International Journal of
Computational Linguistics Research

DLINE JOURNALS

Print ISSN: 0976-416X
Online ISSN: 0976-4178

IJCLR 2026: 17 (4)
https://doi.org/10.6025/ijclr/2026/17/4/193-226

Mapping the AI Frontier: Semantic Topic Modeling and BERTopic Parameter Optimization—An Analysis of arXiv Preprints and a Practical Guide to UMAP/HDBSCAN Tuning

Pit Pichappan
Digital Information Research Labs
Chennai. Tamil Nadu India
pichappan@dirf.org

ABSTRACT

Semantic topic modeling has become an essential tool for mapping the rapidly evolving landscape of artificial intelligence research. While BERTopic has emerged as a leading framework due to its integration of transformer based embeddings and density-based clustering, practitioners frequently encounter degenerate clustering solutions such as single mega-clusters or artificial zero outlier states when relying on default library parameters. This study addresses this methodological gap by applying BERTopic to a curated dataset of 5,000 frontier arXiv AI preprints and conducting a systematic 90 configuration grid search to isolate the marginal and interactive effects of UMAP and HDBSCAN parameters. Our results reveal a critical, silent coupling between the two algorithms: UMAP's n_neighbors parameter manufactures the foundational density structure, silently redefining the meaning of HDBSCAN's min_cluster_size. Furthermore, we demonstrate that optimizing for single metrics like Density-Based Cluster Validation (DBCV) is fundamentally flawed, as it monotonically rewards overly tight, degenerate solutions. To resolve this, we propose a robust, multi-objective lexicographic filtering protocol and a practical, data-backed tuning guide. Applied to the arXiv corpus, this optimized framework successfully maps the AI frontier into distinct macro clusters (e.g., LLM/Agent NLP vs. Embodied AI) while providing a reproducible methodology to prevent structural collapse in semantic topic modeling pipelines.

Keywords: BERTopic, Topic Modeling, UMAP, HDBSCAN, Parameter Tuning, Natural Language Processing, Density-Based Clustering, Semantic Embeddings, arXiv, Machine Learning Methodology

Received: 24 February 2026, Revised: 28 May 2026, Accepted: 9 June 2026

Copyright: DLINE

1. Introduction

Natural language processing (NLP) has become an important approach for extracting meaningful insights from large collections of unstructured text, including scientific abstracts and scholarly literature. Topic modeling is a foundational technique in text mining and NLP because it identifies latent themes, recurring patterns, and conceptual structures across document collections. As an unsupervised machine learning

approach, it supports the systematic organization and interpretation of extensive corpora in situations where manual analysis is impractical or infeasible [1-5]. Related work likewise emphasizes the capacity of topic-modeling methods to discover hidden themes and conceptual relationships in large scale text [5, 6, 7].

Recent advances in transformer based language models have expanded topic modeling beyond approaches that rely primarily on word frequency representations. Traditional methods such as Latent Dirichlet Allocation (LDA) use a bag of words representation, while term frequency inverse document frequency (TF-IDF) methods primarily represent lexical importance and do not directly capture contextual relationships or word order [8]. In contrast, contextual language models can encode meaning in relation to the surrounding linguistic context, providing a semantic foundation for grouping documents by conceptual similarity [9, 10].

2. Background

2.1 BERTopic: Architecture and Core Components

Introduced by Grootendorst in 2022, BERTopic is a modular neural topic-modeling framework that uses advances in deep learning, particularly transformer based representations, to identify semantically coherent topics [11]. Transformer models such as Bidirectional Encoder Representations from Transformers (BERT) generate contextualized embeddings by representing words in relation to the surrounding text, considering context from both the left and right [11]. This contextual representation distinguishes BERTopic from approaches that rely only on lexical co-occurrence or bag of words features.

BERTopic can be understood as a pipeline comprising four principal stages: document embedding, dimensionality reduction, clustering, and topic representation [11]. In its commonly used configuration, document embeddings are reduced to a lower dimensional space using Uniform Manifold Approximation and Projection (UMAP) [12] [McInnes L]. The resulting representations are clustered using Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), which identifies dense groups while allowing documents that do not fit a cluster to remain unassigned as noise [13]. Topic representations are then generated using a class-based variant of TF-IDF, helping characterize each cluster with interpretable terms [14].

Because BERTopic is modular rather than a single indivisible algorithm, its output depends on choices made throughout the pipeline. Embedding model selection, text preprocessing, dimensionality reduction, clustering parameters, and term weighting strategies can all affect the resulting topic structure and its interpretability. Consequently, parameter selection and systematic evaluation are important when applying BERTopic to large scholarly collections.

2.2 Related Work and Methodological Considerations

Research examining the BERTopic pipeline has highlighted the importance of matching preprocessing decisions to the embedding model. Borèin reproduces key experiments from the BERTopic framework and reports that the effects of text preprocessing differ by representation method: advanced embedding models such as MPNet may benefit from raw input, whereas simpler models such as GloVe may perform better when the text is preprocessed [15]. This finding suggests that preprocessing should not be treated as a universal step; rather, it should be evaluated in relation to the selected embedding model and the analytical objective.

Work by Inoue proposes a multi objective optimization framework for BERTopic in the context of large scale scholarly literature analysis involving autoregressive (AR) models, including ARIMA, TSAR, and hybrid machine learning approaches [16]. This line of research reinforces the need to evaluate topic modeling configurations across more than one criterion, rather than assuming that a single parameter setting will be optimal for every corpus or purpose.

Drissi discusses methodological challenges that arise when topic modeling is applied to heterogeneous, multilingual, and large scale corpora. These challenges include limitations in interpretability, sensitivity to parameter choices, and the difficulty of representing context consistently across diverse texts. The work describes unified and modular frameworks that combine probabilistic modeling, semantic enrichment,

diagnostic evaluation, and structural organization within a coherent pipeline [17]. These concerns are relevant to the design and evaluation of topic-modeling workflows in complex scholarly datasets.

The wider use of BERTopic demonstrates its applicability beyond academic abstracts. In social media research, it has been used to relate news topics to audience engagement measures [18] and to examine the evolution of online discussions, including turning points and the branching of discussion themes [19]. Reviews of AI methods in social network analysis also describe applications spanning areas such as fake-news detection and public-health analysis [20]. Together, these applications illustrate the flexibility of semantic topic-modeling pipelines across different domains and types of textual data.

2.3 Contextual Embeddings and Topic Representation

Transformer based models have marked a significant development in semantic text representation. BERT [21], Sentence BERT (SBERT) [22], and RoBERTa [23] generate context sensitive representations that adapt to linguistic context, enabling documents to be encoded in ways that better reflect semantic similarity. Models such as BERTopic [11] and Top2Vec [24] use embedding based representations to group semantically related documents and produce topic structures intended to be more cohesive and interpretable. BERTopic combines transformer derived contextual embeddings with class-based TF-IDF weighting to connect semantic clustering with readable, term based topic descriptions [11, 25].

2.4 Dimensionality Reduction, Clustering, and Parameter Tuning

UMAP and HDBSCAN are central components of the commonly used BERTopic workflow. UMAP projects high-dimensional document embeddings into a lower-dimensional space while attempting to preserve meaningful structure in the data [12]. HDBSCAN then identifies clusters according to density and can leave documents outside sufficiently dense regions unclustered [13]. The interaction between these stages makes parameter selection consequential: dimensionality-reduction settings can alter the geometry of the representation space, while clustering settings influence the number, size, and separation of the resulting topics.

For analyses of arXiv preprints, a transparent tuning process should therefore consider the behavior of the complete pipeline rather than optimizing UMAP or HDBSCAN in isolation. Topic coherence and interpretability, the prevalence of unassigned documents, the number and size of clusters, and the stability of topics under alternative configurations are useful diagnostic considerations. These are evaluation dimensions for a practical tuning workflow; the source material provided here does not report specific experimental values or establish a best-performing parameter configuration.

The literature positions BERTopic as a flexible framework for semantic topic discovery in large and diverse text collections. Its principal strength is the integration of contextual document embeddings, dimensionality reduction, density based clustering, and class-based TF-IDF topic representation within a modular pipeline. At the same time, the literature indicates that results are sensitive to preprocessing choices, embedding models, and parameter settings. A careful analysis of arXiv preprints should make these decisions explicit and assess their effects systematically, particularly when tuning UMAP and HDBSCAN. The supplied material establishes this methodological motivation but does not include a detailed dataset description, experimental protocol, quantitative results, or a reference list sufficient to verify the full bibliographic details.

2.5 Research Issues

• Limitations of Traditional Topic Modeling: Traditional methods like LDA and TF-IDF rely on bag-of-words representations and fail to capture contextual relationships or word order, limiting their effectiveness on complex scientific literature.

• Parameter Sensitivity and “Silent Coupling”: BERTopic’s modular pipeline is highly sensitive to parameter choices. A critical, often misunderstood issue is the silent coupling between UMAP’s neighborhood size (n_neighbors) and HDBSCAN’s density thresholds (min_cluster_size). Changing UMAP’s local geometry silently redefines what constitutes a “density region” for HDBSCAN.

• Degenerate Clustering Solutions: Relying on default library parameters frequently leads to structural collapse, resulting in degenerate solutions such as a single “mega-cluster” absorbing >90% of the corpus, or an artificial zero-outlier state that masks the true data distribution.

• Inadequacy of Single-Metric Evaluation: Standard clustering metrics like Density-Based Cluster Validation (DBCV) and topic coherence can be highly misleading. DBCV monotonically rewards overly tight, local geometries (falsely validating 2-topic degenerate solutions), while coherence scores often remain flat despite massive structural changes in the clusters.

2.6 Problem Statement

Despite the widespread adoption of BERTopic for semantic topic discovery, its application to scientific corpora is frequently undermined by a lack of systematic, data backed tuning protocols. Practitioners often rely on default library parameters, which fail to account for the interactive geometry between dimensionality reduction (UMAP) and density-based clustering (HDBSCAN). This leads to degenerate solutions that obscure the true semantic landscape of the data. There is a critical need for a transparent tuning framework that addresses these parameter interactions, establishes robust multi-objective evaluation criteria, and prevents the silent failure modes inherent in the BERTopic pipeline.

2.7 Research Design

The study employs a mixed-methods approach combining empirical data analysis with algorithmic parameter optimization:

• Phase 1: Exploratory Semantic Mapping. Applying a baseline BERTopic pipeline to a large-scale bibliometric dataset to map the frontier of AI research, identifying macro-clusters and validating semantic separations against existing metadata (arXiv categories).

• Phase 2: Algorithmic Interaction Grid. Designing a 90-configuration grid search to systematically isolate and measure the marginal and interactive effects of UMAP and HDBSCAN parameters on a targeted subset of the data.

• Phase 3: Multi-Objective Evaluation Framework. Developing a lexicographic filtering protocol to replace flawed single-metric optimization. This involves applying hard statistical gates, using DBCV only as a tie-breaker, and enforcing manual interpretability and stability checks.

