Joint Text-and-Image Clustering for Social Science Research

Han Zhang[a*] and Ryan Leung[b]

[a*]Assistant Professor, Department of Sociology & the Watson School for International and Public Affairs, Brown University, RI, USA. *

[b]Ph.D. Student, Department of Sociology, Northwestern University, Evanston, IL, USA.

Abstract

While automated text analysis is getting extremely popular and image analysis is gaining interest, multi-modal analysis that combines both text and image information remains rare. However, many text or image data are intrinsically multi-modal, such as social media posts. This study compares three practical workflows for clustering text–image pairs: (1) label-level combination, which clusters text and image separately and combines the resulting labels; (2) vector-level combination, which clusters concatenated embeddings extracted from each modality; and (3) joint embedding, which clusters unified representations from multimodal embedding models such as CLIP. We also introduce a set of reusable evaluation tools to help researchers compare, validate, and benchmark multimodal clustering workflows: the Adjusted Mutual Information (AMI)

*Correponding Author: Han Zhang, the Watson School, Brown University, Room 242, 111 Thayer St, Providence, RI, 02912 USA. Email: han_zhang4@brown.edu

to assess text-image alignment, the S_DbW index to evaluate number of clusters, and the within-cluster consistency to validate interpretability. We validate the methods on a Chinese protest dataset from social media with 336,921 text-image pairs, and test robustness and scope conditions using a smaller U.S. news dataset on gun violence with 1,297 news headlines. We find that when text and image provide distinct, non-overlapping information, the second and third methods outperform the first. This study serves as a bridge between the text-as-data and image-as-data communities, as well as computational social science.

1 Introduction

Empirical social science has undergone a transformation—from data scarcity to data abundance (Grimmer et al., 2021). Digitized archives, government records, and user-generated content now provide access to millions of text and image documents. This abundance has fueled the development of automated methods, particularly for analyzing text. Over the past decade, topic modeling and other unsupervised techniques have become central tools for uncovering structure in large text corpora (Grimmer and Stewart, 2013; Blei et al., 2003; Wilkerson and Casas, 2017).

Although most methodological innovations have centered on text analysis, the most dramatic surge in data has come from visual content. Visual platforms like Instagram, YouTube, and TikTok now dominate the social media ecosystem (Auxier and Anderson, 2021), surpassing traditional text-centered platforms. However, most existing work focuses on supervised learning tasks—mapping images onto violence, race, protest, or sentiment labels (Steinert-Threlkeld et al., 2022; Casas and Williams, 2017; Williams et al., 2020). By contrast, unsupervised image analysis—and particularly clustering—remains underdeveloped in social science research (Peng, 2018; Zhang and Peng, 2024). This gap is surprising given that unsupervised clustering is often the first step in exploring massive datasets and has driven much of the growth in text-as-data research.

Even more rare is the integration of text and image to perform joint clustering. This gap is particularly problematic because real-world social media posts, news articles, and political communications rarely rely on a single mode of communication—they strategically combine visual and textual elements to frame events and guide interpretation. Decades of psychology research show that people process text and visual information through parallel cognitive channels: visuals are often processed more intuitively, leave longer-lasting impressions, and are quicker to recall, while text conveys abstract and complex ideas more precisely (Paivio, 1990; Sweller et al., 1998). When texts and images convey similar messages, they can improve understanding and persuasion (Mayer, 2002; Powell et al., 2015; Wittenberg et al., 2021), and help information extraction (Steinert-Threlkeld et al., 2022). Other times, texts and images convey independent information (Zhang and Pan, 2019; Casas and Williams, 2019; Joo and Steinert-Threlkeld, 2022). Yet again, images may even contradict text; for instance, Gibson and Zillmann (2000) showed people peaceful vs. violent protest images with the same headlines and found that people's perceptions of legitimacy shifted differently. Thus, analyzing text or image alone risks missing both these reinforcing and countervailing effects, motivating a clustering approach that treats each pair as a unified unit.

We propose an unsupervised multimodal clustering pipeline that converts text and images into low-dimensional embeddings via pre-trained models, then applies standard clustering algorithms (e.g., K-Means and HDBSCAN). Our focus is on the first step: which strategy for converting text–image pairs into numeric representations (i.e., embeddings) produces the most effective clustering solutions? To answer this, we compare three general methods—label-level combination (cluster each modality separately), vector-level combination (concatenate per-modality embeddings before clustering), and joint embedding (use multimodal models to produce unified text–image vectors). While specific embedding models may evolve, these three strategies cover the full range of practical approaches.

To compare how these three embedding methods as well as other modeling choices impact clustering results, we employ a general-purpose evaluation framework that serves both to

compare methods and to optimize clustering within each approach. Our toolkit includes Adjusted Mutual Information (AMI) to quantify text–image alignment (which indicates when joint clustering is necessary), the S_DbW index to score cluster quality and guide selection of the number of clusters, a data-loss metric that tracks the share of observations pruned when clusters become too small to interpret, and human coding to assess topical coherence.

We implemented our clustering pipeline and evaluation framework on a dataset (CASM) containing 336,921 Chinese social media posts discussing offline protest events. Each post includes both textual and visual content. We found that texts and images contain complementary information in this dataset. Both joint approaches (vector-level concatenation and joint embedding) markedly outperform label-level clustering while incurring far less data loss. We also tested our framework on a supplementary dataset with 1,297 gun violence news articles in the US (BU-NEmo). In this dataset, headline text is predictive of image content, so the two joint methods do not show substantive gains over the baseline label-level combination. Taken together, these results indicate that joint clustering is most valuable when the two modalities convey distinct yet complementary information; when they largely overlap, all three methods can produce satisfactory clustering results.

This study bridges the text-as-data (Grimmer and Stewart, 2013) and image-as-data (Joo and Steinert-Threlkeld, 2022; Zhang and Pan, 2019) literatures by proposing a practical framework for joint clustering of multimodal documents. We show that when text and images convey complementary information, joint clustering methods significantly outperform modality-specific clustering. We also provide researchers with diagnostic tools to determine whether their data would benefit from joint analysis and empirical guidelines for when single-modal approaches are sufficient. Our findings, validated on Chinese social media posts about protests and U.S. news coverage of gun violence, offer concrete guidance for scholars working with multimodal data in computational social science (Grimmer et al., 2021).

Section 2 reviews prior work on text-as-data and image-as-data methods and motivates the need for multimodal clustering. Section 3 introduces our three combination strategies

for combining text and image embeddings, along with our evaluation toolkit. Section 4 applies these methods to the CASM protest dataset and presents clustering results. Section 5 provides diagnostic analysis of when joint clustering is most effective. Section 6 replicates the analysis on the gun violence dataset to establish scope conditions and assess robustness. Section 7 concludes.

2 Foundations of Multimodal Clustering

In this section, we establish when multimodal clustering is necessary, review how texts and images are transformed into embeddings, and discuss clustering algorithms commonly used with these representations.

2.1 When Is Multimodal Clustering Necessary?

Sometimes texts and images exhibit a high cross-modality correlation, meaning that one can predict the other with near certainty. In the simplest case, both modalities convey the identical content, such as a social-media post captioned "apple" accompanied by a photograph of that apple. Alternatively, text and image may deliver different types of information yet often co-occur, like a financial news headline that is often paired with a Wall Street photo, though they are not exactly the same. Although these cases represent semantically different relationships, both demonstrate strong cross-modal correlation patterns. In these cases, the necessity of conducting joint text-image clustering is smaller.

In general, however, text and images often convey complementary yet distinct information, so that a single post may communicate different messages through its caption and its photograph. For example, the caption "stay strong" could accompany either a protest scene or a tranquil landscape—two very different messages that text-only clustering would treat as identical. The text caption can express abstract judgments, intentions or emotions—such as concern, solidarity or outrage—while the accompanying image presents concrete visual

details of people, objects and settings without explicit commentary. In these situations, clustering on text alone may miss important distinctions for visual information, and vice versa. Hence, joint clustering becomes necessary.

Having shown why joint clustering can be useful, we next outline the two technical ingredients that enable it—embeddings and clustering algorithms—before detailing our own approach in Section 3.

2.2 Pre-requisites: transform texts and images into embeddings

2.2.1 Text representation

Effective joint clustering depends critically on how texts and images are transformed into vector representations. Traditional text clustering approaches in social science, particularly topic modeling, rely on document-term matrix representations where each document becomes a long, sparse vector (often 10,000+ dimensions) with most elements being zero (Grimmer and Stewart, 2013; Goldberg and Levy, 2014).[1] These sparse representations consume significant computing resources and make multimodal integration challenging.

Modern text embedding techniques offer a solution by using neural networks to create dense, low-dimensional vectors (typically below 1,000 dimensions). Early approaches like word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) used shallow neural networks to create word-level embeddings, which could then be aggregated into document representations. The major breakthrough came with Transformer-based models like BERT (Vaswani et al., 2017), which use deeper neural networks to directly encode sentences, paragraphs, and entire documents into dense vector representations.

2.2.2 Image representations

For images, embedding techniques convert visual content into dense vector representations with dimensions typically ranging from several hundreds to several thousands. The key breakthrough occurred with convolutional neural networks (CNNs), particularly after

AlexNet in 2012 (Krizhevsky et al., 2012) . More recently, Vision Transformers have adapted the Transformer architecture from natural language processing to image analysis (Dosovitskiy et al., 2020), showing competitive or superior performance to CNNs. Recent work has successfully applied these deep learning techniques to automated image clustering for social science applications (Caron et al., 2018; Zhang and Peng, 2024).

2.3 Clustering algorithms

After texts or images are transformed into dense numeric vectors, scholars can apply clustering algorithms to automatically group similar items and assign topic labels. The number of available clustering algorithms is vast, and comprehensive reviews can be found in Hastie et al. (2009).

Since our primary goal is to compare data representation strategies rather than clustering algorithms, we used two widely used methods from different algorithmic families:

- K-Means, a centroid-based method. It assigns points based on distance to cluster centroids.
- HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), a density- and hierarchy-based method introduced by Campello et al. (2013). HDBSCAN does not assume spherical geometry and can detect clusters of arbitrary shapes.

Although clustering algorithms are abundant and varied, our aim is not to exhaustively compare them, nor can we. Instead, we test whether our vectorization strategies yield consistent clustering results across two distinct clustering paradigms. We find that the relative performance of the representation methods remains robust across both K-Means and HDBSCAN. For simplicity, we present the results from K-Means as the main findings in this article and provide metrics from HDBSCAN clustering in the Appendix.

3 Methods for Joint Text-Image Clustering

Our goal is to conduct unsupervised clustering on documents consisting of both text and images. This process involves three main steps:

- Representation learning: Map texts and images into numeric vectors, or *embeddings*, that can be processed by clustering algorithms. This step includes choices about how to encode each modality and how to combine them.
- Grouping: Apply clustering algorithms to assign these embeddings into distinct clusters. This involves selecting an algorithm (e.g., K-Means or HDBSCAN) and setting parameters such as the number of clusters.
- Interpretation and evaluation: Analyze the resulting clusters by assigning descriptive labels and assessing their quality through both internal metrics and human validation.

Our main contribution lies in the representation learning stage. We compare three methods for representing a text-image pair as vectors. Figure 1 illustrates their workflows. The key distinction among them lies in *when* the text and image data are transformed and merged into a joint embedding.

A second contribution lies in our evaluation framework (Section 3.4), which allow us to systematically compare different clustering solutions. We do not claim novelty in the clustering step itself, which is essential to complete the pipeline but is beyond the scope of this article (see Section 2.3 for reviews).

3.1 Baseline Method: label-level combination

The first and simplest method proceeds by running text and image clustering algorithms separately, obtaining cluster labels for each modality, and then combining them together. The left panel of Figure 1 visualizes this approach. Specifically, the original data are first assigned to m distinct categories based on the textual information and n distinct categories

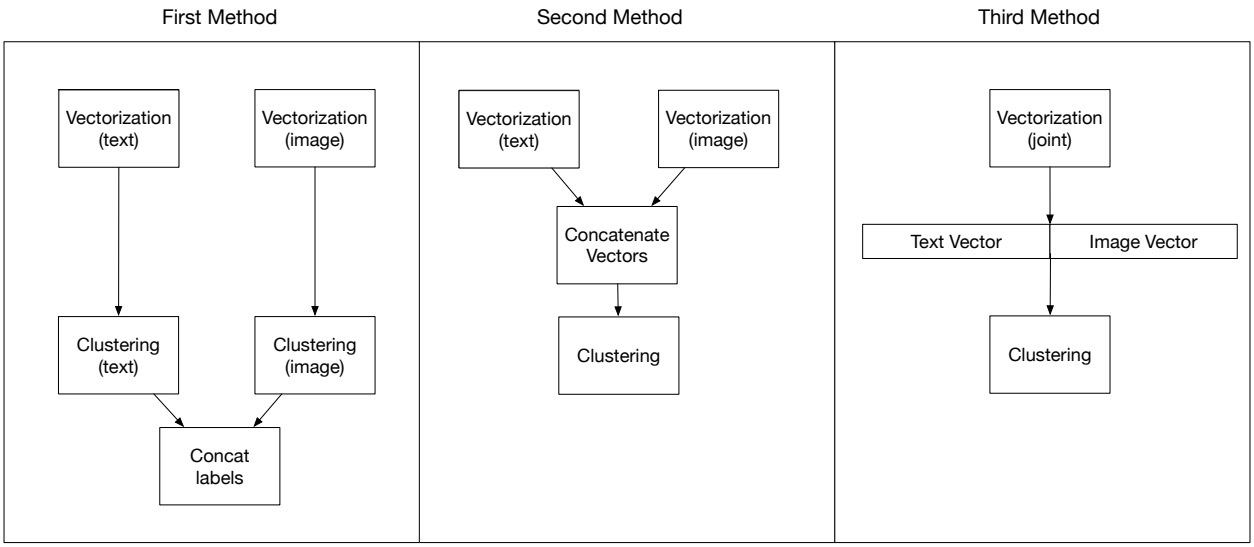


Figure 1: Visualization of three joint text-image clustering methods

based on the images. By taking the concatenation of these categorical labels, we obtain a total of $m \times n$ categories for the data sample.

3.2 When to Use Label-Level combination vs. Joint Clustering?

The effectiveness of label-level combination depends critically on the degree of cross-modal alignment between text and image, as discussed in Section 2.1. To help researchers decide whether joint clustering is needed, we introduce a simple diagnostic that quantifies cross-modal alignment: Adjusted Mutual Information (AMI).

Diagnostic Based on AMI: Do We Need Joint Clustering? Mutual Information (MI) measures how much knowing one variable reduces uncertainty about another. In our context, it quantifies how much information the clustering of one modality (e.g., text) reveals about the clustering of the other (e.g., image). We use Adjusted Mutual Information (AMI), which adjusts for chance alignment and scales the result to fall between 1 (perfect agreement) and 0 (no better than chance). A full mathematical definition is provided in Appendix A.1.

High AMI values (close to 1) suggest strong cross-modal correspondence—each text cluster closely predicts its associated image cluster. In this case, separate clustering on each

modality already captures the joint structure, and label-level combination suffices. When AMI approaches 0, however, a single text topic links to many different visual patterns (and vice-versa), so separate analyses cannot reliably infer one modality from the other. Under such low-correlation conditions, joint text–image clustering methods (vector-level and joint embedding) become essential for exploiting complementary information, as we discuss next.

3.3 Joint clustering

Vector-level combination Vector-level combination aims to merge text and image vectors into a single vector, which is then fed into grouping algorithms (middle panel of Figure 1). We use text embedding algorithms to transform the textual component of a document into a p-dimensional vector, and image embedding algorithms to transform its visual content into a q-dimensional vector. For instance, we can use pre-trained models such as Word2Vec, GloVe, or BERT, or more recent LLM embedding models from OpenAI and Google Gemini to generate text embeddings.[2] Similarly, we use pre-trained models such as ImageNet VGG16 or more recent Google Vertex AI's image or multimodal embedding models to map images into dense vector representations. We then concatenate the p-dimensional text vector and the q-dimensional image vector into a single $(p + q)$-dimensional embedding for clustering.

Joint embedding The third method treats text-image pairs as a unity from the beginning (right panel of Figure 1). This is made possible by the emergence of multimodal models in the past several years. These multi-modal pre-trained models can take image-text pairs (e.g., tweets with both text and images) as their training data, and learn to *jointly* map them into a single embedding vector in the same vector space (see Figure 1). The first breakthrough was the open-sourced Contrastive Language-Image Pre-training (CLIP) model released by OpenAI in 2021, which is trained on over 400 million text-image pairs collected from the Internet (Radford et al., 2021; Srinivasan et al., 2021), and its variants.[3] There have been more commercial multimodal releases since then, such as Google Vertex-AI or

Amazon Titan [4] By training on diverse image–text pairs, multimodal models learn common and uncommon co-occurrence patterns, providing a useful knowledge base for clustering on researchers' specific datasets.

