF1 0.775). Bottom-up agglomerative methods like BERTopic-Hier benefit from merging the two most similar clusters at each step, which preserves local subtree tightness but does not directly optimize document-cluster fit. SCM-OT instead targets con-

tent quality: Split refines impure parents, Classify reassigns boundary documents to better-fitting siblings, and Merge consolidates confirmed duplicates, directly improving the assignment of documents to clusters.

Tradeoff: cluster count, Purity, and PPP. Standard Purity is monotone in cluster count: any method that produces smaller, narrower clusters tends to win it. The three methods we compare differ sharply on this axis: BERTopic-Hier produces 27–42 L1 clusters per dataset, HERCULES uses a *fixed* 10 clusters regardless of dataset, and SCM-OT adapts to 4–7 via forward-pass duplicate-merge and backward-pass consolidation. BERTopic-Hier exploits over-segmentation directly, winning Purity on 20NG and DBpedia despite far worse NMI; PPP's macro-average defuses this asymmetry, and SCM-OT outperforms BERTopic-Hier on PPP by $+0.064$ overall. HERCULES, with its fixed $K{=}10$, leads SCM-OT on PPP in aggregate (0.729 vs. 0.713), winning on 4 of 6 datasets (20NG, AG News, DBpedia, StackExchange); SCM-OT wins on BBC and Tweet, where its adaptive $K{=}4$–7 lands closest to K_{GT}. The 0.016 aggregate gap is not statistically significant (paired Wilcoxon $W{=}59$, $p = 0.26$, $n = 18$) – fixed-$K{=}10$ and adaptive-K trade per-dataset advantages without a clear methodological winner. SCM-OT's adaptive cluster count is itself a contribution: forward duplicate-merge plus backward Merge converge to a dataset-appropriate K without manual tuning, while still winning all multi-level metrics (NMI, H-NMI, hF1).

When SCM helps and hurts. SCM produces the largest gains when the forward pass produces impure clusters amenable to refinement: Tweet Topic ($+0.147$ NMI over Forward) and StackExchange ($+0.127$) both feature semantically overlapping categories that benefit from boundary document reclassification. SCM-OT outperforms SCM-LLM on all 6 datasets in mean NMI, indicating that Sinkhorn OT is not merely a cheaper substitute but often produces *sharper* reassignments than LLM-based classification, likely because the transport plan enforces global capacity constraints that prevent cluster collapse. The smallest gain over Forward occurs on AG News ($+0.012$), where the forward pass already produces strong clusters (NMI 0.567 with $K_{\text{GT}} = 4$); refinement yields diminishing returns when the initial hierarchy is already nearly optimal.

7 Conclusion

We presented LLMTopicTree, a framework for hierarchical topic discovery that decouples topic generation from structural optimization. The core contribution is *Split-Classify-Merge* (SCM), a bottom-up refinement pass that splits impure clusters, reclassifies boundary documents, and merges confirmed duplicates—the first automated backward-pass optimization of document-cluster assignments within a topic hierarchy. *OT-Classify* further replaces LLM-based classification with Sinkhorn optimal transport, eliminating all classify-phase LLM calls and reducing total API costs by up to 23%.

SCM-OT achieves an average NMI of 0.604 across 6 datasets (3 seeds each), outperforming all 6 baselines including TraCo (0.259), SawETM (0.327), BERTopic-Hier (0.430), and HERCULES (0.508), the closest LLM-HTM baseline. SCM-OT also leads on the hierarchy-aware metrics we report – Hierarchical NMI, Per-Parent Purity, and Hierarchical F1 – demonstrating that backward-pass refinement improves quality consistently across all hierarchy levels.

Future directions include adaptive cluster counts to handle high-K datasets, larger embedding models that may improve OT-Classify performance, human evaluation of generated topic labels, and scaling to corpora exceeding 100K documents.

Limitations

We discuss several limitations of our work that constrain its current applicability and that motivate future research.