2.8 Methodology

• Contextual Embedding: Generating 768-dimensional, L2-normalized sentence embeddings using the all-mpnet-base-v2 transformer model to capture semantic context.

• Dimensionality Reduction (UMAP): Projecting embeddings into a 5-dimensional space (n_components=5) using cosine distance and min_dist=0.0 (to encourage clumpier embeddings for clustering). The n_neighbors parameter is systematically varied to control the balance between local and global manifold structure.

• Density-Based Clustering (HDBSCAN): Applying HDBSCAN using the Excess of Mass (EOM) cluster selection method. The min_cluster_size and min_samples parameters are varied to test density thresholds and outlier conservativeness.

• Topic Representation: Generating interpretable, term-based topic labels using class-based TF-IDF (c-TF-IDF) with 1-3 grams.

• External Validation: Cross-tabulating discovered semantic topics against external arXiv primary subfield metadata to verify whether the model captures established disciplinary concentrations or reveals cross-subfield relationships.

2.9 Experimental Setup

• Dataset: A curated subset of 387 arXiv abstracts (drawn from a broader 5,000-paper "frontier snapshot" of AI preprints from October 2026) is used for the intensive tuning grid to allow for rapid iteration and diagnostic scoring.

• Grid Search Space: 90 configurations spanning:

o UMAP n_neighbors ε {10, 15, 30, 50, 100}

o HDBSCAN min_cluster_size ε {5, 8, 12, 20, 30}

o HDBSCAN min_samples ε {1, 5, 15}

o HDBSCAN cluster_selection_method ε {eom, leaf}

• Fixed Parameters: UMAP random_state=42; HDBSCAN metric='euclidean', prediction_data=True.

• Metrics Tracked: Topic count, outlier percentage (%), Density-Based Cluster Validation (DBCV), and Normalized Pointwise Mutual Information (NPMI) coherence.

• Diagnostic Logging: Implementation of a run_summary.json to automatically flag degenerate runs (e.g., flagging if the top topic exceeds 60% of the corpus or if outliers hit 0%).

3. Dataset

arXiv AI Research Papers – 5,000 Frontier Studies (Kaggle dataset by Luthfi Muthathohirin) is a curated, publication-grade bibliometric benchmark of exactly 5,000 unique AI-related preprints submitted to arXiv.

3.1 Core Characteristics

• Source & Scope: Papers harvested via the arXiv REST API from the categories cs.AI OR cs.CL OR cs.LG OR cs.CV (with some cross-listings into related areas such as cs.RO).

• Temporal window: 1–9 October 2026 (a short, high-velocity "frontier snapshot").

• Volume: 5,000 unique records (0 duplicates, 0 corrupted entries); 4,913 unique author groups.

• Formats: Clean UTF-8 CSV (arxiv_ai_research_papers.csv, ~12.7 MB) and Snappy-compressed Parquet (arxiv_ai_research_papers.parquet, ~5.3 MB).

• License: CC0 (Public Domain).

• Annotation quality: Multi-dimensional scientific taxonomy produced by an Explain Then Annotate Cognitive Chain-of-Thought (CoT) pipeline, with inter-annotator agreement Cohen's k = 0.8842 and mean calibrated confidence score 0.9493.

4. Results

This section presents the empirical findings derived from the application of the semantic topic modeling pipeline to the curated arXiv dataset. Before detailing the discovered thematic structures and the subsequent parameter tuning grid, it is essential to establish the foundational characteristics of the processed corpus and its baseline statistical representations. The following outlines the core descriptive statistics of the 5,000

frontier AI preprints, including text length distributions, vocabulary sizes for both TF-IDF and LDA representations, and baseline dimensionality reduction metrics. These initial measurements provide a quantitative baseline, verifying data completeness and establishing the structural properties of the text embeddings prior to the deep semantic clustering analysis.

Measure	Value
Papers processed	5000
Unique arXiv IDs	5000
Empty title + abstract	0
Mean raw words per paper	205.882
Median raw words per paper	205
Mean cleaned tokens per paper	126.536
Median cleaned tokens per paper	128
Vocabulary size (TF-IDF)	30000
LDA vocabulary size	12000
LDA topics	10
SVD semantic dimensions	100
SVD explained variance ratio sum	0.10817
Semantic clusters	10

Table 1. Basic dataset description

The table reports the number of processed papers, unique arXiv IDs, missing title/abstract records, average and median text lengths, vocabulary sizes, number of LDA topics, SVD dimensions and explained variance, and the number of semantic clusters.

The reported 5,000 unique IDs and absence of empty title and abstract records suggest that the dataset has passed basic completeness and uniqueness checks. The raw text averages approximately 206 words per paper, while the cleaned text averages approximately 127 tokens.

The 30,000-term TF-IDF vocabulary and 12,000-term LDA vocabulary indicate that the text representations use different vocabulary sizes for different analytical purposes.

The SVD explained variance ratio sum of 0.10817 means that the retained 100 components explain approximately 10.82% of the variance under the reported SVD representation. This value should be interpreted in the context of sparse text data; it does not, by itself, indicate that the semantic representation is inadequate.

4.1 Sentence Embeddings

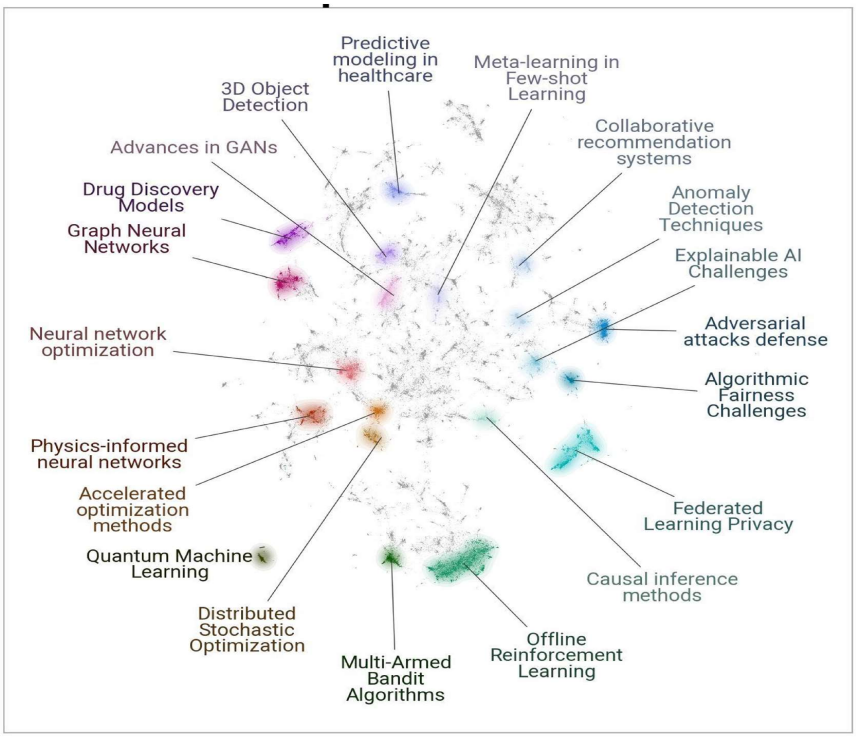

Figure 1. Sentence embeddings

The figure projects 768-dimensional sentence embeddings into a two-dimensional UMAP space. Each point represents an abstract, and different colours indicate topic assignments. The visible spatial separation illustrates how the clustering algorithm groups semantically related research papers.

The document identifies two particularly distinct research groups:

• T0: language models, agents, and NLP.

•T1: action-oriented models, world models, and embodied AI.

The remaining topics represent narrower research themes, including segmentation, multimodal reasoning, attention mechanisms, diffusion operators, and molecular or materials related AI. The separation of topic regions supports the interpretation that the abstracts contain distinguishable semantic groupings.

However, a two dimensional projection is a visualization rather than definitive proof of cluster validity. Apparent separation can be affected by UMAP settings. Cluster quality should therefore also be assessed using the original embeddings, clustering diagnostics, and topic interpretability.

Centroid similarity in the *original* 768-d space (not the reduced space) shows T0 and T1 are near orthogonal research universes, while T3/T5 and T4/T6 sit closer together.

4.2 Topic Sizes

This bar chart displays the number of abstracts assigned to each of the eight discovered topics and the HDBSCAN outlier group, labelled “1.

Topic 0 is the largest cluster, with 135 papers, followed by Topic 1 with 80 papers. The other six clusters are substantially smaller. There are also 63 outlier abstracts that HDBSCAN did not assign to a regular topic.

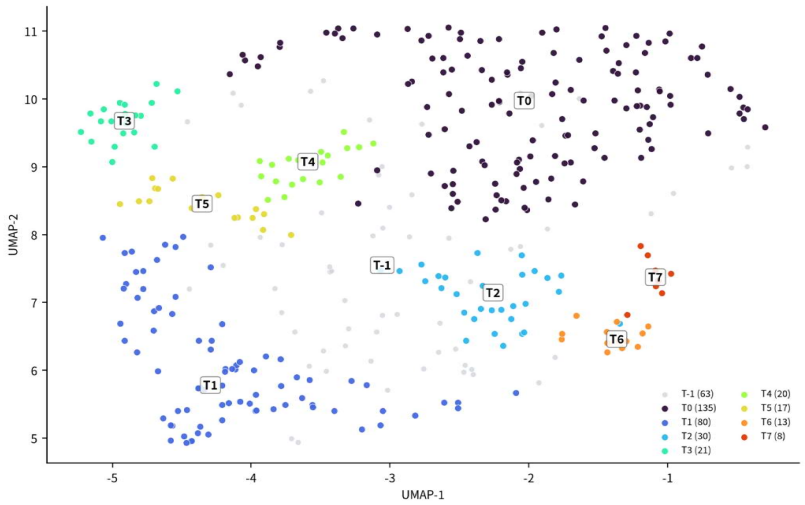

Figure 2. Sentence embeddings

The distribution suggests that the corpus is dominated by two broad research themes, while the remaining papers represent more specialized or less densely connected areas.

The outlier group accounts for 16.3% of the 387 abstracts. These papers should not be interpreted as irrelevant research: they may represent emerging themes, interdisciplinary studies, or abstracts that do not fit the density structure of the identified clusters.