3.3.1 Summary and Practical Guidance

To help researchers choose between the three clustering workflows, we compare their key trade-offs along the following dimensions:

- Theoretical soundness: Joint embedding is the most advanced and theoretically appealing approach. Label-level combination performs no cross-modal learning, while vector-level combination sits in between.

- Decision complexity: Label-level combination requires separate modeling choices for each modality—embedding methods, clustering algorithms, and K values for both text and images—while joint embedding uses a single pipeline; vector-level combination sits in between.

- Model selection flexibility: Label- and vector-level combinations can choose from a wide range of single-modality embedding models. Early-generation models (word2vec, ResNet-Places365) are lightweight and easy to fine-tune and are still widely used in recent social science research, while advanced models offer better performance but cost more to run and are harder to customize (i.e., fine-tuning). Model choice should depend on dataset characteristics: if the dataset matches a model's training domain (e.g., protest scenes with Places365), early-generation embedding models suffice, as we found in our primary dataset; otherwise, advanced models trained on broader datasets typically generalize better (Section 6.5). In contrast, joint embedding offers fewer model options and limited fine-tuning flexibility.

  In contrast, joint embedding has much fewer choices. Moreover, these models are difficult to fine-tune. If flexibility is the priority, then one should choose label- or

vector-level combinations.

- Dimension explosion: Label-level combination multiplies cluster counts—even modest numbers of text and image clusters yield potentially excessive joint clusters. Researchers must either merge or prune these clusters (risking data loss) or face a heavy interpretive burden.

We provide key Python codes for the three methods in Section B for readers interested in the technical details. Notably, extracting these embeddings requires fewer than a hundred lines of code, significantly enhancing the practicality and accessibility for use in research applications.

3.4 Evaluation Criteria

Choosing the right embedding model is just the first step—we also need robust methods to evaluate clustering results (Grimmer and Stewart, 2013). Moreover, an evaluation toolkit is useful because while embedding models evolve rapidly and researchers may choose different models depending on their datasets and computational constraints, the evaluation framework for multimodal clustering remains relatively stable. Text-image pairs present unique evaluation challenges compared to single modalities. We discuss our evaluation toolkit.

3.4.1 Data-driven performance measure of clustering results using S_DbW index

To evaluate internal clustering quality, we use the S_DbW index (Halkidi et al., 2002), which sums two components—each of which we aim to minimize:[5]

- Within-cluster dispersion (lower is better): the average dispersion of data points around their cluster centroids; minimizing this yields tighter, more cohesive clusters.

- Between-cluster density (lower is better): the density of points in the regions between clusters; minimizing this reduces overlap and ensures clusters remain distinct.

Consequently, lower S_DbW values indicate higher-quality groupings, achieving both compactness and clear separation. For the formal definition, see Appendix A.2.

3.4.2 Selecting K using the marginal gain of S_DbW index

S_DbW also guides us through one of the trickiest parts of clustering: deciding how many clusters to use. Too few clusters might hide meaningful differences; too many can create artificial splits and make interpretation harder.

We begin with a small number of clusters and slowly increase it. At each step, we check how much the S_DbW index improves, relative to the increase in the number of clusters. If this relative gain is large, the extra cluster might be worth it. But if the gain becomes small, we likely just add complexity without a real benefit.

We formalize this intuition as the *marginal gain* in S_DbW, calculated as the difference between the absolute changes in index divided by changes in the cluster number from K' to K. We use absolute changes to make it robust to small changes that would inadvertently favor solutions with larger number of clusters.

$$\text{MarginalGain}_{S\_DbW} = |\frac{\Delta S\_DbW}{\Delta K}| = |\frac{S\_DbW(K) - S\_DbW(K')}{K - K'}|$$

We stop increasing K when this marginal gain drops below a threshold (0.01 in this paper), meaning that further increasing the number of clusters no longer improves grouping quality.

3.4.3 Human-coded within-cluster consistency

Optimal clustering solutions in the eyes of machines—smaller S_DbW in our context—may not correspond to the optimal solutions identified by humans (Chang et al., 2009). We follow Zhang and Peng (2024) to calculate the within-cluster consistency of each image clustering method and each choice of K. We measured within-cluster consistency by randomly sampling 10 posts from each cluster and having human coders assign themes based on text-only, image-

only, and joint content. The consistency score represents the proportion of posts in a cluster that match its most frequently assigned theme, with higher scores indicating better thematic coherence. Appendix E provides details of the human coding procedures. Note that this index does not capture the between-cluster density part of S_DbW index so it should be use with other evaluation metrics.

3.4.4 Addressing dimension explosion issues and data loss

A unique problem of label-level combination is dimension explosion. Clustering text and images separately and then combining the labels creates many small clusters—often too small to interpret or use, when cross-modal alignment is low (i.e., low AMI). Researchers could choose a lower number to avoid dimension explosion, but this would risk using too few clusters for each modality.

Joint clustering methods (Methods 2 and 3) avoid this problem by analyzing both modalities simultaneously. Instead of producing many tiny combinations, the algorithm recognizes that several small clusters are actually variations of the same broader theme and groups them together. This produces larger, more interpretable clusters with far less data loss.

If there are indeed dimension explosion issues from label-level combinations, there are two solutions:

Iterative merging One common strategy is to estimate more topics than you expect to need and then manually merge any that look redundant. In practice, rather than computing formal similarity scores, researchers typically inspect an interactive display (e.g. LDAvis; (Sievert and Shirley, 2014)) to spot near-duplicate topics and collapse them into a single theme. Some toolkits—most notably BERTopic—offer built-in commands to streamline this step, but each merge ultimately relies on the analyst's judgment of thematic overlap. Manual label merging is also a valid option when guided by theory and supported by clear coding schemes; our proposed within-cluster consistency measures can help validate and audit such

merging decisions.

Pruning and data loss The second solution is to "prune" the results by keeping only the largest clusters and discarding the rest. This is also a popular choice and has been implemented in popular topic modeling packages such as BERTtopic (Grootendorst, 2020). This approach is easier to implement compared with iterative merging, but the downside of this approach is that it discards data. We formally measure the *data loss* as the percentage of documents dropped if we only keep the top k largest clusters out of a total K clusters.

4 Joint Text-Image Clustering on CASM: process and results

4.1 Dataset

Our main dataset comes from CASM-China, where Zhang and Pan (2019) used supervised methods to identify whether social media posts (with both texts and images) discuss offline protests.CASM-China contains over 136,330 offline protests in China from 2010 to mid-2017, with 273,950 associated Weibo (Chinese Twitter) posts—around 2.01 posts per event on average. Posts can contain up to 9 images, though many contain no images, resulting in 336,921 total images.[6] Zhang and Peng (2024) used a sample of CASM-China (around 5%) to compare image-only clustering approaches. This article extends that work by systematically comparing clustering results using text alone, images alone, and multimodal combinations.

We keep only Weibo posts that have both images and text, and further create image-text pairs from the posts. If a Weibo post contained one paragraph of text but multiple images, we create multiple text-image pairs by associating the same text with each piece of images. In total, we have 336,921 image-text pairs extracted from the Weibo posts. Then we applied our three methods below. The computer details are provided in Appendix Section B.4.

4.2 Label-level combination

Text clustering For text representation in the CASM dataset, we used BERT to extract 512-dimensional document embeddings.[7] Then we removed the stopwords with our own stopword list depending on Jieba and HIT Chinese stopwords list. We manually added some Chinese tokens that are clearly irrelevant to our context. [8] Given that CASM texts are typically concise and informal, we found that standard BERT-based embeddings already performed well. While recent LLMs could potentially improve representation quality, the current embeddings already yielded coherent and interpretable clusters across validation metrics. We then used K-means for grouping these embeddings into clusters.

The text clustering results for $K = 10$ clusters are presented in Table 1. We first examine each modality separately with $K = 10$ to capture sufficient topical variation. However, using the same K for images would result in 100 total clusters for label-level combination, which becomes too much as we will see in next subsection.

Over the ten topics shown in the table, we can observe that three out of the ten clusters (Clusters 2, 3, and 10) are about labor disputes. The observation that labor disputes is the most prevalent protest type is similar to those in the original CASM-China dataset, but we obtained these from clustering whereas the original authors used supervised approaches based on dictionary methods (Zhang and Pan, 2019). Cluster 1 is about protesters using blocking roads as a tactic, which caused traffic jams, and many of the posts were from a third-person point of view. Cluster 4 is a mixed cluster with two topics of textual information: either doctor-patient disputes resulting in family members of patients protesting in hospitals, or posts describing protesters who suffered from violence and received medical treatment. Cluster 5 is about posts describing general protests without enough contextual details. Cluster 6 is about legal enforcement. Interestingly, both Cluster 7 and Cluster 8 are about consumer rights protests, with Cluster 8 more concentrated on posts where homeowners express grievances towards real estate developers or management. Finally, Cluster 9 is about protests against forced evictions and land acquisitions.

Table 1: Words with Highest c-TF-IDF Score in K-Means Text Clustering at $K = 10$. The c-TF-IDF measurement to pick out the most important word tokens in each clusters (Grootendorst, 2020)

Cluster	**# of Posts**	**Terms with Top 10 Highest TF-IDF Score**
1: Road Blocking Protest	11095	Road Blocking 堵路/拦路 Gate 门口 Mobbing 闹事 Somebody 有人 Banner 横幅堵车 Protest 游行 Detour 绕行 City Government 市政府
2: Labor Dispute	6187	Wage 血汗钱 We 我们 People 老百姓 Worker 农民工/民工/工人 Government 政府 Demand 讨要 Company 公司 New Year 过年 Return Home 回家
3: Labor Dispute	19767	Worker 农民工 Wage 工资 Demanding Payment 讨薪 Delayed Payment 拖欠 They 他们 New Year 过年 Return Home 回家 We 我们 Government 政府 Boss 老板
4: Doctor-Patient Disputes / Violence	13550	Hospital 医院 Taxi 出租车 Chengguan 城管 Driver 司机 Family Member 家属 Law Enforcement 执法 Violence 暴力 Police 警察 Car-Owner 车主 In site 现场
5: Protest (General)	5224	Protest 抗议 Link 链接 Webpage 网页 Collective 集体 Demonstration 示威 Student 学生 China Construction Bank 建行 Japan 日本 School 学校 People 民众
6: Law Enforcement	7932	SWAT 特警 Police 警察 Police Car 警车 Escalate 出动 Public Security 公安 Gate 门口 Force 力量 Uncle 叔叔 Today 今天 Webpage 网页
7: Consumer Rights	6808	Defend Rights 维权 Home-owner 业主 We 我们 Ourselves 自己 Car-owner 车主 Defend 维护 Rights 权益 Consumer 消费者 Support 支持 Link 链接
8: Home-Owners	12592	Home-owner 业主 Developer 开发商 Community 小区 Defend Rights 维权 Home Management 物业交房 House 房子 Vanke Real Estate 万科 Price Drop 降价 Issue 问题
9: Forced Eviction	11720	Villager 村民 Government 政府 Forced Eviction 强拆 Eviction 拆迁 Land 土地 Farmer 农民 People 老百姓/百姓我们 Land Acquisition 征地
10: Labor Dispute	17820	Employee 员工 Worker 工人/民工 Demand Payment 讨薪 Boss 老板 Owe 拖欠 Wage 工资 Company 公司 They 他们 Jump of Building 跳楼

Image Clustering For image representation in the CASM dataset, we followed Zhang and Peng (2024) by first transforming images into embeddings using a pre-trained ResNet-18 model trained on the Places365 dataset. ResNet-18 is a widely used convolutional architecture in deep learning (He et al., 2016), and Places365 contains 1.8 million labeled images across 365 scene categories, covering a broad range of indoor and outdoor environments such as streets, squares, conference rooms, kitchens, and legislative chambers (Zhou et al., 2014). This combination is particularly well-suited to our setting: the CASM dataset consists of user-generated protest-related social media posts, where images predominantly depict environmental or situational scenes (e.g., streets, crowds, surveillance infrastructure). Compared to models trained on object-centric datasets like ImageNet, Places365-trained models better capture the spatial and contextual structure of such protest scenes, providing more relevant image embeddings for downstream analysis. The resulting image embeddings are 512-dimensional vectors. Finally, we perform K-Means clustering on the 512-dimensional vectors to identify visual topic groupings.

Figure 2 shows the clustering results from the K-means algorithm. We apply K-Means ($K = 10$).[9] Each row represents a cluster, and we randomly sampled 10 images belonging to the cluster. From the results, we can observe interpretable themes of images in different K-means image clusters: crowd gatherings at Cluster 1, injuries and conflicts at Cluster 2, buildings at Cluster 3, construction plants at Cluster 4, screenshots or contracts at Cluster 6, outdoor protests and police escalations in Cluster 8, indoor collective actions at Cluster 9, and protests with banners at Cluster 10. Except for Cluster 6, all other image clusters consist of photos showing the venues (construction plants, factories, gates of governmental buildings), the involved social actors (workers, homeowners, police, officials, organizations), the intensity of protests (violent clashes, injuries), and the tactics (road blocking, door blocking, holding banners, threatening to commit suicide by jumping off the building) of the protest.

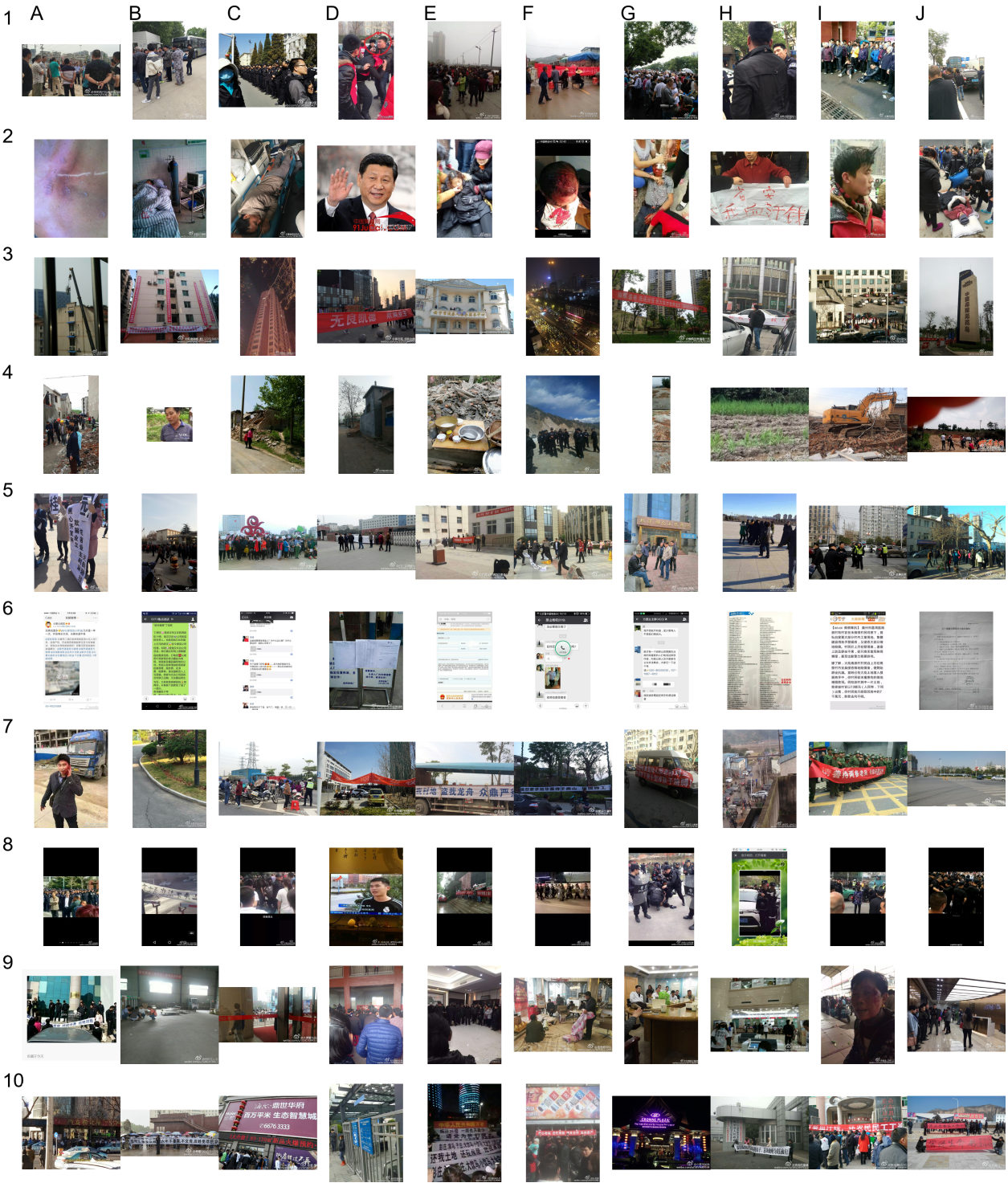


Figure 2: Image-clustering alone; $K = 10$. Each row shows 10 randomly selected images from a single cluster.

