

Assignment_3

May 16, 2025

0.1 Assignment 3

0.1.1 Authors: Jacopo Corrao, Pietro Farina

This assignment consist in developing code that combines LLMs and traditional ML to extract knowledge from the Openreivew review dataset. The goal is to find interesting patterns in reviews (and possibly, why not, in papers) that can give hints to authors about how to write better papers.

We decided to organize the work in the following:

1. Manual inspection of the dataset
2. Data Cleaning
3. Knowledge Extraction from Paper Acceptance Data using BERT-based NLP Techniques
4. Paper Acceptance Analysis
5. Pairwise comparison of papers through LLMs

0.1.2 Setup

```
[3]: from google.colab import userdata  
my_secret_key = userdata.get('API_KEY')
```

```
[4]: !pip install openai pandas openpyxl
```

Requirement already satisfied: openai in /usr/local/lib/python3.11/dist-packages (1.78.1)

Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)

Requirement already satisfied: openpyxl in /usr/local/lib/python3.11/dist-packages (3.1.5)

Requirement already satisfied: anyio<5,>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from openai) (4.9.0)

Requirement already satisfied: distro<2,>=1.7.0 in /usr/local/lib/python3.11/dist-packages (from openai) (1.9.0)

Requirement already satisfied: httpx<1,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from openai) (0.28.1)

Requirement already satisfied: jiter<1,>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from openai) (0.9.0)

Requirement already satisfied: pydantic<3,>=1.9.0 in /usr/local/lib/python3.11/dist-packages (from openai) (2.11.4)

Requirement already satisfied: sniffio in /usr/local/lib/python3.11/dist-

packages (from openai) (1.3.1)
 Requirement already satisfied: tqdm>4 in /usr/local/lib/python3.11/dist-packages (from openai) (4.67.1)
 Requirement already satisfied: typing-extensions<5,>=4.11 in /usr/local/lib/python3.11/dist-packages (from openai) (4.13.2)
 Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.0.2)
 Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.9.0.post0)
 Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
 Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
 Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.11/dist-packages (from openpyxl) (2.0.0)
 Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<5,>=3.5.0->openai) (3.10)
 Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx<1,>=0.23.0->openai) (2025.4.26)
 Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx<1,>=0.23.0->openai) (1.0.9)
 Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx<1,>=0.23.0->openai) (0.16.0)
 Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<3,>=1.9.0->openai) (0.7.0)
 Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<3,>=1.9.0->openai) (2.33.2)
 Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<3,>=1.9.0->openai) (0.4.0)
 Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)

```

[3]: from google.colab import drive

drive.mount("/content/drive")

file_path = "/content/drive/MyDrive/KnowledgeDiscoveryAndPatternExtraction/
↪open_review_dataset.xlsx"
  
```

Mounted at /content/drive

1 Manual Inspection of the Dataset

```
[5]: import pandas as pd

# Load the Excel file
excel_file = pd.ExcelFile(file_path)

# Display sheet names
print(excel_file.sheet_names)

# Load each sheet into a dictionary
sheets = {sheet_name: excel_file.parse(sheet_name) for sheet_name in excel_file.
          sheet_names}

# Prepare summaries for each sheet
sheet_summaries = {}
for name, df in sheets.items():
    summary = {
        "columns": df.columns.tolist(),
        "head": df.head().to_dict(orient="records")
    }
    sheet_summaries[name] = summary
for sum in sheet_summaries:
    print(sheet_summaries[sum])
```

```
['Sheet1', 'Sheet2', 'Sheet3', 'Sheet4', 'Sheet6', 'Sheet5']
{'columns': ['title', 'Unnamed: 1', 'keywords', 'E', 'F', 'G', 'decision', 'J',
'K', 'rate', 'T'], 'head': [{'title': '#Exploration: A Study of Count-Based
Exploration for Deep Reinforcement Learning | OpenReview', 'Unnamed: 1': '06 Nov
2016 (modified: 10 Jan 2017)', 'keywords': 'Keywords:###Deep learning,
Reinforcement Learning, Games', 'E': 'Conflicts:###berkeley.edu,
eecs.berkeley.edu, openai.com, ugent.be', 'F': 93, 'G': '06 Feb 2017',
'decision': 'Decision:###Reject', 'J': 1458, 'K': '22 Dec 2016 10 Jan 2017',
'rate': 4, 'T': 1655}, {'title': '#Exploration: A Study of Count-Based
Exploration for Deep Reinforcement Learning | OpenReview', 'Unnamed: 1': '06 Nov
2016 (modified: 10 Jan 2017)', 'keywords': 'Keywords:###Deep learning,
Reinforcement Learning, Games', 'E': 'Conflicts:###berkeley.edu,
eecs.berkeley.edu, openai.com, ugent.be', 'F': 93, 'G': '06 Feb 2017',
'decision': 'Decision:###Reject', 'J': 1458, 'K': '19 Dec 2016 (modified: 24 Jan
2017) 10 Jan 2017 24 Jan 2017', 'rate': 6, 'T': 4910}, {'title': '#Exploration:
A Study of Count-Based Exploration for Deep Reinforcement Learning |
OpenReview', 'Unnamed: 1': '06 Nov 2016 (modified: 10 Jan 2017)', 'keywords':
'Keywords:###Deep learning, Reinforcement Learning, Games', 'E':
'Conflicts:###berkeley.edu, eecs.berkeley.edu, openai.com, ugent.be', 'F': 93,
'G': '06 Feb 2017', 'decision': 'Decision:###Reject', 'J': 1458, 'K': '17 Dec
2016 10 Jan 2017', 'rate': 7, 'T': 1427}, {'title': 'A Baseline for Detecting
Misclassified and Out-of-Distribution Examples in Neural Networks | OpenReview',
'Unnamed: 1': '05 Nov 2016 (modified: 29 Nov 2018)', 'keywords':
```

'Keywords:###Computer vision', 'E': 'Conflicts:###uchicago.edu, ttic.edu', 'F': 103, 'G': '06 Feb 2017', 'decision': 'Decision:###Accept (Poster)', 'J': 274, 'K': '20 Dec 2016 14 Jan 2017', 'rate': 6, 'T': 273}, {'title': 'A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks | OpenReview', 'Unnamed: 1': '05 Nov 2016 (modified: 29 Nov 2018)', 'keywords': 'Keywords:###Computer vision', 'E': 'Conflicts:###uchicago.edu, ttic.edu', 'F': 103, 'G': '06 Feb 2017', 'decision': 'Decision:###Accept (Poster)', 'J': 274, 'K': '20 Dec 2016 (modified: 20 Dec 2016) 14 Jan 2017', 'rate': 6, 'T': 1575}}]

{ 'columns': [' ', 'title', 'rate', 'reviewLength', 'decision', 'avgscore'], 'head': [{ ' ': 1025, 'title': '"Style" Transfer for Musical Audio Using Multiple Time-Frequency Representations | OpenReview', 'rate': 7, 'reviewLength': 4071, 'decision': 'Decision:###Reject', 'avgscore': nan}, { ' ': 1026, 'title': '"Style" Transfer for Musical Audio Using Multiple Time-Frequency Representations | OpenReview', 'rate': 6, 'reviewLength': 4451, 'decision': 'Decision:###Reject', 'avgscore': nan}, { ' ': 1027, 'title': '"Style" Transfer for Musical Audio Using Multiple Time-Frequency Representations | OpenReview', 'rate': 4, 'reviewLength': 5481, 'decision': 'Decision:###Reject', 'avgscore': nan}, { ' ': 2522, 'title': '3C-GAN: AN CONDITION-CONTEXT-COMPOSITE GENERATIVE ADVERSARIAL NETWORKS FOR GENERATING IMAGES SEPARATELY | OpenReview', 'rate': 4, 'reviewLength': 1261, 'decision': 'Decision:###Reject', 'avgscore': nan}, { ' ': 2523, 'title': '3C-GAN: AN CONDITION-CONTEXT-COMPOSITE GENERATIVE ADVERSARIAL NETWORKS FOR GENERATING IMAGES SEPARATELY | OpenReview', 'rate': 4, 'reviewLength': 5867, 'decision': 'Decision:###Reject', 'avgscore': nan}]]

{ 'columns': ['title', 'submit_date', 'decision', 'rate', 'keywords'], 'head': [{ 'title': '\$A^*\$ sampling with probability matching | OpenReview', 'submit_date': '28 Sep 2018 (modified: 21 Dec 2018)', 'decision': 'Reject', 'rate': 6, 'keywords': nan}, { 'title': '\$A^*\$ sampling with probability matching | OpenReview', 'submit_date': '28 Sep 2018 (modified: 21 Dec 2018)', 'decision': 'Reject', 'rate': 3, 'keywords': nan}, { 'title': '\$A^*\$ sampling with probability matching | OpenReview', 'submit_date': '28 Sep 2018 (modified: 21 Dec 2018)', 'decision': 'Reject', 'rate': 5, 'keywords': nan}, { 'title': '(Unconstrained) Beam Search is Sensitive to Large Search Discrepancies | OpenReview', 'submit_date': '28 Sep 2018 (modified: 21 Dec 2018)', 'decision': 'Reject', 'rate': 5, 'keywords': 'Keywords:###beam search, sequence models, search, sequence to sequence'}, { 'title': '(Unconstrained) Beam Search is Sensitive to Large Search Discrepancies | OpenReview', 'submit_date': '28 Sep 2018 (modified: 21 Dec 2018)', 'decision': 'Reject', 'rate': 7, 'keywords': 'Keywords:###beam search, sequence models, search, sequence to sequence'}}]

{ 'columns': ['title', 'publish_time', 'keywords', 'tL_DL', 'paper_decision_time', 'decision', 'paper_decision_commentlength', 'review_publish_time', 'rate', 'review_score_one', 'review_score_two', 'review_score_three', 'review_contentlength'], 'head': [{ 'title': '{COMPANYNAME}11K: An Unsupervised Representation Learning Dataset for Arrhythmia Subtype Discovery | OpenReview', 'publish_time': '26 Sep 2019 (modified: 20 Dec 2019)', 'keywords': 'Keywords:###representation learning, healthcare, medical, clinical, dataset, ecg, cardiology, heart, discovery, anomaly detection, out of distribution', 'tL_DL': 'TL;DR:###We release a dataset constructed from single-lead ECG data from 11,000 patients who were prescribed

to use the {DEVICENAME}(TM) device.', 'paper_decision_time': '20 Dec 2019 (modified: 20 Dec 2019)', 'decision': 'Reject', 'paper_decision_commentlength': 297, 'review_publish_time': '22 Oct 2019 (modified: 06 Nov 2019) 13 Nov 2019 (modified: 15 Nov 2019)', 'rate': 3, 'review_score_one': 'Review Assessment: Thoroughness In Paper Reading:###I read the paper at least twice and used my best judgement in assessing the paper.', 'review_score_two': 'Review Assessment: Checking Correctness Of Experiments:###N/A', 'review_score_three': 'Review Assessment: Checking Correctness Of Derivations And Theory:###N/A', 'review_contentlength': 1766}, {'title': '{COMPANYNAME}11K: An Unsupervised Representation Learning Dataset for Arrhythmia Subtype Discovery | OpenReview', 'publish_time': '26 Sep 2019 (modified: 20 Dec 2019)', 'keywords': 'Keywords:###representation learning, healthcare, medical, clinical, dataset, ecg, cardiology, heart, discovery, anomaly detection, out of distribution', 'tL_DL': 'TL;DR:###We release a dataset constructed from single-lead ECG data from 11,000 patients who were prescribed to use the {DEVICENAME}(TM) device.', 'paper_decision_time': '20 Dec 2019 (modified: 20 Dec 2019)', 'decision': 'Reject', 'paper_decision_commentlength': 297, 'review_publish_time': '22 Oct 2019 (modified: 15 Nov 2019) 13 Nov 2019', 'rate': 3, 'review_score_one': 'Review Assessment: Thoroughness In Paper Reading:###I read the paper thoroughly.', 'review_score_two': 'Review Assessment: Checking Correctness Of Experiments:###I carefully checked the experiments.', 'review_score_three': 'Review Assessment: Checking Correctness Of Derivations And Theory:###I assessed the sensibility of the derivations and theory.', 'review_contentlength': 4246}, {'title': '3D-SIC: 3D Semantic Instance Completion for RGB-D Scans | OpenReview', 'publish_time': '26 Sep 2019 (modified: 28 Nov 2019)', 'keywords': 'Keywords:###3d reconstruction, rgb-d scanning, 3d learning, 3d scene understanding', 'tL_DL': 'TL;DR:###From an incomplete RGB-D scan of a scene, we aim to detect the individual object instances comprising the scene and infer their complete object geometry.', 'paper_decision_time': '08 Nov 2019', 'decision': 'Reject', 'paper_decision_commentlength': 0, 'review_publish_time': '05 Nov 2019 (modified: 06 Nov 2019)', 'rate': 3, 'review_score_one': 'Review Assessment: Thoroughness In Paper Reading:###I read the paper at least twice and used my best judgement in assessing the paper.', 'review_score_two': 'Review Assessment: Checking Correctness Of Experiments:###I did not assess the experiments.', 'review_score_three': 'Review Assessment: Checking Correctness Of Derivations And Theory:###N/A', 'review_contentlength': 2850}, {'title': '3D-SIC: 3D Semantic Instance Completion for RGB-D Scans | OpenReview', 'publish_time': '26 Sep 2019 (modified: 28 Nov 2019)', 'keywords': 'Keywords:###3d reconstruction, rgb-d scanning, 3d learning, 3d scene understanding', 'tL_DL': 'TL;DR:###From an incomplete RGB-D scan of a scene, we aim to detect the individual object instances comprising the scene and infer their complete object geometry.', 'paper_decision_time': '08 Nov 2019', 'decision': 'Reject', 'paper_decision_commentlength': 0, 'review_publish_time': '28 Oct 2019 (modified: 06 Nov 2019)', 'rate': 3, 'review_score_one': 'Review Assessment: Thoroughness In Paper Reading:###I read the paper thoroughly.', 'review_score_two': 'Review Assessment: Checking Correctness Of Experiments:###I assessed the sensibility of the experiments.', 'review_score_three': 'Review Assessment: Checking Correctness Of Derivations And Theory:###N/A',

```

'review_contentlength': 3077}, {'title': '3D-SIC: 3D Semantic Instance
Completion for RGB-D Scans | OpenReview', 'publish_time': '26 Sep 2019
(modified: 28 Nov 2019)', 'keywords': 'Keywords:###3d reconstruction, rgb-d
scanning, 3d learning, 3d scene understanding', 'tL_DL': 'TL;DR:###From an
incomplete RGB-D scan of a scene, we aim to detect the individual object
instances comprising the scene and infer their complete object geometry.',
'paper_decision_time': '08 Nov 2019', 'decision': 'Reject',
'paper_decision_commentlength': 0, 'review_publish_time': '22 Oct 2019
(modified: 06 Nov 2019)', 'rate': 6, 'review_score_one': 'Review Assessment:
Thoroughness In Paper Reading:###I read the paper at least twice and used my
best judgement in assessing the paper.', 'review_score_two': 'Review Assessment:
Checking Correctness Of Experiments:###I carefully checked the experiments.',
'review_score_three': 'Review Assessment: Checking Correctness Of Derivations
And Theory:###I carefully checked the derivations and theory.',
'review_contentlength': 1780}}]
{'columns': ['rate', 'arxiv', 'keywords', 'title', 'cite', 'avgscore',
'fangcha'], 'head': [{'rate': 2.0, 'arxiv': -1, 'keywords': 'maximum mean
discrepancy, kernelized Stein Discrepancy, RKHS, two-sample test, empirical
estimator, discrete distributions', 'title': 'a generalized probability kernel
on discrete distributions and its application in two-sample test', 'cite': 0,
'avgscore': 2.0, 'fangcha': 0.5}, {'rate': 2.0, 'arxiv': -1, 'keywords':
'adversarial robustness, resisting adversarial examples', 'title': 'towards
counteracting adversarial perturbations to resist adversarial examples', 'cite':
0, 'avgscore': 2.0, 'fangcha': 0.5}, {'rate': 2.25, 'arxiv': 'failed',
'keywords': 'Distributed Machine Learning, Federated Learning, Distributed
Averaging Consensus', 'title': 'consensus driven learning', 'cite': 0,
'avgscore': 2.25, 'fangcha': 0.6875}, {'rate': 2.25, 'arxiv': -1, 'keywords':
'Graph embedding, Theory, Topology, Functional analysis', 'title': '$graph
embedding via topology and functional analysis$', 'cite': 0, 'avgscore': 2.25,
'fangcha': 0.1875}, {'rate': 2.25, 'arxiv': -1, 'keywords': 'text generation,
knowledge graph', 'title': 'ketg: a knowledge enhanced text generation
framework', 'cite': 0, 'avgscore': 2.25, 'fangcha': 0.1875}]}
{'columns': ['rate', 'decision', 'keywords', 'title', 'citenum', 'cite',
'avgscore', 'fangcha'], 'head': [{'rate': nan, 'decision': 'Reject', 'keywords':
'continuous extensions, algorithmic reasoning, set functions, machine learning,
combinatorial optimization, image classification', 'title': 'neural extensions:
training neural networks with set functions', 'citenum': -1.0, 'cite': 0,
'avgscore': 0.0, 'fangcha': 0.0}, {'rate': nan, 'decision': 'Reject',
'keywords': 'Nearest neighbor search, tree algorithms, graph cuts, random
projections', 'title': 'cluster tree for nearest neighbor search', 'citenum':
nan, 'cite': 0, 'avgscore': 0.0, 'fangcha': 0.0}, {'rate': nan, 'decision':
'Reject', 'keywords': 'minimizing parameter l2 norm, representation cost,
implicit bias', 'title': 'inductive bias of multi-channel linear convolutional
networks with bounded weight norm', 'citenum': 9.0, 'cite': 9, 'avgscore': 0.0,
'fangcha': 0.0}, {'rate': nan, 'decision': 'Reject', 'keywords': 'extensive form
games, network extensive form games, online learning, optimistic gradient
descent ascent', 'title': 'fast convergence of optimistic gradient ascent in
network zero-sum extensive form games', 'citenum': -1.0, 'cite': 0, 'avgscore':