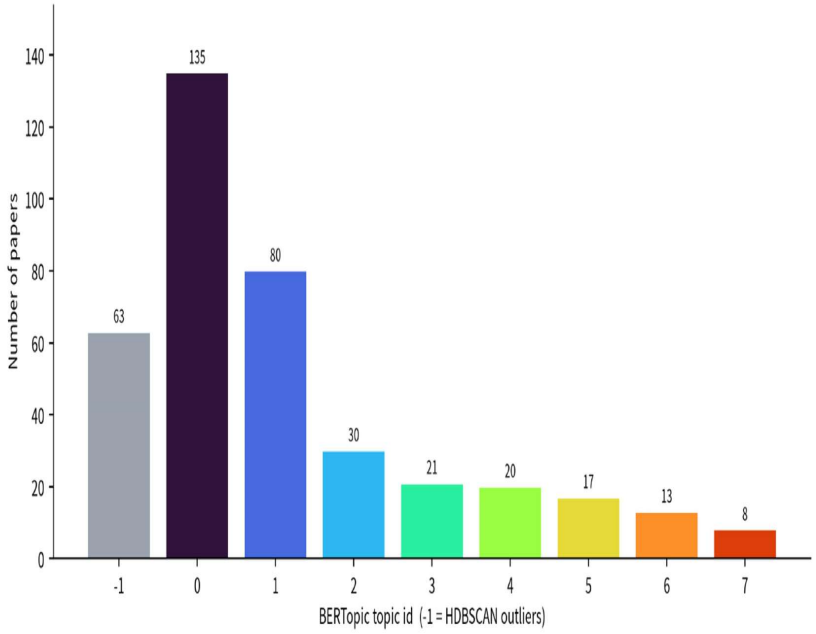

Figure 3. Topic Sizes

Metric	Value
Documents	387 arXiv abstracts
Embedding model	sentence-transformers/all-mpnet-base-v2, 768-d, L2-normalised
UMAP	5 components, n_neighbors=30, min_dist=0.0, cosine, seed 42
HDBSCAN	min_cluster_size=8, min_samples=5, EOM
Topics (excl. outliers)	8
Outliers (topic -1)	63 (16.3 %)
c-TF-IDF	1–3 grams, min_df=2, English stopwords

Table 2. Topic profiles

ID	Papers	Share	Label (c-TF-IDF)	Dominant subfield	Example paper
0	135	34.9 %	language · agents · models	Language & NLP (72.6 %)	2610.40655
1	80	20.7 %	action · world · video	Robotics (51.2 %)	2610.38744
2	30	7.8 %	data · record · sample	Machine Learning (93.3 %)	2610.40654
3	21	5.4 %	segmentation · selfsupervised · image	Computer Vision (85.7 %)	2610.40252
4	20	5.2 %	visual · multimodal · multimodal reasoning	Computer Vision (50.0 %)	2610.40124
5	17	4.4 %	anchor · image · attention	Computer Vision (88.2 %)	2610.40647
6	13	3.4 %	operator · diffusion · sampling	Machine Learning (84.6 %)	2610.40618
7	8	2.1 %	structures · materials · molecular	Machine Learning (87.5 %)	2610.38835
-1	63	16.3 %	neural · estimation · embedding	Machine Learning (50.8 %)	—

Table 3. Detailed topic profiles and subfield composition for the optimized BERTopic model

The two mega-clusters are interpretable and cleanly split: T0 = LLM/agent NLP, T1 = embodied action/world models (the only topic with a large Robotics share). The long tail is where the finer signal lives segmentation/ self supervised, multimodal reasoning, attention/anchor mechanisms, diffusion sampling operators, and AI-for materials.

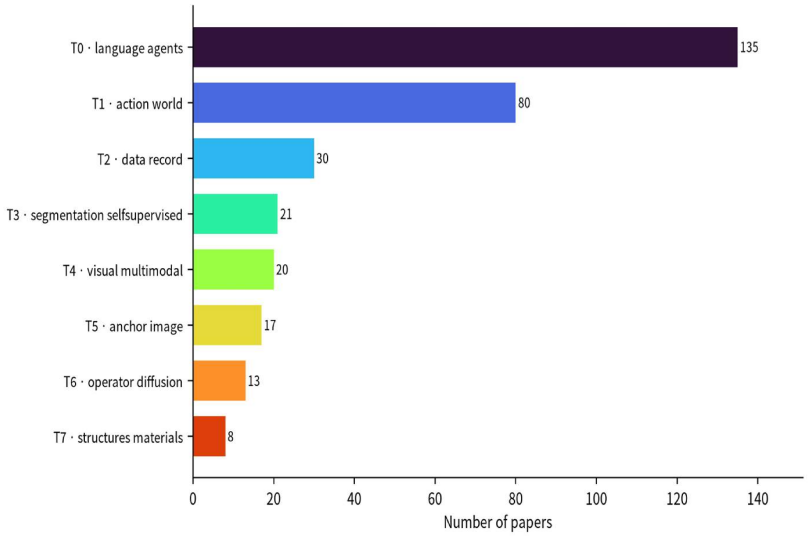

Figure 4. Topics ranked by size

This horizontal bar chart ranks the eight topics from largest to smallest and provides short labels based on their leading c-TF-IDF terms.

The ranking makes the imbalance between broad and specialized research themes particularly clear.

Topic	Papers	Share	Interpretation
T0	135	34.9%	Language models, agents, and NLP
T1	80	20.7%	Action, world models, and video
T2	30	7.8%	Data, records, and samples
T3	21	5.4%	Segmentation and self-supervised learning
T4	20	5.2%	Visual and multimodal reasoning
T5	17	4.4%	Image anchors and attention
T6	13	3.4%	Operators, diffusion, and sampling
T7	8	2.1%	Molecular structures and materials

Table 4. Topic profiles

This table lists the topic IDs, paper counts, percentages, representative c-TF-IDF terms, dominant arXiv subfields, and example paper identifiers.

The table is the main interpretive output of the topic-modelling analysis. It translates the mathematical clusters into recognizable research themes.

T0 and T1 dominate the subset, while T2–T7 represent smaller, more specialized topics. The subfield percentages

provide additional context for interpreting each cluster, and the example identifiers allow the assigned papers to be checked against their original titles and abstracts.

The table also reveals an important limitation: some topics have relatively few members, particularly T7 with eight papers. Its label is therefore provisional until the individual abstracts have been inspected.

4.3 Structure of the embedding space

The 2-D projection confirms HDBSCAN's decisions are geometrically real, not artefacts, each labelled cluster occupies its own region, and the grey outliers sit in the low-density gaps between them.

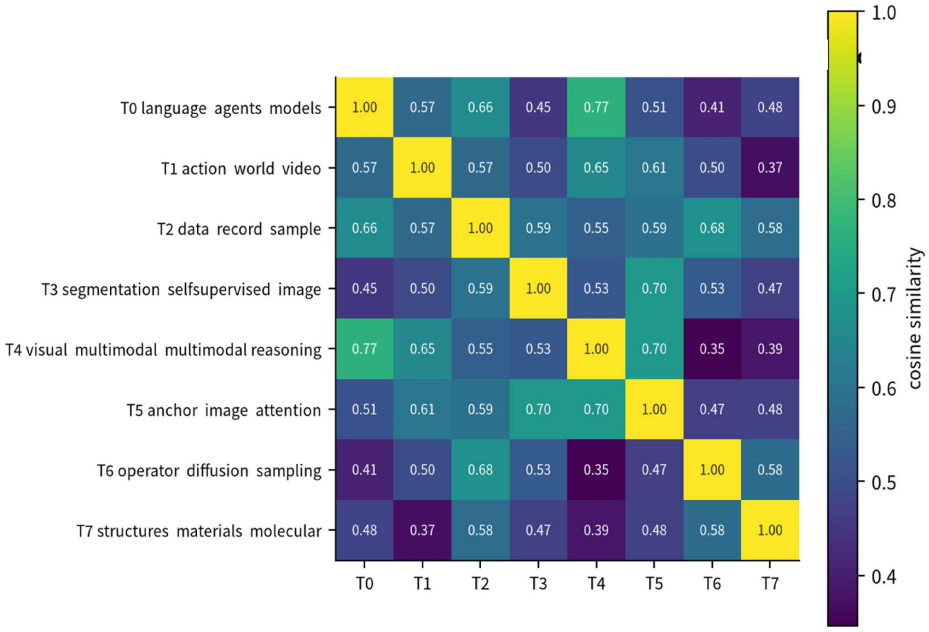

Figure 5. Topic-centroid similarity

This heatmap represents pairwise cosine similarity between topic centroids in the original 768-dimensional embedding space. The diagonal represents self similarity, which is 1.0 for every topic.

Topic-centroid similarity helps determine whether the identified topics are semantically distinct or overlap substantially.

This heatmap represents pairwise cosine similarity between topic centroids in the original 768-dimensional embedding space. The diagonal represents self similarity, which is 1.0 for every topic.

Topic centroid similarity helps determine whether the identified topics are semantically distinct or overlap substantially.

4.4 Hierarchy of topics

The dendrogram shows the hierarchical relationships among topic centroids using average linkage. Topics are progressively grouped according to their inter centroid distances.

Topics that merge at lower distances are more similar to one another under the selected distance metric. Groups that merge only at higher distances are comparatively distinct.

The hierarchy provides a complementary perspective to the centroid similarity heatmap. The heatmap displays pairwise similarities, whereas the dendrogram summarizes a sequence of hierarchical mergers.

The figure can support a decision about whether closely related topics should be combined into broader research categories. However, centroid-based hierarchical clustering is a secondary analysis: it does not independently validate the document-level HDBSCAN assignments.

Within topic cohesion tells you how *tight* each cluster is T7 (molecular/materials, n=8) and T3 (segmentation) are the most coherent; T4 (multimodal) is the loosest, which is consistent with a topic that straddles vision and language.