Label-level combination With separate clustering results for each modality, we create the final category by concatenating text and image group labels. For instance, if a social media post is assigned into the text category describing "environment" and the image category describing "banner", then the joint category will be "environment & banner".

Figure 3 shows a heatmap of these combinations, with each cell representing the percentage of posts in that joint cluster. To highlight potential text-image alignments, we reordered the heatmap's columns so that high values along the main diagonal (compared to other cells in the same row or column) would indicate strong one-to-one correspondence between text and image categories. [10]

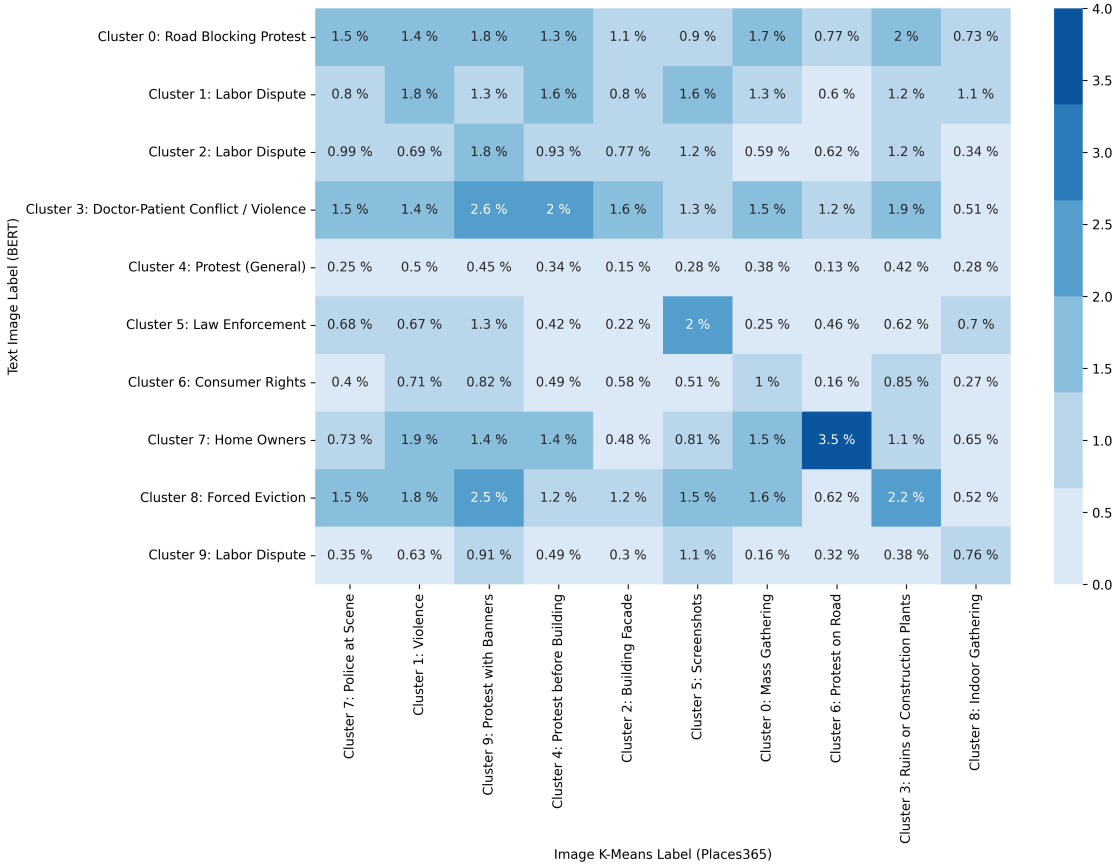


Figure 3: Heatmap for the proportion of classes in label-level combination (label-level combination). The X-axis are image categories and Y-axis are text categories. Each cell indicates the ratio between the number of text-image pairs in this particular category and the total number of text-image pairs. We permute columns (via the Hungarian algorithm) to align matching text–image clusters along the diagonal for easier visual inspection.

4.2.1 Low modality correlation motivates the necessity of joint embedding

The diagonal structure in Figure 3 reveals the degree of alignment between text and image clusters. When text and image modalities are strongly correlated, we expect to see a clear diagonal pattern with high values. Conversely, a more dispersed pattern indicates weaker correlation, suggesting that text and image content convey different information.

Upon visual inspection, we observe that the diagonal values are not particularly high compared to off-diagonal cells, suggesting texts and images are not strongly correlated in our dataset. To formally test this intuition, we computed the AMI between text and image labels. The resulting AMI is 0.029, very close to zero, which confirms near-independence between modalities. This lack of cross-modal alignment creates challenges for label-level combination, as we end up with many smaller clusters.

4.2.2 Selecting the Right K Using Marginal Gains of S_DbW

To identify a common and reasonable value of K for this dataset, we applied the marginal gain rule to clustering solutions with $K = 9\ (3^2), 16\ (4^2), \cdots 100\ (10^2)$. Figure 4 shows how marginal gains in the S_DbW index change as K increases.

We observe that marginal gains drop substantially at $K = 25$ for all three methods. Increasing K beyond this point yields minimal improvements in S_DbW, while adding interpretive burden. This suggests that our earlier use of $K = 100$ in the label-level combination may have been excessive. Consequently, we reduce the number of clusters to $K = 25$ and apply this value uniformly across the all methods, including the other vector-level combination and joint embedding methods. For consistency, we also rerun the label-level combination using $K = 25$ (i.e., 5 clusters per modality). This alignment allows for a fair comparison, ensuring that observed differences stem from the embedding strategies rather than from variations in cluster count.

We acknowledge that $K = 25$ may not be the optimal value for every method. Each method may perform better with a different K not included in our grid search. Moreover,

the marginal gain in the S_DbW index is just one of many possible criteria for selecting K. In applied research, analysts may explore a wider range of values and make final choices for each particular clustering method separately. For this methodological study, we fix $K = 25$ in our main results because it offers a reasonable balance across all three methods (as supported by S_DbW scores) and enables fair methodological comparison.

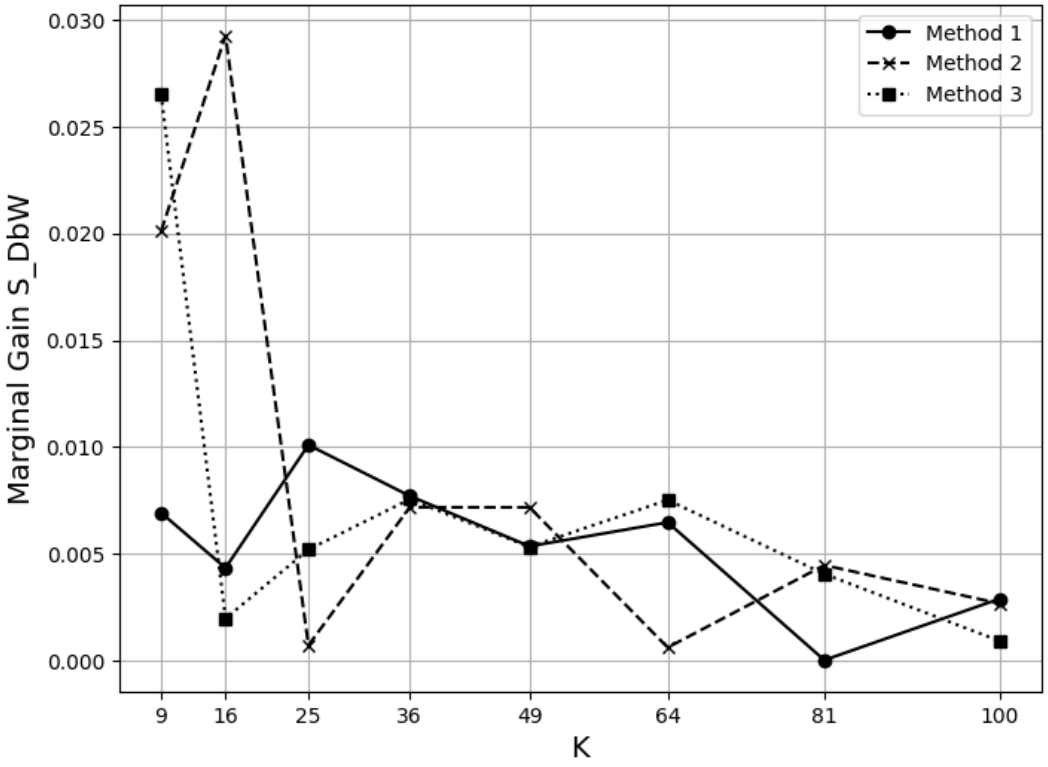


Figure 4: Marginal absolute *S_DbW* gain (Y-axis) as a function of the number of clusters (K) for CASM. At $K = 25$, all three methods reach a marginal gain below 0.01, indicating diminishing returns.

4.3 Vector-level combination

We used the same embedding models for text (BERT) and images (ResNet-Places365) to obtain embeddings for each modality. We then concatenated the two to form a 1,024-dimensional document vector. We further used Uniform Manifold Approximation and Projection (UMAP) to project the 1024-dimensional vectors into a 50-dimensional space (McInnes

et al., 2020). This reduction step is mainly to reduce computational cost. For instance, K-means's runtime scales quadratically with the embedding dimension; clustering directly on the full 1024-dimensional vectors would take over 400 times longer than on the reduced 50-dimensional vector. The final 50-dimensional vectors are clustered using K-means.

The clustering result is presented in Figure 5. We show five pairs for each of the first 5 largest clusters, ranked by the size of clusters. For the cluster visualizations in Figure 5 and following, we randomly sample five image-text pairs within the 10% observations that are closest to the cluster centroids. With this strategy, we can have a direct view for the data-points that are most representative of the clusters. The remaining 20 clusters can be viewed online due to space limitations.[11]

Figure 5 shows that vector-level combination is able to find common, large clusters. For example: the largest cluster—road-block protests—comprises 6.38% of the dataset. These demonstrations—where workers block major roads to demand owed wages—are among the most common forms of collective action in China (Zhang and Pan, 2019; Cai, 2010). By contrast, label-level combination fragments this cluster across six separate clusters, based on the combinations of text and image cluster labels:

- Discussions around labor disputes on owed wages belong to text cluster 2, 3 and 10 (see Table 1; and rows of Figure 3).
- Protesters holding banners are captured in Cluster 3 and 8 according to the image clustering (Figure 2).

The superior performance of vector-level combination stems from its ability to use fewer clusters ($K = 25$) while still identifying large, meaningful clusters. With label-level combination, we face a dilemma: using too few clusters would miss small clusters such as Cluster 6, Consumer Rights in text clustering. Conversely, using even a moderate number of clusters (10 each) correctly identifies small clusters but creates redundancy with multiple similar

concept clusters appearing separately. Vector-level combination effectively resolves this challenge in low text-image alignment scenarios.

4.4 Joint embedding

We used OpenAI's CLIP-ViT-B32 multimodal model to turn image-text pairs into a 1024-dimensional vector, and again reducing that to 50 dimensions. [12] We then use K-means with $K = 25$. Similarly, we show the top 5 clusters (each cluster shows 5 documents) in Figure 6. The rest 20 topics are shown in the Appendix.[13]

We can see that the top clusters are quite similar to that from vector-level combination: labor disputes with people blocking roads, protests at government offices. Vector-level combination identified protest with building facades (cluster 17), and bystanders complaining that they saw a protest blocking roads (cluster 22). On the other hand, joint embedding has protests portraying indoor images of dormitories or rental rooms accompanying grievances about housing or unpaid salary (cluster 18). It also portrays scenes in which uniformed officers confront or monitor the demonstrators (cluster 3). Both methods produce similar cluster types, but their relative sizes differ, so each method's top 5 contains different specific clusters.

Theoretically, the third method should be superior to the second in extracting more meaningful embeddings. We find that both methods yield reasonable results and it is harder to tell specifically which one is performing better. Hence we conduct evaluations using the tools we introduce in Section 3.4.

cluster 1: Labor Protest with Road Blocking (N = 21504, 6.38%)

Employees laid off from a well-known bookstore were compelled to demand their rights at the Urban Star in Chengdu. Not a single representative from the ...

Could this be what they call the troublesome villagers, blocking the road at the hospital entrance?

Fighting for Workers' Rights: A Tough Journey

Intersection of Avenue Two and East Street, featuring a road-blocking performance art. Drivers, please turn around early.

In a report from Chengdu, sanitation workers blocked a road to demand unpaid wages. At the intersection of two streets, dozens of workers used garbage trucks to ...

cluster 17: Protest with Building Facades (N = 20577, 6.11%)

The traffic jams on the way home.

Is anyone managing the road blockage in the village?

Every year before the end of the year, there's always a big confrontation, with banners and blocked roads to demand payment.

Recently, on Wuhu Road, there was a notable banner protest for demanding unpaid wages. This approach is certainly better than forming a human blockade and ...

It's unclear why there's such a banner! I'm here in Shaoxing at a specific location...

cluster 20: Labor Disputes with Road Blocking (N = 19251, 5.71%)

A group of migrant workers blocked the road at the entrance of a residential area, reportedly due to unpaid wages.

Title: First Encounter with Workers Blocking the Road Location: Lumen Avenue Story: I experienced a situation where workers were obstructing the road on Lumen ...

On the morning of August 13, 2015, around 10 o'clock, several individuals blocked the entrance to the government building in a district of a city. According to ...

[User Submission] Another road blocked in protest of unpaid wages!

A group of individuals protested due to unpaid wages from a construction site, blocking the road and affecting all passing vehicles. It was really ...

cluster 3: Protest at Government Buildings (N = 19233, 5.71%)

Crowds can be dangerous, so it's best not to surround someone sitting in a car.

Workers seeking unpaid wages appealed to the city government but were unexpectedly suppressed by over 300 people coordinated by officials in the black market and ...

A netizen reported: This morning (February 13), several workers hung a banner at a marketing center to demand their wages. Details are unclear.

It's alarming how an anonymous beauty and hair salon is deceiving its employees and taking their hard-earned money. Is there no authority in China to address this? ...

Today, I brought over a hundred workers, and both the labor bureau and the city government visited, but still, no one is addressing the issue. They continue to side with ...

cluster 22: Bystander View of Road Blocking (N = 16916, 5.02%)

An anonymous source reported: A group of 19 cleaning workers detained a luxury car for four days to protest unpaid wages of 410,000 RMB owed by their employer. They ...

Hey, you're blocking the road!

Are you blocking the road and causing trouble every day?

Stuck in endless traffic, when will I finally get home... I'm at: [web link]

Traffic jam again! I'm on East Dongfeng Road.

Figure 5: The top 5 largest clusters in vector-level combination (BERT with ResNet-Places365) under K-Means (K = 25). Each row contains 5 samples from the 10% samples that are closest to the cluster centers for each cluster. The total number of clusters are 25.

cluster 13: Protest with Road Blocking (N = 28375, 8.42%)

Real Estate Company, you're unbelievable and have made headlines. Does your CEO know? Do you really feel okay about maliciously withholding wages from ...

Look at these coal miners. Whose fault is it that they've reached this point today? We urgently need someone to stand up for us! Stop trampling on our spirits! We demand ...

It seems the entrance of the construction site is blocked today. Can any management staff or government officials take care of this? Location: Riverside North ...

I'm at the snowy scene of Century Square in front of the city government of a new district, located on Qibin Avenue.

These idiots! Does blocking the road solve anything?

cluster 18: Claiming Rights with Living Conditions (N = 26609, 7.90%)

Although most of the owners of Fuli No. 10 have already left, the owners of Phase II have also come to defend their rights. Since they didn't have an appointment, the ...

Enhanced Version After Rights Protection

A construction company has been delaying the wages of migrant workers. When approached about the unpaid salaries, the response was, "I have the money, but I just won't ...

Here is the translated and anonymized text: In Henan, 150 investors from Luoyang have been defrauded and are petitioning for justice. In Yunnan, 80 petitioners ...