```

```
0.0, 'fangcha': 0.0}, {'rate': nan, 'decision': 'Reject', 'keywords': 'neuron
type classification, convolutional neural network, electrophysiology', 'title':
'neocortical cell type classification from electrophysiology recordings using
deep neural networks', 'citenum': -1.0, 'cite': 0, 'avgscore': 0.0, 'fangcha':
0.0}}]
```

We have a dataset split in six sheets, each representing different years of the review process for some conference.

Sheet	Rows	Columns	Notes
Sheet1	1495	22	Many unnamed columns (e.g., 'Unnamed: 1', 'F', 'G'), likely noisy; includes abstract, decision, review, rates
Sheet2	2849	12	Some columns in Chinese (' ' = index, ' ' = variance); has derived stats like avgscore, confidence sum
Sheet3	4733	12	Includes submission dates, IDs, abstract/keywords, rates, and decisions
Sheet4	7769	19	Rich metadata (e.g., review times, reviewer level, comment length), cleaner naming
Sheet5	3457	25	Multiple reviewers' scores, citation info, and a mix of identifiers
Sheet6	2966	26	Very similar to Sheet5 but with some extra fields like 'arxiv', more structured reviewer columns

```
[6]: import json

# Create a prompt with sheet summaries
prompt = "I have an Excel file with multiple sheets, this dataset represent the
↳openreview dataset with information about the review process of some papers,
↳my objective is to extract information that could help write better papers.
↳Each sheet has the following structure:\n\n"

for name, summary in sheet_summaries.items():
    columns = ', '.join(summary['columns'])
    first_row = summary['head'][0] if summary['head'] else {}
    first_row_str = json.dumps(first_row)
    prompt += f"[Sheet: {name} | Columns: {columns} ] "

prompt += "Can you compare these sheets, highlight the differences in their
↳structures, explain the different labels, and suggest how I might align them
↳for a unified analysis? I want you to provide a deep detailed explanations
↳of the previous points"

# Get GPT-4's response
# response = client.chat.completions.create(
#     model="openai/gpt-4",
#     messages=[
#         {"role": "system", "content": "You are a data analyst."},
#         {"role": "user", "content": prompt}
```

```
# ]
# )

# Print GPT-4's response
# print(response.choices[0].message.content)
```

Output of previous request:

Let's take a look at the structure of each of the Excel sheets:

****Sheet1**** has columns such as title1, abstract, keywords, title, decision, rate, confidence, 1

****Sheet2**** has a mix of Chinese and English columns like (Number), title, confidence, rate (

****Sheet3**** has auto_increase, id, title, submit_date, decision, Confidence, rate1, rate, review

****Sheet4**** has more details about paper review process and timelines, abstract, keyword, title:

****Sheet5**** and ****Sheet6**** seem to be identical in their structure, containing detailed paper r

Now, comparing these sheets:

1. Some columns such as 'title', 'review', 'rate', 'confidence', 'decision', 'abstract', 'keyw
2. Sheet1 has multiple unnamed columns that may make merging with other sheets challenging.
3. Sheet2 has some columns in Chinese which need translation or mapping with other English col
4. Sheet2, Sheet5, and Sheet6 have columns that provide aggregate information or metrics deriv
5. Sheets 4, 5, and 6 provide more process control level data related to paper review process a
6. Sheets 5 and 6 have more citation and referencing related details like 'citenum', 'cite', 'l
7. The 'id' and 'paper_id' could communicate between different sheets for combination, if they

Based on this, here are some recommendations to align the sheets for a unified analysis:

1. Investigate the unnamed columns: Before proceeding with any data alignment, it would be wor
2. Translate the Chinese language columns: To make merging of the datasets easier, translate tl
3. Standardize column naming: You may need to consider renaming or aliasing columns that repres
4. Merge using common columns: Consider merging datasets based on common columns such as 'title
5. Handling derived metrics: Columns that contain derived information like 'avgscore' , 'fangel
6. Combining reviewing process-related data: Consider merging reviewer-related data like 'revi

1.0.1 Columns Ranking

We can explore the data and the columns to see the most meaningful ones.

High Importance (Core columns, appear in 3 sheets):

- title (6): Identifier for papers.
- decision (5): Outcome of the review process: acceptance or rejection.
- abstract (4): Insights on the content of the paper.
- review (4): Full review text.
- rate, rate1: Numerical review scores.
- confidence, avgscore, keywords: Reviewer confidence, mean score.
- keywords: another insight on the content of the paper.

Medium Importance (Moderate usage or derived metrics): * paper_id, rate0 to rate6: Breakdown of scores per reviewer. * reviewer0 to reviewer4: Individual reviewers. * cite, fangcha (variance), chairs, link1, Title1.

Low Importance / Noisy / One-off, these appear only once or are ambiguous: * Columns like e, f, g, unnamed: 1, unnamed: 11, etc. * Chinese-only: , , rate — likely derivatives of others. * Specialized metadata: tl_dl, titlelength, review_publish_time, review_score_one, arxiv. These should be discarded or deprioritized in correlation analysis unless later needed for specific modeling.

Overall, some columns names were missing or imprecise, requiring to manually explore the data to be able to rename them accordingly.

1.0.2 Notes

From sheets 1 to 4 we have a **review-per-row**, meaning that each row represents a single evaluation from a reviewer; thus, a paper is represented by more rows. For sheets 5 and 6, instead, we have a **paper-per-row** meaning that the different comments from the reviewers are in different columns of the same rows. This discrepancy between sheets might mislead the analysis, however, we decided to keep the representation as in the original data, due to time constraints. In general, **review-per-row** can be used for fine-grained analysis of individual reviewer feedback and it enables statistical summaries like average score, disagreement, or sentiment. On the other hand, **paper-per-row** is easier for summarizing overall reception and it is a cleaner format for descriptive analytics, e.g., decision outcome based on scores or comments.

Sheets 5 and 6, includes for each paper the link to the webpage describing the corresponding review process, by following the link we discovered that also the submitted paper was available. We decided to retrieve some papers to perform a **manual information extraction** via LLMs described in the last section.

2 Data Cleaning

2.1 Objective of the Work

The purpose of this cleaning procedure is to prepare a consistent and noise-free dataset for analyzing the characteristics that distinguish accepted from rejected papers on OpenReview. The ultimate goal is to identify useful patterns to improve the writing and structure of a scientific paper.

2.2 Cleaning Steps and Motivations

2.2.1 1. Column Standardization

- Column names referring to the same concept but labeled differently were unified (e.g., abstract_field → abstract, keyword_field → keywords, title1 → title).

- Purpose: ensure consistency across sheets to facilitate data merging and cross-analysis.

2.2.2 2. Review Unification

- Columns such as reviewer0, reviewer1, etc., were concatenated into a single review column.
- Purpose: provide a compact and comprehensive textual representation of the feedback received by each paper.

2.2.3 3. Handling of rate Columns

- In sheets with multiple rating columns (e.g., rate1, rate2), the average was calculated, ignoring zeros.
- Purpose: obtain a meaningful and synthetic measure of the paper’s evaluation.

2.2.4 4. Standardization of the decision Column

- Multiple decision variants (e.g., decision1, decision123) were merged into a single decision column.
- Numerical values (“0”, “1”) were converted into textual labels: “Reject” and “Accept”.
- Purpose: obtain a consistent categorical column for outcome classification.

2.2.5 5. Removal of Low-Coverage Columns

- Columns filled in less than 1% of the rows were eliminated.
- Purpose: remove noisy and uninformative dimensions for statistical modeling.

2.2.6 6. Removal of Columns Containing Links

- Columns containing URLs (e.g., http://...) were removed as they are not relevant for analysis.

2.2.7 7. Final Cleanup of Redundant Columns

- In Sheet2, the column accept was renamed to decision for semantic consistency.
- In Sheet5, the chairs column was removed.
- In all sheets, only the title column was kept; duplicates like title1 were dropped.

2.2.8 8. Basic Text Cleaning

- Basic text preprocessing was applied: converting to lowercase, removing special characters, extra spaces, and irrelevant text portions.
- Purpose: ensure uniformity in textual fields for subsequent NLP or classification tasks.

2.2.9 9. Removal of Non-Significant Columns

- Throughout the process, columns deemed irrelevant to the goal of identifying patterns useful for improving paper acceptance probability were removed.

2.3 Conclusion

This data cleaning process has transformed a heterogeneous dataset into a clean, consistent, and analytically usable version. It now enables reliable statistical or machine learning analysis to discover patterns among paper features and their acceptance likelihood. This forms a solid basis for providing actionable and data-driven advice to researchers preparing a scientific paper.

3 Knowledge Extraction from Paper Acceptance Data using BERT-based NLP Techniques

3.1 1. Introduction

The primary objective is to extract actionable insights that can guide researchers in crafting more effective paper submissions by leveraging natural language processing (NLP) techniques and machine learning-based topic modeling.

3.1.1 1.1 Dataset Overview

The dataset includes the following key fields: - **title**: Title of the paper. - **keywords**: Keywords assigned by authors. - **decision**: Outcome of the submission (“Accept” or “Reject”). - **rate / avgscore**: Numerical score given by reviewers. - **review_publish_time, paper_decision_time**: Timestamps for tracking timelines. - **review_contentlength, paper_decision_commentlength**: Length of review and final decision comments.

3.2 2. Methodology

3.2.1 2.1 Data Preprocessing

Data was extracted from Excel sheets containing multiple years of conference submissions. Each sheet was processed independently to ensure modularity and flexibility across different editions of the same conference.

Key preprocessing steps included: - Cleaning and standardizing keyword formatting (`Keywords:###BERT, model compression` → `["BERT", "model compression"]`) - Extracting clean decisions (Accept / Reject) - Computing title lengths and extracting n-grams (bigrams, trigrams)

3.2.2 2.2 Natural Language Processing with BERT

2.2.1 Topic Modeling via BERTopic To identify thematic differences between accepted and rejected papers, we employed **BERTopic**, a state-of-the-art topic modeling technique based on BERT embeddings. This method allows for: - Unsupervised clustering of titles into semantically coherent topics - Comparison of dominant topics in accepted vs. rejected papers - Insight into which themes are most associated with successful submissions

BERTopic leverages transformer-based contextual embeddings to capture nuanced semantic relationships between paper titles, offering a richer alternative to traditional methods like LDA.

2.2.2 Keyword Analysis We analyzed the frequency and distribution of keywords in accepted and rejected papers. In particular, we calculated: - Absolute frequency of each keyword - Relative lift: how much more likely a keyword appears in accepted papers compared to all papers

This allowed us to identify keywords strongly correlated with acceptance.

2.2.3 Power Words in Titles Using bigram and trigram analysis, we identified frequent combinations of words appearing in paper titles. These were compared across accepted and rejected sets to uncover linguistic patterns that may influence reviewer perception.

2.2.4 Correlation Between Review Scores and Outcomes Where available, we analyzed numerical review scores (`rate` or `avgscore`) to determine: - Average scores for accepted vs. rejected papers - Distribution of scores within each group - Threshold values that correlate with acceptance

3.3 6. Future Work

- **Sentiment and linguistic pattern analysis using LLMs:**
Feed the full review texts and final decision comments into a Large Language Model (e.g., GPT, Llama, Mistral) to extract qualitative differences between accepted and rejected papers. This could highlight:
 - Common reasons for rejection (e.g., lack of novelty, insufficient experiments)
 - Phrases or argument structures associated with acceptance
 - Tone, clarity, and persuasiveness in reviewer feedback

This would complement the current quantitative approach with deeper qualitative insights that are difficult to extract via traditional NLP methods.