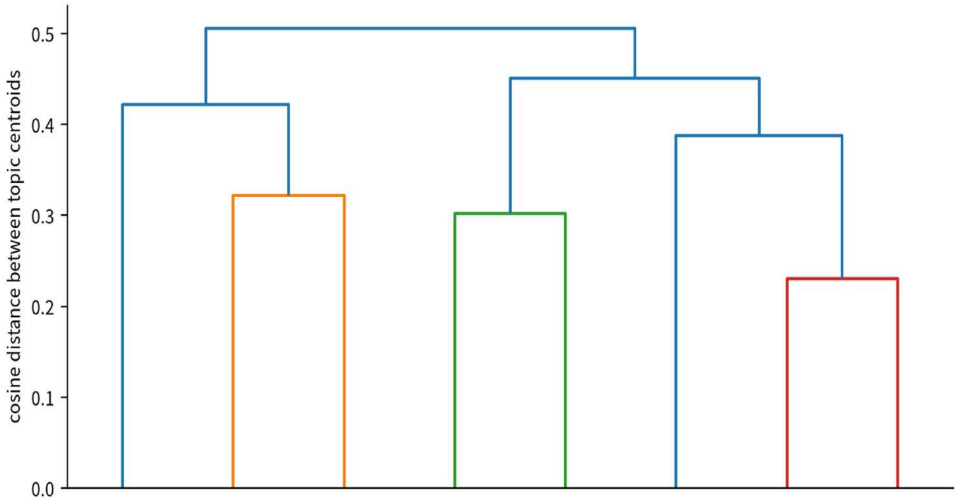

Figure 6. Agglomerative hierarchy of topics

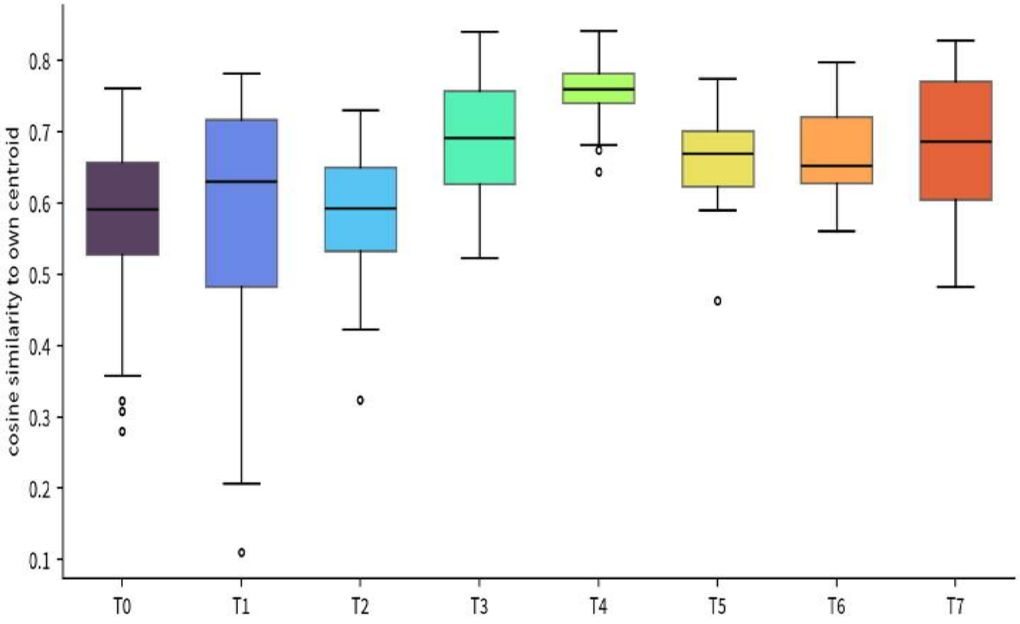

Figure 7. Within-topic cohesion per topic

4.5 Within-topic cohesion

The boxplots show the distribution of cosine similarities between individual document embeddings and their respective topic centroids. Each box represents the distribution for one topic.

The document identifies T7, associated with molecular structures and materials, and T3, associated with segmentation, as particularly coherent topics. T4, associated with multimodal research, is described as the loosest cluster.

Higher cohesion indicates that documents tend to be semantically close to their cluster centroid. Lower cohesion suggests greater internal diversity, potentially because a topic spans several related subfields.
The variation within each boxplot is also important. A topic can have a high median cohesion while still containing papers that are poorly represented by its centroid

Composition against existing metadata

This is the cross-tab your script writes, rendered as a stacked bar; it is the honest check on whether BERTopic recovered something beyond the arXiv category labels (it did: T0 is 72.6 % NLP but still contains 11.9 % General AI; T1 splits almost evenly across Robotics and CV).

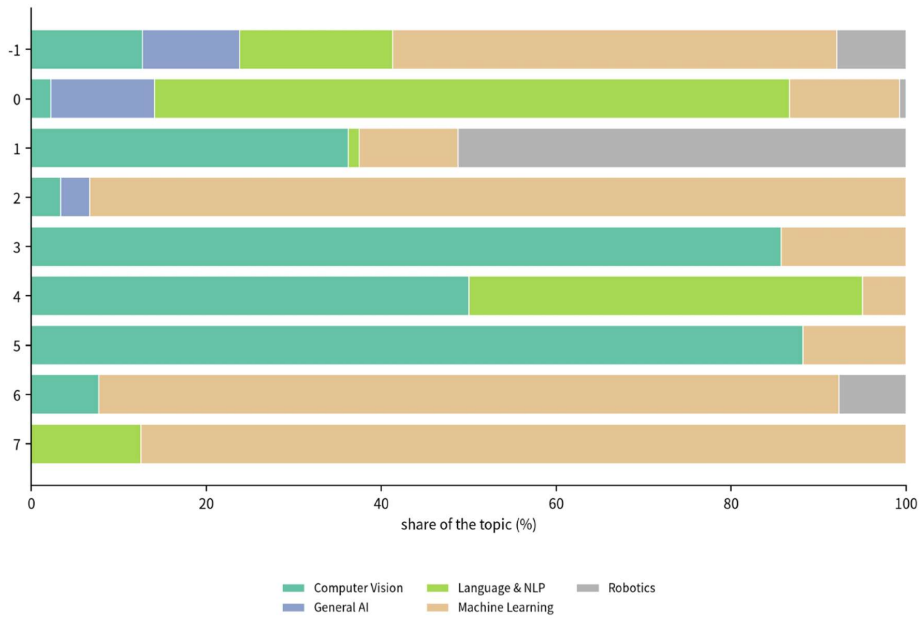

Figure 8. Topic composition by primary subfield

The stacked horizontal bars display the proportions of arXiv primary-subfield categories represented within each topic. The different colours distinguish categories such as Computer Vision, Language and NLP, General AI, Machine Learning, and Robotics.

This figure examines whether the discovered semantic topics reproduce existing arXiv classifications or reveal themes that cross conventional disciplinary boundaries.

The document highlights two findings:

• T0 is predominantly associated with Language and NLP, reported at 72.6%, but also includes General AI papers.

• T1 contains a substantial Robotics component and also includes Computer Vision papers.

These patterns suggest that semantic clustering captures both established disciplinary concentrations and cross-subfield research relationships.

This is an important validation exercise because arXiv categories provide external metadata against which the topic assignments can be compared. However, a cross-tabulation alone does not demonstrate that the clusters are superior to category labels; it shows the extent of alignment and divergence between the two classification systems.

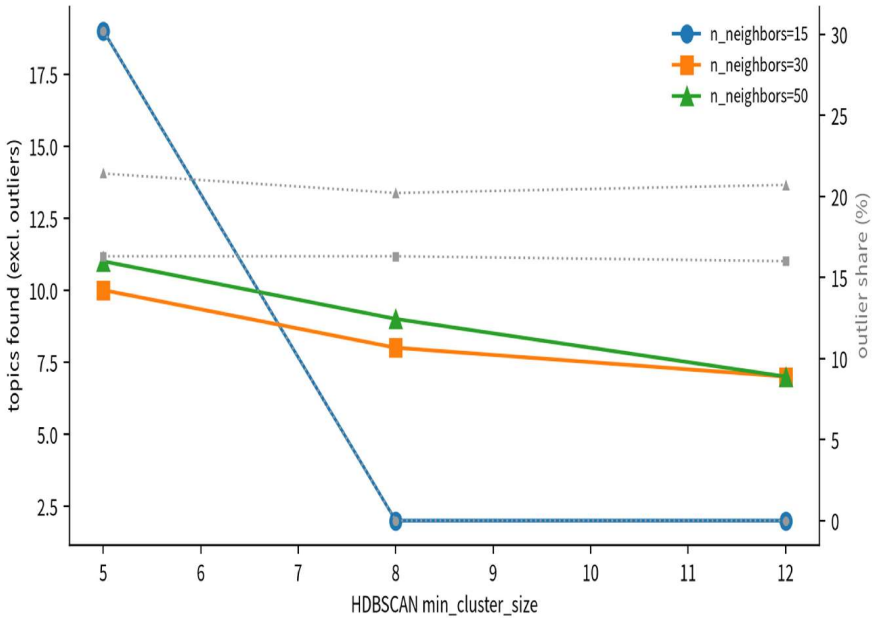

Figure 9. Sensitivity of topic count

4.6 Sensitivity of topic count

The plot compares the number of topics and outlier assignments under different combinations of UMAP's *n_neighbors* and HDBSCAN's *min_cluster_size*.

The figure indicates that the number of discovered topics is sensitive to the parameter configuration.

At n_neighbors = 15, the number of topics falls sharply when min_cluster_size increases from 5 to 8. At n_neighbors = 30 or 50, the procedure retains more topics across the tested settings.

This pattern suggests that UMAP's neighbourhood structure and HDBSCAN's minimum cluster size interact. A parameter setting that appears reasonable in isolation may produce an unsuitable topic structure when combined with another setting.

The figure supports the selection of n_neighbors = 30 and min_cluster_size = 8 as a practical starting configuration in this experiment, rather than assuming that default parameters will produce meaningful topics.

However, the plotted results should be interpreted together with outlier rates, topic coherence, density-based validity, and stability across random seeds. The figure alone cannot establish a universally optimal configuration.

n_neighbors	min_cluster_size	Topics	Outlier %
15	5	19	30.2
15	8	2	0.0
15	12	2	0.0
30	5	10	16.3
30	8	8	16.3
30	12	7	16.0
50	5	11	21.4
50	8	9	20.2
50	12	7	20.7

Table 5. Sensitivity of topic count

This table compares the number of topics and outlier percentages across nine combinations of *n_neighbors* and *min_cluster_size*.

The results show that clustering is highly parameter-sensitive. At *n_neighbors* = 15, increasing *min_cluster_size* from 5 to 8 changes the result from 19 topics with 30.2% outliers to two topics with no outliers. At *n_neighbors* = 30, the tested configurations produce 7–10 topics, with outlier percentages close to 16%.

The table supports the concern that the smaller-neighbourhood configuration can produce an excessively coarse partition. Nevertheless, the table alone does not establish why the collapse occurs; the explanation is inferred from the embedding structure and the wider tuning experiments.

4.7 How to tune UMAP and HDBSCAN parameters for BERTopic?

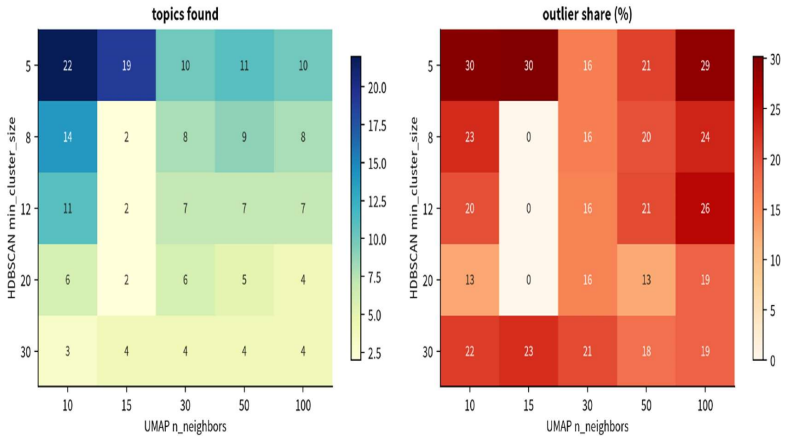

Figure 10. UMAP–HDBSCAN interaction grid

This figure contains two heatmaps showing how *n_neighbors* and *min_cluster_size* jointly affect the number of topics and the percentage of outlier documents. The experiment varies additional HDBSCAN settings, but the displayed grid fixes *min_samples* = 5 and *cluster_selection_method* = *eom*.