At the request of the boss, the outsourced production of the item has been successfully completed! Hey boss, it's time to settle the payment~ Wages for workers ...

cluster 3: Police Dispatchments at Protest (N = 25518, 7.57%)

Citizens petitioned, but were met with heavy-handed police action.

The family asked the hospital if there was still hope for their child. The hospital replied that there was no hope unless a miracle occurred. The family offered 5 ...

Here's a related story involving migrant workers. When trying to obtain their overdue wages, they were instead falsely accused of harassment by a female employer. ...

The once-renowned DairyPro has forced its employees to seek government intervention, while the group's leadership remains indifferent. We need to survive, we ...

For ten years, the journey to defend rights has been long and arduous. Information about these efforts has been repeatedly blocked and suppressed. Please help spread ...

cluster 6: Marching on Roads (N = 24532, 7.28%)

The result was that municipal enforcement came out and drove him away. The crowd protested!

Is this the style of our hometown, Pingdingshan—delaying and not paying wages? | Main Avenue

A user has reported: "The road is blocked again. The residents of the Golden Waterfront Community on New Fu Road have been living here for two years without heating. There ...

Police brutality!

As the New Year approaches, migrant workers have no money to celebrate, and the problem remains unresolved. They've trusted the party and the government and followed legal ...

cluster 16: Claiming Rights at Government Buildings (N = 20606, 6.12%)

For two consecutive days, I passed by the county government entrance and saw migrant worker friends holding banners to demand their hard-earned wages. Unfortunately, ...

A construction project at a certain high school in Tangshan was completed in 2010. However, payment still hasn't been made to the workers. It's almost the end of the ...

The New Year is approaching, and once again, migrant workers are enduring the hardship of trying to get their pay! Location: Central Plaza, Changsha.

This morning around 9 o'clock, seven to eight hundred retired workers from a local port group gathered to protest due to housing subsidy issues. It's said that some ...

At the entrance of a residential community in Changsha, there was a blockage by construction workers. On August 11th, a user on a real estate forum posted that in the ...

Figure 6: The Top 5 biggest clusters in joint embedding (Multimodal Model) under K-Means. Each row contains 5 samples from the 10% samples that are closest to the cluster centers for each cluster. The total number of clusters are 25.

5 Evaluations

5.1 Data-driven performance measure of clustering results using S_DbW index

We first calculated the S_DbW index, where a lower score indicates better clustering performance. For every embedding combination method, we also used HDBSCAN instead of K-means to group the same embeddings.[14] To ensure a fair comparison, we set K to 25 (5 for each modality in label-level combination).

Table 2 shows that both vector-level combination and joint embedding consistently achieve lower S_DbW scores than label-level combination. This pattern is true regardless of using K-means or HDBSCAN for grouping. This pattern holds across a wide range of K values (Figure 7) for K-means, providing quantitative evidence that label-level combination produces lower-quality clusters. From the results in Figure 7, among the two joint-clustering methods, joint embedding slightly outperforms vector-level combination under K-Means when $K \leq 25$, but performs slightly worse when $K > 25$.

Cluster Algorithm	**Method 1: label-level combination**	**Method 2: vector-level combination**	**Method 3: joint embedding**
K-Means	1.07466	0.76451	0.63964
HDBSCAN	0.38097	0.16604	0.16708

Table 2: S_DbW Score for Different Feature Extraction Scheme and Algorithm, setting K as 25 for all cells in this table. Note that this choice is to ensure comparisons across all conditions; it may not be the optimal choice for real tasks.

5.2 Data Loss

Our earlier diagnostic shows that the CASM dataset exhibits low cross-modality correlation, which increases the risk of dimensional explosion. To understand how each workflow handles high cluster granularity, here we evaluate the case under dimensional explosion when the

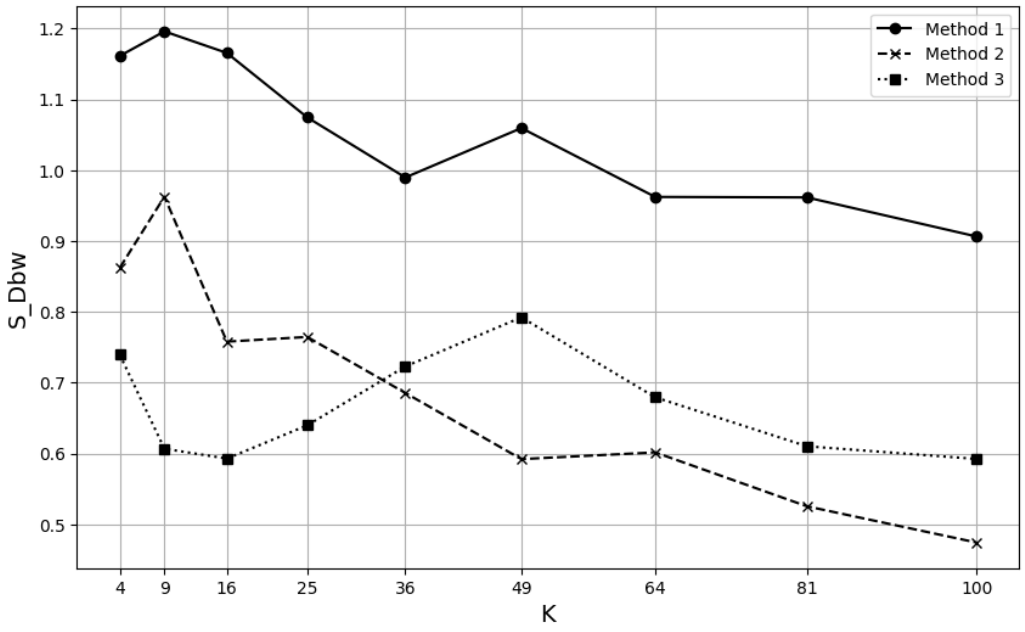
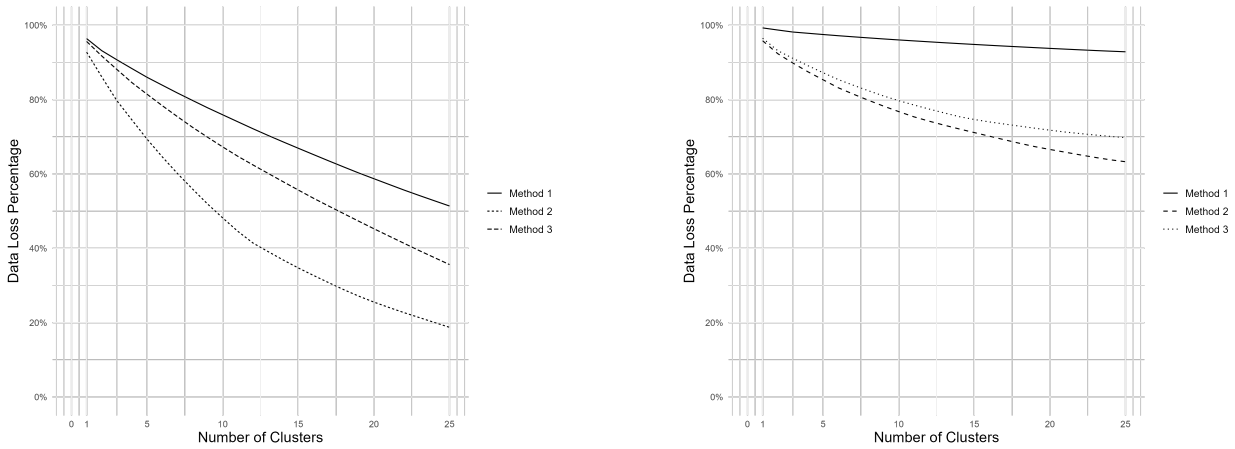


Figure 7: S_DbW as a function of number of clusters on the CASM dataset using K-means as the grouping algorithm.

K for K-Means at both modalities are 10, resulting in 100 different combinations of labels to prune, and comparing this workflow with vector combination and joint embedding when K-Means setting $K = 100$. We also provide the data loss rate in pruning by selecting top-N clusters under the clustering algorithm of HDBSCAN. Consistent with this, Figure 8 shows that label-level combination results in substantially higher data loss due to the proliferation of small, unaligned clusters (see Appendix C.3 for more details). In contrast, vector-level combination and joint embedding produce far fewer small clusters, leading to significantly lower data loss.

5.3 Comparing Clustering Algorithms

While clustering algorithm selection is not the primary focus of this study, it still influences the results. The S_DbW scores presented in Table 2 show that HDBSCAN outperforms K-Means by producing more compact and well-separated clusters. This advantage is expected, as K-Means assigns all observations—including noisy or hard-to-cluster points—to a cluster,



(a) K-Means

(b) HDBSCAN

Figure 8: Data loss (Y-axis) as a function of the number of top clusters retained (ranked by size) for each method.

whereas HDBSCAN is able to identify and discard such points as noise.

However, this strength comes at a cost: HDBSCAN results in substantially higher data loss (Figure 8). For label-level combination in particular, the magnitude is notable—if we were to rerun the results in Section 4.2 using HDBSCAN, the data loss would be nearly double that of K-Means.

Lastly, for each set of results presented in the main text, we also provide corresponding versions using HDBSCAN in the Appendix. For label-level combination, the results of text clustering are discussed in Appendix C.1, and image clustering results are in Appendix C.2. Results for vector-level combination are shown in Figure D.1, and results for joint embedding are presented in Figure D.2. We find that, overall, HDBSCAN identifies similar topic contents to those found by K-Means when other parameters are held constant. Human inspection did not reveal substantial differences between the two.

Since there is no clear winner, we use K-Means in our main analysis to reduce data loss and maintain consistent interpretability across all methods. We acknowledge, however, that our comparison is limited. Future work could extend this analysis by evaluating a broader range of clustering algorithms.

5.4 Human Validation

Last, we relied on human coders to calculate within-cluster consistency for three methods, with K ranging from 20, 25, and 30. Coding procedure is briefly described in Section 3.4.3 and in greater detail in Section E. The higher the within-cluster consistency, the more thematically similar the images within a category are, indicating a stronger clustering result.

Figure 9 shows the average within-cluster consistency. The full result, which includes each cluster's within-cluster consistency, is shown in Figure E.1. Joint embedding yields the best-performing model ($K = 20$), but also the worst-performing model ($K = 25$) in terms of maximizing within-cluster consistency.

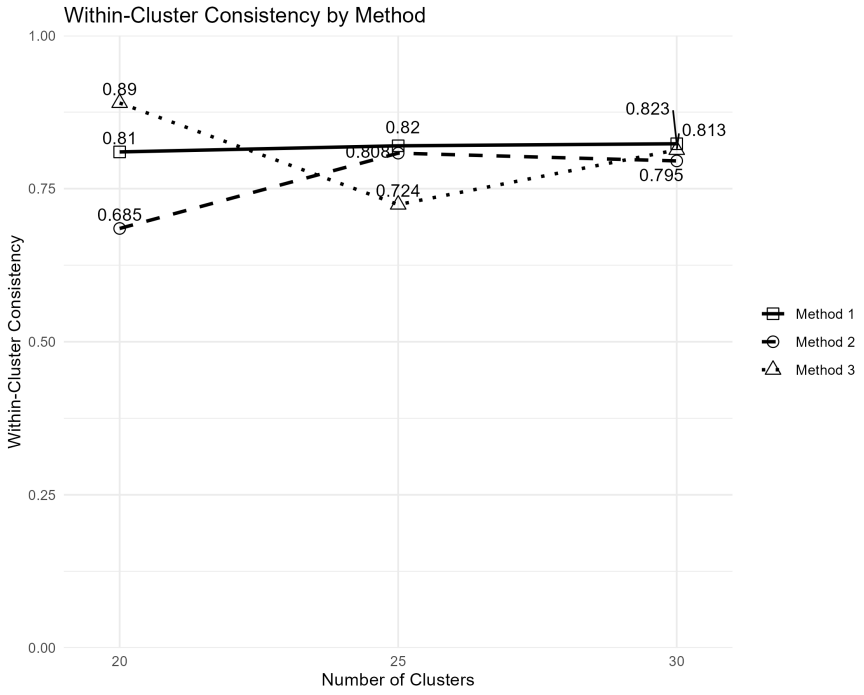


Figure 9: Average human-coded within-cluster consistency by methods and the number of clusters K

Overall, we find no clear winner between vector-level combination and joint embedding. While joint embedding is more theoretically appealing and performs better when $K \leq 25$, vector-level combination achieves better S_DbW scores when $K > 25$ and produces slightly more interpretable clusters. Human validation results for joint embedding are mixed across

different configurations. Importantly, both methods consistently outperform the first label-level combination across all metrics: internal clustering quality, ease of interpretation, data retention, and theoretical soundness.

6 Joint Text-Image Clustering Robustness Across Domains

To evaluate how our multimodal clustering methods perform in different content domains, we replicate our analysis on a second dataset: BU-NEmo, a multimodal dataset of U.S. gun violence news comprising 1,297 headline–image pairs from 840 articles across 21 media outlets (Reardon et al., 2022). Compared with CASM-China, this dataset differs across several dimensions: it is smaller, English-language, U.S.-based, and produced in a professional journalistic setting. These differences allow us to test the robustness of the joint text-image clustering pipeline and to assess how well the proposed clustering strategies generalize across sociopolitical and media contexts.

Importantly, we expect cross-modal alignment to be stronger in BU-NEmo, since journalists deliberately select images to illustrate headlines. In such settings, the benefit of joint clustering may be reduced. This section therefore also serves as a test of the scope conditions under which joint clustering adds value.

6.1 Embedding models

For label-level and vector-level combinations, we embed text with OpenAI's `text-embedding-3` model and images with Google Vertex-AI's multimodal embedding service.[15] Joint embedding applies this same Google multimodal model directly to the image-text pairs.

We moved away from the ResNet (Places365) and CLIP embeddings used in our CASM analysis for two main reasons. First, the embedding models we adopt here are trained on image–text corpora that are several orders of magnitude larger than those used for earlier

models. These larger training sets often yield more robust and general-purpose embeddings, which is advantageous in the context of BU-NEmo's diverse and professionally curated content. In contrast, our main CASM dataset consists of informal, user-generated content centered on protest scenes, where domain-aligned models like Places365 (scene classification) does not prevent us from finding meaningful categories. CASM is also over 300 times larger than BU-NEmo, making earlier-generation embedding models substantially more computationally and economically efficient for large-scale processing. We did compare these newer embeddings with older embedding's results in Section 6.5.

6.2 Clustering

We used the marginal gain methods to select K. It appears here that $K = 25$ is also a good choice (See Figure F.2 in the appendix). We used k-means to perform clustering.[16]

6.3 Results

Label-level combination Text-only clustering groups news headlines into five themes: (1) political debate over gun-control laws, (2) coverage of specific mass-shooting incidents, (3) school-safety concerns, (4) NRA and gun-sale regulation stories, and (5) disputes over 3-D-printed guns (Table F.1). Image-only clustering groups the photos into five themes: (1) advocacy and protest imagery, (2) community vigils and memorials, (3) policy-debate scenes (speeches, rallies, 3-D-printed guns), (4) justice-process visuals (courtrooms, mugshots, mourning), and (5) law-enforcement and security response (Figure F.3).

Cross-modal alignment As we expected, the adjusted mutual information (AMI) between text-only and image-only clusters is 0.166 when $K = 5$ for each modality,[17] compared with 0.029 on CASM, indicating moderate correspondence. Therefore, we predict that the added value of joint clustering will be rather limited for this dataset.

Joint clustering results Indeed, vector-level combination and joint embedding largely recover the core themes identified by label-level combination, such as political debate, mass shootings, and memorials. The detailed results are shown in Figure F.6 and F.7. This is expected from near-zero AMI scores. However, they also capture additional distinctions not observed in label-level results. For example, they more clearly separate visual coverage of victims from that of suspects in hate-crime reporting. Joint embedding further isolates smaller, specific narratives—such as celebrity-led school-shooting vigils at music award ceremonies and political events like the Kavanaugh nomination—that were previously merged into broader clusters.

6.4 Data-driven performance measure of clustering results using S_DbW index

Joint embedding achieves the lowest (best) S_DbW score (0.841), followed by label-level combination (0.856) and vector-level combination (0.879).[18] The differences among methods are modest, with S_DbW scores differing by no more than 4%—unlike the CASM dataset, where S_DbW scores were halved when shifting from label-level combination to the other two methods.

For data loss, label-level combination performs best: only about 8% of documents are pruned when reduced to the top 25 clusters, whereas vector-level combination and joint embedding discard slightly more observations (Figure F.8). Overall, these findings suggest that when AMI is high—indicating strong cross-modal alignment—all three approaches offer similar clustering performance.