- **Automated classification** of paper abstracts to predict acceptance likelihood
- Integration of **citation graphs** to assess prior work coverage
- Analysis of **reviewer engagement** through comment length and detail level
- Correlation between **reviewer agreement** and final decision outcome

```
[6]: !pip install pandas BERTopic Counter numpy
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages  
(2.2.2)
```

```
Collecting BERTopic
```

```
  Downloading bertopic-0.17.0-py3-none-any.whl.metadata (23 kB)
```

```
Collecting Counter
```

```
  Downloading Counter-1.0.0.tar.gz (5.2 kB)
```

```
  Preparing metadata (setup.py) ... done
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages  
(2.0.2)
```

```
Requirement already satisfied: python-dateutil>=2.8.2 in  
/usr/local/lib/python3.11/dist-packages (from pandas) (2.9.0.post0)
```

Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)

Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)

Requirement already satisfied: hdbscan>=0.8.29 in /usr/local/lib/python3.11/dist-packages (from BERTopic) (0.8.40)

Requirement already satisfied: plotly>=4.7.0 in /usr/local/lib/python3.11/dist-packages (from BERTopic) (5.24.1)

Requirement already satisfied: scikit-learn>=1.0 in /usr/local/lib/python3.11/dist-packages (from BERTopic) (1.6.1)

Requirement already satisfied: sentence-transformers>=0.4.1 in /usr/local/lib/python3.11/dist-packages (from BERTopic) (4.1.0)

Requirement already satisfied: tqdm>=4.41.1 in /usr/local/lib/python3.11/dist-packages (from BERTopic) (4.67.1)

Requirement already satisfied: umap-learn>=0.5.0 in /usr/local/lib/python3.11/dist-packages (from BERTopic) (0.5.7)

Requirement already satisfied: scipy>=1.0 in /usr/local/lib/python3.11/dist-packages (from hdbscan>=0.8.29->BERTopic) (1.15.3)

Requirement already satisfied: joblib>=1.0 in /usr/local/lib/python3.11/dist-packages (from hdbscan>=0.8.29->BERTopic) (1.5.0)

Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.11/dist-packages (from plotly>=4.7.0->BERTopic) (9.1.2)

Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from plotly>=4.7.0->BERTopic) (24.2)

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)

Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>=1.0->BERTopic) (3.6.0)

Requirement already satisfied: transformers<5.0.0,>=4.41.0 in /usr/local/lib/python3.11/dist-packages (from sentence-transformers>=0.4.1->BERTopic) (4.51.3)

Requirement already satisfied: torch>=1.11.0 in /usr/local/lib/python3.11/dist-packages (from sentence-transformers>=0.4.1->BERTopic) (2.6.0+cu124)

Requirement already satisfied: huggingface-hub>=0.20.0 in /usr/local/lib/python3.11/dist-packages (from sentence-transformers>=0.4.1->BERTopic) (0.31.2)

Requirement already satisfied: Pillow in /usr/local/lib/python3.11/dist-packages (from sentence-transformers>=0.4.1->BERTopic) (11.2.1)

Requirement already satisfied: typing_extensions>=4.5.0 in /usr/local/lib/python3.11/dist-packages (from sentence-transformers>=0.4.1->BERTopic) (4.13.2)

Requirement already satisfied: numba>=0.51.2 in /usr/local/lib/python3.11/dist-packages (from umap-learn>=0.5.0->BERTopic) (0.60.0)

Requirement already satisfied: pynndescent>=0.5 in /usr/local/lib/python3.11/dist-packages (from umap-learn>=0.5.0->BERTopic) (0.5.13)

Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-

packages (from huggingface-hub>=0.20.0->sentence-transformers>=0.4.1->BERTopic) (3.18.0)

Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.20.0->sentence-transformers>=0.4.1->BERTopic) (2025.3.2)

Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.20.0->sentence-transformers>=0.4.1->BERTopic) (6.0.2)

Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.20.0->sentence-transformers>=0.4.1->BERTopic) (2.32.3)

Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.11/dist-packages (from numba>=0.51.2->umap-learn>=0.5.0->BERTopic) (0.43.0)

Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic) (3.4.2)

Requirement already satisfied: Jinja2 in /usr/local/lib/python3.11/dist-packages (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic) (3.1.6)

Collecting nvidia-cuda-nvrtc-cu12==12.4.127 (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic)

Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)

Collecting nvidia-cuda-runtime-cu12==12.4.127 (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic)

Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)

Collecting nvidia-cuda-cupti-cu12==12.4.127 (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic)

Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)

Collecting nvidia-cudnn-cu12==9.1.0.70 (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic)

Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)

Collecting nvidia-cublas-cu12==12.4.5.8 (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic)

Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)

Collecting nvidia-cufft-cu12==11.2.1.3 (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic)

Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)

Collecting nvidia-curand-cu12==10.3.5.147 (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic)

Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)

Collecting nvidia-cusolver-cu12==11.6.1.9 (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic)

Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)
 Collecting nvidia-cusparselt-cu12==12.3.1.170 (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic)
 Downloading nvidia_cusparselt_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)
 Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/lib/python3.11/dist-packages (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic) (0.6.2)
 Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/python3.11/dist-packages (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic) (2.21.5)
 Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic) (12.4.127)
 Collecting nvidia-nvjitlink-cu12==12.4.127 (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic)
 Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
 Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/dist-packages (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic) (3.2.0)
 Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/dist-packages (from torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic) (1.13.1)
 Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from sympy==1.13.1->torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic) (1.3.0)
 Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers<5.0.0,>=4.41.0->sentence-transformers>=0.4.1->BERTopic) (2024.11.6)
 Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/dist-packages (from transformers<5.0.0,>=4.41.0->sentence-transformers>=0.4.1->BERTopic) (0.21.1)
 Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-packages (from transformers<5.0.0,>=4.41.0->sentence-transformers>=0.4.1->BERTopic) (0.5.3)
 Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2->torch>=1.11.0->sentence-transformers>=0.4.1->BERTopic) (3.0.2)
 Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub>=0.20.0->sentence-transformers>=0.4.1->BERTopic) (3.4.2)
 Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub>=0.20.0->sentence-transformers>=0.4.1->BERTopic) (3.10)
 Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub>=0.20.0->sentence-transformers>=0.4.1->BERTopic) (2.4.0)
 Requirement already satisfied: certifi>=2017.4.17 in

```

/usr/local/lib/python3.11/dist-packages (from requests->huggingface-
hub>=0.20.0->sentence-transformers>=0.4.1->BERTopic) (2025.4.26)
Downloading bertopic-0.17.0-py3-none-any.whl (150 kB)
150.6/150.6 kB
10.8 MB/s eta 0:00:00
Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl
(363.4 MB)
363.4/363.4 MB
4.7 MB/s eta 0:00:00
Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl (13.8 MB)
13.8/13.8 MB
102.4 MB/s eta 0:00:00
Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl (24.6 MB)
24.6/24.6 MB
87.2 MB/s eta 0:00:00
Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl (883 kB)
883.7/883.7 kB
47.8 MB/s eta 0:00:00
Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl
(664.8 MB)
664.8/664.8 MB
2.9 MB/s eta 0:00:00
Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl
(211.5 MB)
211.5/211.5 MB
5.9 MB/s eta 0:00:00
Downloading nvidia_curand_cu12-10.3.5.147-py3-none-
manylinux2014_x86_64.whl (56.3 MB)
56.3/56.3 MB
12.7 MB/s eta 0:00:00
Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-
manylinux2014_x86_64.whl (127.9 MB)
127.9/127.9 MB
7.3 MB/s eta 0:00:00
Downloading nvidia_cusparses_cu12-12.3.1.170-py3-none-
manylinux2014_x86_64.whl (207.5 MB)
207.5/207.5 MB
6.0 MB/s eta 0:00:00
Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl (21.1 MB)
21.1/21.1 MB
56.2 MB/s eta 0:00:00
Building wheels for collected packages: Counter
  Building wheel for Counter (setup.py) ... done
  Created wheel for Counter: filename=Counter-1.0.0-py3-none-any.whl size=5392

```


sha256=1b8eca98a5e1549c735471639a6390c6ca67b5df09485c985429f7e232aa8546
Stored in directory: /root/.cache/pip/wheels/08/5b/a0/8f15503db6a45a1d8747bf0f1438411cb37484ac4dfdfe6c0b
Successfully built Counter
Installing collected packages: Counter, nvidia-nvjitlink-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, nvidia-cuda-cupti-cu12, nvidia-cublas-cu12, nvidia-cusparse-cu12, nvidia-cudnn-cu12, nvidia-cusolver-cu12, BERTopic
Attempting uninstall: nvidia-nvjitlink-cu12
Found existing installation: nvidia-nvjitlink-cu12 12.5.82
Uninstalling nvidia-nvjitlink-cu12-12.5.82:
Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
Attempting uninstall: nvidia-curand-cu12
Found existing installation: nvidia-curand-cu12 10.3.6.82
Uninstalling nvidia-curand-cu12-10.3.6.82:
Successfully uninstalled nvidia-curand-cu12-10.3.6.82
Attempting uninstall: nvidia-cufft-cu12
Found existing installation: nvidia-cufft-cu12 11.2.3.61
Uninstalling nvidia-cufft-cu12-11.2.3.61:
Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
Attempting uninstall: nvidia-cuda-runtime-cu12
Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
Attempting uninstall: nvidia-cuda-nvrtc-cu12
Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
Attempting uninstall: nvidia-cuda-cupti-cu12
Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
Attempting uninstall: nvidia-cublas-cu12
Found existing installation: nvidia-cublas-cu12 12.5.3.2
Uninstalling nvidia-cublas-cu12-12.5.3.2:
Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
Attempting uninstall: nvidia-cusparse-cu12
Found existing installation: nvidia-cusparse-cu12 12.5.1.3
Uninstalling nvidia-cusparse-cu12-12.5.1.3:
Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
Attempting uninstall: nvidia-cudnn-cu12
Found existing installation: nvidia-cudnn-cu12 9.3.0.75
Uninstalling nvidia-cudnn-cu12-9.3.0.75:
Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
Attempting uninstall: nvidia-cusolver-cu12
Found existing installation: nvidia-cusolver-cu12 11.6.3.83
Uninstalling nvidia-cusolver-cu12-11.6.3.83:
Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83

Successfully installed BERTopic-0.17.0 Counter-1.0.0 nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidia-cuda-runtime-cu12-12.4.127 nvidia-cudnn-cu12-9.1.0.70 nvidia-cufft-cu12-11.2.1.3 nvidia-curand-cu12-10.3.5.147 nvidia-cusolver-cu12-11.6.1.9 nvidia-cuspars-cu12-12.3.1.170 nvidia-nvjitlink-cu12-12.4.127

```
[1]: import pandas as pd
from bertopic import BERTopic
from collections import Counter
import numpy as np
import re

# Extract n-grams (bigrams/trigrams) from titles
def extract_ngrams(texts, n=2):
    tokens = [re.split(r'\s+', t.lower()) for t in texts]
    ngrams = []
    for tok in tokens:
        for i in range(len(tok)-n+1):
            ngrams.append(tuple(tok[i:i+n]))
    return Counter(ngrams)

# Function to load all sheets
def load_all_sheets(file_path):
    xls = pd.ExcelFile(file_path, engine='openpyxl')
    all_data = []
    for sheet in xls.sheet_names:
        df = pd.read_excel(xls, sheet_name=sheet)
        if 'title' in df.columns and 'decision' in df.columns:
            df['sheet'] = sheet # Add sheet name for traceability
            all_data.append(df)
    return pd.concat(all_data, ignore_index=True)

# Function to clean decision values ('Accept' or 'Reject')
def clean_decision(decision):
    if isinstance(decision, str):
        if 'Accept' in decision:
            return 'Accept'
        elif 'Reject' in decision:
            return 'Reject'
    return None

# Function to extract keywords
def extract_keywords(keywords_str):
    if isinstance(keywords_str, str):
        return [kw.strip() for kw in keywords_str.split(',') if kw.strip()]
    return []
```

```

# Function to count words in title
def count_words(title):
    return len(str(title).split())

# Function to run global topic modeling
def run_topic_modeling(titles):
    print("\nRunning Global Topic Modeling...")
    vectorizer = CountVectorizer(ngram_range=(1, 2), stop_words='english')
    model = BERTopic(language="english", vectorizer_model=vectorizer,
    ↪min_topic_size=10, verbose=False)
    topics, probs = model.fit_transform(titles)
    print(model.get_topic_info().head(10))
    return model

# Function to count global keywords
def count_global_keywords(df, col='keywords'):
    all_keywords = [item for sublist in df[col] for item in sublist]
    return Counter(all_keywords)

# Helper function to avoid KeyError
def safe_explode(df, col):
    """Internal function to handle optional columns"""
    if col in df.columns:
        return df.explode(col)[col].dropna().tolist()
    else:
        return []

# 1. Prepare data from a sheet
def prepare_sheet_data(df_sheet):
    print("Preparing sheet data...")
    optional_cols = ['title', 'keywords', 'decision', 'rate', 'publish_time',
    ↪'review_contentlength']
    relevant_cols = [col for col in optional_cols if col in df_sheet.columns]
    if 'title' not in df_sheet.columns:
        print("Warning: Required column 'title' is missing. Cannot proceed.")
        return None
    df = df_sheet[relevant_cols].copy()
    # Clean keywords
    if 'keywords' in df.columns:
        df['keywords'] = df['keywords'].astype(str).str.replace('Keywords:###',
    ↪',', regex=False).str.split(', ')
        df['keywords'] = df['keywords'].apply(lambda x: [kw.strip() for kw in x
    ↪if isinstance(kw, str) and kw.strip()] if isinstance(x, list) else [])
    # Create decision_clean only if decision column exists
    if 'decision' in df.columns:
        df['decision_clean'] = df['decision'].astype(str).str.
    ↪extract('(Accept|Reject)')

```

```

else:
    df['decision_clean'] = None
    print("Warning: Column 'decision' not found. Accept/Reject analysis_
↳skipped.")
    print(f>Data prepared: {len(df)} valid rows.")
    return df

# 2. Comparative analysis between accepted and rejected papers
def analyze_accepted_vs_rejected(df_prepared, sheet_name, summary_data):
    print(f"\nCOMPARATIVE ANALYSIS - SHEET '{sheet_name}'")
    has_decision = 'decision_clean' in df_prepared.columns and_
↳df_prepared['decision_clean'].notna().any()
    if not has_decision:
        print("No valid decisions ('Accept'/'Reject') found. Skipping Accept/
↳Reject analysis.")
        accepted = pd.DataFrame()
        rejected = pd.DataFrame()
    else:
        accepted = df_prepared[df_prepared['decision_clean'] == 'Accept']
        rejected = df_prepared[df_prepared['decision_clean'] == 'Reject']
        print(f>Total papers: {len(df_prepared)}")
        print(f>Accepted: {len(accepted)} ({len(accepted)/len(df_prepared):.
↳1%})")
        print(f>Rejected: {len(rejected)} ({len(rejected)/len(df_prepared):.
↳1%})")

    # Initialize dictionary for this sheet
    summary_data[sheet_name] = {
        "total": len(df_prepared),
        "accepted_count": len(accepted),
        "rejected_count": len(rejected),
        "all_keywords": [],
        "accepted_keywords": [],
        "rejected_keywords": [],
        "title_lengths": [],
        "accepted_title_lengths": [],
        "rejected_title_lengths": [],
        "titles": [],
        "accepted_titles": [],
        "rejected_titles": [],
        "rates": [],
        "accepted_rates": [],
        "rejected_rates": [],
        "review_lengths": [],
        "accepted_review_lengths": [],
        "rejected_review_lengths": [],
        "submit_dates": [],

```

```

        "accepted_submit_dates": [],
        "rejected_submit_dates": [],
    }

    # Keyword Analysis
    if 'keywords' in df_prepared.columns and df_prepared['keywords'].apply(len).
    ↪sum() > 0:
        all_kw = df_prepared.explode('keywords')['keywords'].value_counts()
        print("\nTop 10 Most Common Keywords:")
        print(all_kw.head(10))
        if has_decision:
            acc_kw = accepted.explode('keywords')['keywords'].value_counts().
            ↪head(10)
            rej_kw = rejected.explode('keywords')['keywords'].value_counts().
            ↪head(10)
            print("\nTop Keywords in Accepted Papers:")
            print(acc_kw)
            print("\nTop Keywords in Rejected Papers:")
            print(rej_kw)
            summary_data[sheet_name]["all_keywords"] = safe_explode(df_prepared,
            ↪'keywords')
            summary_data[sheet_name]["accepted_keywords"] = safe_explode(accepted,
            ↪'keywords')
            summary_data[sheet_name]["rejected_keywords"] = safe_explode(rejected,
            ↪'keywords')
        else:
            print("\nNo keywords available for analysis.")
            summary_data[sheet_name]["all_keywords"] = []
            summary_data[sheet_name]["accepted_keywords"] = []
            summary_data[sheet_name]["rejected_keywords"] = []

    # Title Length Analysis
    def title_length(title):
        return len(str(title).split())

    df_prepared['title_len'] = df_prepared['title'].apply(title_length)
    mean_len_all = df_prepared['title_len'].mean()
    print(f"\nAverage number of words in titles: Total={mean_len_all:.1f}")
    summary_data[sheet_name]["title_lengths"] = df_prepared['title_len'].
    ↪tolist()
    summary_data[sheet_name]["titles"] = df_prepared['title'].astype(str).
    ↪dropna().tolist()

    if has_decision:
        mean_len_accept = df_prepared.loc[df_prepared['decision_clean'] ==
        ↪'Accept', 'title_len'].mean()