Fixed branching factor. LLMTopicTree uses a fixed $K{=}10$ clusters per level, which constrains applicability to corpora whose ground-truth granularity does not match this branching factor. Preliminary experiments on Banking77 ($K_{\text{GT}}{=}77$, severe under-segmentation) and Yahoo Answers ($K_{\text{GT}}{=}10$, semantically ambiguous Q&A categories) revealed that our method under-performs when the ground-truth class count diverges substantially from K. Adaptive cluster-count selection per level (e.g., via the BIC, gap statistic, or LLM-guided silhouette assessment) is an important direction for future work.

Run-to-run variance. All main-table results are reported as 3-seed mean$\pm$std (18 runs per method). Observed standard deviations span $\pm$0.01–0.10

NMI across (method, dataset) cells, with the largest variance on 20NG (SCM-OT ± 0.10) and Tweet (SCM-OT ± 0.09); SCM-OT vs. Forward is significant by Wilcoxon signed-rank ($n = 18$, $W = 18$, $p = 0.0019$). LLM non-determinism (Bedrock does not expose seed parameters) is the dominant source of variance for our methods; we mitigate this with 3 independent seeds per configuration. A larger seed budget (e.g., 5–10) would tighten the confidence intervals further.

LLM cost and dependence. Each run requires 400–900 LLM API calls; OT-Classify reduces this by up to 23% but does not eliminate it. Using a proprietary LLM (Claude 4.5 Haiku via AWS Bedrock) limits reproducibility for groups without API access; a future direction is exploring whether smaller open-weight models (e.g., Llama 3.1-8B-Instruct) suffice for the labeling and validation phases. Prompt templates and per-stage token accounting are reported in Appendix D to support replication.

Embedding choice. All methods use `all-MiniLM-L6-v2` sentence embeddings (384 dimensions). Stronger encoders (e.g., E5-large, NV-Embed, Matryoshka representations) may improve the forward pass and shift the relative effectiveness of OT-Classify versus LLM-based classification, since OT relies on the geometry of the embedding space. We have not tuned encoder choice per dataset.

Single-level flat evaluation. Our primary clustering evaluation operates at L1. Per-level NMI results are reported in the appendix, but a comprehensive multi-level evaluation framework—e.g., hierarchical NMI, tree edit distance, or CEDS/HSR-style structure metrics—is left to future work.

Domain coverage. We evaluate on text classification benchmarks spanning news, social media, and Q&A; we do not evaluate on long-form documents, code corpora, or non-English text. The cosine-similarity merge prefilter ($\cos > 0.95$) and outlier-detection thresholds may need recalibration for substantially different domains.

Risks. Our framework can amplify any biases present in the LLM's labeling and validation decisions, particularly for sensitive domains. We did not conduct a human evaluation of the generated topic labels for offensive or biased content; production deployments should include such review.

Ethics Statement

This work uses six publicly available text-classification benchmarks (20 Newsgroups, AG News, BBC News, DBpedia, Tweet Topic, StackExchange) under their respective open licenses, and one open-source hierarchical topic-modeling baseline (HERCULES via the official `pyhercules` package). No human-subjects research was conducted; ground-truth labels were used only for automated evaluation, never exposed to any method during tree construction. The LLM backbone (Claude 4.5 Haiku, accessed via AWS Bedrock) is used for topic labeling, validation, and merge confirmation; no personally identifying information was processed beyond what is already public in the source datasets. We acknowledge that LLM-generated topic labels can encode biases present in the underlying language model; mitigations are discussed in the Limitations section. Total compute: approximately 234 multi-seed runs ($\sim$12 CPU-hours on commodity AWS instances, $\sim$\$30 in Bedrock API costs).

References

David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035.

David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. 2010. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2):1–30. Originally presented at NIPS 2003.

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

Chia-Hsuan Chang, Jui-Tse Tsai, Yi-Hang Tsai, and San-Yih Hwang. 2024. LITA: An efficient LLM-assisted iterative topic augmentation framework. *arXiv preprint arXiv:2412.12459*. Accepted at PAKDD 2025.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*.