6.5 Testing alternative embedding configurations

To assess whether the newer embedding models actually improve clustering performance, we also tested the configurations previously used in our CASM study:

- New: Google Vertex-AI for images; OpenAI for texts.
- Same as the first dataset: ResNet-Places365 for images paired with BERT for text
- Same image embedding; text upgraded: ResNet-Places365 for images paired with OpenAI's text embedding for text.[19]

Table 3 shows that switching to older ResNet (Places365) and BERT embeddings degraded clustering performance. For every method, using newer embedding models improves clustering performance, highlighting the advantage of general-purpose embedding models trained on diverse datasets over task-specific models.

Importantly, the performance gains from newer embeddings are comparable to or smaller than the gains from different combination approaches. In fact, using older embeddings with the theoretically most advanced approach (joint embedding, Method 3) yields results nearly identical to using newer embeddings with label-level combination method.

Table 3: Embedding performance comparison using S_DbW score (K=25). Method 3 has no entry in the third row because it uses joint embeddings and cannot vary text and image embedding models separately.

Embedding Combination Method	**Method 1: label-level combination**	**Method 2: vector-level combination**	**Method 3: joint Embedding**
New	0.856	0.879	0.841
Same as first dataset	0.901	0.899	0.844
Same image embedding; text embedding upgraded	0.890	0.895	NA

Summary We replicated our joint text–image clustering approach on the BU-NEmo gun violence dataset to test its generalizability and refine the scope conditions under which joint clustering adds value. In this dataset—where text and image content are more strongly aligned—the added benefits of joint clustering were less pronounced. Joint methods still produced coherent groupings, but their differences with label-level combination were small.

Ultimately, the choice of method should reflect the degree of cross-modal correlation and the analytical goals of the researcher.

7 Conclusions

This study proposes and compares three methods for unsupervised clustering of multimodal data containing both text and images—such as social media posts or news articles. We recommend using pre-trained models to transform each modality into dense vector representations (embeddings), followed by standard clustering algorithms such as K-Means or HDBSCAN. Through a combination of internal clustering metrics, diagnostic tools, and human validation, we show that clustering based on joint text–image embeddings consistently outperforms the simple alternative of clustering each modality separately and combining their labels (label-level combination), when texts and images convey different information that cannot be predicted from each other.

We compare two main approaches for constructing joint representations: vector-level combination, which concatenates embeddings from separate text and image embedding models; and joint embedding, which uses multimodal models trained to project both modalities into a shared semantic space. While joint embedding is often considered superior from a machine learning perspective, we find that its empirical advantage is modest and context-dependent.

Our contribution extends beyond strategies to turn texts and images into numeric representations. We introduce a set of practical evaluation tools for researchers working with multimodal clustering: (1) a diagnostic based on Adjusted Mutual Information (AMI) to assess text–image alignment, (2) the use of marginal gains of S_DbW index to determine the optimal number of clusters, and (3) a human-coded within-cluster consistency check to validate interpretability. These tools help clarify when joint clustering is most useful and how to evaluate it rigorously.

By applying our clustering pipeline and evaluation toolkit, we reveal sensitivity at all three steps of the pipeline. Combination strategy matters most: switching from label-level combination to either joint method reduced S_Dbw errors by nearly half on the Chinese protest dataset (CASM). Embedding model choice also affects results, but changes in embedding models generally had a smaller impact on clustering quality than the choice of combination strategy. Clustering algorithm and K value matter as well, though once K is selected via the marginal-gain rule, K-Means and HDBSCAN produced broadly similar thematic structures.

To assess clustering robustness, we recommend checks at three steps. For embeddings, rerun the pipeline with alternative text, image, or joint embedding models; stable cluster structure suggests robust results. For embedding combination, report AMI scores and compare at least two combination strategies. For clustering, visualize diagnostic metrics, ensure clusters persist across reasonable K values, and test additional clustering algorithms. While not every check is necessary, even a subset provides valuable validation.

This study contributes to the computational social science literature by bridging the gap between the text-as-data and image-as-data communities. Although both approaches have gained traction independently, they are rarely integrated despite many datasets being inherently multimodal. Our work demonstrates how the two can be combined and evaluated systematically, offering a roadmap for future research. As digital content becomes increasingly multimodal, the need for multimodal analytical frameworks will only grow.

Finally, while our study focuses on clustering with text and image data, our framework is extensible. Future work might incorporate other modalities—such as audio, video, or geospatial context—and evaluate clustering strategies that can scale across these richer forms of data. We also acknowledge that the space of multimodal AI models is evolving rapidly. Rather than exhaustively benchmarking every model, our goal is to equip researchers with a practical toolkit to make informed, context-sensitive choices about representation, clustering, and evaluation. We hope this work encourages scholars to treat multimodal content in its

natural form rather than reducing it prematurely to one modality or analytical lens.

Acknowledgement

We thank Yilang Peng, Lincoln Quillian, Jaye Seawright, Oscar Stuhler and the anonymous reviewers for their advice to improve this paper. We thank Daphne Ying Deng and Jiayi Su for their research assistance.

Funding

The author(s) declare that they received no funding for this research.

ENDNOTES

[1]A Twitter post or newspaper article typically uses only 1% of English vocabulary; the remaining 99% results in zeros in the document-term matrix.

[2]Topic-model-based representations like LDA are not appropriate here, because they produce very high-dimensional sparse vectors (often over 10,000 dimensions), incompatible with the much lower dimensionality of typical image embeddings.

[3]`https://github.com/mlfoundations/open_clip`

[4]OpenAI has not updated their multi-modal embedding models since CLIP.

[5]Popular choices such as the Elbow method only captures within-cluster dispersion.

[6]Some posts lack images and contain only text. Some images are corrupted or truncated to 0 bytes. We only use data that can be processed by the transfer learning models.

[7]For preprocessing, we used Jieba package in tokenization of the posts into word tokens, `https://github.com/fxsjy/jieba`

[8]The detailed stop words list is available at our online repository: `https://osf.io/gwbv6/`

[9]The result using the same set of image vectors but using HDBSCAN as the clustering algorithm is shown in Appendix C.2

[10]Specifically, we take the matrix of category-pair proportions, negate it, and feed it to the Hungarian algorithm to find the permutation that minimizes the total off-diagonal cost—equivalently, maximizes the

diagonal trace on the original matrix.

[11]https://osf.io/gwbv6/

[12]The model description is at: https://huggingface.co/sentence-transformers/clip-ViT-B-32-multilingual-v1.

[13]https://osf.io/gwbv6/

[14]For the HDBSCAN parameters, we set the minimum cluster size to 1000 and α to 0.6. A lower α value reflects a higher tolerance for including less-dense points in clusters. Because HDBSCAN's hierarchical structure allows some data points to remain unclustered (i.e., treated as outliers), we excluded these unclustered points when calculating the S_DbW index.

[15]https://cloud.google.com/vertex-ai/generative-ai/docs/embeddings/get-multimodal-embeddings. OpenAI does not currently offer a dedicated image-embedding model.

[16]In our testing of this dataset, we found that using $K = 10$ in each modality ($K = 100$) could find more meaningful clusters and achieve lower S_DbW scores. We provide the detailed results for $K = 100$ in Appendix F.4. For comparison purposes, we still use $K = 25$ in the main text.

[17]AMI is 0.232 when $K = 10$ for each modality.

[18]When using $K = 10 \times 10$, the S_DbW scores improve across all methods: 0.722 for label-level combination, 0.759 for vector-level combination, and 0.701 for joint embedding. Although higher K improves cluster compactness, it also substantially increases interpretation burden.

[19]text-embedding-3.

REFERENCES

Auxier, Brooke and Monica Anderson. 2021. "Social media use in 2021." *Pew Research Center* 1:1–4.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. "Latent Dirichlet Allocation." *J. Mach. Learn. Res.* 3:993–1022.

Cai, Yongshun. 2010. *Collective Resistance in China: Why Popular Protests Succeed or Fail*. Stanford University Press.

Campello, Ricardo J. G. B., Davoud Moulavi, and Joerg Sander. 2013. "Density-Based

Clustering Based on Hierarchical Density Estimates." In *Advances in Knowledge Discovery and Data Mining*, edited by Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, Lecture Notes in Computer Science, pp. 160–172, Berlin, Heidelberg. Springer.

Caron, Mathilde, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. "Deep Clustering for Unsupervised Learning of Visual Features." In *Computer Vision – ECCV 2018*, edited by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, Lecture Notes in Computer Science, pp. 139–156, Cham. Springer International Publishing.

Casas, Andreu and Nora Webb Williams. 2017. "Images That Matter: Online Protests and the Mobilizing Role of Pictures." SSRN Scholarly Paper ID 2832805, Social Science Research Network, Rochester, NY.

Casas, Andreu and Nora Webb Williams. 2019. "Images That Matter: Online Protests and the Mobilizing Role of Pictures." *Political Research Quarterly* 72:360–375.

Chang, Jonathan, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. 2009. "Reading Tea Leaves: How Humans Interpret Topic Models." In *Advances in Neural Information Processing Systems 22*, edited by Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, pp. 288–296. Curran Associates, Inc.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* .

Gibson, Rhonda and Dolf Zillmann. 2000. "Reading between the Photographs: The Influence of Incidental Pictorial Information on Issue Perception." *Journalism & Mass Communication Quarterly* 77:355–366.

Goldberg, Yoav and Omer Levy. 2014. "Word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method." *arXiv:1402.3722 [cs, stat]* .

Grimmer, Justin, Margaret E. Roberts, and Brandon M. Stewart. 2021. "Machine Learning for Social Science: An Agnostic Approach." *Annual Review of Political Science* 24:null.

Grimmer, Justin and Brandon M. Stewart. 2013. "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts." *Political Analysis* 21:267–297.

Grootendorst, Maarten. 2020. "BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics."

Halkidi, Maria, Yannis Batistakis, and Michalis Vazirgiannis. 2002. "Clustering validity checking methods: Part II." *ACM Sigmod Record* 31:19–27.

Hastie, Trevor, Robert Tibshirani, Jerome Friedman, T. Hastie, J. Friedman, and R. Tibshirani. 2009. *The Elements of Statistical Learning*, volume 2. Springer.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep Residual Learning for Image Recognition." In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Joo, Jungseock and Zachary C Steinert-Threlkeld. 2022. "Image as data: Automated content analysis for visual presentations of political actors and events." *Computational Communication Research* 4.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Networks." In *Advances in Neural Information Processing Systems 25*, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, pp. 1097–1105. Curran Associates, Inc.

Mayer, Richard E. 2002. "Multimedia learning." In *Psychology of learning and motivation*, volume 41, pp. 85–139. Elsevier.

McInnes, Leland, John Healy, Steve Astels, et al. 2017. "hdbscan: Hierarchical density based clustering." *J. Open Source Softw.* 2:205.

McInnes, Leland, John Healy, and James Melville. 2020. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction."

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pp. 3111–3119, USA. Curran Associates Inc.

Paivio, Allan. 1990. *Mental representations: A dual coding approach.* Oxford university press.

Peng, Yilang. 2018. "Same candidates, different faces: Uncovering media bias in visual portrayals of presidential candidates with computer vision." *Journal of Communication* 68:920–941.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning. 2014. "Glove: Global vectors for word representation." In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Powell, Thomas E, Hajo G Boomgaarden, Knut De Swert, and Claes H De Vreese. 2015. "A clearer picture: The contribution of visuals and text to framing effects." *Journal of communication* 65:997–1017.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. "Learning Transferable Visual Models From Natural Language Supervision." *CoRR* abs/2103.00020.

Reardon, Carley, Sejin Paik, Ge Gao, Meet Parekh, Yanling Zhao, Lei Guo, Margrit Betke, and Derry Tanti Wijaya. 2022. "BU-NEmo: an Affective Dataset of Gun Violence News." In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 2507–2516, Marseille, France. European Language Resources Association.

Sievert, Carson and Kenneth Shirley. 2014. "LDAvis: A method for visualizing and interpreting topics." In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pp. 63–70.

Srinivasan, Krishna, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. "WIT: Wikipedia-based Image Text Dataset for Multimodal Multilingual Machine Learning." *CoRR* abs/2103.01913.

Steinert-Threlkeld, Zachary C, Alexander M Chan, and Jungseock Joo. 2022. "How state and protester violence affect protest dynamics." *The Journal of Politics* 84:798–813.

Sweller, John, Jeroen JG Van Merrienboer, and Fred GWC Paas. 1998. "Cognitive architecture and instructional design." *Educational psychology review* 10:251–296.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention is all you need." *Advances in neural information processing systems* 30.

Vinh, Nguyen Xuan, Julien Epps, and James Bailey. 2009. "Information theoretic measures for clusterings comparison: is a correction for chance necessary?" In *Proceedings of the 26th annual international conference on machine learning*, pp. 1073–1080.

Wilkerson, John and Andreu Casas. 2017. "Large-scale computerized text analysis in political science: Opportunities and challenges." *Annual Review of Political Science* 20:529–544.

Williams, Nora Webb, Andreu Casas, and John D Wilkerson. 2020. *Images as data for social*

science research: An introduction to convolutional neural nets for image classification. Cambridge University Press.

Wittenberg, Chloe, Ben M Tappin, Adam J Berinsky, and David G Rand. 2021. "The (minimal) persuasive advantage of political video over text." *Proceedings of the National Academy of Sciences* 118:e2114388118.

Zhang, Han and Jennifer Pan. 2019. "Casm: A deep-learning approach for identifying collective action events with text and image data from social media." *Sociological Methodology* 49:1–57.

Zhang, Han and Yilang Peng. 2024. "Image clustering: An unsupervised approach to categorize visual data in social science research." *Sociological Methods & Research* 53:1534–1587.

Zhou, Bolei, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. 2014. "Learning deep features for scene recognition using places database." *Advances in neural information processing systems* 27.

Appendix A Evaluation Details

A.1 Measuring Clustering Alignment with Adjusted Mutual Information

Mutual information (MI) is a widely used information criterion that quantifies how much knowing one clustering reduces uncertainty about the other. However, computing the raw mutual information between text and image clustering can be misleading, since even two random partitions will share some information purely by chance. This caveat becomes more salient when the number of clusters increases.

To address this issue, we use the adjusted mutual information proposed by Vinh et al. (2009).[20]

A.1.1 Formal Definition

Let $T(x)$ and $I(x)$ denote the text-only and image-only cluster assignments of a document x. To quantify the alignment between these clustering solutions, we compute the Adjusted Mutual Information (AMI):

$$\mathrm{AMI}(T, I) = \frac{\mathrm{MI}(T, I) - \mathbb{E}[\mathrm{MI}(T, I)]}{\max\{H(T), H(I)\} - \mathbb{E}[\mathrm{MI}(T, I)]} \tag{1}$$

where the mutual information between T and I is defined as:

$$\mathrm{MI}(T, I) = \sum_{t \in T} \sum_{i \in I} p(t, i) \log \frac{p(t, i)}{p(t) \cdot p(i)} \tag{2}$$

and the entropy is defined as:

$$H(T) = -\sum_{t \in T} p(t) \log p(t) \tag{3}$$

A.1.2 Implementation Details

Let n_{ij} be the number of samples in both text cluster T_i and image cluster I_j. We define:

$$a_i = \sum_j n_{ij} \quad \text{(size of text cluster } T_i\text{)} \tag{4}$$

$$b_j = \sum_i n_{ij} \quad \text{(size of image cluster } I_j\text{)} \tag{5}$$

$$N = \sum_{i,j} n_{ij} \quad \text{(total number of samples)} \tag{6}$$

The term $\mathbb{E}[\text{MI}(T, I)]$ represents the baseline MI one would expect if the two clusterings were independent but shared the same cluster-size distributions:

$$\mathbb{E}[\text{MI}(T, I)] = \sum_{i=1}^{|T|} \sum_{j=1}^{|I|} \sum_{n_{ij}} P(n_{ij}) \cdot \frac{n_{ij}}{N} \cdot \log\left(\frac{N \cdot n_{ij}}{a_i \cdot b_j}\right) \tag{7}$$

where $P(n_{ij})$ is the hypergeometric probability of observing n_{ij} co-occurrences given fixed marginals a_i and b_j.

By subtracting this chance-level MI and then normalizing by the maximum possible MI (also above chance), AMI rescales our measure so that:

- AMI $= 0$ when the two clusterings agree no more than random chance.
- AMI $= 1$ when they agree perfectly.