```

```

        mean_len_reject = df_prepared.loc[df_prepared['decision_clean'] == 'Reject', 'title_len'].mean()
        print(f"    Accepted={mean_len_accept:.1f}, Rejected={mean_len_reject:.1f}")

        summary_data[sheet_name]["accepted_title_lengths"] = df_prepared.loc[df_prepared['decision_clean'] == 'Accept', 'title_len'].tolist()
        summary_data[sheet_name]["rejected_title_lengths"] = df_prepared.loc[df_prepared['decision_clean'] == 'Reject', 'title_len'].tolist()
        summary_data[sheet_name]["accepted_titles"] = accepted['title'].astype(str).dropna().tolist()
        summary_data[sheet_name]["rejected_titles"] = rejected['title'].astype(str).dropna().tolist()

    # Average Rating Analysis
    if 'rate' in df_prepared.columns:
        df_prepared['rate'] = pd.to_numeric(df_prepared['rate'], errors='coerce')
        mean_rate_all = df_prepared['rate'].mean()
        print(f"\nAverage rating: Total={mean_rate_all:.1f}")
        summary_data[sheet_name]["rates"] = df_prepared['rate'].dropna().tolist()

        if has_decision:
            mean_rate_accept = df_prepared.loc[df_prepared['decision_clean'] == 'Accept', 'rate'].mean()
            mean_rate_reject = df_prepared.loc[df_prepared['decision_clean'] == 'Reject', 'rate'].mean()
            print(f"    Accepted={mean_rate_accept:.1f}, Rejected={mean_rate_reject:.1f}")
            summary_data[sheet_name]["accepted_rates"] = df_prepared.loc[df_prepared['decision_clean'] == 'Accept', 'rate'].dropna().tolist()
            summary_data[sheet_name]["rejected_rates"] = df_prepared.loc[df_prepared['decision_clean'] == 'Reject', 'rate'].dropna().tolist()
        else:
            print("\nColumn 'rate' not present. Rating analysis skipped.")

    # Review content length analysis
    if 'review_contentlength' in df_prepared.columns:
        df_prepared['review_contentlength'] = pd.to_numeric(df_prepared['review_contentlength'], errors='coerce')
        summary_data[sheet_name]["review_lengths"] = df_prepared['review_contentlength'].dropna().tolist()

        if has_decision:
            summary_data[sheet_name]["accepted_review_lengths"] = df_prepared.loc[df_prepared['decision_clean'] == 'Accept', 'review_contentlength'].dropna().tolist()

```

```

        summary_data[sheet_name]["rejected_review_lengths"] = df_prepared.
↳loc[df_prepared['decision_clean'] == 'Reject', 'review_contentlength'].
↳dropna().tolist()

    # Submission date analysis
    if 'publish_time' in df_prepared.columns:
        df_prepared['submit_date'] = pd.to_datetime(df_prepared['publish_time'].
↳str.split(' ').str[0], errors='coerce')
        summary_data[sheet_name]["submit_dates"] = df_prepared['submit_date'].
↳dropna().tolist()
        if has_decision:
            summary_data[sheet_name]["accepted_submit_dates"] = df_prepared.
↳loc[df_prepared['decision_clean'] == 'Accept', 'submit_date'].dropna().
↳tolist()
            summary_data[sheet_name]["rejected_submit_dates"] = df_prepared.
↳loc[df_prepared['decision_clean'] == 'Reject', 'submit_date'].dropna().
↳tolist()

    return accepted, rejected

# 3. Topic Modeling on Titles
def topic_modeling_titles_for_sheet(accepted, rejected, sheet_name):
    print(f"\nTopic Modeling on Titles - SHEET '{sheet_name}'")
    min_docs_for_topic_modeling = 5

    def run_topic_model(titles, label):
        if len(titles) < min_docs_for_topic_modeling:
            print(f"Fewer than {min_docs_for_topic_modeling} titles for {label}.
↳ Skipped.")
            return
        try:
            model = BERTopic(language="english",
↳min_topic_size=min_docs_for_topic_modeling, verbose=False)
            topics, _ = model.fit_transform(titles)
            info = model.get_topic_info().head(6)
            print(f"\nTop Topics - {label}:")
            print(info)
        except Exception as e:
            print(f"Error in topic modeling for {label}: {e}")

    if not accepted.empty and 'title' in accepted.columns:
        titles = accepted['title'].astype(str).dropna().tolist()
        run_topic_model(titles, "Accepted")
    if not rejected.empty and 'title' in rejected.columns:
        titles = rejected['title'].astype(str).dropna().tolist()
        run_topic_model(titles, "Rejected")

```

```

# 4. Global Aggregated Analysis
def global_analysis(summary_data):
    print("\nGlobal Aggregated Analysis Across All Sheets")
    all_global_kw = []
    acc_global_kw = []
    rej_global_kw = []
    all_global_titles = []
    acc_global_titles = []
    rej_global_titles = []
    all_global_rates = []
    acc_global_rates = []
    rej_global_rates = []
    title_lengths_all = []
    title_lengths_acc = []
    title_lengths_rej = []
    review_lengths_all = []
    review_lengths_acc = []
    review_lengths_rej = []
    submit_dates_all = []
    submit_dates_acc = []
    submit_dates_rej = []

    for sheet in summary_data:
        data = summary_data[sheet]
        all_global_kw.extend(data["all_keywords"])
        acc_global_kw.extend(data["accepted_keywords"])
        rej_global_kw.extend(data["rejected_keywords"])
        all_global_titles.extend(data["titles"])
        acc_global_titles.extend(data["accepted_titles"])
        rej_global_titles.extend(data["rejected_titles"])
        all_global_rates.extend(data["rates"])
        acc_global_rates.extend(data["accepted_rates"])
        rej_global_rates.extend(data["rejected_rates"])
        title_lengths_all.extend(data["title_lengths"])
        title_lengths_acc.extend(data["accepted_title_lengths"])
        title_lengths_rej.extend(data["rejected_title_lengths"])
        review_lengths_all.extend(data["review_lengths"])
        review_lengths_acc.extend(data["accepted_review_lengths"])
        review_lengths_rej.extend(data["rejected_review_lengths"])
        submit_dates_all.extend(data["submit_dates"])
        submit_dates_acc.extend(data["accepted_submit_dates"])
        submit_dates_rej.extend(data["rejected_submit_dates"])

    # Global Statistics
    print("\nGlobal Statistics:")
    print(f"Total papers: {len(all_global_titles)}")

```



```

    print(f"Average number of words in titles: {np.mean(title_lengths_all):.1f}")
    if title_lengths_acc:
        print(f"Average number of words in accepted titles: {np.mean(title_lengths_acc):.1f}")
    if title_lengths_rej:
        print(f"Average number of words in rejected titles: {np.mean(title_lengths_rej):.1f}")

    # Ratings
    if all_global_rates:
        print(f"Global average rating: {np.mean(all_global_rates):.1f}")
        if acc_global_rates:
            print(f"Average rating for accepted papers: {np.mean(acc_global_rates):.1f}")
        if rej_global_rates:
            print(f"Average rating for rejected papers: {np.mean(rej_global_rates):.1f}")

    # Global Keywords
    print("\nTop 10 Global Keywords:")
    print(Counter(all_global_kw).most_common(10))
    print("\nTop 10 Keywords in Accepted Papers:")
    print(Counter(acc_global_kw).most_common(10))
    print("\nTop 10 Keywords in Rejected Papers:")
    print(Counter(rej_global_kw).most_common(10))

    # N-gram Analysis
    print("\nMost Common Bigrams in Titles:")
    print(extract_ngrams(all_global_titles, n=2).most_common(10))
    print("\nMost Common Trigrams in Titles:")
    print(extract_ngrams(all_global_titles, n=3).most_common(10))

    # Global Topic Modeling
    print("\nGlobal Topic Modeling on All Titles:")
    if len(all_global_titles) >= 5:
        model = BERTopic(language="english", min_topic_size=5, verbose=False)
        topics, _ = model.fit_transform(all_global_titles)
        print(model.get_topic_info().head(6))
    else:
        print("Not enough titles for global topic modeling.")

    # Topic vs Decision
    if len(all_global_titles) >= 5 and len(acc_global_titles) > 0 and len(rej_global_titles) > 0:
        print("\nGlobal Topic Modeling + Topic-Decision Association:")

```

```

titles = acc_global_titles + rej_global_titles
decisions = ['accept'] * len(acc_global_titles) + ['reject'] *
↳len(rej_global_titles)

df_global = pd.DataFrame({
    'title': titles,
    'decision': decisions
})

model = BERTopic(language="english", min_topic_size=5, verbose=False)
topics, probs = model.fit_transform(df_global['title'])
df_global['topic'] = topics

topic_decisions = df_global.groupby('topic')['decision'].
↳value_counts(normalize=True).unstack(fill_value=0).sort_values(by='accept',
↳ascending=False)
print(topic_decisions)

print("\nTop Topics with Acceptance Keywords:")
for topic_id in topic_decisions.index[:5]:
    keywords = model.get_topic(topic_id)
    if isinstance(keywords, list):
        keyword_list = ", ".join([word for word, _ in keywords[:5]])
    else:
        keyword_list = "N/A"
    print(f"Topic {topic_id}: {keyword_list}")

topic_keywords = {topic: model.get_topic(topic) for topic in
↳df_global['topic'].unique()}

high_accept_topics = topic_decisions[topic_decisions['accept'] > 0.7].
↳index.tolist()
print("\nKeywords associated with high acceptance topics:")
success_keywords = []
for topic_id in high_accept_topics:
    keywords = model.get_topic(topic_id)
    if isinstance(keywords, list):
        words = [word for word, _ in keywords]
        success_keywords.extend(words)
        keyword_line = ", ".join(words[:5])
    else:
        keyword_line = "N/A"
    print(f"Topic {topic_id} → {keyword_line}")

print("\nMost frequent keywords in winning topics:")
print(Counter(success_keywords).most_common(10))

```

```

else:
    print("Not enough data for Topic vs Decision analysis.")

# Review Length Analysis
if review_lengths_all:
    print("\nReview Length Analysis:")
    print(f"- Global average: {np.mean(review_lengths_all):.1f}")
    if review_lengths_acc:
        print(f"- Accepted: {np.mean(review_lengths_acc):.1f}")
    if review_lengths_rej:
        print(f"- Rejected: {np.mean(review_lengths_rej):.1f}")

# Temporal Analysis
if submit_dates_all:
    print("\nTemporal Analysis:")
    submit_df = pd.DataFrame({'date': submit_dates_all})
    submit_df['month_year'] = pd.to_datetime(submit_df['date']).dt.
    to_period('M')
    print(submit_df['month_year'].value_counts().sort_index())
    if submit_dates_acc and submit_dates_rej:
        accept_df = pd.DataFrame({'date': submit_dates_acc, 'type':
    'accept'})
        reject_df = pd.DataFrame({'date': submit_dates_rej, 'type':
    'reject'})
        combined = pd.concat([accept_df, reject_df])
        monthly = combined.resample('M', on='date').value_counts().
    unstack(fill_value=0)
        print(monthly)

# --- Main Execution ---
if __name__ == "__main__":
    file_path = "/content/drive/MyDrive/KnowledgeDiscoveryAndPatternExtraction/
    cleaned_dataset.xlsx"
    sheet_names = None
    try:
        xls = pd.ExcelFile(file_path, engine='openpyxl')
        sheet_names = xls.sheet_names
        print(f"\nFile loaded successfully. Available sheets: {sheet_names}")
    except FileNotFoundError:
        print(f"File '{file_path}' not found.")
        exit()
    except Exception as e:
        print(f"Error opening Excel file: {e}")
        exit()

summary_data = {} # To collect aggregated data

```

```

for sheet_name in sheet_names:
    print(f"\n{'='*40}\nStarting analysis: {sheet_name}\n{'='*40}")
    df_sheet = pd.read_excel(xls, sheet_name=sheet_name)
    if df_sheet.empty:
        print(f"Sheet '{sheet_name}' is empty. Skipping.")
        continue
    df_prepared = prepare_sheet_data(df_sheet)
    if df_prepared is None or df_prepared.empty:
        print(f"No valid data for sheet '{sheet_name}'.")
        continue
    accepted, rejected = analyze_accepted_vs_rejected(df_prepared,
↪sheet_name, summary_data)
    topic_modeling_titles_for_sheet(accepted, rejected, sheet_name)
    print(f"{'='*40}\nFinished analysis: {sheet_name}\n{'='*40}")

# Final aggregated analysis
if summary_data:
    global_analysis(summary_data)
else:
    print("No useful data for aggregated analysis.")

```

File loaded successfully. Available sheets: ['Sheet1', 'Sheet2', 'Sheet3', 'Sheet4', 'Sheet6', 'Sheet5']

```

=====
Starting analysis: Sheet1
=====

```

Preparing sheet data...
Data prepared: 1495 valid rows.

COMPARATIVE ANALYSIS - SHEET 'Sheet1'
Total papers: 1495
Accepted: 607 (40.6%)
Rejected: 742 (49.6%)

Top 10 Most Common Keywords:

keywords	
Deep learning	1197
Unsupervised Learning	323
Computer vision	317
Natural language processing	315
Applications	251
Supervised Learning	233
Optimization	192
Reinforcement Learning	175
Theory	131
Transfer Learning	95

Name: count, dtype: int64

Top Keywords in Accepted Papers:

keywords

Deep learning	524
Natural language processing	137
Unsupervised Learning	123
Computer vision	99
Reinforcement Learning	95
Optimization	93
Applications	90
Supervised Learning	61
Theory	56
Transfer Learning	41

Name: count, dtype: int64

Top Keywords in Rejected Papers:

keywords

Deep learning	597
Computer vision	188
Unsupervised Learning	173
Natural language processing	163
Applications	134
Supervised Learning	122
Optimization	81
Reinforcement Learning	66
Theory	57
Transfer Learning	48

Name: count, dtype: int64

Average number of words in titles: Total=9.3

Accepted=8.8, Rejected=9.7

Average rating: Total=5.7

Accepted=6.9, Rejected=4.8

Topic Modeling on Titles - SHEET 'Sheet1'

modules.json: 0%| | 0.00/349 [00:00<?, ?B/s]

config_sentence_transformers.json: 0%| | 0.00/116 [00:00<?, ?B/s]

README.md: 0%| | 0.00/10.5k [00:00<?, ?B/s]

sentence_bert_config.json: 0%| | 0.00/53.0 [00:00<?, ?B/s]

config.json: 0%| | 0.00/612 [00:00<?, ?B/s]

Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed.
Falling back to regular HTTP download. For better performance, install the
package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`

WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance, install the package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`

```
model.safetensors:  0%|          | 0.00/90.9M [00:00<?, ?B/s]
tokenizer_config.json:  0%|          | 0.00/350 [00:00<?, ?B/s]
vocab.txt:  0%|          | 0.00/232k [00:00<?, ?B/s]
tokenizer.json:  0%|          | 0.00/466k [00:00<?, ?B/s]
special_tokens_map.json:  0%|          | 0.00/112 [00:00<?, ?B/s]
config.json:  0%|          | 0.00/190 [00:00<?, ?B/s]
```

Top Topics - Accepted:

	Topic	Count	Name \
0	-1	7	-1_residuals_fractalnet_ultradeep_without
1	0	40	0_optimization_to_imaginationbased_metacontrol
2	1	25	1_pruning_cnns_pooling_convolutional
3	2	20	2_quantization_ternary_trained_network
4	3	18	3_language_natural_level_compose
5	4	18	4_quasirecurrent_chaos_automatic_long

Representation \

0	[residuals, fractalnet, ultradeep, without, de...
1	[optimization, to, imaginationbased, metacontr...
2	[pruning, cnns, pooling, convolutional, convne...
3	[quantization, ternary, trained, network, towa...
4	[language, natural, level, compose, character,...
5	[quasirecurrent, chaos, automatic, long, extra...

Representative_Docs

0	[FractalNet: Ultra-Deep Neural Networks withou...
1	[Hyperband: Bandit-Based Configuration Evaluat...
2	[Faster CNNs with Direct Sparse Convolutions a...
3	[Trained Ternary Quantization OpenReview, Tr...
4	[Program Synthesis for Character Level Languag...
5	[Automatic Rule Extraction from Long Short Ter...