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.

Zhibin Duan, Dongsheng Wang, Bo Chen, Chaojie Wang, Wenchao Chen, Yewen Li, Jie Ren, and

Mingyuan Zhou. 2021. Sawtooth factorial topic embeddings guided gamma belief network. In *Proceedings of the 38th International Conference on Machine Learning*.

Jonas Golde, Nicolaas Jedema, RaviKiran Krishnan, and Phong Le. 2026. TaxMorph: Hierarchical text classification with LLM-refined taxonomies. *arXiv preprint arXiv:2601.18375*.

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

Masaru Isonuma, Junichiro Mori, Danushka Bollegala, and Ichiro Sakata. 2020. Tree-structured neural topic model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Wannes Janssens, Matthias Bogaert, and Dirk Van den Poel. 2025. LLM-assisted topic reduction for BERTopic on social media data. *arXiv preprint arXiv:2509.19365*.

Priyanka Kargupta, Nan Zhang, Yunyi Zhang, Rui Zhang, Prasenjit Mitra, and Jiawei Han. 2025. TaxoAdapt: Aligning LLM-based multidimensional taxonomy construction to evolving research corpora. *arXiv preprint arXiv:2506.10737*.

Svetlana Kiritchenko, Stan Matwin, and A. Fazel Famili. 2005. Functional annotation of genes using hierarchical text categorization. In *Proceedings of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology (held at ISMB)*. Introduces hierarchical precision/recall/F1 used widely in subsequent hierarchical text classification work.

Ari Kobren, Nicholas Monath, Akshay Krishnamurthy, and Andrew McCallum. 2017. An online hierarchical algorithm for extreme clustering. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of EACL*, pages 530–539.

Sahil Mishra, Kumar Arjun, and Tanmoy Chakraborty. 2025. Rank, chunk and expand: Lineage-oriented reasoning for taxonomy expansion. In *Findings of the Association for Computational Linguistics: ACL 2025*. ArXiv:2505.13282.

Fionn Murtagh and Pedro Contreras. 2012. Algorithms for hierarchical clustering: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97.

Gábor Petneházi and Bernadett Aradi. 2025. HERCULES: Hierarchical embedding-based recursive clustering using LLMs for efficient summarization. *arXiv preprint arXiv:2506.19992*.

Chau Minh Pham, Alexander Hoyle, Simeng Sun, Philip Resnik, and Mohit Iyyer. 2024. TopicGPT: A prompt-based topic modeling framework. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Carlos N. Silla and Alex A. Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1–2):31–72.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *Proceedings of the 5th International Conference on Learning Representations*.

Dominik Stammbach, Vilém Zouhar, Alexander Hoyle, Mrinmaya Sachan, and Elliott Ash. 2023. Revisiting automated topic model evaluation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Xiaobao Wu, Fengjun Pan, and Anh Tuan Luu. 2023. Towards the TopMost: A topic modeling system toolkit. *arXiv preprint arXiv:2309.06908*.

Xiaobao Wu, Fengjun Pan, Thong Nguyen, Yichao Feng, Chaoqun Liu, Cong-Duy Nguyen, and Anh Tuan Luu. 2024. On the affinity, rationality, and diversity of hierarchical topic modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*. ArXiv:2401.14113.

Yishi Xu, Dongsheng Wang, Bo Chen, Ruiying Lu, Zhibin Duan, and Mingyuan Zhou. 2022. HyperMiner: Topic taxonomy mining with hyperbolic embedding. In *Advances in Neural Information Processing Systems*.

Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 103–114.

Mian Zhong, Pristina Wang, and Anjalie Field. 2025. HICode: Hierarchical inductive coding with large language models. *arXiv preprint arXiv:2509.17946*.