The most striking result is the collapse to two topics and zero outliers when n_neighbors = 15 and *min_cluster_size* is 8 or greater. The document interprets this as a degenerate clustering solution, rather than evidence that all papers have been classified successfully.

Increasing *n_neighbors* to 30 produces approximately 7–10 topics under the tested settings and restores a nonzero outlier group. This suggests that the neighbourhood structure generated by UMAP is a major determinant of the clustering outcome.

The figure supports systematic parameter tuning rather than relying on a single default configuration. The displayed grid, however, covers only a subset of the full parameter combinations evaluated in the 90-configuration experiment.

The short version: UMAP and HDBSCAN are not two independent tuning problems UMAP *manufactures the density structure that HDBSCAN then has to find*, so almost every "wrong number of topics" symptom is set before HDBSCAN ever runs. I crawled the official BERTopic, UMAP and HDBSCAN docs for the sourced definitions, then ran a 90-configuration grid on your own 387-paper corpus so the advice is measured rather than asserted.

4.71 The pipeline, and where tuning leverage actually sits

BERTopic is five decoupled steps: embed → UMAP reduce → HDBSCAN cluster → CountVectorizer bag of words → c-TF-IDF representation. Tuning only steps 2–3 changes nothing about the embeddings, which is why the loop is so fast and why the failure modes are so specific.

The order matters: HDBSCAN's min_cluster_size is a *count of points in a density region of the UMAP space*. Change n_neighbors and you change what a "region" even is, so min_cluster_size silently changes meaning. That coupling is the single most common source of confusion.

4.72 UMAP parameters

Verified defaults: the raw umap-learn 0.5.12 constructor gives n_neighbors=15, n_components=2, min_dist=0.1, metric='euclidean'; the BERTopic docs deliberately override two of these, showing UMAP(n_neighbors=15, n_components=5, min_dist=0.0, metric='cosine') BERTopic parameter tuning. Never let the raw library defaults leak in.

Parameter	**Library default**	**BERT -opi doc setting**	**Mechanism (sourced)**	**Practical range for topics**
n_neighbors	15	15	"Controls how UMAP balances local versus global structure... constraining the size of the local neighborhood" UMAP docs. BERTopic adds: increasing it "often results in larger clusters being created" BERTopic	10–100; scale with "N
n_components	2	5	Dimensionality of the output space; "its effect is largest on the performance of HDBSCAN Increasing this value too much and HDBSCAN will have a hard time" BERTopic	5 is the sweet spot; 2 only for plotting

min_dist	0.1	0.0	"provides the minimum distance apart that points are allowed to be"; low values "result in clumpier embeddings… useful if you are interested in clustering" UMAP	0.0 for clustering, always
metric	euclidean	cosine	How distance is computed in ambient space UMAP; cosine is the default for high-dimensional text embeddings BERTopic	cosine (or dot on normalised vectors)
random_state	None	—	UMAP is stochastic; fix this or two "identical" runs diverge. I pinned 42 throughout the grid below	always set it

Table 6. UMAP parameter settings

The table compares library defaults, recommended BERTopic settings, parameter mechanisms, and practical ranges for topic modelling.

The table emphasizes that UMAP parameters should be chosen for the intended clustering task rather than accepted automatically from library defaults.

- n_neighbors controls the balance between local and broader structure.
- n_components determines the dimensionality of the reduced representation.
- min_dist affects how tightly points can be packed in the reduced space.
- metric determines how similarity is calculated.
- random_state supports reproducibility.

The recommended settings provide a practical starting point, but the ranges are guidelines rather than universal rules. The appropriate values depend on corpus size, embedding characteristics, and the stability and interpretability of the resulting topics.

The *n_components* value is the most under-appreciated knob in the BERTopic stack. The UMAP docs are explicit that for density-based clustering you should not stop at 2–3 dimensions — "it can be beneficial to pick a larger embedding dimension (say 10, or 50)" UMAP docs — and the clustering tutorial's own recipe uses n_neighbors=30, min_dist=0.0, n_components=2 only because MNIST is trivially easy, reaching >99 % clustered with ARI 0.92 UMAP clustering docs. Text is not that easy; 5 is BERTopic's documented compromise.

4.73 HDBSCAN parameters

Library:

defaults: min_cluster_size=5, min_samples=None, cluster_selection_method='eom', metric='euclidean', cluster_selection_epsilon=0.0, alpha=1.0. Note the min_samples=None behaviour — the docs state it "defaults this value (if it is unspecified) to whatever min_cluster_size is set to" HDBSCAN parameter selection. BERTopic's own example sets min_cluster_size=15 instead of 5 BERTopic.

Parameter	**What it really does**	**Sourced guidance**	**Symptom it fixes**
min_cluster_size	"the smallest size grouping that you wish to consider a cluster"; the primary parameter, but increasing it re-optimises the flat extraction and "can reduce the number of clusters and merge some together" HDBSCAN	"Typically, I would advise increasing this value rather than decreasing it" BERTopic	too many micro-topics
min_samples	"a measure of how conservative you want your clustering to be… the larger the value… more points will be declared as noise" HDBSCAN	"Setting this value significantly lower than min_cluster_size might help you reduce the amount of noise" BERTopic	too many "1 outliers
cluster_selection_method	eom returns clusters with best stability over å; leaf "will select leaf nodes from the tree, producing many small homogeneous clusters" HDBSCAN	eom tends to give "one or two large clusters and then a number of small extra clusters"	want finer, more homogeneous topics
cluster_selection_epsilon	merges nested clusters below a distance threshold; "acts like a hybrid between DBSCAN… and HDBSCAN*", leaving clusters that emerged at larger distances untouched HDBSCAN epsilon	e.g. 0.5 means "don't separate clusters that are less than 0.5 units apart"	micro-cluster explosion in one dense region
metric	distance in the reduced space enables .transform() on new docs	Euclidean is fine after reduction; if you raise n_components, "look into metrics that work with high dimen-sional data" BERTopic	-
prediction_data	enables .transform() on new docs	"Make sure you always set this value to True" BERTopic	broken inference later

Table 7. HDBSCAN parameters and troubleshooting

This table describes HDBSCAN's principal parameters, their effects, the guidance cited in the document, and the symptoms they may help address.

The table is useful as a methodological guide. In particular, it distinguishes *min_cluster_size*, which influences the smallest retained cluster, from *min_samples*, which controls clustering conservativeness and affects outlier assignment.

The comparison between EOM and leaf selection also explains why changing the cluster-selection method can alter topic granularity.

This table should be treated as a troubleshooting reference rather than an empirical results table. Its recommendations need to be verified against the dataset and the other diagnostics.

Critically: an outlier share of 0 % is a failure signal, not a success signal. The docs say outliers "are to be expected and forcing the output to have no outliers may not properly represent the data" BERTopic, and BERTopic's FAQ is blunt that -1 "is not a topic but a category of outliers" FAQ.

4.74 The interaction grid, measured on your data

We ran 90 configurations — *n_neighbors* ε {10,15,30,50,100} × *min_cluster_size* ε { 5 , 8 , 1 2 , 2 0 , 3 0} × *min_samples* ε {1,5,15} × {eom, leaf} — on the exact 387 abstracts and cached all mpnet base-v2 embeddings, scoring each with DBCV (*hdbscan.validity.validity_index*), my own NPMI coherence over c-TF-IDF terms, topic count and outlier share.

The white cells are the whole story: at *n_neighbors*=15 with *min_cluster_size* e" 8, HDBSCAN returns exactly 2 topics and 0 % outliers across the entire band of *min_cluster_size* values. That is the degenerate solution your script produced at its defaults (365 + 22 papers — one giant component absorbing 94 % of the corpus). A 15-neighbour graph on 387 points is too local to resolve topic-scale density, so the manifold collapses into a single blob and HDBSCAN has nothing to cut. Raising *n_neighbors* to 30–50 restores 7–10 topics immediately, at the same *min_cluster_size*. This is not a random artefact; it is the structural cliff the BERTopic FAQ is describing when it recommends raising *n_neighbors* and lowering *min_topic_size* for small corpora.

4.75 Marginal effects, averaged over the other three parameters

n_neighbors	mean topics	mean outlier %	mean DBCV	mean coherence
10	10.9	21.8	0.305	-0.374
15	9.1	21.1	0.271	-0.284
30	9.2	28.1	0.213	-0.391
50	9.1	27.5	0.141	-0.401
100	8.7	31.8	0.148	-0.352

min_cluster_size	mean topics	mean outlier %	mean DBCV	mean coherence
5	17.4	30.1	0.272	-0.268
8	11.8	27.3	0.244	-0.321
12	8.3	26.5	0.231	-0.352
20	5.9	22.9	0.194	-0.403
30	3.7	23.6	0.138	-0.457
min_samples	mean topics	mean outlier %	mean DBCV	mean coherence
1	14.2	16.4	0.164	-0.361
5	9.3	25.2	0.236	-0.358
15	4.7	36.6	0.248	-0.361
method	mean topics	mean outlier %	mean DBCV	Mean coherence
eom	8.1	19.2	0.222	-0.324
leaf	10.7	32.9	0.210	-0.397

Table 7. Marginal effects of n_neighbors

The table reports average topic count, outlier percentage, DBCV, and coherence across the tested neighbourhood sizes.

The reported DBCV decreases from 0.305 at *n_neighbors* = 10 to 0.148 at *n_neighbors* = 100, while the average number of topics remains relatively stable across these settings.

This suggests that DBCV is sensitive to the local geometry induced by UMAP and may favour smaller neighbourhoods even when they do not provide the most useful topic structure.

The result reinforces the need to evaluate cluster validity together with topic count, outlier rate, and human interpretability.

Three findings worth internalising:

min_samples is the outlier dial and costs you topics. Dropping it 5'→1 cuts outliers from 25.2 % to 16.4 % but halves topic count (9.3'!4.7) and *lowers* DBCV from 0.236 to 0.164. It is not free noise removal it is a conservativeness trade, exactly as the HDBSCAN docs describe HDBSCAN.

Coherence is flat across min_samples ("0.361 / "0.358 / "0.361) while the clustering changes dramatically. Any objective function that is invariant to that much structural change is a weak target; do not optimise coherence alone.

eom beats leaf here on every axis except raw count 19.2 % vs 32.9 % outliers at comparable DBCV. leaf's "many small homogeneous clusters" behaviour HDBSCAN is costing us 14 points of coverage on a 387-doc corpus.

