Appendices: Supplemental Material

The appendices contain many experiments and results that supplement the core idea of the main paper and may be of interest to some readers. As mentioned in the paper the generated motif graph does not necessarily ensure full connectivity; to attempt to alleviate that three different component connection methods were tried and are elaborated on in Section A. As the intent of this application is to be used in the real-world, example run-times are presented in Section B. Following this in Section C are the comprehensive tables containing every accuracy score collected throughout the experimentation for this work. A more in-depth discussion and exploration of how the clusters, both in their content distribution and number, are effected by various hyper-parameters is given in Section D. On the heels of this, in Section E.1, follows a discussion on the graph and centroid structures which are heavily effected by the feature type chosen. Section F includes additional qualitative results as well as a more specific results graph accounting for second-order metrics such as the number of edges in a graph. Section G has a list of all the users whose Telegram channels were scraped in order to form the data set. This is followed by a comparison between the kind of output a CBIR system might return and the output returned by a motif mining pipeline in Section H. And finally, the normalizing factor in the motif mining formalization is described in detail in Section I.

A. Graph Connection

The baseline Erdős-Rényi model takes a parameter p, which specifies the probability of adding an edge between any two components. The weights assigned to these new edges are proportional to the average weights of the edges in the two components being connected. We find p so that the expected number of new edges added to the graph is linear in the initial number of components. This avoids needlessly changing the density of the graph.

The *Best* and *Average* connection strategies work similarly to the Erdős-Rényi approach but with different strategies for determining when components get connected with each other. Given N_C total components and a proposed pair of components C_i and C_j, these algorithms compare the components by extracting their vertices' associated feature vectors. The cosine similarity of these feature vectors then determines the similarity between the two components. The *Best* approach assigns a similarity to the pair (C_i, C_j) based on the *most similar* pair of vertices found from C_i and C_j. The *Average* approach, by contrast, assigns a similarity to (C_i, C_j) based on the *average similarity* of their corresponding vertex pairs. In either case, we then find a threshold θ so that the number of pairs (C_i, C_j) with similarity scores above θ is proportional to N_C. Those pairs of components are then connected as follows: the *Best* adds edges between those vertices that had the most similar feature vectors; the *Average* approach randomly connects k-many pairs of vertices (by default, $k = 1$). These new edges are weighted in proportion to the components' similarity.

B. Runtimes

The various run times for the pipeline vary widely depending on what feature type is used for extraction. PHASH and SURF features were the quickest due to their ease of parallelization and in SURF's case, its ability to run on a GPU. MOBILE features are noticeably slower but still much faster than VGG features, which took more than twice as long as MOBILE features on the Indonesia data set. It is for this reason primarily the we recommend against using VGG features. PHASH and SURF features, while fast, achieved low scores on both the Reddit and Ukraine data set in their individual forms, and slightly higher in the combination of the two. Surprisingly the SURF_PHASH score on the Indonesia data set was quite high and comparable to the top scores. It's unclear whether this was a fluke, due to some quirk in the data set, or due to the increasing size of the data set. More work needs to be done to explore this but if speed was of the absolute essence it would be worth trying this feature combination to explore a sufficiently large data set. If speed does not matter as much we recommend a variety of the MOBILE features. It is important to keep in mind that the MOBILE features by themselves will be limited to a number of clusters equal to the number of centroids the OPQ index is initialized with and thus we prefer the SURF_MOBILE combination which allows for more clusters, thus achieving a better image/cluster ratio.

Adding images to the index is extremely quick and should not be a serious consideration when exploring motif mining. On the other hand, the graph creation, connection, and clustering have serious run time implications. Local feature querying is significantly slower than the global features due to the voting required to map back to the images from the features retrieved from the index. On top of this, the graph connection process is costly. Note that the run-times for this portion of the pipeline include all three connection methods and in practice only one would need to be used. Even with the connection methods sped up using dynamic programming the BEST and AVG connection methods still averaged seven hours approximately for the Indonesia data set. We don't believe this is worth the CPU time due to no noticeable increase in the accuracy scores on the resulting graphs. Human observers seem not to notice whether or not a graph has been connected prior to clustering. The clustering run times include all three methods on all four graphs and therefore, in an implementation in which one were to run only a single combinations, are of no serious concern.

An important note is that these run-times are from just one set of experiments and therefore should only be used as rough guidelines to how long one might expect the pipeline to run. Times might vary depending on the hardware and other activities on the machine. These run times were collected on a machine with an Intel Xeon E5-2620 v3 (12) @ 3.200GHz (CPU), 256 GB of RAM and a Titan X and Titan Z (GPUs).