This adjustment ensures that our clustering alignment metric is robust to the number of clusters and provides a meaningful measure of agreement beyond what would be expected by chance.

A.2 Definition of S_DbW index

Let $D = \{v_i|1, ..., n\}$ be a partition of the data S into n distinct clusters by a clustering algorithm, and v_i is the corresponding centroid of each cluster.

Definition 1. *Inter − cluster Density* (ID): This index measures the average density among the clusters in the region in relation with the densities of the clusters. It is defined as below. The point v_i, v_j are the centroids of clusters c_i, c_j, and u_{ij} are the middle point of the line segment defined by v_i, v_j.

$$Dens\_bw = \frac{1}{n \cdot (n-1)} \sum_{i=1}^{n} \Big(\sum_{j=1,i}^{n} \frac{density(u_{ij})}{max(density(v_i), density(v_j))} \Big)$$

The density function of the neighborhood u is defined as below. The term *stdev* denotes the average standard deviance of all the clusters and is defined as $stdev = \frac{1}{n} \sqrt{\sum_{i=1}^{n} \|\sigma(\mathbf{v_i})\|}$. n_{ij} represents the number of tuples that are contained in the union set of cluster c_i and c_j.

$$Density(u) = \sum_{l=1}^{n_{ij}} f(x_l, u);$$

The neighborhood of a data point x is defined to be a hyper-sphere with center u and radius the average standard deviation of the clusters, *stdev*. It is defined by the function $f(x, u)$. The point x belongs to the neighborhood u if its distance is smaller or equal o the average standard deviation of the clusters.

$$f(x,u) = \begin{cases} 0, distance(x,u) > stdev \\ 1, otherwise \end{cases}$$

Definition 2. Intra-cluster Variance : This is defined by the average scattering, or the average of variances, of the clusters. σ(S) denotes the overall variance of the dataset.

$$Scatter = \frac{\frac{1}{n} \sum_{i=1}^{n} \|\sigma(\mathbf{v_i})\|}{\|\sigma(\mathbf{S})\|}$$

Finally, the S_DbW index is defined as the sum of the scatterness score and the inter-cluster density. The lower S_DbW index indicates the better performance. A low S_DbW index indicates the partition of the data have comparatively high compactness of data over each

clusters, and the clusters are all well separated between different clusters.

$$S\_DbW = Dens\_bw + Scatter$$

Appendix B Extracting embeddings for CASM dataset

B.1 Code for Generating Text Embeddings

```
from sentence_transformers import SentenceTransformer, util


# Custom Model for Transfer Learning
model_name = 'sentence-transformers/distiluse-base-multilingual-cased-v1'


# Text Model for CLIP Text Embeddings
# model_name = 'sentence-transformers/clip-ViT-B-32-multilingual-v1'


text_model = SentenceTransformer(model_name)
text_list = [...] # Pseudocode for packing the input texts into a list
text_embeddings = text_model.encode(text_list)


# Generated text embeddings could be feed to the next step
```

B.2 Code for Generating ResNet-Places365 Image Embeddings

```
import os
import pandas as pd
import numpy as np
```

```
import torch
from torch.autograd import Variable as V
import torchvision.models as models
from torchvision import transforms as trn
from torch.nn import functional as F
import cv2
from PIL import Image

# Helper functions
def recursion_change_bn(module):
    if isinstance(module, torch.nn.BatchNorm2d):
        module.track_running_stats = 1
    else:
        for i, (name, module1) in enumerate(module._modules.items()):
            module1 = recursion_change_bn(module1)
    return module

def hook_feature(module, input, output):
    features_blobs.append(np.squeeze(output.data.cpu().numpy()))

def returnTF():
# Resize the input to make sure feed into the model
    tf = trn.Compose([
        trn.Resize((224,224)),          trn.ToTensor(),          trn.Normalize(
    ])
    return tf
```

```
def load_model():
    # this model has a last conv feature map as 14x14
    model_file = './places365/wideresnet18_places365.pth.tar'
    if not os.access(model_file, os.W_OK):
        os.system('wget http://places2.csail.mit.edu/models_places365/'
        + model_file)
        os.system('wget https://raw.githubusercontent.com/csailvision/places
        master/wideresnet.py')

    import wideresnet
    model = wideresnet.resnet18(num_classes=365)
    checkpoint = torch.load(model_file, map_location=lambda storage,
loc: storage)
    state_dict = {str.replace(k,'module.',''): v for k,v in
    checkpoint['state_dict'].items()}
    model.load_state_dict(state_dict)

    # hacky way to deal with the upgraded batchnorm2D and avgpool layers
    for i, (name, module) in enumerate(model._modules.items()):
        module = recursion_change_bn(model)
    model.avgpool = torch.nn.AvgPool2d(kernel_size=14, stride=1, padding=0)

    model.eval()

    # This is the last convolution layer of the resnet
    features_names = ['layer4','avgpool']
    for name in features_names:
```

```
        model._modules.get(name).register_forward_hook(hook_feature)
    return model

# Loading the ResNet weights for Places365
model = load_model()
tf = returnTF()
params = list(model.parameters())
weight_softmax = params[-2].data.numpy()
weight_softmax[weight_softmax<0] = 0

if __name__ == '__main__':
    images_link = [...] # List of image directories
    image_embeddings = [] # Empty list to store image embeddings

# Start to generate embeddings
    for i in range(images_link):
        features_blobs = []
        img = Image.open(images_link[i])
        try:
            input_img = V(tf(img).unsqueeze(0))
            logit = model.forward(input_img)
            h_x = F.softmax(logit, 1).data.squeeze()
            probs, _ = h_x.sort(0, True)
            image_embeddings.append(features_blobs[1])
        except:
            image_embeddings.append(None)
```

B.3 Code for Generating CLIP Image Embeddings

```
import torch
import clip
import numpy as np
from PIL import Image, ImageFile

ImageFile.LOAD_TRUNCATED_IMAGES = True
device = "cuda" if torch.cuda.is_available() else "cpu"
image_dir_list = [...] # List of directory of Images
image_embeddings = [] # Empty list to store the CLIP image embeddings

# Specify training for CLIP model to use
image_model, preprocess = clip.load("ViT-B/32", device=device)

# Image should be load in batch to prevent running out of GPU Memory

for i in range(len(image_list)):
# Read Image
    image = Image.open(image_dir_list[i])
    image = preprocess(image).unsqueeze(0).to(device)

# Get CLIP image embedding and store it to CPU
    with torch.no_grad():
        image_features = model.encode_image(image)
    image_features = image_features.cpu()
    image_embeddings.append(image_features)
```

```
# Empty cached image
    del image, image_features
    torch.cuda.empty_cache()

image_embeddings = np.array(image_embeddings)
```

B.4 Training details for CASM dataset

In the training we used a computer with the following configuration:

- CPU: Intel(R) Xeon(R) Gold 5215 CPU, 2.50GHz with 40 cores
- Internal memory: 256GB
- GPU: 5 NVIDIA GeForce RTX 3090

This setup is much smaller in scale than a full university HPC cluster (which might have hundreds or thousands of cores across many nodes and much newer GPU models), but more powerful than what most individual researchers would have dedicated access to. Our computer specifications are more than adequate for this task. If researchers have university access, they could utilize their institutional HPC resources. If not, they could use commercial cloud computing services such as AWS's P3 or P4d instances, or Google Cloud's A2 instances with similar GPU capabilities.

Appendix C Robustness checks for label-level combination

C.1 Text Clustering with HDBSCAN

Table C.1 presents the clustering results using HDBSCAN. We did not use this method in our main analysis because the HDBSCAN algorithm is less frequently used in social sciences

and produces too many topics, as we will see later. From the main text, we also know that HDBSCAN has a lower S_DbW index than the K-means algorithm, which is why some data scientists prefer it over the simple K-means algorithm. Specifically, we used the flat clustering feature of the Python HDBSCAN package (McInnes et al., 2017) to fix the number K of outgoing clusters by extracting clusters from the condensed hierarchical tree. This step ensures comparability of performance across different clustering algorithms. We do find that HDBSCAN's clustering solutions find easy-to-interpret clusters with issues of focus and tactics that are not separated in the K-Means analysis: Cluster 6 of teacher strikes, Cluster 8 of taxi-driver strikes, Cluster 9 of environmental protests, Cluster 11 of peaceful sit-ins or hunger strikes, and Cluster 12 of economic fraud. HDBSCAN is a better clustering solution compared to K-Means if researchers want to have a more fine-grained understanding of the topics within the data, but is a worse solution in terms of data loss.

Table C.1: HDBSCAN Result with BERT Sentence Embedding, Selecting Top 14 Biggest Topics

Topic Label	**Cluster ID**	**Terms with Top 14 Highest TF-IDF Score**
Migrant Worker	0	Migrant Worker 农民工/民工 Seeking Wage 讨薪 Wage 工资 Owing 拖欠 Blood Money 血汗钱 Worker 工人 Boss 老板 Company 公司 Unpaid Wages 欠薪
Real Estate	1	Property Owner 业主 Real Estate Developer 开发商 Defending Rights 维权 Neighborhood 小区 Property Management 物业 Wanke 万科 Real Estate 楼盘 Delivery 交房 Shenzhen 深圳 House 房子
Forced Eviction	2	Villager 村民 Demolition 拆迁 Forced Eviction 强拆 Land 土地 Farmer 农民 Government 政府 Land Acquisition 征地 Forced 强行 Township Government 镇政府 Reparation 补偿
Traffic / Road Blocking	3	Road Blocking 堵路/拦路 Door 门口 Mobbing 闹事 City Government 市政府 Banner 横幅 Detour 绕行 Traffic Jam 堵车 Demonstration 示威 Traffic 交通
Police	4	SWAT 特警 Police 警察/公安/警察叔叔 Police Car 警车 Armed Police 武警 Deployment 出动 Door 门口 Force 力量 Mobbing 闹事
General Protests	5	Defending Rights 维权 People 老百姓/百姓 Government 政府 Rights 权益/权利 Defend 维护 Society 社会 China 中国 Law 法律
Teacher Strike	6	Teacher 教师/老师 School 学校 Student 学生 Parents 家长 Protest 抗议 School Strike 罢课 Collective 集体 Banner 横幅 Defending Rights 维权
Doctor-Patient Disputes	7	Hospital 医院 Doctor 医生 Family Member 家属 Patient 患者/病人 Yinao 医闹 Medical Worker 医护人员 Death 死亡 Mobbing 闹事 Banner 横幅
Taxi Strike	8	Taxi 出租车 Driver 司机 Ride-Hailing 专车 Strike 罢工 Didi 滴滴 Blocking 围堵 Unlicensed Taxi 黑车 Taxi-hailing 打车 Shutdown 停运 Taxi Driver 的哥
Environmental Protest	9	Dalian 大连 Waste Incineration 垃圾焚烧 Pollution 污染 Protest 抗议 Villager 村民 Chemical Factory 化工厂 PX(p-Xylene) Hangzhou 杭州 Fujia 福佳 Dahua Group 大化
Protest Tactics	11	Protest 抗议 Demonstration 示威 Voiceless 无声 Ineffective 无效 Door 门口 Opposition 反对 Collective 集体 Banner 横幅 Hunger Strike 绝食 Body 身体
Economic Frauds	12	Blood Money 血汗钱 People 老百姓 Fraud 诈骗 Company 公司 Investment 投资 Secured 担保 Government 政府 Fundraising 集资 Liar 骗子 Investor 投资人
Car Owners	13	Car Owner 车主 4S Dealers 4s 店 Car Fair 车展 Defending Rights 维权 Mercedes-Benz 奔驰 BMW 宝马 Volkswagen 大众 Consumer 消费者 Banner 横幅 Driver 司机
Law Enforcement Violence	14	Chengguan 城管 Law Enforcement 执法 Violence 暴力 Vendor 小贩/商贩 Beat 打人 Yan'an 延安 Neck 脖子 Law Enforcement Personnels 执法人员 Guangzhou 广州 Choking 掐住

C.2 Image Clustering with HDBSCAN

Figure C.1 shows the clustering results from the HDBSCAN algorithm. For simplicity, we only display the top 10 largest clusters generated by HDBSCAN, with each row representing a cluster. We randomly sampled 10 images from each cluster. Based on the results, we can identify interpretable topics within the visual information in the clustered images, such as screenshots or photos of documents, injuries to protesters, crowd gatherings (often with police presence), road blockages, blockades of government buildings, and protests involving banners. These findings are similar to those obtained from the K-means clustering.

C.3 Data Loss of label-level combination

To compare approaches, we set $K = 100$ directly for vector-level combination and joint embedding. For label-level combination, we used 10 clusters each for text and image, yielding 100 combined clusters. Figure 8 confirms our prediction: regardless of whether K-means (left panel) or HDBSCAN (right panel) serves as the final clustering algorithm, vector-level combination and joint embedding consistently produce significantly lower data loss than label-level combination. The magnitude of this data loss is substantial—selecting just the top 25 clusters under K-means excludes over 50% of observations, while the same selection under HDBSCAN excludes over 90%. Comparing the two joint methods, joint embedding slightly outperforms vector-level combination with marginally lower data loss rates.

Appendix D Robustness checks for vector-level combination and joint embedding

We provide the visualizations for the 10 biggest clusters with the clustering algorithm of HDBSCAN in Figure D.1 and Figure D.2 as robustness check. For these two results, we are using the same embedding scheme, and the flat clustering option for HDBSCAN to set

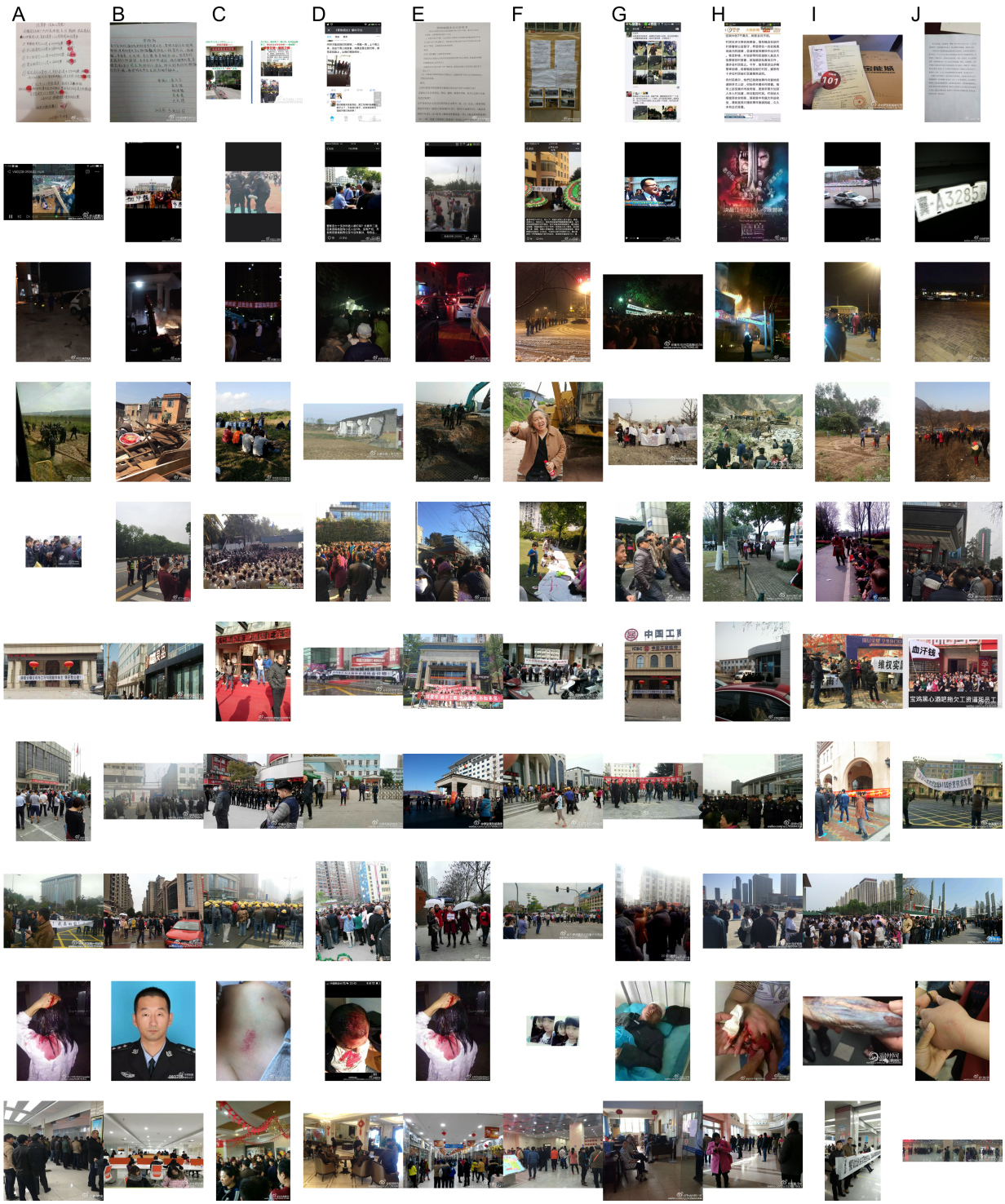


Figure C.1: The HDBSCAN clustering for image clustering. Each row show a random sample of ten images from one cluster. Only clusters with size over 100 are displayed due to space limit.