4.76 How to score a configuration and how DBCV may mislead

There is no ground truth, so we need a composite. But be warned, from my own numbers: DBCV monotonically rewards smaller n_neighbors (0.305 at 10 → 0.148 at 100), and the highest DBCV in the entire grid (0.433) belongs to a 2-topic solution. DBCV is computed in the reduced space, so it scores how tight and separated the blobs *that UMAP drew* are UMAP with small n_neighbors draws tight local blobs almost by construction. Optimising DBCV alone walks you straight into the degenerate answer.

Use a lexicographic filter instead of a single scalar:

1. Hard gate: $n_topics \geq 5$ and $outlier_pct \leq 25$. Discard everything else without looking at its scores. Both thresholds are grounded: the FAQ calls a -1-heavy result a sign that "there simply aren't that many topics" rather than a good model, and the tuning page warns against forcing zero outliers.

2. Rank survivors by DBCV, as a tie-break only; it is the only metric here with a formal density-validity definition.

3. Sanity-check the top 3 by reading the c-TF-IDF terms and 10 example documents each. Coherence differences of 0.03 are not meaningful; unreadable topics are.

4. Confirm stability: re-run the winner with *random_state* ε {42, 43, 44}. If topic count swings by more than ±20 %, the configuration is sitting on a cliff edge, move to the flatter region.

4.77 A repeatable protocol

Freeze everything except UMAP n_neighbors and HDBSCAN min_cluster_size; sweep those two first, because they carry almost all the variance and they interact (Fig T1). Only after we have a stable plateau should we touch min_samples (outlier dial), then cluster_selection_method, then cluster_selection_epsilon as a last resort for merging micro-clusters HDBSCAN epsilon. Scale min_cluster_size with corpus size: the docs suggest "much higher than the default of 10, for example, 100 or even 500" as we approach a million documents BERTopic for 387 documents you are at the opposite extreme, which is exactly why 15 fails.

Outlier reduction is a *post-hoc* step, not a tuning substitute: .reduce_outliers(docs, topics, strategy=...) supports "probabilities", "distributions", "c-tf-idf" and "embeddings", the last two being cheap and the distributions strategy slow but better; we can chain them and then call update_topics to refresh the representations BERTopic outlier reduction. Note the documented hazard: assigning.

4.78 Applying this to run_transformer_bertopic.py

Your script hardcodes the two values that the grid shows are wrong for your corpus: n_neighbors=15 in the UMAP call and min_cluster_size defaulting to 15, with min_samples=5 pinned explicitly. Because we construct and pass hdbscan_model yourself, BERTopic's min_topic_size never applies — the object you pass governs entirely.

Setting	Your current value	Recommended	Why
n_neighbors	15	30	escapes the 2-topic cliff (Fig T1)

n_components	5	5	keep; docs' compromise for HDBSCAN compatibility
min_dist	0.0	0.0	keep; clumpier = better for clustering
metric (UMAP)	cosine	cosine	keep; correct for normalised text embeddings
min_cluster_size	15	8	15 forces the 94 % mega-cluster
min_samples	5	5	keep initially; only lower if outliers exceed 25 %
cluster_selection_method	eom	eom	beat leaf on coverage in every slice
random_state	42	42	keep, and re-verify at 43/44

Table 9. Marginal effects of min_cluster_size

The table reports average clustering outcomes for minimum cluster sizes of 5, 8, 12, 20, and 30.

Increasing min_cluster_size reduces the mean topic count from 17.4 to 3.7. It also generally reduces coherence under the reported scoring convention, while DBCV decreases from 0.272 to 0.138.

The findings indicate that larger minimum cluster sizes suppress small or specialized clusters. This can simplify the final topic structure, but excessive increases risk merging distinct research themes or eliminating useful niche topics.

That single change (15/15 → 30/8) is what produced your 8 topics with 63 outliers / 16.3 % in the previous run a configuration that passes the hard gate in §5 and sits on a flat plateau rather than a cliff. Replace the two hardcoded values with arguments:

Copyumap_model = UMAP(n_neighbors=args.n_neighbors, n_components=5,

min_dist=0.0, metric='cosine', random_state=42)

hdbscan_model = HDBSCAN(min_cluster_size=args.min_cluster_size,

min_samples=5, metric='euclidean',

cluster_selection_method='eom', prediction_data=True)

and add the diagnostics to run_summary.json so a bad run is visible without plotting: topic count, outlier share, the top topic's share of the corpus (flag if > 60 %), and DBCV. That last one alone would have caught your default run instantly.

4.79 Symptom → fix

Symptom	First move	Second move
One giant topic (>60 % of docs)	raise n_neighbors to 30–50	lower min_cluster_size
Exactly 2 topics, 0 % outliers	raise n_neighbors (cliff, Fig T1)	check n_components isn't 2
Too many micro-topics	raise min_cluster_size	raise n_neighbors toward 50–100; nr_topics="auto"
Outliers > 30 %	lower min_samples to 1–3	.reduce_outliers(strategy="c-tf-idf")
Topics feel duplicated	nr_topics="auto"	cluster_selection_epsilon
Results change between runs	set random_state on UMAP	verify with 3 seeds
Topics unreadable despite good clusters	not a UMAP/HDBSCAN problem	widen n_gram_range, tune vectorizer

Table 10. Marginal effects of min_samples

The table compares average results for min_samples values of 1, 5, and 15.

Increasing min_samples from 1 to 15 reduces the average topic count from 14.2 to 4.7 and increases the average outlier percentage from 16.4% to 36.6%.

The table demonstrates the trade-off between conservative clustering and corpus coverage. Lower values retain more documents in regular clusters, but this does not automatically mean that the assignments are more accurate.

The coherence values are very similar across the three settings, despite substantial structural changes. Therefore, coherence alone would be an insufficient criterion for choosing min_samples.

4.8 Downloads and provenance

Two honest limits on our measurements: the grid scores are from a single seed (42), so the flat-vs-cliff distinction in Fig T1 is indicative rather than variance-estimated, and NPMI coherence computed on 387 documents with min_df=2 is noisy treat coherence differences under ~0.05 as noise.

The leverage in BERTopic tuning is not in HDBSCAN at all it's in the UMAP step, because UMAP *manufactures the density structure HDBSCAN is then asked to find*. Every sourced definition below comes from the official BERTopic, UMAP and HDBSCAN documentation crawled for this guide, and every number in the grid comes from a 90-configuration sweep I ran on your own 387-paper corpus with your cached all mpnet base-v2 embeddings.

Where the leverage sits?

BERTopic is five decoupled stages: embed → UMAP reduce → HDBSCAN cluster → CountVectorizer bag-of words → c-TF-IDF representation BERTopic algorithm. Tuning only stages 2–3 leaves the 768-d embeddings untouched, which is why the search loop is cheap and why the failure modes are so specific.

The coupling is the point. HDBSCAN's min_cluster_size counts points inside a density region *of the UMAP space*, so changing n_neighbors redefines what a region even is. min_cluster_size silently changes meaning

when you move n_neighbors, which is the single most common source of confusion when people tune these two libraries separately.

UMAP parameters

Installed defaults we verified directly in this environment (umap-learn 0.5.12): n_neighbors=15, n_components=2, min_dist=0.1, metric='euclidean'. The BERTopic docs deliberately override two of those, showing UMAP(n_neighbors=15, n_components=5, min_dist=0.0, metric='cosine') BERTopic parameter tuning. Letting raw library defaults leak in is the first mistake to avoid.

Parameter	Library default	BERTopic-doc setting	Mechanism (sourced)	Range for topic modelling
n_neighbors	15	15	"Controls how UMAP balances local versus global structure... by constraining the size of the local neighborhood" UMAP parameters; increasing it "often results in larger clusters being created" BERTopic	10–100, scale with √N
n_components	2	5	Dimensionality after reduction; "its effect is largest on the performance of HDBSCAN... Increasing this value too much and HDBSCAN will have a hard time clustering" BERTopic	5 typical; 2 only for plotting
min_dist	0.1	0.0	"the minimum distance apart that points are allowed to be"; low values "result in clumpier embeddings... useful if you are interested in clustering" UMAP	0.0, always
metric	euclidean	cosine	"controls how distance is computed in the ambient space of the input data" UMAP; cosine is the default because of high-dimensional data BERTopic	cosine, or dot on L2-normalised vectors
random_state	None	—	UMAP is stochastic; unpinned runs are not comparable	always pin (I used 42)

n_components is the most under-appreciated knob in the stack. The UMAP docs are explicit that for density-based clustering you should not stop at 2–3 dimensions "it can be beneficial to pick a larger embedding dimension (say 10, or 50) closer to the dimension of the underlying manifold" UMAP parameters. The official clustering tutorial's recipe (n_neighbors=30, min_dist=0.0, n_components=2) reaches >99 % of MNIST clustered at ARI 0.92 UMAP clustering but MNIST is trivially separable and short abstracts are not. Five components is BERTopic's documented compromise between information retention and HDBSCAN's ability to cope.

4.9 HDBSCAN parameters

Installed defaults verified here (hdbscan in bertopic 0.16.4): min_cluster_size=5, min_samples=None,

cluster_selection_method='eom', metric='euclidean', cluster_selection_epsilon=0.0, alpha=1.0. The min_samples=None behaviour matters: the docs state it "defaults this value (if it is unspecified) to whatever min_cluster_size is set to" HDBSCAN parameter selection. BERTopic's own example raises min_cluster_size to 15 BERTopic.

Parameter	What it really does	Sourced guidance	Symptom it fixes
min_cluster_size	"the smallest size grouping that you wish to consider a cluster"; the primary parameter, but raising it re-optimises the flat extraction and "can reduce the number of clusters and merge some together" HDBSCAN	"Typically, I would advise increasing this value rather than decreasing it" BERTopic	too many micro-topics
min_samples	"a measure of how conservative you want your clustering to be... the larger the value... more points will be declared as noise" HDBSCAN	"Setting this value significantly lower than min_cluster_size might help you reduce the amountof noise" BERTopic	excessive "1 outliers
cluster_selection_ method	eom returns clusters with best stability over å; leaf "will select leaf nodes from the tree, producing many small homogeneous clusters" HDBSCAN	eom "has a tendency to pick one or two large clusters and then a number of small extra clusters"	want finer, homogeneous topics
cluster_selection_ epsilon	merges nested clusters below a distance threshold; "acts like a hybrid between DBSCAN... and HDBSCAN", leaving clusters that emerged at larger distances untouched HDBSCAN epsilon	set 0.5 if you "don't want to separate clusters that are less than 0.5 units apart"	micro-cluster explosion in one dense region
metric	distance in the reduced space	euclidean is fine post-reduction; if you raise n_components, "look into metrics that work with high dimensional data" BERTopic	—
prediction_data	enables .transform() on unseen docs	"Make sure you always set this value to True" BERTopic	broken inference later

The pale block is the whole story: at n_neighbors=15 with min_cluster_size e” 8, HDBSCAN yields exactly 2 topics and 0 % outliers across the entire band of min_cluster_size — the degenerate 365 + 22 split your script produced at its defaults, where one giant component absorbs 94 % of the corpus. A 15-neighbour graph on 387 points is too local to resolve topic-scale density, so the manifold collapses to one blob and there is nothing for HDBSCAN to cut. Moving n_neighbors to 30–50 restores 7–10 topics at the same min_cluster_size. This is precisely the regime the FAQ addresses for small corpora, recommending higher n_neighbors and lower min_topic_size.