A Forward Pass Pseudocode

B Full Results

Tables 5–10 report all metrics across the 6 evaluation datasets, as 3-seed means. NMI, ARI, Purity are computed at L1 vs. ground truth; H-NMI averages NMI across L1, L2, L3; PPP is the macro-average L1 majority-class purity; hF1 is hierarchical F1 (Kiritchenko et al., 2005; Silla and Freitas, 2011). Best per column in **bold**.

Algorithm 3 Forward Pass: Hierarchical Discovery

Require: $\mathcal{D}$, levels L, clusters k, min size $\tau_{\min}$
Ensure: Initial tree $\mathcal{T}^{(0)}$

```
V_0 ← {root}; D_root ← D
for l = 1 to L do
    for each v ∈ V_{l−1} in parallel do
        if |D_v| < τ_min then mark terminal; continue
        end if
        {c_i} ← KMeans++(D_v, k, φ)
        Label clusters in batches of 5 via M
        ch(v) ← ForwardMerge({c_i}, M)
    end for
    V_l ← ⋃_{v∈V_{l−1}} ch(v)
end for
RelabelInternal(T^(0), M)
Validate(T^(0), M)
return T^(0)
```

On cluster count and Purity. BERTopic-Hier produces 4–7× more L1 clusters than SCM-OT (e.g., 27–42 vs. 6 on most datasets), inflating standard Purity at the cost of NMI/ARI. Our proposed Per-Parent Purity (PPP) macro-averages over clusters and is robust to this asymmetry.

C SCM Convergence

SCM converges rapidly in practice. Across all evaluation datasets, the first round accounts for >90% of all document movements and cluster merges; the second round yields <0.005 additional NMI improvement on average and the third round triggers early stopping (zero changes) in most cases. This motivates our default of R=3 rounds with early stopping.

The level cap at L_2 (`SCM_MAX_LEVEL`=2) is critical for preventing pathological behavior at deeper levels. Without the cap, applying full SCM at L_3 on AG News increases API cost 3.6× (521 → 2,602 calls) and the L3 leaf count from 100 to over 500, while NMI remains flat. Splitting fine-grained clusters yields fragmentation without quality gains. The splits at L_3 fragment already-pure clusters into near-duplicate sub-topics, creating redundant siblings faster than the merge phase can consolidate them. Restricting L_3+ to merge-only preserves the benefits of deep-level deduplication without the fragmentation cost.

D Configuration Details

Table 11 lists all hyperparameters used in our experiments.

Method	NMI↑	ARI↑	Purity↑	H-NMI↑	PPP↑	hF1↑
SCM-OT (ours)	**.528**	**.302**	.367	**.514**	.495	.565
Forward (ours)	.450	.195	.293	.476	.415	.542
SCM-LLM (ours)	.433	.203	.314	.465	.400	**.566**
hLDA	.428	.209	.337	.417	.460	.525
HERCULES	.397	.140	.269	.464	**.501**	.522
BERTopic-Hier	.381	.165	**.383**	.378	.427	.413
TraCo	.354	.172	.276	.399	.318	.459
SawETM	.318	.152	.245	.411	.308	.422
HyperMiner	.307	.149	.230	.414	.229	.419

Table 5: 20 Newsgroups (K=20, N=5,000). 3-seed mean.

Method	NMI↑	ARI↑	Purity↑	H-NMI↑	PPP↑	hF1↑
SCM-OT (ours)	**.579**	.602	**.827**	**.437**	.801	.850
SCM-LLM (ours)	.575	**.660**	.826	.435	.764	.855
Forward (ours)	.567	.573	.822	.426	**.806**	**.859**
HERCULES	.449	.343	.726	.382	.802	.832
BERTopic-Hier	.378	.191	.810	.382	.777	.820
HyperMiner	.373	.373	.673	.370	.659	.766
SawETM	.345	.333	.614	.362	.587	.753
hLDA	.219	.172	.610	.126	.585	.621
TraCo	.064	.029	.309	.094	.581	.361

Table 6: AG News (K=4, N=5,000). 3-seed mean.