$K = 20$. The result for vector-level combination is shown at Figure D.1, and the result for joint embedding with CLIP is shown at FigureD.2.

Figure D.1: The Top 10 biggest clusters in vector-level combination (BERT + ResNet-Places365) with HDBSCAN flat clustering ($K = 20$).

D.1 joint embedding in with HDBSCAN

Figure D.2: The Top 10 biggest clusters in joint embedding (CLIP) with HDBSCAN flat clustering ($K = 20$).

Appendix E Human validation

We propose within-cluster consistency to measure how well clustering solutions identify internally coherent clusters. The procedure follows a similar approach to Zhang and Peng (2024), but extends to settings where the original data (social media posts) contain both text and images.

- Choosing documents to code: For each of the three methods (label-level combination, vector-level combination, and joint embedding), and for each K value tested (K=20, 25, 30), we randomly selected 10 original social media posts containing both text and images from each cluster. This stratified random sampling ensured representation across all clusters regardless of size, yielding a total of 3 methods $\times$ (20 + 25 + 30) clusters $\times$ 10 posts = 2,250 posts.

- Text Theme Identification: Coders assigned a common "text" theme for each cluster based on textual content only. They then gave specific labels to individual posts within the cluster.

- Image Theme Identification: Coders assigned a common "image" theme for each cluster based on visual content only. They then gave specific labels to individual images within the cluster.

- Joint Theme Identification: Coders assigned a common "joint" theme for each cluster based on the integrated content. This labeling occurred after separate text and image coding to allow comparison between unimodal and multimodal interpretations.

- Coder training: At the beginning of the coding process, three coders independently coded 10 randomly selected clusters (10 posts per each cluster) to establish baseline consistency. We then resolved discrepancies through discussion and refined our coding manual. Following this calibration phase, two coders separately completed the remaining coding.

- Cluster Label Assignment: After the human coding, the first authors picked out those descriptions as cluster names by exact keyword overlap (e.g., “road blockade,” “banner protest”) or, when wording differs, by a two-out-of-three author vote on semantic equivalence. Then, the authors rephrased the cluster labels with terms in natural language for improving interpretability.

- Consistency Score Calculation: The within-cluster consistency score (α) represents the proportion of posts in a cluster that match its primary theme (defined as the most frequently assigned theme by coders). If there are 7 posts whose theme is ”protests related to fraud in front of government office” out of the 10 sampled posts, then the consistency score is 0.7. Higher scores indicate greater thematic coherence within clusters.

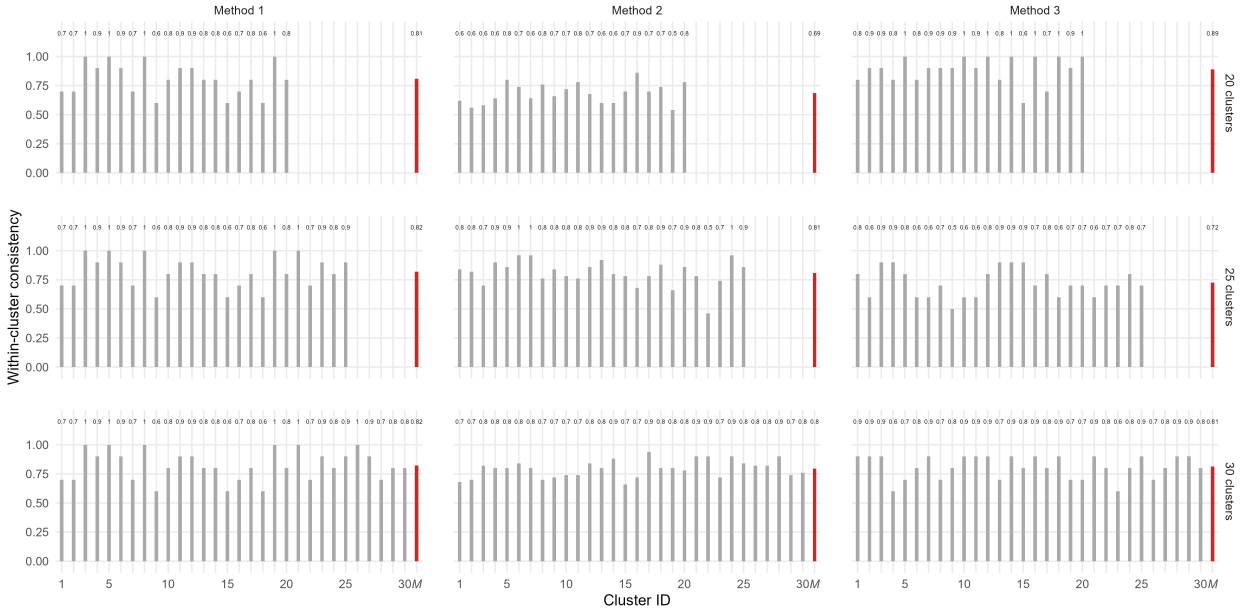


Figure E.1: Within-cluster consistency. Average within-cluster consistency (M) is highlighted in red and the exact values of the average within-cluster consistency is shown on the top of each bar. Nineteen clustering solutions are shown, varying by the number of clusters and methods to map images into vectors.

Appendix F Detailed results for gun violence dataset

F.1 Cross-modality correlations and selecting K

Figure F.1 shows the heatmap of cross prevalence with label-level combination (label-level combination). The rows list the OpenAI text labels and the columns list the Google Multimodal image cluster. Darker cells along the reordered diagonal show that certain image-cluster labels consistently accompany the same text-cluster labels, revealing moderate cross-modal alignment in this professional news dataset.

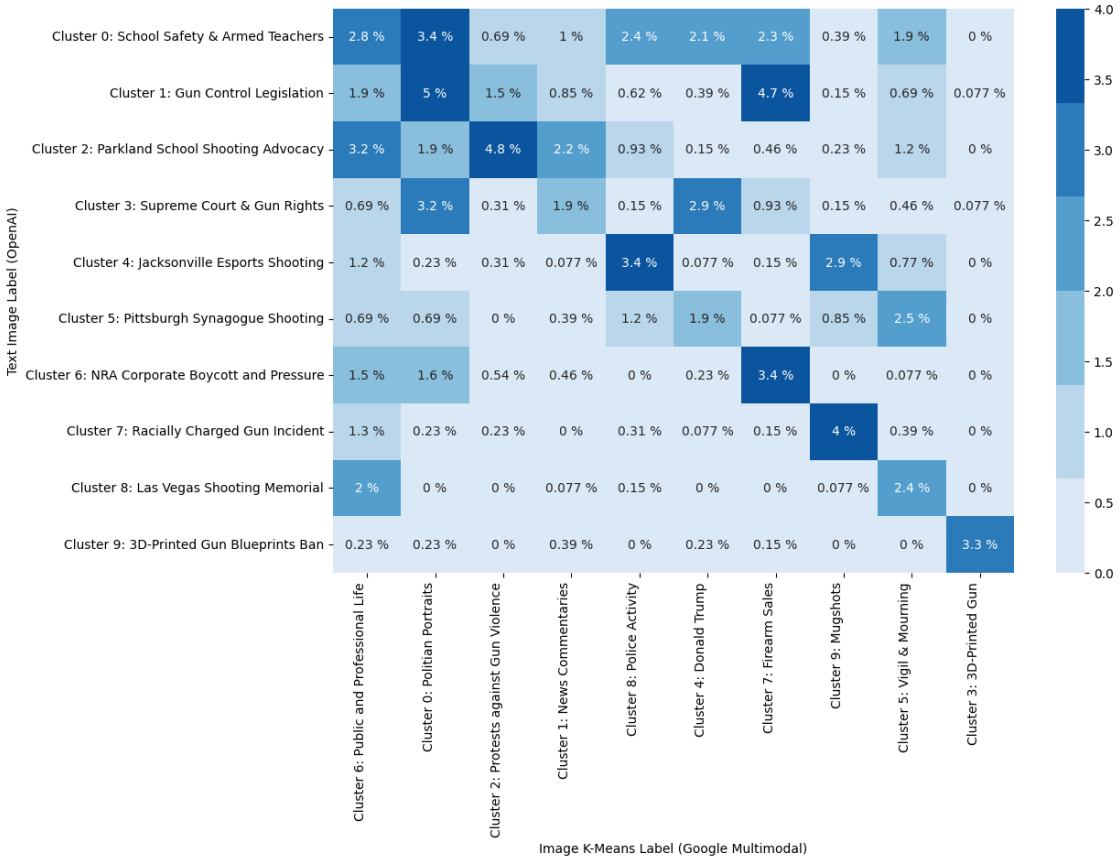


Figure F.1: Permuted co-occurrence heatmap between text labels (OpenAI) and image clusters (Google Multimodal) on the BU-NEmo dataset. Each cell shows the percentage of headline–image pairs that fall into the corresponding text–image label combination. Stronger alignment appears as darker diagonal blocks, indicating which image clusters most frequently accompany each text topic.

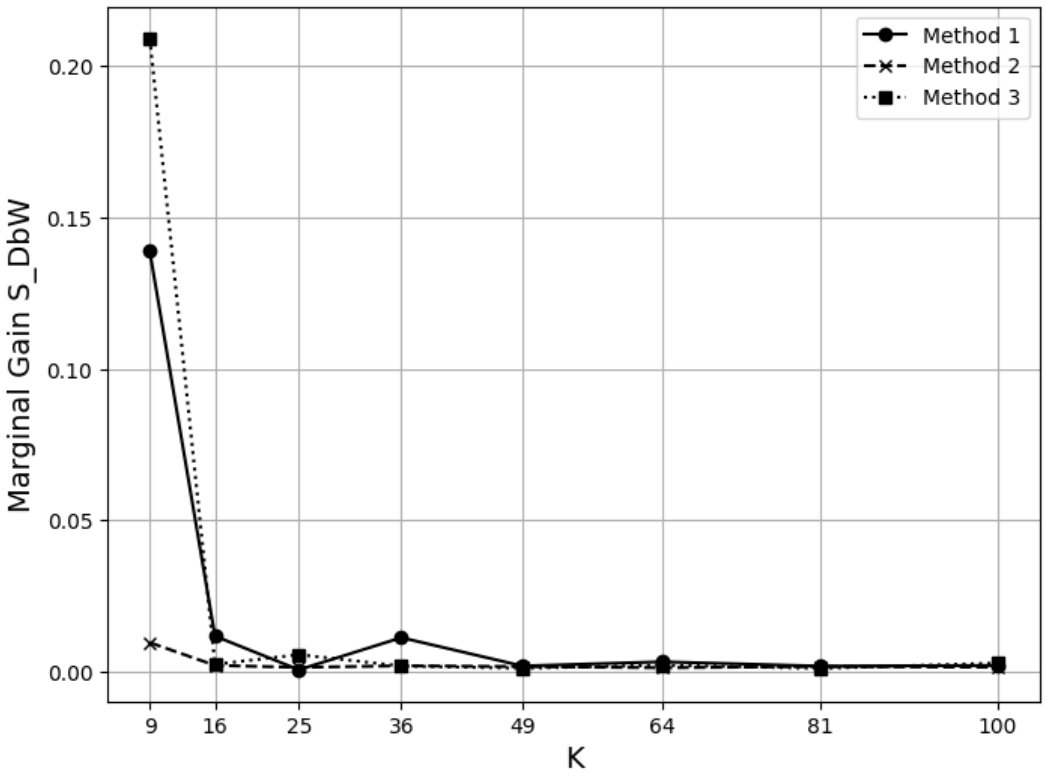


Figure F.2: Marginal absolute S_DbW gain for BU_NEmo dataset related to gun violence.

F.2 Results for the gun violence dataset (BU_NEmo): label-level combinations

F.3 label-level combination (K = 5 * 5)

For label-level combination, Table F.1 shows the textual clusters, and Figure F.3 shows the image clusters. The content for the text clusters varies in the scope for different events: from more generic news about politicians and judges' stances and actions on gun law and policies (Cluster 1), to the reports of specific shooting events and their aftermaths (Cluster 2, 4, 5, 7, 8). On the other hand, the image clusters capture visual frames from the portraits of politicians or new hosts (Cluster 0, 1, 4), to the scenes of public commemorations and vigils (Cluster 5, 6), police dispatchs (Cluster 8), trials (Cluster 9), or protests (Cluster 2) as the aftermath of shooting events.

For the image clusters, we can see that using K = 5 heavily undercuts the interpretability

ID of Topic	Posts	Words with Top 10 Highest TF-IDF Score.
0: Politicians on Gun Control	409	gun, trump, guns, control, nra, florida, california, new, laws, shooting
1: Shooting Events News Report	371	shooting, pittsburgh, synagogue, victims, man, suspect, police, shooter, black, gunman
2: School Safety	354	gun, school, shooting, violence, students, control, parkland, shootings, kavanaugh, mass
3: NRA and Gun Sale Regulation	104	nra, gun, sales, makers, business, ties, york, bank, delta, amid
4: 3-D Printed Guns	59	3d, 3dprinted, blueprints, guns, printed, judge, gun, blocks, release, plans

Table F.1: Text Clustering Results ($K = 5$) with OpenAI text embeddings and K-Means on BU-NEmo Dataset

of the clustering scheme. Since we set the number of K too low, the internal variance of meaning within the same cluster becomes too large, and it becomes hard for human annotators to assign labels for those clusters.

Figure F.3: Image Clustering Results ($K = 5$) with Google Multimodal image embeddings and K-Means on BU-NEmo Dataset.

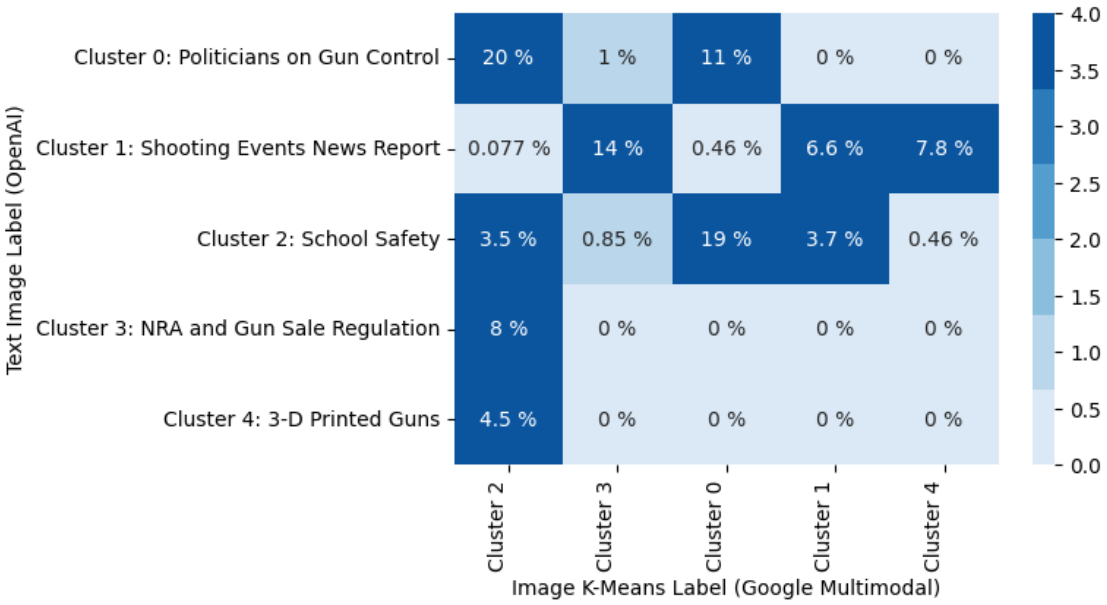


Figure F.4: Permuted co-occurrence heatmap between text labels (OpenAI) and image clusters (Google Multimodal) on the BU-NEmo dataset at $K = 5$ for each modality.