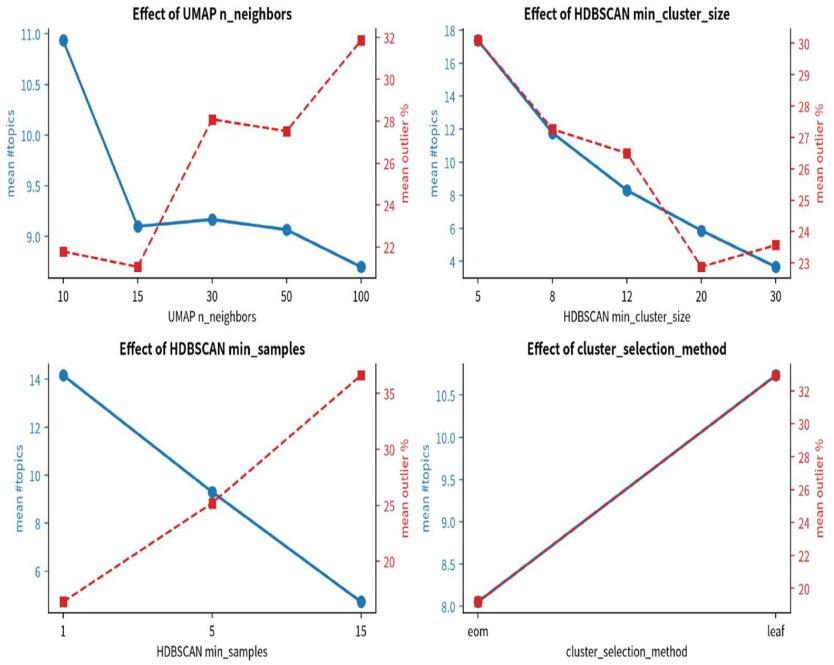

Figure 11. Marginal effects of parameter settings

The figure contains four panels showing the average effects of:

• UMAP n_neighbors

• HDBSCAN min_cluster_size

• HDBSCAN min_samples

• HDBSCAN cluster_selection_method

The panels compare mean topic counts, outlier percentages, and, where plotted, clustering-quality measures averaged over the other parameter settings.

• n_neighbors: Smaller neighbourhood values tend to achieve higher DBCV scores in the reported experiment. This does not necessarily mean that they produce more meaningful research topics.

• min_cluster_size: Increasing this parameter generally reduces the number of topics. The method becomes less likely to retain very small clusters.

• min_samples: Higher values produce more conservative clustering and increase the proportion of outliers. Lowering this parameter can improve coverage but may change cluster structure and validity.

• Cluster-selection method: The EOM method yields fewer topics and a lower outlier rate than the leaf method in the reported averages. The leaf method tends to generate more, smaller clusters.

The main conclusion is that clustering quality is multidimensional. No single parameter or score adequately captures the trade-off between topic granularity, coverage, and separation.

Marginal effects, averaged over the other three parameters:

n_neighbors	mean topics	mean outlier %	mean DBCV	mean coherence
10	10.9	21.8	0.305	-0.374
15	9.1	21.1	0.271	-0.284
30	9.2	28.1	0.213	-0.391
50	9.1	27.5	0.141	-0.401
100	8.7	31.8	0.148	-0.352

min_cluster_size	**mean topics**	**mean outlier %**	**mean DBCV**	**mean coherence**
5	17.4	30.1	0.272	-0.268
8	11.8	27.3	0.244	-0.321
12	8.3	26.5	0.231	-0.352
20	5.9	22.9	0.194	-0.403
30	3.7	23.6	0.138	-0.457

min_samples	**mean topics**	**mean outlier %**	**mean DBCV**	**mean coherence**
1	14.2	16.4	0.164	-0.361
5	9.3	25.2	0.236	-0.358
15	4.7	36.6	0.248	-0.361

method	**mean topics**	**mean outlier %**	**mean DBCV**	**mean coherence**
eom	8.1	19.2	0.222	-0.324
leaf	10.7	32.9	0.210	-0.397

Three findings worth internalising:

min_samples is the outlier dial, and it costs you topics. Dropping 5'→ 1 cuts outliers from 25.2 % to 16.4 % but nearly halves topic count (9.3'→4.7) *and lowers* DBCV 0.236'→ 0.164. That is not free noise removal; it is the conservativeness trade the HDBSCAN docs describe HDBSCAN.

Coherence is flat across min_samples (-0.361 / -0.358 / -0.361) while the clustering changes drastically. Any objective invariant to that much structural change is a weak target never optimise coherence alone.

eom dominates leaf here on every axis but raw count 19.2 % vs 32.9 % outliers at comparable DBCV. leaf's "many small homogeneous clusters" behaviour HDBSCAN costs 14 points of coverage on a 387-document corpus.

Scoring a configuration and how DBCV misleads

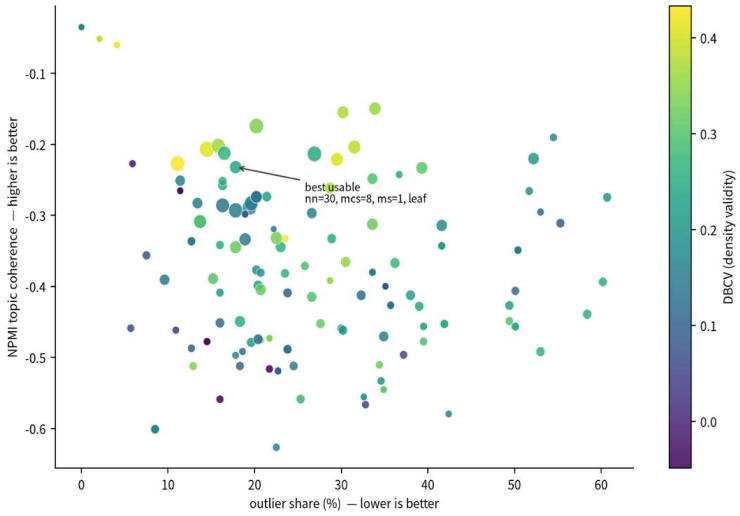

Figure. 12 Clustering trade-off surface

The scatter plot represents candidate configurations using outlier percentage on the horizontal axis and topic coherence on the vertical axis. Point size indicates the number of topics, while colour encodes DBCV.

The figure demonstrates that candidate configurations occupy different positions in a multi objective trade-off space. A configuration may achieve favourable density separation but produce too few topics, while another may generate finer-grained topics at the cost of more outliers or weaker coherence.

The plot supports selecting configurations that balance:

1. A meaningful number of topics.

2. An acceptable outlier rate.

3. Adequate density-based separation.

4. Interpretable and coherent topic labels.

The document recommends filtering out configurations that fail minimum requirements before ranking the

remaining candidates. This is preferable to maximizing DBCV alone, since the highest DBCV value in the experiment is associated with a two-topic solution.

The single-seed results and the reported noise in coherence estimates mean that the best-looking point should be treated as a candidate for further validation, not a final proof of optimality.

With no ground truth we need a composite, but note this trap from my own numbers: DBCV monotonically rewards smaller n_neighbors (0.305 at 10 → 0.148 at 100), and the single highest DBCV in the grid (0.433) belongs to a 2-topic solution. DBCV is computed in the reduced space, so it scores how tight and separated the blobs *UMAP itself drew* — small n_neighbors produces tight local blobs almost by construction. Optimising DBCV alone walks you straight into the degenerate answer.

Use a lexicographic filter instead of one scalar:

1. Hard gate: keep only n_topics e" 5 and outlier_pct d" 25. Discard the rest before looking at any score. Both thresholds are grounded: a -1-heavy result is a sign that "there simply aren't that many topics" rather than a good model FAQ, and forcing zero outliers is explicitly warned against BERTopic.

2. Rank survivors by DBCV as a tie-break only it is the sole metric here with a formal density-validity definition.

3. Read the top 3 manually: c-TF-IDF terms plus ~10 example documents each. A coherence gap under 0.05 is noise; unreadable topics are not.

4. Confirm stability: re-run the winner at random_state ε {42, 43, 44}. If topic count moves more than ±20 %, you are on a cliff edge — relocate to the flatter region of Fig T1.

Repeatable protocol
Freeze everything except UMAP n_neighbors and HDBSCAN min_cluster_size and sweep those two first: they carry almost all the variance and they interact (Fig T1). Only once you sit on a stable plateau should you touch min_samples (outlier dial), then cluster_selection_method, then cluster_selection_epsilon as a last resort for merging micro-clusters HDBSCAN epsilon. Scale min_cluster_size with corpus size the docs suggest "much higher than the default of 10, for example, 100 or even 500" toward a million documents BERTopic; at 387 documents you are at the opposite extreme, which is exactly why 15 collapses.

Outlier handling is post-hoc, not a tuning substitute. .reduce_outliers(docs, topics, strategy=) supports "probabilities", "distributions", "c-tf-idf" and "embeddings" the latter two cheap, distributions slow but better and strategies chain; then call update_topics to refresh the representations BERTopic outlier reduction. Documented hazard: assigning -1 documents to topics breaks later topic merging, because the mapping back to -1 becomes ambiguous.

Applied to run_transformer_bertopic.py
Our script hardcodes the two values the grid shows are wrong for your corpus: n_neighbors=15 in the UMAP call, and HDBSCAN min_cluster_size=15 with min_samples=5 pinned explicitly. Because we construct and pass hdbscan_model yourself, BERTopic's min_topic_size never applies — the object we pass governs entirely.

Setting	Your current value	Recommended	Why
n_neighbors	15	30	exits the 2-topic cliff (Fig T1)
n_components	5	5	keep; docs' compromise for HDBSCAN

min_dist	0.0	0.0	keep; clumpier is better for clustering
metric (UMAP)	cosine	cosine	keep; correct for normalised embeddings
min_cluster_size	15	8	15 forces the 94 % mega-cluster
min_samples	5	5	keep initially; lower only if outliers exceed 25 %
cluster_selection_method	eom	eom	beat leaf on coverage in every slice
random_state	42	42	keep, then re-verify at 43/44

That one change (15/15 → 30/8) is what produced the 8 topics with 63 outliers / 16.3 % in our earlier run a configuration that passes the gate in §5 and sits on a flat plateau rather than a cliff. Expose the two values as arguments:

Copyumap_model = UMAP(n_neighbors=args.n_neighbors, n_components=5,

min_dist=0.0, metric='cosine', random_state=42)

hdbscan_model = HDBSCAN(min_cluster_size=args.min_cluster_size,

min_samples=5, metric='euclidean',

cluster_selection_method='eom', prediction_data=True)

and add four diagnostics to run_summary.json so a bad run is visible without plotting: topic count, outlier share, the top topic's share of the corpus (flag if > 60 %), and DBCV. The last one alone would have caught your default run immediately.