Top Topics - Rejected:

	Topic	Count	Name \
0	-1	21	-1_graph_changing_dictionary_graphs
1	0	22	0_loss_residual_empirical_surfaces
2	1	21	1_pruning_coarse_chess_concepts
3	2	19	2_recurrent_layer_connected_combining
4	3	18	3_features_fewshot_controllers_communicating
5	4	18	4_recurrent_continuous_comparators_dependencies

Representation \

- 0 [graph, changing, dictionary, graphs, adaption...
- 1 [loss, residual, empirical, surfaces, analysis...
- 2 [pruning, coarse, chess, concepts, divide, den...
- 3 [recurrent, layer, connected, combining, analy...
- 4 [features, fewshot, controllers, communicating...
- 5 [recurrent, continuous, comparators, dependenc...

Representative_Docs

- 0 [NEUROGENESIS-INSPIRED DICTIONARY LEARNING: ON...
- 1 [An Empirical Analysis of Deep Network Loss Su...
- 2 [The Incredible Shrinking Neural Network: New ...
- 3 [A Way out of the Odyssey: Analyzing and Combi...
- 4 [Communicating Hierarchical Neural Controllers...
- 5 [Dynamic Neural Turing Machine with Continuous...

=====

Finished analysis: Sheet1

=====

=====

Starting analysis: Sheet2

=====

Preparing sheet data...

Data prepared: 2849 valid rows.

COMPARATIVE ANALYSIS - SHEET 'Sheet2'

Total papers: 2849

Accepted: 1051 (36.9%)

Rejected: 1519 (53.3%)

No keywords available for analysis.

Average number of words in titles: Total=9.7

Accepted=9.4, Rejected=9.8

Average rating: Total=5.4

Accepted=6.4, Rejected=4.6

Topic Modeling on Titles - SHEET 'Sheet2'

Top Topics - Accepted:

	Topic	Count	Name \
0	-1	16	-1_examplebased_existing_insufficiency_pixelln
1	0	33	0_word_lstm_sentence_text
2	1	21	1_classification_analyze_betweenclass_rewiring
3	2	21	2_quantization_structured_variational_alternating
4	3	19	3_recurrent_skip_state_sequence

5 4 18 4_metalearning_bayes_competitive_attentive

Representation \

0 [examplebased, existing, insufficiency, pixeln...
1 [word, lstms, sentence, text, representations,...
2 [classification, analyze, betweenclass, rewiri...
3 [quantization, structured, variational, altern...
4 [recurrent, skip, state, sequence, diffusion, ...
5 [metalearning, bayes, competitive, attentive, ...

Representative_Docs

0 [On the insufficiency of existing momentum sch...
1 [Beyond Word Importance: Contextual Decomposit...
2 [An image representation based convolutional n...
3 [Variational Network Quantization | OpenReview...
4 [Initialization matters: Orthogonal Predictive...
5 [Meta-Learning and Universality: Deep Represen...

Top Topics - Rejected:

	Topic	Count	Name \
0	-1	18	-1_recommendations_optimizer_sessionbased_twol...
1	0	34	0_generative_improve_adversarial_collaborative
2	1	30	1_state_reward_robot_modelbased
3	2	28	2_perturbations_saliency_ensembles_methods
4	3	25	3_embeddings_word_connected_compute
5	4	25	4_optimizing_generative_space_latent

Representation \

0 [recommendations, optimizer, sessionbased, two...
1 [generative, improve, adversarial, collaborati...
2 [state, reward, robot, modelbased, reinforceme...
3 [perturbations, saliency, ensembles, methods, ...
4 [embeddings, word, connected, compute, alone, ...
5 [optimizing, generative, space, latent, ration...

Representative_Docs

0 [Theoretical properties of the global optimize...
1 [Image Quality Assessment Techniques Improve T...
2 [TOWARDS ROBOT VISION MODULE DEVELOPMENT WITH ...
3 [DETECTING ADVERSARIAL PERTURBATIONS WITH SALI...
4 [On the Use of Word Embeddings Alone to Repres...
5 [Optimal transport maps for distribution prese...

=====
Finished analysis: Sheet2

=====
Starting analysis: Sheet3

=====

Preparing sheet data...

Data prepared: 4733 valid rows.

COMPARATIVE ANALYSIS - SHEET 'Sheet3'

Total papers: 4733

Accepted: 1526 (32.2%)

Rejected: 3207 (67.8%)

Top 10 Most Common Keywords:

keywords

deep learning	449
nan	387
reinforcement learning	265
Deep Learning	181
Reinforcement Learning	162
optimization	118
unsupervised learning	115
generative models	111
representation learning	106
meta-learning	94

Name: count, dtype: int64

Top Keywords in Accepted Papers:

keywords

deep learning	145
reinforcement learning	105
nan	66
unsupervised learning	52
Deep Learning	51
optimization	51
generative models	49
Reinforcement Learning	46
meta-learning	43
representation learning	42

Name: count, dtype: int64

Top Keywords in Rejected Papers:

keywords

nan	321
deep learning	304
reinforcement learning	160
Deep Learning	130
Reinforcement Learning	116
optimization	67
representation learning	64
generalization	64
unsupervised learning	63

generative models 62

Name: count, dtype: int64

Average number of words in titles: Total=9.8

Accepted=9.6, Rejected=9.9

Average rating: Total=5.3

Accepted=6.6, Rejected=4.7

Topic Modeling on Titles - SHEET 'Sheet3'

Top Topics - Accepted:

	Topic	Count	Name \
0	-1	11	-1_odds_may_accuracy_be
1	0	27	0_generative_adversarial_computation_clouds
2	1	27	1_learn_bounces_distinct_dependencies
3	2	24	2_quantized_quantization_defensive_discretized
4	3	23	3_metalearning_unsupervised_parts_discovery
5	4	21	4_graph_capsule_community_line

	Representation \
0	[odds, may, accuracy, be, role, overparametriz...
1	[generative, adversarial, computation, clouds,...
2	[learn, bounces, distinct, dependencies, locat...
3	[quantized, quantization, defensive, discretiz...
4	[metalearning, unsupervised, parts, discovery,...
5	[graph, capsule, community, line, powerful, pr...

	Representative_Docs
0	[Robustness May Be at Odds with Accuracy Ope...
1	[Learning Localized Generative Models for 3D P...
2	[Generating Multiple Objects at Spatially Dist...
3	[ProxQuant: Quantized Neural Networks via Prox...
4	[Unsupervised Discovery of Parts, Structure, a...
5	[Predict then Propagate: Graph Neural Networks...

Top Topics - Rejected:

	Topic	Count	Name \
0	-1	103	-1_research_crossentropy_poor_problem
1	0	42	0_na_muss_immer_man
2	1	39	1_language_document_pretraining_auxiliary
3	2	35	2_convolutional_advertisement_backdrop_column
4	3	30	3_codebook_binding_decisions_dictionary
5	4	27	4_generative_models_counterfactuals_boxes

	Representation \
0	[research, crossentropy, poor, problem, margin...
1	[na, muss, immer, man, cakewalk, withdrawn, in...

2 [language, document, pretraining, auxiliary, m...
 3 [convolutional, advertisement, backdrop, colum...
 4 [codebook, binding, decisions, dictionary, exp...
 5 [generative, models, counterfactuals, boxes, b...

Representative_Docs

0 [Using Deep Siamese Neural Networks to Speed u...
 1 [NA | OpenReview, NA | OpenReview, NA | OpenRe...
 2 [Language Modeling Teaches You More Syntax tha...
 3 [IEA: Inner Ensemble Average within a convolut...
 4 [Pearl: Prototype lEArning via Rule Lists | Op...
 5 [Tinkering with black boxes: counterfactuals u...

Finished analysis: Sheet3

Starting analysis: Sheet4

Preparing sheet data...

Data prepared: 7769 valid rows.

COMPARATIVE ANALYSIS - SHEET 'Sheet4'

Total papers: 7769

Accepted: 2056 (26.5%)

Rejected: 5713 (73.5%)

Top 10 Most Common Keywords:

keywords

nan	867
deep learning	494
reinforcement learning	412
Deep Learning	318
Reinforcement Learning	260
representation learning	209
meta-learning	122
GAN	107
optimization	106
natural language processing	105

Name: count, dtype: int64

Top Keywords in Accepted Papers:

keywords

nan	200
deep learning	170
reinforcement learning	136
Deep Learning	78
Reinforcement Learning	69

generalization	45
optimization	44
graph neural networks	43
representation learning	41
meta-learning	40

Name: count, dtype: int64

Top Keywords in Rejected Papers:

keywords	
nan	667
deep learning	324
reinforcement learning	276
Deep Learning	240
Reinforcement Learning	191
representation learning	168
GAN	88
meta-learning	82
natural language processing	77
generative models	74

Name: count, dtype: int64

Average number of words in titles: Total=9.9
Accepted=9.9, Rejected=9.9

Average rating: Total=4.2
Accepted=6.2, Rejected=3.4

Topic Modeling on Titles - SHEET 'Sheet4'

Top Topics - Accepted:

	Topic	Count	Name \
0	-1	49	-1_use_advil_criticality_minibatches
1	0	25	0_capabilities_respect_triangle_inequality
2	1	25	1_inductive_structured_representation_graphsaint
3	2	24	2_adversarial_against_examples_robustness
4	3	22	3_knowledge_answering_question_over
5	4	21	4_languages_lottery_compositional_lifelong

Representation \

0	[use, advil, criticality, minibatches, stiefel...
1	[capabilities, respect, triangle, inequality, ...
2	[inductive, structured, representation, graphs...
3	[adversarial, against, examples, robustness, a...
4	[knowledge, answering, question, over, reasoni...
5	[languages, lottery, compositional, lifelong, ...

Representative_Docs

0	[To Relieve Your Headache of Training an MRF, ...
---	---

- 1 [An Inductive Bias for Distances: Neural Nets ...
- 2 [Inductive and Unsupervised Representation Lea...
- 3 [Fooling Detection Alone is Not Enough: Advers...
- 4 [Learning to Retrieve Reasoning Paths over Wik...
- 5 [Playing the lottery with rewards and multiple...

Top Topics - Rejected:

	Topic	Count	Name \
0	-1	224	-1_spike_states_dsvic_kolmogorov
1	0	35	0_datasets_compatibility_adverserial_corpora
2	1	34	1_recurrent_entanglement_fan_focused
3	2	31	2_rl_adaptability_acutum_if
4	3	31	3_ticket_directional_mixup_practical
5	4	29	4_automodulators_both_bivariate_doautoencoder

Representation \

- 0 [spike, states, dsvic, kolmogorov, adaptation,...
- 1 [datasets, compatibility, adverserial, corpora...
- 2 [recurrent, entanglement, fan, focused, comput...
- 3 [rl, adaptability, acutum, if, entropic, lagra...
- 4 [ticket, directional, mixup, practical, innova...
- 5 [automodulators, both, bivariate, doautoencode...

Representative_Docs

- 0 [White Box Network: Obtaining a right composit...
- 1 [Generative Adversarial Networks For Data Scar...
- 2 [Decoupling Hierarchical Recurrent Neural Netw...
- 3 [SMiRL: Surprise Minimizing RL in Entropic Env...
- 4 [Boosting Ticket: Towards Practical Pruning fo...
- 5 [Implicit ?-Jeffreys Autoencoders: Taking the ...

Finished analysis: Sheet4

Starting analysis: Sheet6

Preparing sheet data...

Warning: Column 'decision' not found. Accept/Reject analysis skipped.

Data prepared: 2966 valid rows.

COMPARATIVE ANALYSIS - SHEET 'Sheet6'

No valid decisions ('Accept'/'Reject') found. Skipping Accept/Reject analysis.

Top 10 Most Common Keywords:

keywords

nan 384

deep learning 137

reinforcement learning	114
Reinforcement Learning	85
Deep Learning	83
representation learning	80
robustness	49
Representation Learning	44
self-supervised learning	42
generalization	41

Name: count, dtype: int64

Average number of words in titles: Total=7.9

Average rating: Total=5.4

Topic Modeling on Titles - SHEET 'Sheet6'

=====

Finished analysis: Sheet6

=====

=====

Starting analysis: Sheet5

=====

Preparing sheet data...

Data prepared: 3457 valid rows.

COMPARATIVE ANALYSIS - SHEET 'Sheet5'

Total papers: 3457

Accepted: 1095 (31.7%)

Rejected: 2362 (68.3%)

Top 10 Most Common Keywords:

keywords

nan	459
reinforcement learning	118
deep learning	111
Reinforcement Learning	87
representation learning	69
Deep Learning	66
self-supervised learning	50
Graph Neural Networks	46
generalization	46
Federated Learning	46

Name: count, dtype: int64

Top Keywords in Accepted Papers:

keywords

nan	119
reinforcement learning	40

Reinforcement Learning	30
deep learning	26
representation learning	23
Deep Learning	19
Graph Neural Networks	17
Transformer	16
robustness	14
generalization	14

Name: count, dtype: int64

Top Keywords in Rejected Papers:

keywords	
nan	340
deep learning	85
reinforcement learning	78
Reinforcement Learning	57
Deep Learning	47
representation learning	46
self-supervised learning	36
Federated Learning	34
generalization	32
robustness	31

Name: count, dtype: int64

Average number of words in titles: Total=8.3
Accepted=8.3, Rejected=8.4

Average rating: Total=5.2
Accepted=6.6, Rejected=4.5

Topic Modeling on Titles - SHEET 'Sheet5'

Top Topics - Accepted:

	Topic	Count	Name \
0	-1	255	-1_of_neural_and_the
1	0	144	0_reinforcement_policy_offline_learning
2	1	62	1_adversarial_robustness_robust_attacks
3	2	60	2_graph_networks_neural_graphs
4	3	39	3_transformers_vision_transformer_token
5	4	30	4_gans_generative_synthesis_gan

Representation \

0	[of, neural, and, the, for, networks, training..
1	[reinforcement, policy, offline, learning, gam...
2	[adversarial, robustness, robust, attacks, aga...
3	[graph, networks, neural, graphs, equivariant,...
4	[transformers, vision, transformer, token, vis...
5	[gans, generative, synthesis, gan, maps, image...

```

                                Representative_Docs
0 [temporal efficient training of spiking neural...
1 [imitation learning by reinforcement learning,...
2 [towards understanding the robustness against ...
3 [space-time graph neural networks, topological...
4 [how do vision transformers work?, quadtree at...
5 [do not escape from the manifold: discovering ...

Top Topics - Rejected:
  Topic  Count                                Name \
0      -1    672                                -1_learning_data_for_of
1       0    114                                0_graph_networks_graphs_convolutional
2       1     58    1_reinforcement_exploration_distribut...
3       2     43                                2_transformers_vision_transformer_are
4       3     42                                3_language_bert_translation_models
5       4     36                                4_attention_mechanism_visual_selfattention

```

```

                                Representation \
0 [learning, data, for, of, in, and, the, with, ...
1 [graph, networks, graphs, convolutional, neura...
2 [reinforcement, exploration, distributional, m...
3 [transformers, vision, transformer, are, sharp...
4 [language, bert, translation, models, nlp, mac...
5 [attention, mechanism, visual, selfattention, ...

```

```

                                Representative_Docs
0 [domain-wise adversarial training for out-of-d...
1 [on locality in graph learning via graph neura...
2 [exploring the robustness of distributional re...
3 [polyvit: co-training vision transformers on i...
4 [dict-bert: enhancing language model pre-train...
5 [deep dynamic attention model with gate mechan...