Method	NMI↑	ARI↑	Purity↑	H-NMI↑	PPP↑	hF1↑
SCM-OT (ours)	**.796**	**.778**	**.951**	.617	**.959**	**.954**
Forward (ours)	.776	.745	.908	.604	.925	.944
SCM-LLM (ours)	.738	.711	.887	.597	.925	.942
HERCULES	.717	.671	.891	.590	.922	.933
BERTopic-Hier	.623	.446	.913	.624	.880	.919
SawETM	.613	.566	.740	**.656**	.767	.843
hLDA	.595	.487	.806	.435	.792	.834
TraCo	.575	.517	.703	.562	.707	.875
HyperMiner	.162	.096	.393	.134	.479	.467

Table 7: BBC News (K=5, N=2,225). 3-seed mean.

Method	NMI↑	ARI↑	Purity↑	H-NMI↑	PPP↑	hF1↑
SCM-OT (ours)	**.728**	**.510**	.537	**.718**	.633	.834
SCM-LLM (ours)	.694	.495	.527	.705	.652	**.835**
HERCULES	.693	.463	.556	.689	**.691**	.812
Forward (ours)	.691	.482	.490	.706	.551	.800
BERTopic-Hier	.613	.454	**.705**	.610	.679	.726
hLDA	.586	.440	.562	.473	.585	.668
HyperMiner	.455	.298	.386	.450	.411	.541
TraCo	.434	.333	.401	.429	.395	.434
SawETM	.426	.259	.343	.519	.411	.593

Table 8: DBpedia (K=14, N=5,000). 3-seed mean.

Method	NMI↑	ARI↑	Purity↑	H-NMI↑	PPP↑	hF1↑
SCM-OT (ours)	**.548**	**.541**	**.844**	**.384**	**.800**	.784
SCM-LLM (ours)	.453	.502	.751	.353	.666	**.791**
Forward (ours)	.401	.418	.680	.335	.648	.787
HERCULES	.381	.301	.701	.323	.796	.783
BERTopic-Hier	.298	.119	.742	.299	.751	.761
TraCo	.127	.098	.540	.100	.583	.547
SawETM	.088	.105	.489	.114	.454	.578
hLDA	.078	.048	.526	.078	.531	.611
HyperMiner	.062	.053	.498	.077	.524	.559

Table 9: Tweet Topic (K=6, N=4,400). 3-seed mean.

Method	NMI↑	ARI↑	Purity↑	H-NMI↑	PPP↑	hF1↑
SCM-OT (ours)	**.443**	**.220**	**.401**	.510	.588	**.664**
HERCULES	.412	.101	.314	**.529**	**.661**	.628
SCM-LLM (ours)	.333	.126	.279	.482	.532	.647
Forward (ours)	.316	.115	.201	.481	.310	.572
BERTopic-Hier	.284	.107	.316	.285	.380	.353
hLDA	.240	.136	.272	.165	.269	.292
SawETM	.172	.073	.153	.204	.257	.231
HyperMiner	.027	.008	.080	.043	.110	.134
TraCo	.001	.000	.050	.002	.172	.056

Table 10: StackExchange (K=20, N=5,000). 3-seed mean.

Component	Parameter	Value
Forward pass	Clusters per level (k)	10
	Max levels (L)	3
	Min cluster size ($\tau_{\min}$)	50
	Sample size	5,000
SCM	Accuracy threshold (τ)	0.70
	Split sub-clusters (k_s)	3
	Acceptance margin (δ_{acc})	0.05
	Merge cosine pre-filter	0.95
	Max rounds (R)	3
	Max level (`SCM_MAX_LEVEL`)	2
OT-Classify	Sinkhorn regularization (ε)	0.05
	Move margin (δ)	0.25
	Capacity buffer	15%
LLM	Model	Claude 4.5 Haiku
	Temperature	0.1
	Max tokens	128
	Max workers	20
Embeddings	Model	all-MiniLM-L6-v2
	Dimensions	384

Table 11: Full hyperparameter configuration. All experiments use identical settings unless otherwise noted.