F.4 label-level combination (K = 10 * 10)

For label-level combination, Table F.2 shows the textual clusters, and Figure F.5 shows the image clusters. The content for the text clusters varies in the scope for different events: from more generic news about politicians and judges' stances and actions on gun law and policies (Cluster 1), to the reports of specific shooting events and their aftermaths (Cluster 2, 4, 5, 7, 8). On the other hand, the image clusters capture visual frames from the portraits of politicians or news hosts (Cluster 0, 1, 4), to the scenes of public commemorations and vigils (Cluster 5, 6), police dispatches (Cluster 8), trials (Cluster 9), or protests (Cluster 2) as the aftermath of shooting events.

Figure F.5 shows the image clusters produced by label-level combination (label-level combination) of the gun violence dataset with $k = 10$.

ID of Topic	Posts	Words with Top 10 Highest TF-IDF Score.
0: School Safety and Armed Teachers	219	school, shootings, guns, shooting, teachers, mass, schools, trump, gun, doctors
1: Gun Control Legislation	207	gun, california, new, ban, governor, control, signs, laws, assault, florida
2: Parkland School Shooting Advocacy	196	students, gun, violence, parkland, control, school, shooting, survivors, march, florida
3: Supreme Court and Gun Rights	141	gun, trump, nra, kavanaugh, control, amendment, second, brett, guns, trumps
4: Jacksonville Esports Shooting	117	jacksonville, madden, tournament, esports, shooting, game, gaming, gamers, security, cancels
5: Pittsburgh Synagogue Shooting	108	synagogue, pittsburgh, shooting, trump, suspect, antisemitic, bowers, hate, attack, robert
6: NRA Corporate Boycott and Pressure	102	nra, gun, sales, business, makers, ties, york, delta, bank, amid
7: Racially Charged Gun Incident	87	man, black, police, shooting, shooter, school, suspect, gunman, white, teen
8: Las Vegas Shooting Memorial	61	vegas, victims, shooting, las, year, anniversary, honor, thousand, oaks, mass
9: 3D-Printed Gun Blueprints Ban	59	3d, 3dprinted, blueprints, printed, guns, judge, blocks, gun, release, plans

Table F.2: Text Clustering Results ($K = 10$) with OpenAI text embeddings and K-Means on BU-NEmo Dataset

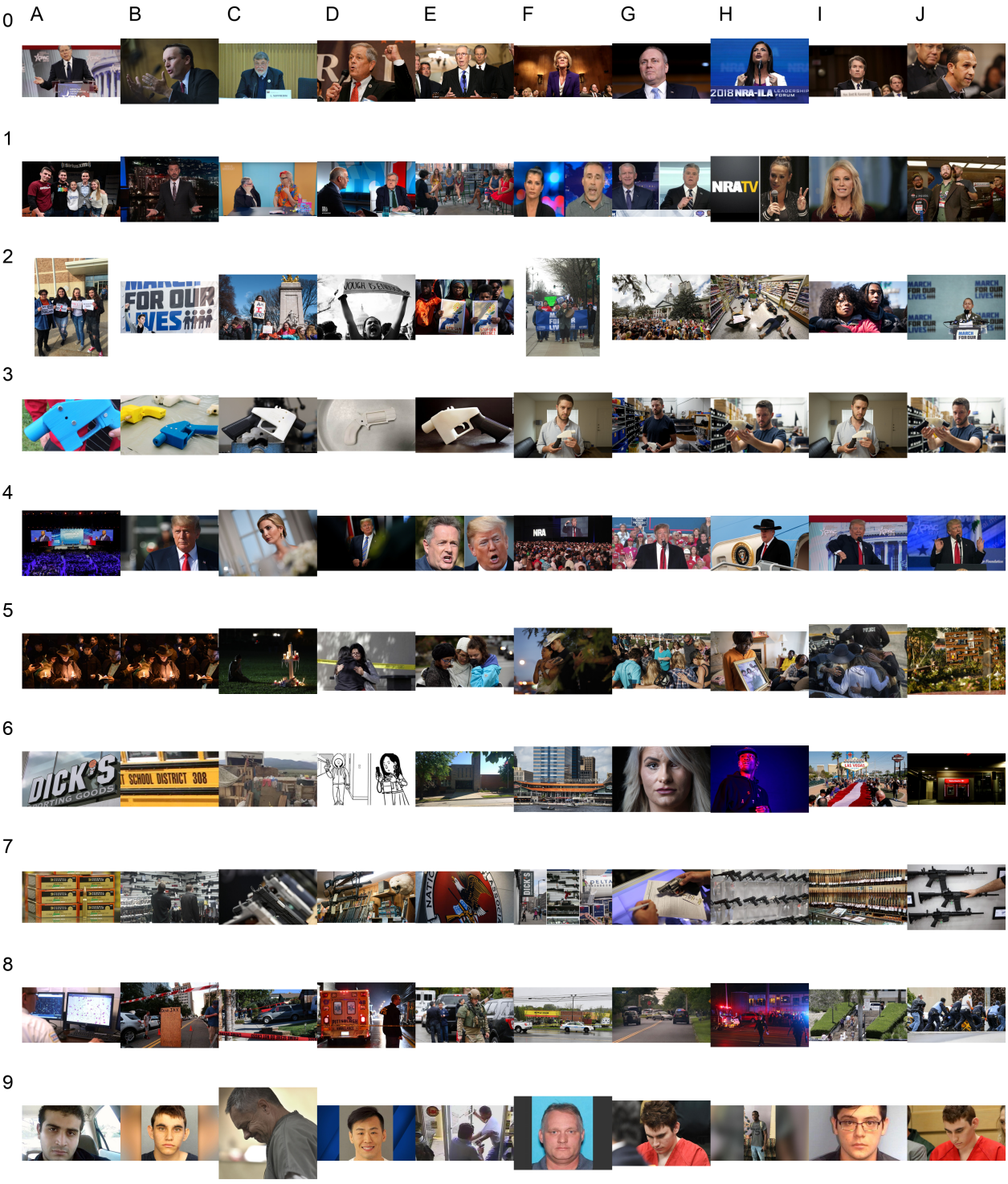


Figure F.5: Image Clustering Results ($K = 10$) with Google Multimodal image embeddings and K-Means on BU-NEmo Dataset.

F.5 Joint clustering results: vector-level combination and joint embeddings

Figure F.6 shows the 5 largest clusters ranked by their sizes for vector-level combination. Figure F.7 shows the 5 largest clusters ranked by their sizes for vector-level combination. The rest clusters can be viewed online due to space limitations.

F.6 Data loss for the gun violence dataset (BU_NEmo)

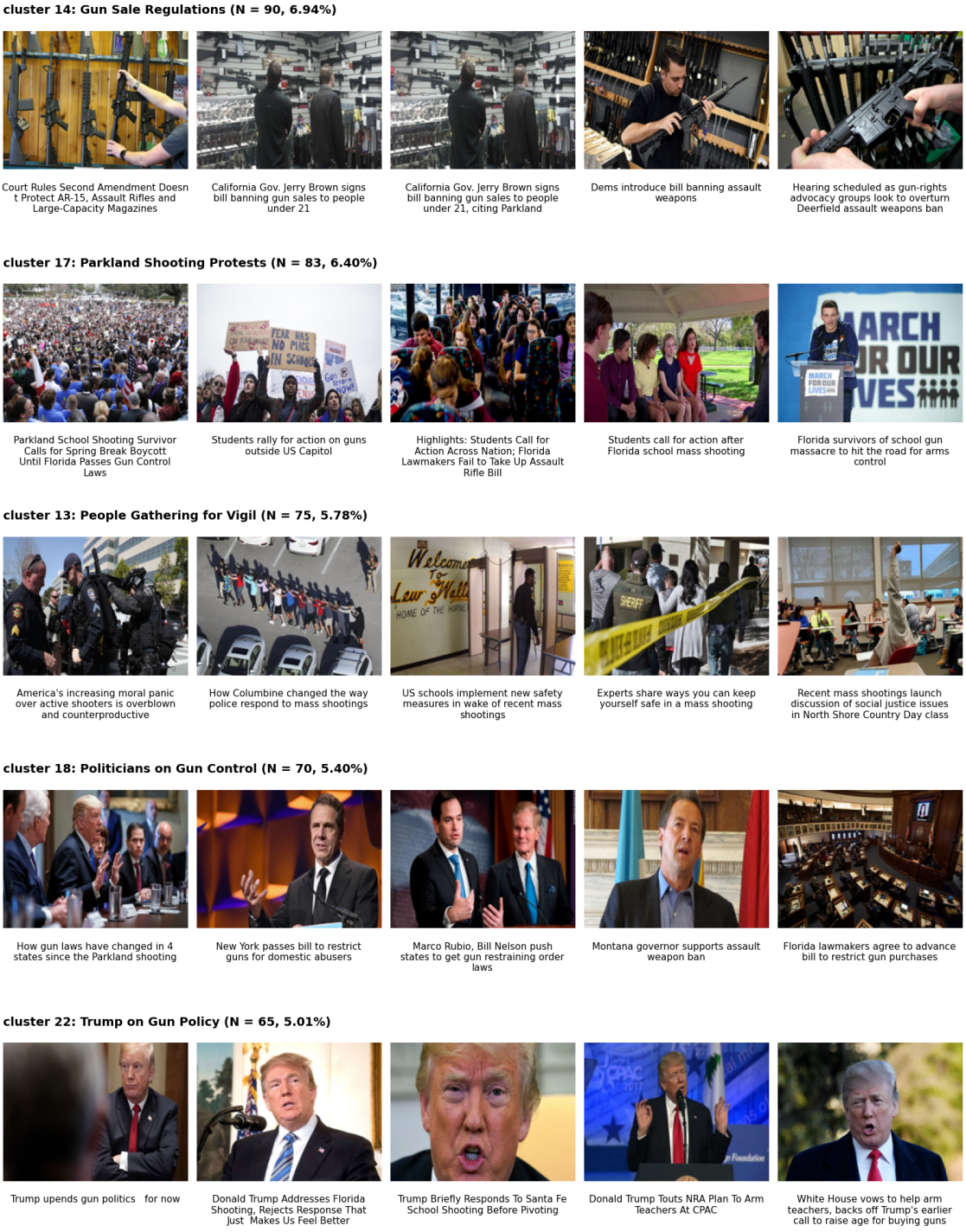


Figure F.6: Representative images and headlines for the five largest clusters produced by vector-level combination (Method 2) on the BU-NEmo dataset. Each panel shows five exemplar headline–image pairs illustrating the thematic coherence of each cluster. We used $k = 25$ in the clustering.

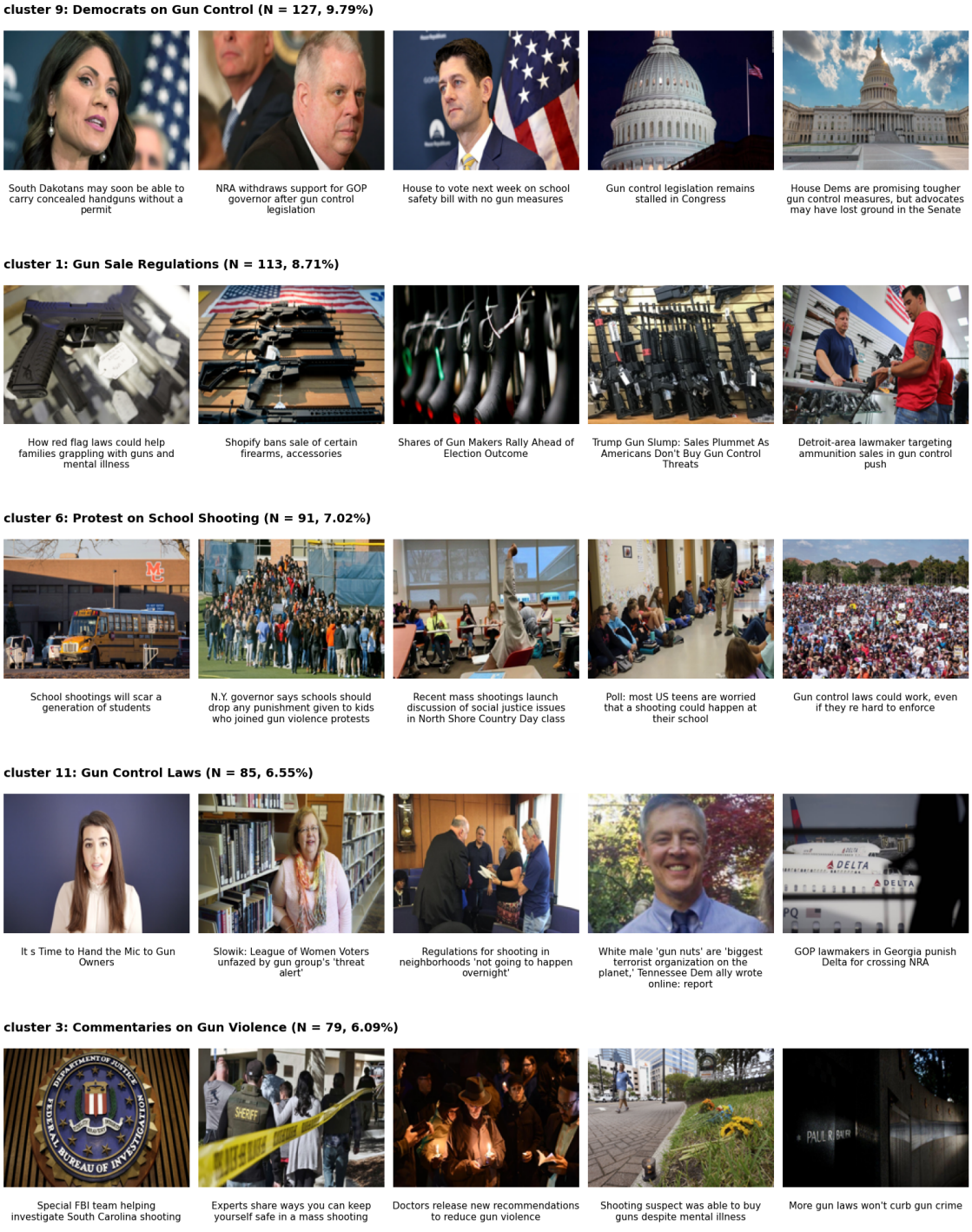
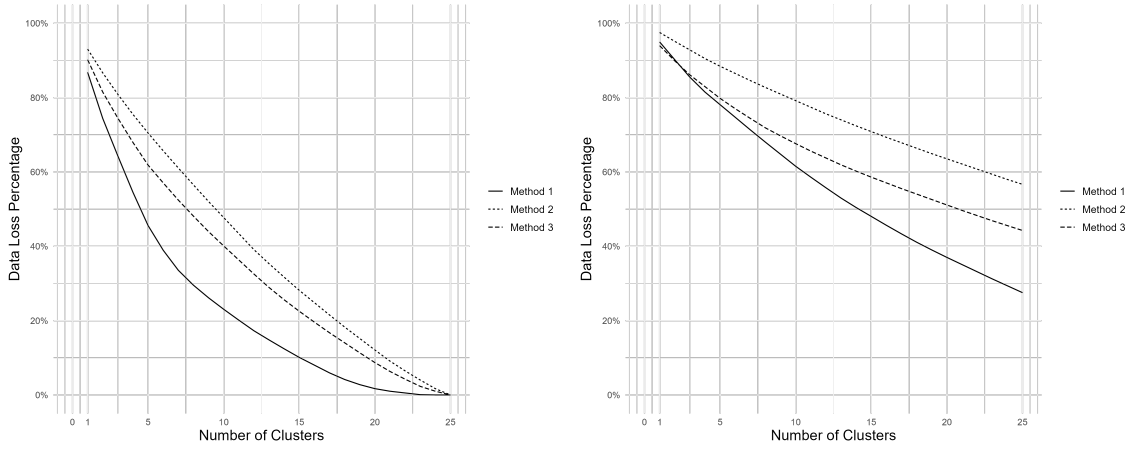


Figure F.7: Representative images and headlines for the five largest clusters produced by joint embedding (Method 3) on the BU-NEmo dataset. Each panel shows five exemplar headline–image pairs illustrating the thematic coherence of each cluster. We used $k = 25$ in the clustering.



(a) K = 25 (b) K = 100

Figure F.8: Data-loss curves on the BU-NEmo dataset, showing the fraction of documents discarded when retaining only the top k clusters for each method. label-level combination (label-level combination, blue) exhibits the lowest data-loss at every k, dropping under 10% even at $k = 25$. By contrast, vector-level combination (concatenated embedding, orange) and joint embedding (joint embedding, green) lose a larger share of observations as they prune small or noisy clusters. The data loss curves at $k = 100$ is also attached.