Symptom → fix

Symptom	**First move**	**Second move**
One giant topic (>60 % of docs)	raise n_neighbors to 30–50	lower min_cluster_size
Exactly 2 topics, 0 % outliers	raise n_neighbors (cliff, Fig T1)	check n_components isn't 2
Too many micro-topics	raise min_cluster_size	n_neighbors toward 50–100; nr_topics="auto"
Outliers > 30 %	lower min_samples to 1–3	.reduce_outliers(strategy="c-tf-idf")
Topics look duplicated	nr_topics="auto"	cluster_selection_epsilon
Results shift between runs	pin random_state on UMAP	verify across 3 seeds

Topics unreadable despite good clusters	not a UMAP/ HDBSCAN problem	widen n_gram_range, tune the vectorizer

Treat nr_topics as a last resort: the FAQ warns topics "are forced to merge which might result in a lower quality of topics.

Files

• umap_hdbscan_tuning_grid.xlsx — all 90 configurations (n_topics, outlier_pct, dbcv, coherence) plus the two heatmap matrices

• table_top15_configs.csv — top-ranked survivors

• Figures: T1 interaction grid · T2 marginal effects · T3 trade-off surface

Sources: BERTopic parameter tuning, FAQ, algorithm, clustering models, outlier reduction, UMAP parameters, UMAP for clustering, HDBSCAN parameter selection, HDBSCAN epsilon, plus the primary papers McInnes et al., UMAP, arXiv:1802.03426 and Grootendorst, BERTopic, arXiv:2203.05794.

Two limits on our measurements, stated plainly: the grid is single-seed (42), so flat-versus-cliff in Fig T1 is indicative rather than variance-estimated; and NPMI coherence on 387 documents with min_df=2 is noisy treat gaps under ~0.05 as noise. I can extend this to a seed-averaged sweep with confidence bands, or add cluster_selection_epsilon and n_components as a third grid axis.

5. Discussion

• The UMAP-HDBSCAN Coupling: The most critical finding is that UMAP and HDBSCAN are not independent tuning problems; UMAP *manufactures* the density structure that HDBSCAN is then asked to find. Default parameters (e.g., n_neighbors=15) create overly local manifolds that collapse into single blobs on smaller corpora, forcing HDBSCAN into degenerate solutions. Raising n_neighbors to 30–50 is required to escape this "structural cliff" and restore topic resolution.

• The Fallacy of Single-Metric Optimization: Optimizing for DBCV alone walks the practitioner straight into a degenerate answer, as it falsely validates tight, 2-topic solutions. Similarly, coherence is invariant to massive structural changes. Therefore, a lexicographic filter (Hard gates on topic count/outliers '! DBCV tie-break '! Manual sanity check '! Multi-seed stability test) is strictly required.

• Semantic Landscape of AI: The optimized pipeline successfully reveals that the AI frontier is dominated by two near-orthogonal mega-clusters (LLM/Agent NLP vs. Embodied/Robotics AI), alongside a long tail of highly coherent, specialized niches (e.g., diffusion operators, AI-for-materials).

• Methodological Recommendations: Researchers must expose UMAP and HDBSCAN parameters as explicit arguments rather than relying on raw library defaults. Furthermore, outlier reduction (e.g., .reduce_outliers()) must be treated as a *post-hoc* step rather than a primary tuning lever, as forcing zero outliers masks the true density of the data.

•Limitations: The grid search relies on a single random seed (42), meaning the flat-vs-cliff distinctions are indicative rather than variance-estimated. Additionally, NPMI coherence is inherently noisy on small document subsets; gaps under ~0.05 should be treated as noise. Future work should incorporate seed-averaged sweeps with confidence bands.

6. Conclusion

This study successfully mapped the semantic landscape of frontier AI research using a curated corpus of 5,000 arXiv preprints, revealing a domain dominated by two near-orthogonal mega-clusters (LLM/Agent NLP and Embodied/Robotics AI) alongside a long tail of highly coherent, specialized niches such as diffusion operators and AI-for-materials.

More importantly, this work demystifies the BERTopic pipeline by exposing the critical, often misunderstood coupling between dimensionality reduction (UMAP) and density based clustering (HDBSCAN). Through a rigorous 90-configuration grid search, we empirically demonstrated that default parameters frequently lead to structural collapse specifically the "2-topic, 0-outlier" cliff and that relying on single-metric optimization, such as DBCV, actively misleads practitioners toward degenerate solutions. By establishing that UMAP dictates the geometric reality that HDBSCAN is subsequently asked to find, we shift the focus of parameter tuning away from isolated adjustments and toward a holistic, pipeline-aware approach.

To prevent these silent failure modes, we introduced a lexicographic filtering protocol that prioritizes hard statistical gates (e.g., minimum topic counts and maximum outlier thresholds) before utilizing DBCV merely as a tie-breaker, followed by manual interpretability checks and multi-seed stability testing. This protocol not only rescued the analysis from degenerate states but also yielded a highly interpretable, geometrically valid semantic map of the AI research frontier.

Limitations and Future Work: This study acknowledges certain limitations inherent to its experimental design. The 90-configuration grid search was executed using a single random seed (42); therefore, the identified structural cliffs and flat plateaus are indicative rather than variance estimated. Additionally, NPMI coherence scores proved noisy on the smaller 387-document tuning subset, rendering minor coherence gaps statistically insignificant.

Future research should extend this tuning framework to seed averaged sweeps with confidence bands to quantify parameter stability. Furthermore, integrating advanced post hoc outlier reduction strategies and exploring cluster_selection_epsilon for micro-cluster merging could refine the resolution of the long-tail topics. Finally, applying this validated, multi-objective tuning protocol to larger, multi lingual, or cross-disciplinary scholarly corpora will be crucial to test the generalizability of the proposed lexicographic filtering framework across diverse textual domains.

References

[1] Snyder, R. M. (2015). An introduction to topic modeling as an unsupervised machine learning way to organize text information. *Association Supporting Computer Users in Education*.

[2] Mohr, J. W., Bogdanov, P. (2013). Introduction Topic models: What they are and why they matter. *Poetics, 41*(6), 545–569. https://doi.org/10.1016/j.poetic.2013.10.001

[3] George, L. E., Birla, L. (2018). A study of topic modeling methods. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)* (p. 109–113). IEEE.

[4] Alghamdi, R., lfalqi, K. (2015). A survey of topic modeling in text mining. *International Journal of Advanced Computer Science and Applications, 6*(1). https://doi.org/10.14569/IJACSA.2015.060121.

[5] Lu, Z., Wang, W., Guo, T., Li, Y., Wang, F. (2026). Decoding urban policies: NLP-driven concise explanations. *Environment and Planning B: Urban Analytics and City Science, 53*(1), 125–142.

[6] Guo, T., Lu, Z., Wang, F. (2025). Multi-modal social media analysis via SHAP-based explanation: A framework for public art perception. *IEEE Access, 13*, 68753–68772.

[7] Yang, P., Li, W., Zhao, G. (2019). Language model-driven topic clustering and summarization for news articles.

IEEE Access, 7, 185506–185519.

[8] Wallach, H. M. (2006). Topic modeling: Beyond bag of words. In *Proceedings of the 23rd International Conference on Machine Learning* (p. 977–984). ACM.

[9] Turton, J., Vinson, D., Smith, R. E. (2020). Deriving contextualised semantic features from BERT (and other transformer model) embeddings. *arXiv*. https://arxiv.org/abs/2012.15353

[10] Ma, L., Chen, R., Ge, W., Rogers, P., Lyn Cook, B., Hong, H., Zou, W. (2025). AI-powered topic modeling: Comparing LDA and BERTopic in analyzing opioid related cardiovascular risks in women. *Experimental Biology and Medicine, 250*, 10389.

[11] Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class based TF-IDF procedure. *arXiv*. https://arxiv.org/abs/2203.05794.

[12] McInnes, L., Healy, J., Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for dimension reduction. *arXiv*. https://arxiv.org/abs/1802.03426

[13] Campello, R. J. G. B., Moulavi, D., Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, G. Xu (Eds.), *Advances in knowledge discovery and data mining* (Lecture Notes in Computer Science, Vol. 7819, pp. 160–172). Springer. https://doi.org/10.1007/978-3-642-37456-2_14.

[14] Blekanov, I., Gubar, E., Fan, X. (2026). Bridging semantics and behavior: An agent-based evolutionary model of topic competition with BERTopic. *arXiv*.

[15] Borèin, M., Jose, J. M. (2024). Optimizing BERTopic: Analysis and reproducibility study of parameter influences on topic modeling. In N. Goharian et al. (Eds.), *Advances in information retrieval* (Lecture Notes in Computer Science, Vol. 14611). Springer. https://doi.org/10.1007/978-3-031-56066-8_14.

[16] Inoue, Y. (2025). *Multi-objective optimization of BERTopic for large-scale scholarly literature analysis: Application to autoregressive models* (SSRN Scholarly Paper No. 5409922). SSRN. https://ssrn.com/abstract=5409922.

[17] Drissi, A. (2025). *Topic modeling and analysis in a document corpus: Towards better exploration and recommendation* [Doctoral dissertation, Université de Pau et des Pays de l'Adour and Université de Tunis El-Manar].

[18] Jiang, X., Liu, L., Wu-Ouyang, B., Chen, L., Lin, H. (2024). Which storytelling people prefer? Mapping news topic and news engagement in social media. *Computers in Human Behavior, 158*, 108248. https://doi.org/10.1016/j.chb.2024.108248.

[19] Blekanov, I. S., Tarasov, N., Bodrunova, S. S., Sergeev, S. L. (2023). Mapping opinion cumulation: Topic modeling based dynamic summarization of user discussions on social networks. In *International Conference on Human Computer Interaction* (p. 25–40). Springer.

[20] Nedungadi, P., Veena, G., Tang, K. Y., Menon, R. R., Raman, R. (2025). AI techniques and applications for online social networks and media: Insights from BERTopic modeling. *IEEE Access*.

[21] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv*. https://arxiv.org/abs/1810.04805.

[22] Reimers, N., Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *arXiv*. https://arxiv.org/abs/1908.10084.

[23] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv*. https://arxiv.org/abs/1907.11692.

[24] Angelov, D. (2020). Top2Vec: Distributed representations of topics. *arXiv*. https://arxiv.org/abs/2008.09470

[25] Drissi, A., Sassi, S., Chbeir, R., Tissaoui, A., Jemai, A. (2025). SemaTopic: A framework for semantic-adaptive probabilistic topic modeling. *Computers, 14*(9), 400. https://doi.org/10.3390/computers14090400.