```

=====

Finished analysis: Sheet5

=====

Global Aggregated Analysis Across All Sheets

Global Statistics:

Total papers: 23269

Average number of words in titles: 9.3

Average number of words in accepted titles: 9.4

Average number of words in rejected titles: 9.6

Global average rating: 5.0

Average rating for accepted papers: 6.5

Average rating for rejected papers: 4.1

Top 10 Global Keywords:

```
[('nan', 2097), ('Deep learning', 1359), ('deep learning', 1191),
('reinforcement learning', 909), ('Reinforcement Learning', 769), ('Deep
Learning', 648), ('representation learning', 464), ('Unsupervised Learning',
410), ('Computer vision', 345), ('Natural language processing', 321)]
```

Top 10 Keywords in Accepted Papers:

```
[('Deep learning', 548), ('nan', 385), ('deep learning', 341), ('reinforcement
learning', 281), ('Reinforcement Learning', 240), ('Deep Learning', 148),
('Natural language processing', 138), ('Unsupervised Learning', 135),
('Optimization', 121), ('representation learning', 106)]
```

Top 10 Keywords in Rejected Papers:

```
[('nan', 1328), ('Deep learning', 713), ('deep learning', 713), ('reinforcement
learning', 514), ('Reinforcement Learning', 430), ('Deep Learning', 417),
('representation learning', 278), ('Unsupervised Learning', 234), ('Computer
vision', 211), ('Natural language processing', 167)]
```

Most Common Bigrams in Titles:

```
[('|', 'openreview'), 16846), (('neural', 'networks'), 2059), (('networks',
'|'), 1896), (('learning', '|'), 1482), (('reinforcement', 'learning'), 1234),
(('learning', 'with'), 604), (('deep', 'neural'), 571), (('neural', 'network'),
570), (('learning', 'to'), 539), (('deep', 'learning'), 443)]
```

Most Common Trigrams in Titles:

```
[(('networks', '|', 'openreview'), 1896), (('learning', '|', 'openreview'),
1482), (('neural', 'networks', '|'), 994), (('reinforcement', 'learning', '|'),
523), (('deep', 'neural', 'networks'), 449), (('models', '|', 'openreview'),
408), (('deep', 'reinforcement', 'learning'), 274), (('generative',
'adversarial', 'networks'), 231), (('convolutional', 'neural', 'networks'),
218), (('graph', 'neural', 'networks'), 217)]
```

Global Topic Modeling on All Titles:

	Topic	Count	Name \
0	-1	3460	-1_advantage_normalized_relu_sampling
1	0	96	0_federated_personalized_clients_agnostic
2	1	89	1_continual_hcnet_dualnetwork_orderrobust
3	2	80	2_adaptation_domain_wildly_multitarget
4	3	78	3_architecture_search_channellevel_cnas
5	4	76	4_active_coreset_mixmatch_oracle

	Representation \
0	[advantage, normalized, relu, sampling, anomal...
1	[federated, personalized, clients, agnostic, p...
2	[continual, hcnet, dualnetwork, orderrobust, o...
3	[adaptation, domain, wildly, multitarget, self...
4	[architecture, search, channellevel, cnas, rep...
5	[active, coreset, mixmatch, oracle, introduc...

Representative_Docs

```

0 [Latent forward model for Real-time Strategy g...
1 [adaptive personalized federated learning, ada...
2 [Continual Learning via Neural Pruning | OpenR...
3 [Distribution Matching Prototypical Network fo...
4 [BETANAS: Balanced Training and selective drop...
5 [Training Data Distribution Search with Ensemb...

```

Global Topic Modeling + Topic-Decision Association:

decision accept reject

topic

1368	1.0	0.0
1354	1.0	0.0
765	1.0	0.0
1323	1.0	0.0
1321	1.0	0.0
...
886	0.0	1.0
883	0.0	1.0
882	0.0	1.0
521	0.0	1.0
1124	0.0	1.0

[1409 rows x 2 columns]

Top Topics with Acceptance Keywords:

Topic 1368: paying, twostream, bout, swim, zebrafish
 Topic 1354: monotonic, chunkwise, multihead, attention, openreview
 Topic 765: mutual, maximization, estimation, information, perspective
 Topic 1323: neurquri, inspector, answerability, requirement, reading
 Topic 1321: perexample, routing, solve, diversity, depth

Keywords associated with high acceptance topics:

Topic 1368 → paying, twostream, bout, swim, zebrafish
 Topic 1354 → monotonic, chunkwise, multihead, attention, openreview
 Topic 765 → mutual, maximization, estimation, information, perspective
 Topic 1323 → neurquri, inspector, answerability, requirement, reading
 Topic 1321 → perexample, routing, solve, diversity, depth
 Topic 1320 → holstep, logic, proving, theorem, higherorder
 Topic 1296 → normal, anomaly, projection, subspace, autoencoding
 Topic 1307 → repair, localize, program, jointly, dynamic
 Topic 1281 → pan, dilations, 2sphere, altaz, tshaped
 Topic 660 → multilingual, nonautoregressive, decoupled, translators, translation
 Topic 1294 → texttospeech, clarinet, wave, voice, scaling
 Topic 1293 → transforms, scattering, pruned, diffusion, graphs
 Topic 961 → pertensor, fixedpoint, proxquant, backpropagation, operators
 Topic 1274 → maml, rapid, reuse, effectiveness, train

Topic 971 → freeform, electron, ffjord, reversible, paths
 Topic 1140 → strokenet, artistic, painting, environment, style
 Topic 1151 → warm, sgdr, gapaware, staleness, restarts
 Topic 1204 → identifiability, transformers, verification, robustness, on
 Topic 1202 → mixout, structbert, finetune, incorporating, structures
 Topic 1159 → photo, introspective, editing, consistency, regularization
 Topic 715 → esmaml, metaoptimization, shorthorizon, hessianfree, mcmc
 Topic 1327 → speed, skimrnn, structuraljump lstm, reading, via
 Topic 1335 → encourage, eliminate, singularities, connections, skip
 Topic 1179 → stealing, queryefficient, poisoning, dnn, defenses
 Topic 1353 → capsules, dotproduct, em, routing, inverted
 Topic 1161 → carlo, monte, correlated, categorical, autoencoding
 Topic 1350 → matchlstm, pointer, comprehension, augmented, answer
 Topic 1348 → l0, regularization, sparse, through, networks
 Topic 502 → considering, programmatic, probing, trajectories, semantics
 Topic 1237 → nonlinearities, substantially, create, bad, surfaces
 Topic 621 → expectationbased, picking, go, objectives, before
 Topic 597 → recursion, combinator, lattices, programmerinterpreters, procedures
 Topic 1246 → autoaugment, local, features, improving, robust
 Topic 1086 → retinal, primate, ganglion, prosthesis, responses
 Topic 1261 → about, physicsasinversegraphics, objectoriented, physical, interactions
 Topic 1259 → approximability, vc, quantized, relu, bounds
 Topic 1257 → incomplete, factorised, misgan, discriminators, observations
 Topic 1248 → horizon, infinite, infinitehorizon, doubly, offpolicy
 Topic 1047 → white, channel, noisy, noise, analysis
 Topic 1336 → spectrum, computation, spectral, under, normalization
 Topic 1337 → autocurricula, tool, use, environments, disentangling
 Topic 1311 → deployment, specialize, each, once, bit
 Topic 857 → maxaffine, parallelizing, spline, length, twin
 Topic 1218 → explain, efficiently, execution, logic, module
 Topic 1222 → overlearning, reveals, complexvalued, attributes, obfuscation
 Topic 1229 → pointprediction, io, fitting, conservative, quantifying
 Topic 1215 → nas, Own, spare, frustratingly, hard
 Topic 1263 → stiefel, riemannian, cayley, transform, manifold
 Topic 934 → adaptively, am, inputadaptive, mad, wins
 Topic 1048 → convexcombinatorial, frankwolfe, whats, mixed, problem
 Topic 317 → coattention, dcn, coarsegrain, finegrain, multievidence
 Topic 410 → varibad, bayesadaptive, remapping, very, good
 Topic 688 → taskfree, uncertaintyguided, nonstationary, dirichlet, continual
 Topic 852 → dictionary, l4based, noodl, coefficient, dynamical
 Topic 819 → sketch, drawings, sketchode, strokebased, ssrgnns
 Topic 805 → skill, dynamicsaware, discovery, cic, lipschitzconstrained
 Topic 1016 → competition, emergent, capabilities, coordination, compounding
 Topic 1002 → reset, leave, trace, traces, backtracking
 Topic 299 → normalizers, sigma, delta, extending, defensive
 Topic 1101 → transductive, acquisition, bayes, synthetic, metalearning
 Topic 1133 → grained, nadpex, repeat, fine, repetition

Topic 1117 → spectralnet, spectral, unifying, hyperspectral, datadriven
 Topic 1083 → powersoftwo, relaxed, discretized, nonuniform, discretization
 Topic 1069 → realistic, comprehensive, compositional, referencelimited, permutation
 Topic 1089 → music, longterm, steerability, structure, generating
 Topic 1032 → points, critical, forms, pure, analytical
 Topic 1033 → deepdsl, compilationbased, compilation, domainspecific, expedited
 Topic 461 → deductive, expressions, neuralguided, evaluations, program
 Topic 1308 → prioritized, experience, replay, modelaugmented, forget
 Topic 1292 → discriminatoractorcritic, addressing, inefficiency, gener, ative
 Topic 1264 → systematic, situated, required, emergent, scan
 Topic 1193 → strategies, use, measuring, pretraining, assessment
 Topic 1352 → formal, finite, automata, languages, representing
 Topic 258 → texture, bagoflocalfeatures, imagenettrained, works, cnmbased
 Topic 269 → optimized, contextadaptive, hyperprior, drasic, compression
 Topic 675 → variations, factors, concepts, controlling, few
 Topic 609 → hedged, box, embeddings, order, prototype
 Topic 371 → lossless, hilloc, latentvariable, bits, back
 Topic 763 → decentralized, arbitrary, communication, bandwidth, communicate
 Topic 731 → corruptions, integer, common, benchmarking, issues
 Topic 336 → perturbandparse, cshapley, syntaxdirected, lshapley, structured
 Topic 424 → deephoyer, sparser, sparsity, singular, tying
 Topic 404 → skills, motivation, transitional, curricula, selfplay
 Topic 873 → koopman, operators, biological, modelbased, compositional
 Topic 472 → storedprogram, kanerva, mogrifier, multigrid, pseudoinverse
 Topic 605 → nasbench102, nasbench1shot1, scope, reproducible, extending
 Topic 1010 → distributioninterpolation, off, trade, realism, interpolation
 Topic 1025 → visceral, riskaversion, physiological, graphenhanced, misaligned
 Topic 996 → flowqa, grasping, comprehension, conversational, fusionnet
 Topic 594 → describe, programmer, scenes, learner, neurosymbolic
 Topic 573 → programs, represent, invocations, hoppity, invocmap
 Topic 581 → binarized, resnetlike, security, skip, attacking
 Topic 571 → cgans, graphic, layoutgan, layouts, wireframe
 Topic 589 → adds, backpropaganda, selforganization, gsnr, truth
 Topic 936 → unlikelihood, degeneration, curious, skipthought, advcodec
 Topic 524 → compose, finite, words, sentences, state
 Topic 538 → sequencetosequence, misesfisher, von, outputs, amortized
 Topic 119 → diet, fat, metricoptimized, thin, introspectionaccelerating
 Topic 663 → symmetrydriven, salesperson, plaplacian, condensation, traveling
 Topic 668 → applicationoriented, forgetting, backpropagated, comprehensive, during
 Topic 646 → contingencyaware, subjective, sense, open, complex
 Topic 643 → thump, rattle, swoosh, recomposing, demonstrations
 Topic 630 → accented, speech, driven, stylization, tts
 Topic 657 → collision, cleverer, transformationbased, videoflow, video
 Topic 205 → protein, cryoem, reconstructing, simulator, transferrable
 Topic 234 → grounded, babyai, domqnet, turker, mechanical
 Topic 433 → compressed, sensing, nonconvolutional, untrained, subsampling

Topic 1112 → theoretic, maxmig, crowds, cotraining, joint
Topic 108 → scratch, densitydiversity, n2n, lossconditional, once
Topic 259 → nesterov, scale, reject, illegal, intriguing
Topic 247 → pacbayesian, noiseresilience, bounds, nonvacuous, spectrallynormalized

Most frequent keywords in winning topics:
[('', 16), ('openreview', 11), ('neural', 6), ('machine', 5), ('sequence', 5), ('with', 5), ('attention', 4), ('adversarial', 4), ('generative', 4), ('generalization', 4)]

Review Length Analysis:
- Global average: 2561.2
- Accepted: 2537.9
- Rejected: 2569.6

4 Paper Acceptance Analysis

Objective: Identify patterns and best practices from a dataset of accepted and rejected scientific papers to provide actionable insights for improving paper quality and increasing acceptance rates.

4.1 1. Dataset Overview

Metric	Value
Total Papers Analyzed	23,269
Accepted Papers	7,435 (31.9%)
Rejected Papers	15,834 (68.1%)
Average Word Count in Titles	9.3 words
- Accepted	9.4 words
- Rejected	9.6 words
Average Rating	5.0
- Accepted	6.5
- Rejected	4.1

This section provides a general summary of the dataset, including the total number of papers analyzed, acceptance and rejection rates, average title length, and average ratings. It gives context and scale to the analysis.

4.2 2. Key Insights: What Works, What Doesn't

4.2.1 Topics Associated with High Acceptance Rates

The following topics show **100% acceptance rates** based on topic modeling:

Topic ID	Keywords
616	recursion, combinator, lattices, programmerinterpreters
1320	explain, execution, logic, module
673	subset, homotopy, correspondence, transport
1317	warm, sgdr, gapaware, staleness
1325	retinal, prosthesis, ganglion, primate

4.2.2 Keywords Found in Accepted Papers

Keyword	Frequency
Deep learning	548
Reinforcement Learning	281
Natural Language Processing	138
Unsupervised Learning	135
Optimization	121
Representation Learning	106

4.2.3 Keywords Commonly Found in Rejected Papers

Keyword	Frequency
Deep learning	713
deep learning	713
representation learning	278
self-supervised learning	36

Highlights the topics and keywords most strongly associated with accepted or rejected papers. This helps identify trending and well-received research areas versus overused or less-impactful terms.

4.3 3. Title Structure & Length

- **Average title length:** 9.3 words
- **Accepted papers:** ~9.4 words
- **Rejected papers:** ~9.6 words

4.3.1 Most Common N-Grams in Titles:

- **Bigrams:** neural networks, learning with, deep learning
- **Trigrams:** deep neural networks, generative adversarial networks, graph neural networks

Analyzes the average length and common linguistic patterns (bigrams and trigrams) found in paper titles. Offers guidance on how to craft effective, standards-aligned titles that resonate with reviewers.

4.4 4. Quality = Acceptance

There is a **strong correlation between high ratings and acceptance**: - **Accepted papers average rating: 6.5** - **Rejected papers average rating: 4.1**

Demonstrates the strong correlation between higher average ratings and paper acceptance. Emphasizes that perceived quality — through clarity, rigor, and relevance — significantly impacts the likelihood of acceptance.

4.5 5. Review Length

Category	Average Length (characters)
Global	2561
Accepted	2538
Rejected	2569

Compares the average length of peer reviews across accepted and rejected papers. Suggests that review length is not a decisive factor in acceptance, but rather the content and strength of the feedback matters more.

5 Pairwise comparison of papers through LLMs

```
[7]: def clean_titles_of_openreview(sheets):  
    """  
    Removes the word 'openreview' from the 'title' column in all sheets.  
  
    Parameters:  
        sheets (dict): Dictionary of DataFrames keyed by sheet name.  
  
    Returns:  
        dict: Cleaned sheets with updated titles.  
    """  
    cleaned = {}  
  
    for sheet_name, df in sheets.items():  
        df = df.copy()  
        df.columns = df.columns.str.lower().str.strip()  
  
        if 'title' in df.columns:
```

```

        df['title'] = df['title'].astype(str).str.replace(r'openreview', ' ',
↪'', case=False, regex=True).str.strip()

        cleaned[sheet_name] = df

    return cleaned

sheets = clean_titles_of_openreview(sheets)

```

In this section, we explore how large language models (LLMs) can be used to manually extract useful insights from research papers. Specifically, we focus on both accepted and rejected papers from the dataset to identify patterns or information that could contribute to writing stronger submissions.

```

[8]: from collections import defaultdict

def find_titles_in_multiple_sheets(sheets):
    """
    Identifies paper titles that appear in more than one sheet.

    Parameters:
        sheets (dict): Dictionary of DataFrames keyed by sheet name.

    Returns:
        dict: Dictionary mapping each duplicate title to the list of sheet_
↪names it appears in.
    """
    title_map = defaultdict(set)

    for sheet_name, df in sheets.items():
        df.columns = df.columns.str.lower().str.strip()
        if 'title' not in df.columns:
            continue

        titles = df['title'].dropna().str.strip().unique()
        for title in titles:
            title_map[title].add(sheet_name)

    duplicates = {title: sorted(list(sheet_names)) for title, sheet_names in_
↪title_map.items() if len(sheet_names) > 1}
    return duplicates

# Example usage:
duplicate_titles = find_titles_in_multiple_sheets(sheets)
print(f"Found {len(duplicate_titles)} titles in multiple sheets:")
for title, sheet_names in duplicate_titles.items():
    print(f"- '{title}' appears in: {sheet_names}")

```

Found 43 titles in multiple sheets:

- 'Data augmentation instead of explicit regularization |' appears in: ['Sheet2', 'Sheet4']
- 'Efficient Exploration through Bayesian Deep Q-Networks |' appears in: ['Sheet2', 'Sheet3']
- 'Graph2Seq: Scalable Learning Dynamics for Graphs |' appears in: ['Sheet2', 'Sheet3']
- 'Massively Parallel Hyperparameter Tuning |' appears in: ['Sheet2', 'Sheet3']
- 'Open Loop Hyperparameter Optimization and Determinantal Point Processes |' appears in: ['Sheet2', 'Sheet3']
- 'Value Propagation Networks |' appears in: ['Sheet2', 'Sheet3']
- 'withdrawn |' appears in: ['Sheet2', 'Sheet3']
- 'Dataset Distillation |' appears in: ['Sheet3', 'Sheet4']
- 'Deep Imitative Models for Flexible Inference, Planning, and Control |' appears in: ['Sheet3', 'Sheet4']
- 'Double Neural Counterfactual Regret Minimization |' appears in: ['Sheet3', 'Sheet4']
- 'Pushing the bounds of dropout |' appears in: ['Sheet3', 'Sheet4']
- 'Unified recurrent network for many feature types |' appears in: ['Sheet3', 'Sheet4']
- 'Unsupervised Meta-Learning for Reinforcement Learning |' appears in: ['Sheet3', 'Sheet4']
- 'Weakly-supervised Knowledge Graph Alignment with Adversarial Learning |' appears in: ['Sheet3', 'Sheet4']
- 'max-affine spline insights into deep network pruning' appears in: ['Sheet5', 'Sheet6']
- 'class-weighted evaluation metrics for imbalanced data classification' appears in: ['Sheet5', 'Sheet6']
- 'autonomous learning of object-centric abstractions for high-level planning' appears in: ['Sheet5', 'Sheet6']
- 'towards understanding label smoothing' appears in: ['Sheet5', 'Sheet6']
- 'apollo: an adaptive parameter-wised diagonal quasi-newton method for nonconvex stochastic optimization' appears in: ['Sheet5', 'Sheet6']
- 'poisoned classifiers are not only backdoored, they are fundamentally broken' appears in: ['Sheet5', 'Sheet6']
- 'learning to actively learn: a robust approach' appears in: ['Sheet5', 'Sheet6']
- 'on the certified robustness for ensemble models and beyond' appears in: ['Sheet5', 'Sheet6']
- 'ariel: volume coding for sentence generation comparisons' appears in: ['Sheet5', 'Sheet6']
- 'novel policy seeking with constrained optimization' appears in: ['Sheet5', 'Sheet6']
- 'almost tight l0-norm certified robustness of top-k predictions against adversarial perturbations' appears in: ['Sheet5', 'Sheet6']
- 'is deeper better? it depends on locality of relevant features' appears in: ['Sheet5', 'Sheet6']
- 'learning to solve multi-robot task allocation with a covariant-attention based neural architecture' appears in: ['Sheet5', 'Sheet6']

- 'learning-augmented sketches for hessians' appears in: ['Sheet5', 'Sheet6']
- 'stability analysis of sgd through the normalized loss function' appears in: ['Sheet5', 'Sheet6']
- 'optimization variance: exploring generalization properties of dnns' appears in: ['Sheet5', 'Sheet6']
- 'concentric spherical gnn for 3d representation learning' appears in: ['Sheet5', 'Sheet6']
- 'l2e: learning to exploit your opponent' appears in: ['Sheet5', 'Sheet6']
- 'classify and generate reciprocally: simultaneous positive-unlabelled learning and conditional generation with extra data' appears in: ['Sheet5', 'Sheet6']
- 'augmented sliced wasserstein distances' appears in: ['Sheet5', 'Sheet6']
- 'mqtransformer: multi-horizon forecasts with context dependent and feedback-aware attention' appears in: ['Sheet5', 'Sheet6']
- 'wiring up vision: minimizing supervised synaptic updates needed to produce a primate ventral stream' appears in: ['Sheet5', 'Sheet6']
- 'neurosymbolic deep generative models for sequence data with relational constraints' appears in: ['Sheet5', 'Sheet6']
- 'direct evolutionary optimization of variational autoencoders with binary latents' appears in: ['Sheet5', 'Sheet6']
- 'relational learning with variational bayes' appears in: ['Sheet5', 'Sheet6']
- 'bridging the gap: providing post-hoc symbolic explanations for sequential decision-making problems with inscrutable representations' appears in: ['Sheet5', 'Sheet6']
- 'open-world semi-supervised learning' appears in: ['Sheet5', 'Sheet6']
- 'align-rudder: learning from few demonstrations by reward redistribution' appears in: ['Sheet5', 'Sheet6']
- 'information condensing active learning' appears in: ['Sheet5', 'Sheet6']

Since only sheet 5 and 6 has links to the submitted paper we can just focus on those:

```
[9]: def find_common_titles_between_sheets(sheets, sheet1_name, sheet2_name):
    """
    Finds papers with the same title in two specified sheets.

    Parameters:
        sheets (dict): Dictionary of DataFrames keyed by sheet name.
        sheet1_name (str): Name of the first sheet.
        sheet2_name (str): Name of the second sheet.

    Returns:
        list: List of titles common to both sheets.
    """
    df1 = sheets[sheet1_name].copy()
    df2 = sheets[sheet2_name].copy()

    df1.columns = df1.columns.str.lower().str.strip()
    df2.columns = df2.columns.str.lower().str.strip()
```

```

titles1 = df1['title'].dropna().str.strip().unique()
titles2 = df2['title'].dropna().str.strip().unique()

common_titles = list(set(titles1).intersection(titles2))
return common_titles

# Example usage:
common_titles = find_common_titles_between_sheets(sheets, 'Sheet5', 'Sheet6')
print(f"Common titles between Sheet5 and Sheet6: {len(common_titles)}")
for title in common_titles:
    print("-", title)

```

Common titles between Sheet5 and Sheet6: 29

- concentric spherical gnn for 3d representation learning
- direct evolutionary optimization of variational autoencoders with binary latents
- poisoned classifiers are not only backdoored, they are fundamentally broken
- learning to actively learn: a robust approach
- max-affine spline insights into deep network pruning
- almost tight l0-norm certified robustness of top-k predictions against adversarial perturbations
- is deeper better? it depends on locality of relevant features
- towards understanding label smoothing
- bridging the gap: providing post-hoc symbolic explanations for sequential decision-making problems with inscrutable representations
- align-rudder: learning from few demonstrations by reward redistribution
- mqtransformer: multi-horizon forecasts with context dependent and feedback-aware attention
- novel policy seeking with constrained optimization
- learning to solve multi-robot task allocation with a covariant-attention based neural architecture
- information condensing active learning
- classify and generate reciprocally: simultaneous positive-unlabelled learning and conditional generation with extra data
- wiring up vision: minimizing supervised synaptic updates needed to produce a primate ventral stream
- stability analysis of sgd through the normalized loss function
- autonomous learning of object-centric abstractions for high-level planning
- l2e: learning to exploit your opponent
- apollo: an adaptive parameter-wised diagonal quasi-newton method for nonconvex stochastic optimization
- on the certified robustness for ensemble models and beyond
- relational learning with variational bayes
- ariel: volume coding for sentence generation comparisons
- optimization variance: exploring generalization properties of dnns
- neurosymbolic deep generative models for sequence data with relational constraints
- learning-augmented sketches for hessians

- class-weighted evaluation metrics for imbalanced data classification
- augmented sliced wasserstein distances
- open-world semi-supervised learning

```
[10]: def extract_title_info_from_sheets(sheets, titles):
    """
    For each given title, find and collect its rows from all sheets where it
    appears.

    Parameters:
        sheets (dict): Dictionary of DataFrames keyed by sheet name.
        titles (list or set): List of titles to search for.

    Returns:
        dict: Dictionary where keys are titles and values are lists of
        (sheet_name, DataFrame) pairs.
    """
    title_info = {}
    normalized_titles = {t.strip().lower(): t for t in titles} # Normalize for
    matching, preserve original

    for norm_title, original_title in normalized_titles.items():
        matching_entries = []
        for sheet_name, df in sheets.items():
            df.columns = df.columns.str.lower().str.strip()
            if 'title' not in df.columns:
                continue

            # Safe matching without index misalignment
            mask = df['title'].astype(str).str.strip().str.lower() == norm_title
            matches = df[mask]

            if not matches.empty:
                matching_entries.append((sheet_name, matches))

        if matching_entries:
            title_info[original_title] = matching_entries

    return title_info

# Example usage:
# titles = list of duplicate paper titles
info_by_title = extract_title_info_from_sheets(sheets, common_titles)
```

Let's filter on the accepted papers

```

[11]: import numpy as np

def is_accepted(decision):
    """
    Determines if a decision value indicates acceptance.
    Accepts variants like strings ("accept", "Accept (Poster)") or numbers (1, 0, ...,
    np.int64(1)).
    """
    if isinstance(decision, (int, np.integer)):
        return decision == 1
    elif isinstance(decision, str):
        return 'accept' in decision.lower()
    return False

def filter_accepted_papers_with_sheets(info_by_title):
    """
    Filters for accepted papers and lists the sheets where they were accepted.

    Parameters:
        info_by_title (dict): Dictionary of (title -> [(sheet_name, df), ...])

    Returns:
        dict: title -> list of sheet names where paper was accepted
    """
    accepted = {}

    for title, entries in info_by_title.items():
        accepted_sheets = []
        for sheet_name, df in entries:
            df.columns = df.columns.str.lower().str.strip()
            if 'decision' in df.columns:
                decisions = df['decision']
                if any(is_accepted(val) for val in decisions):
                    accepted_sheets.append(sheet_name)
        if accepted_sheets:
            accepted[title] = accepted_sheets

    return accepted

# Example usage:
accepted_titles_with_sheets = filter_accepted_papers_with_sheets(info_by_title)

# Print results
for title, sheets in accepted_titles_with_sheets.items():
    print(f" '{title}' accepted in sheets: {sheets}")

```

'almost tight l0-norm certified robustness of top-k predictions against adversarial perturbations' accepted in sheets: ['Sheet5']

'bridging the gap: providing post-hoc symbolic explanations for sequential decision-making problems with inscrutable representations' accepted in sheets: ['Sheet5']

'wiring up vision: minimizing supervised synaptic updates needed to produce a primate ventral stream' accepted in sheets: ['Sheet5']

'autonomous learning of object-centric abstractions for high-level planning' accepted in sheets: ['Sheet5']

'on the certified robustness for ensemble models and beyond' accepted in sheets: ['Sheet5']

'relational learning with variational bayes' accepted in sheets: ['Sheet5']

'augmented sliced wasserstein distances' accepted in sheets: ['Sheet5']

'open-world semi-supervised learning' accepted in sheets: ['Sheet5']

We report here all the links for simplicity:

Title	Link of Rejected (sheet 6)	Link of Accepted (sheet 5)
Relational Learning with Variational Bayes	https://openreview.net/forum?id=KjLrDj5jvN	https://openreview.net/forum?id=Az-7gJc6lpr
On the Certified Robustness for Ensemble Models and Beyond	https://openreview.net/forum?id=U0pW321hK	https://openreview.net/forum?id=tUa4REjGjTf
Autonomous Learning of Object-Centric Abstractions for High-Level Planning	https://openreview.net/forum?id=7Y6nB0v1qr	https://openreview.net/forum?id=rrWeE9ZDw__
Almost Tight L0-norm Certified Robustness of Top-k Predictions against Adversarial Perturbations	https://openreview.net/forum?id=QVpmQW073	https://openreview.net/forum?id=gJLEXY3ySpu
Bridging the Gap: Providing Post-Hoc Symbolic Explanations for Sequential Decision-Making Problems with Inscrutable Representations	https://openreview.net/forum?id=TFBmEkic7e	https://openreview.net/forum?id=o-1v9hdSult
Wiring Up Vision: Minimizing Supervised Synaptic Updates Needed to Produce a Primate Ventral Stream	https://openreview.net/forum?id=51AopRgoZa1w	https://openreview.net/forum?id=g1SzIRLQXMM
Open-world Semi-supervised Learning	https://openreview.net/forum?id=6V0pnpP5XZmc	https://openreview.net/forum?id=O-r8LOR-CCA
Augmented Sliced Wasserstein Distances	https://openreview.net/forum?id=ot9bYHwvUL	https://openreview.net/forum?id=iMqTLyfwN0O

After retrieving all the papers, we asked to the GPT-4o model to perform a **comparative analysis** between the accepted and rejected versions. In particular we asked:

1. **Abstract Comparison:** How did the abstracts differ in clarity, structure, or emphasis? Which keywords or framing might have made the accepted version stronger?

2. **Introduction & Motivation:** Was the problem stated more clearly or urgently in the accepted paper? Did the rejected version fail to establish novelty or significance?
3. **Related Work:** Was the literature coverage more thorough or recent in the accepted version? Did it better position the paper within the current research landscape?
4. **Methodology & Experiments:** Were experiments more comprehensive, reproducible, or better visualized? Did the rejected paper lack clarity or detail in implementation?
5. **Results & Discussion:** Was there more rigorous statistical analysis? Was the accepted version better at interpreting results or anticipating criticism?
6. **Conclusion:** Did it better highlight contributions or suggest compelling future work?

We have also asked to evaluate the **quantitative and structural insights**: 1. **Length Analysis:** Word counts per section: Were key sections longer or more detailed in the accepted version? 2. **Figure/Table Use:** More/better diagrams, clearer presentation of data? 3. **Citation Density** Did the accepted version cite more or higher-impact works?

Another aspect in the analysis was the **style and writing quality**: 1. **Readability Scores:** Which version reads more clearly or is closer to the target audience? 2. **Lexical Richness:** Does the accepted version use more domain-specific, impactful language? 3. **Passive vs. Active Voice:** Better engagement and clarity in accepted paper?

Finally, we asked to check **review comments** to see if it was possible to align reviewer comments with specific parts of the paper to identify: * Common criticism themes * Specific fixes that led to acceptance * Overlooked suggestions in the rejected version

Let's see an example of the output:

5.1 Model output on: Relational Learning with Variational Bayes

5.1.1 Abstract Comparison

Clarity, Structure, and Emphasis 1. **Accepted version:** More polished, better flow, and uses formal transitions between ideas. Immediately connects relational learning to human cognition and AI significance before presenting the method. 2. **Rejected version:** Conveys similar content but lacks tight phrasing and structural polish. Phrases like “we propose an unsupervised learning method...” feel less guided.

Keywords/Framing 1. **Accepted:** Introduces “unsupervised,” “probabilistic graphical model,” and “relational processing tasks” with more authoritative framing. 2. **Summary:** Accepted version is stronger due to better balance between domain motivation and technical contribution. Rejected version reads more like an early draft.

5.1.2 Introduction & Motivation

Problem Statement Clarity 1. **Accepted:** Provides a crisp, formal definition (APA citation), concrete example (circles and squares), and contrasts psychological vs. ML definitions early. 2. **Rejected:** Uses similar content but lacks clear transitions and doesn't make the gap in current ML methods explicit early on.

Novelty & Significance 1. **Accepted:** Clearly states that current ML methods often rely on absolute properties and highlights this work as filling a critical gap. 2. **Rejected:** Mentions novelty but less convincingly, and doesn't emphasize what's missing in existing approaches.

5.1.3 Related Work

Coverage and Currency 1. **Accepted:** Rich discussion of supervised relational learning, SRL methods, cognitive tasks, and unsupervised disentanglement. Uses more recent and varied citations. 2. **Rejected:** Shares many citations with the accepted version but lacks deep discussion of how existing methods fall short on decoupled relational learning.

Positioning 1. **Accepted:** Clearly positions the contribution at the intersection of PGM and unsupervised relational representation. Better differentiates from neural relational inference and disentangled learning. 2. **Rejected:** Less effective at articulating the unique niche of the work.

5.1.4 Methodology & Experiments

Clarity and Reproducibility 1. **Accepted:** Details assumptions on latent variables, ELBO training, independence constraints, and introduces RPDA (relation-preserving data augmentation) as a novel regularization. 2. **Rejected:** Contains same VRL core but fewer diagrams, lacks mutual information terms, and discusses optimization issues less rigorously.

Visualization & Experiment Depth 1. **Accepted:** Uses multiple datasets (MNIST, Omniglot, Yale Faces, RAVDESS). Includes comparisons against baselines and ablation studies. 2. **Rejected:** Only presents a synthetic MNIST-based experiment with qualitative results and limited testing.

5.1.5 Results & Discussion

Statistical Rigor 1. **Accepted:** Reports clustering accuracy, error bars, multiple trials, and ablation studies to assess RPDA and latent design choices. 2. **Rejected:** Primarily qualitative figures with minimal quantitative evaluation and no ablation analysis.

Interpretation & Anticipating Criticism 1. **Accepted:** Addresses overfitting, shortcut path issues, and provides mitigation strategies. Discusses failure cases explicitly. 2. **Rejected:** Acknowledges optimization concerns but doesn't propose solutions in depth.

5.1.6 Conclusion

Highlighting Contributions & Future Work 1. **Accepted:** Stresses contributions in disentangling relational and absolute features, real-world applications, and even philosophical boundaries (e.g., VRL's blindness to coupling). 2. **Rejected:** Lists contributions but lacks foresight and clarity in summarizing future directions.

5.1.7 Quantitative & Structural Insights

Length Analysis 1. **Accepted:** More detailed in methodology, related work, and experiments. 2. **Rejected:** Shorter in scope and omits extended discussions (e.g., RPDA or variational lower bounds).

Figure/Table Use 1. **Accepted:** Includes scatter plots, reconstructions, relational mappings, ablation diagrams, and a comprehensive result table (12+ methods). 2. **Rejected:** Fewer visuals, no baselines, and lacks ablation support.

Citation Density 1. **Accepted:** Denser and more modern references (e.g., InfoGAN, NRI, RPDA, contrastive learning). 2. **Rejected:** Sparse citations and lacks key modern references (e.g., BYOL, vector arithmetic).

5.1.8 Style and Writing Quality

Readability 1. **Accepted:** Clear, concise, and professionally written. Technical terms are introduced smoothly. 2. **Rejected:** Wordy with awkward phrasing. Needs more editorial polish.

Lexical Richness 1. **Accepted:** Uses precise domain-specific terminology (e.g., “mutual information regularization,” “non-identifiability”). 2. **Rejected:** Relies on general or repetitive language.

Passive vs. Active Voice 1. **Accepted:** More active constructions (“We propose...”, “Our contribution is...”) lead to stronger engagement. 2. **Rejected:** More passive language (“is proposed”), which weakens impact.

5.1.9 Review Comments (Inferred from Revisions)

Common Criticism Themes (Implied) - Insufficient evaluation and lack of comprehensive testing. - Missing mitigation strategies for known issues (e.g., shortcut path, coupling). - Reduced clarity and reproducibility due to sparse technical detail.

Specific Fixes That Led to Acceptance - Introduction of RPDA. - Mutual information penalty to enforce latent independence. - Expanded experiments (more datasets, baseline comparisons, quantitative metrics). - Better structured and clearer presentation.

Overlooked Suggestions in Rejected Version - Full disentanglement of latent variables (z and b), acknowledged as an ongoing limitation. - Inclusion of real-world data evaluations. - Use of baseline models for comparison.

5.2 Collected Results

We repeated the same questions for each of the pair of papers. This allowed us to extract more specific information about the reviewed papers, allowing for a fine-grain analysis.

We then instructed the model to aggregate relevant comparative studies, distill key patterns, and generate evidence-based recommendations for improving paper quality.

Based on the reviews and comparisons of accepted and rejected papers in your dataset, here are several actionable insights for improving paper acceptance odds, along with evidence from the documents:

Clear and Substantial Empirical Evidence Matters

- Accepted papers tend to provide **comprehensive empirical evaluations** with statistically significant improvements over baselines. For instance, the accepted ASWD paper reports

detailed performance benchmarks across multiple datasets and training conditions, clearly demonstrating its superiority over alternatives with metrics like FID scores and runtime evaluations

- In contrast, the rejected version of the same work showed weaker relative improvements and lacked comparable robustness in experimental validation

Tip: Include ablation studies, multiple datasets, and statistically sound comparisons to strengthen the empirical section.

Well-Articulated Problem Setting and Novelty

- Successful papers precisely define a **new, relevant problem setting**. For example, the accepted ORCA paper introduces “open-world semi-supervised learning” and distinguishes it well from adjacent concepts like zero-shot learning or open-set recognition, giving it strong novelty and motivation.
- The rejected version lacked clarity in positioning the work as a unique contribution compared to known problems like robust SSL or novel class discovery.

Tip: Clearly define the scope of the problem, contrast with existing work, and motivate its real-world importance.

Strong Theoretical Backing Helps

- Accepted papers often offer **rigorous theoretical contributions** that support empirical claims. For instance, the accepted certified robustness paper provides formal proofs and conditions for ensemble robustness using randomized smoothing.

Tip: Where applicable, include proofs, bounds, or theoretical justifications that provide general insights beyond empirical results.

Explainability and User Studies Add Value

- Papers that include human-in-the-loop evaluations or explanations tend to be well-received when appropriate. The accepted explanation paper included IRB-approved user studies with clear hypotheses and statistically supported preferences from users, boosting the paper’s credibility and impact.
- The rejected version of this work lacked similarly convincing user feedback and depth in explanation evaluation.

Tip: For human-centric tasks, complement technical work with user validation or real-world deployment data.

Review Language Signals Quality In reviews, phrases often linked to acceptance include: * “rigorous theoretical analysis” * “comprehensive experiments” * “significantly outperforms baseline methods” * “clearly defines a novel problem setting”

Whereas rejections often correlate with phrases like: * “unclear contribution” * “limited novelty” * “insufficient empirical validation” * “hard to follow motivation”

Tip: Use these phrases as a litmus test: if a reviewer might not say your work “significantly outperforms baselines” or “clearly defines a novel problem,” revisit those sections.

Several **thematic trends** in the dataset suggest that certain research topics are more prone to acceptance or rejection at venues like ICLR: * Trendy topics like adversarial robustness and SSL still need clear novelty or real-world alignment to be accepted. * Cross-disciplinary ideas—especially those blending theory, application, and user interaction—appear to be favored. * Papers without a well-differentiated contribution, even in hot topics, are more vulnerable to rejection.

*From
the
com-
para-
tive
anal-
ysis
on
the 8
pairs
of pa-
pers,
can
you
sum-
ma-
rize
the
most
sig-
nifi-
cant
im-
prove-
ments
among
them,
to
give
gen-
eral
hints
to all
au-
thors?*

Substantially
Strengthened
Em-
piri-
cal
Eval-
ua-
tion

Across
al-
most
all
pairs,
ac-
cepted
ver-
sions
intro-
duced:
*

More
datasets
(e.g.,
CI-
FAR,
Ima-
geNet,
MNIST
vari-
ants,
Minecraft).
*

More
com-
pre-
hen-
sive
met-
rics
(ac-
cu-
racy,
FID,
Wasser-
stein
dis-
tance,
ro-
bust-
ness
bounds).
*

Com-
pari-
son
against
stronger
or
more
di-

Hint:

Show

per-

for-

mance

across

mul-

tiple

set-

tings

and

iso-

late

con-

tribu-

tions.

Re-

view-

ers

re-

ward

breadth

and

re-

pro-

ducibil-

ity.

Clearer

Prob-

lem

Fram-

ing

and

Nov-

elty

Posi-

tion-

ing.

Accepted

pa-
pers
more
ex-
plic-
itly:
* De-
fined
the
prob-
lem
set-
ting
and
how
it ex-
tends
or
dif-
fers
from
prior
work.
*

Placed
their
con-
tribu-
tion
in
con-
text,
some-
times
in-
clud-
ing
new
ter-
mi-
nol-
ogy
(e.g.,
“open-
world
semi-
supervised
learn-
ing”,
“ex-
plana-

Hint:

Don't

as-

sume

read-

ers

will

infer

your

novelty—

define

it,

com-

pare

it,

and

moti-

vate

it

clearly.

Improved

The-

oret-

ical

In-

sight

or

Jus-

tifi-

ca-

tion

For

theory-

heavy

pa-

pers,

ac-

cepted

ver-

sions:

*

Added
proof
sketches
or
com-
plete
proofs.
*

Intro-
duced
tight
bounds
or
con-
di-
tions
(e.g.,
certi-
fied
ro-
bust-
ness,
injec-
tiv-
ity).
*

Linked
the-
ory
with
de-
sign
choices
(e.g.,
why a
neu-
ral
map-
ping
was
injec-
tive).

Hint:

Even
in
application-
heavy
pa-
pers,
theo-
reti-
cal
ground-
ing
in-
creases
trust
and
inter-
pretabil-
ity.

Increased

**Al-
go-
rith-
mic
and
Ar-
chi-
tec-
tural
Clar-
ity**

Accepted
ver-
sions
often
added:

*

Pseudo-
code
or
algo-
rithm
boxes.
* De-
tailed
archi-
tec-
tural
dia-
grams
or
hy-
per-
pa-
rame-
ter
se-
tups.
* De-
sign
ratio-
nales
(why
this
model,
loss,
or
ini-
tial-
iza-
tion?).

Hint:

Help

re-

view-

ers

and

read-

ers

re-

pro-

duce

or

build

on

your

work

with

clar-

ity

and

com-

plete-

ness.

Inclusion

of

Hu-

man

Stud-

ies

or

Real-

World

Use

Cases

(Where

Ap-

pli-

ca-

ble)

For
pa-
pers
in-
volv-
ing
hu-
man
fac-
tors
or
sym-
bolic
rea-
son-
ing:

* Ac-
cepted
ver-
sions
added
IRB-
approved
user
stud-
ies,
often
with
sta-
tisti-
cally
sig-
nifi-
cant
find-
ings.
* De-
scribed
de-
ploy-
ment
impli-
ca-
tions
(e.g.,
in
game
AI or
ex-
plain-
able
plan-
ning).

Hint:

When
hu-
mans
are
in-
volved,
even
small-
scale
user
feed-
back
can
vali-
date
use-
ful-
ness
and
clar-
ity.

Summary:**Top****5****Gen-
eral****Im-
prove-
ments**

Theme
Description
Stronger
Experiments
More datasets, metrics, ablations, baselines
Clear Novelty Framing
Explicit differentiation from prior work; clearer motivation
Theoretical or Conceptual Rigor
Conditions, proofs, or formal

From the comparative analysis on the 8 pairs of papers, can you summarize the most significant quantitative improvements among them, to give general hints to all authors?

From the comparative analysis of the 8 accepted–rejected paper pairs in your dataset, several quantitative improvements consistently contributed to acceptance. These improvements span performance metrics, experimental scope, statistical validation, and reporting clarity. Here’s a synthesis of the most impactful quantitative enhancements that authors should consider:

Significantly Expanded Empirical Coverage

Aspect	Rejected Version	Accepted Version
# of Datasets	1–2	3–5+ (e.g., CIFAR-10, ImageNet, MNIST, Minecraft)
# of Baselines	2–3	4–6+ (including strong recent models)
# of Metrics	Accuracy or one primary metric	Accuracy + FID + certified radius + runtime + uncertainty

Hint: Triple your experimental scope. Evaluate on more diverse datasets and metrics to demonstrate robustness.

Addition of Ablation Studies and Sensitivity Analysis

Component	Rejected Version	Accepted Version
Ablation Studies	Often missing	Present in all accepted versions
# of Ablation Variants	0–2	3–6+ (e.g., loss terms, architecture parts, data sizes)
Hyperparameter Sensitivity	Rarely included	Grid tested (e.g., margin parameters, noise levels)

Hint: For every architectural or loss-function element, show its effect quantitatively.

Substantial Gains in Key Metrics Over Baselines

Paper Domain	Metric	Improvement (Accepted vs Rejected)
Open-World SSL (ORCA)	Accuracy on unseen	+96% on ImageNet unseen classes
Sliced Wasserstein Distance	FID in image tasks	10–20% better than SWD and GSWD
Certified Robustness	Certified accuracy	+3–7% in L -radius robustness over prior SOTA

Paper Domain	Metric	Improvement (Accepted vs Rejected)
Planning Explanation	User task success rate	+35% improvement in user task performance

Hint: Ensure your improvements are both large in magnitude and well-justified statistically (confidence intervals, standard deviations, etc.).

Statistical Significance and Reporting Discipline

Reporting Style	Rejected Version	Accepted Version
Standard Deviations	Sometimes omitted	Reported for nearly all metrics
Significance Tests	Absent or anecdotal	p-values reported in user studies, comparisons
Repetitions	1–3 runs	5–10+ runs (with mean \pm std)

Hint: Report results across multiple seeds and include statistical tests where applicable.

More Detailed Runtime and Resource Analysis

Runtime Benchmarking	Rejected Version	Accepted Version
Training Time	Not measured or vague	Included in tables/charts
Inference Time	Rare	Measured on multiple hardware settings
Memory/Scalability	Not explored	Occasionally benchmarked

Hint: If your method claims scalability or practicality, quantify it in runtime or memory benchmarks.

Summary: Quantitative Improvement Targets

Dimension	Suggested Target for Strong Submissions
# Datasets	3 diverse and standard datasets
# Baselines	4, including most recent strong ones
# Evaluation Metrics	3 (accuracy + task-specific + secondary)
Ablation Studies	3 configurations (each with metric impact)
Experimental Runs	5 seeds per experiment with std reporting
Metric Gains vs Baselines	10–20% unless margins are already tight
Statistical Tests (if human)	p-values or confidence intervals ($\alpha = 0.05$)
Runtime/Memory	Included if efficiency is a selling point

These quantitative refinements are not only persuasive to reviewers but also signal research maturity.

5.3 Feedbacks on the methodology

Although allowing for a fine-grain analysis, this approach has three main flaws: 1. **time consuming**: requires to manually download each pair of papers from the openreview website. 2. **expensive**: we are feeding the LLMs the two original PDFs for each comparative study. The input prompt with the files can vary from ~20K to ~38K tokens, while the output analysis requires ~2K tokens. We can reach around ~40K tokens, which is ~\$0.44 to analyze a single paper. 3. **data scarcity**: only a few papers in the entire dataset had both accepted and rejected versions available for download. While this smaller sample allows for fine-grained analysis, it may not be sufficient to generalize findings across the whole dataset.