Feature Extraction (Total Runtime)	Reddit (10586)	Ukraine (16433)	Indonesia (44612)
PHASH (CPU, PE=6)	00:01:02 (00:01:17)	00:00:17 (00:00:36)	00:01:59 (00:02:47)
MOBILE (CPU, PE=1)	00:41:10 (00:41:38)	00:39:51 (00:40:27)	02:04:54 (02:06:27)
VGG (GPU, PE=1)	01:41:31 (01:41:54)	02:25:53 (02:26:27)	06:58:22 (06:59:48)
SURF (GPU, PE=6)	00:05:08 (00:20:27)	00:06:20 (00:25:14)	00:14:05 (01:02:55)
SURF_PHASH (GPU/CPU, PE=6)	00:06:30 (00:23:36)	00:06:31 (00:29:12)	00:14:59 (01:15:10)
SURF_MOBILE (GPU/CPU PE=1)	01:14:49 (1:39:40)	00:56:30 (01:19:26)	02:46:13 (03:47:54)
SUF_VGG (GPU, PE=1)	02:17:33 (02:30:37)	02:35:49 (02:58:35)	07:33:56 (08:35:24)

Table 1. CPU, GPU indicates which device the feature extraction was performed on. PE gives the number of parallel processes used during the feature extraction. Due to its low overhead, PHASH is trivial to parallelize which decreases the time needed to extract features. Times are expressed in the "hours:minutes:seconds" format.

Index Add	Reddit (10586)	Ukraine (16433)	Indonesia (44612)
PHASH	00:00:02	00:00:02	00:00:02
MOBILE	00:00:02	00:00:03	00:00:03
VGG	00:00:03	00:00:02	00:00:02
SURF	00:00:31	00:00:42	00:02:46
SURF_PHASH	00:00:31	00:00:43	00:01:55
SURF_MOBILE	00:00:32	00:00:44	00:02:43
SURF_VGG	00:00:25	00:00:43	00:02:08

Table 2. The time spent to add all feature vectors to the index, for each feature type. Times are expressed in the "hours:minutes:seconds" format.

Graph/Cluster Creation	Reddit (10586)	Ukraine (16433)	Indonesia (44612)
PHASH	00:00:04/00:12:05/00:02:45	00:00:04/00:14:29/00:02:35	00:00:13/02:14:17/00:16:17
MOBILE	00:00:05/00:17:24/00:03:47	00:00:05/00:25:55/00:03:23	00:00:11/02:30:59/00:06:08
VGG	00:00:04/00:12:08/00:03:13	00:00:05/00:28:13/00:03:45	00:00:09/02:22:41/00:13:45
SURF	00:43:46/00:00:28/00:07:29	00:49:41/00:04:05/00:08:34	04:51:25/00:39:42/00:21:42
SURF_PHASH	00:32:10/00:17:18/00:03:28	00:59:48/01:44:55/00:05:30	07:11:42/14:02:41/00:13:53
SURF_MOBILE	00:43:46/00:13:07/00:03:59	01:08:49/02:33:48/00:05:38	07:57:32/14:53:36/00:15:54
SURF_VGG	00:30:07/00:09:58/00:01:53	01:04:06/01:50:55/00:09:00	05:43:41/18:56:54/00:13:42

Table 3. Times spent to create the clusters and mine the motifs, for each feature type. Times are expressed in the "hours:minutes:seconds" format.

C. Imposter-Host Accuracy Tables

Below are the full tabular results for the Imposter-Host test accuracy scores. The scores marked as N/A were invalid due to there be a number of clusters equal to the number of images in the data set (thus there was no reason to run the task). There is no apparent pattern to which graph connection method observers preferred, and for that reason we recommend against using them, for run-time purposes. However, if time is of no concern, a number of top scores were produced using the BEST connection method, which could be useful for other data. While Markov clustering produced the highest scores we recommend the Louvain method due to the better spread of images amongst the clusters.

Reddit	Louvain	Markov	Spectral
PHASH	23.46% - AVG 25.18% - BEST 24.87% - ER 25.57% - REG	38.73% - AVG 23.63% - BEST N/A - ER N/A - REG	19.52% - AVG 20.34% - BEST 23.53% - ER 18.83% - REG
MOBILE	46.62% - AVG 65.11% - BEST 58.96% - ER 57.86% - REG	45.31% - AVG 59.55% - BEST 60.43% - ER 56.49% - REG	39.44% - AVG 48.88% - BEST 49.39% - ER 46.45% - REG
VGG	21.93% - AVG 61.13% - BEST 57.27% - ER 62.00% - REG	34.28% - AVG 57.48% - BEST 64.25% - ER 58.76% - REG	24.61% - AVG 13.59% - BEST 24.79% - ER 14.95% - REG
SURF	23.29%	35.08%	39.62%
SURF_PHASH	21.49% - AVG 21.96% - BEST 25.68% - ER 26.65% - REG	29.22% - AVG 32.09% - BEST 36.23% - ER 33.26% - REG	20.49% - AVG 23.41% - BEST 23.74% - ER 08.77% - REG
SURF_MOBILE	40.28% - AVG 58.88% - BEST 55.63% - ER 56.22% - REG	63.83% - AVG 64.32% - BEST 64.96% - ER 64.67% - REG	21.26% - AVG 22.79% - BEST 17.77% - ER 26.94% - REG
SURF_VGG	37.94% - AVG 54.01% - BEST 45.87% - ER 53.90% - REG	44.01% - AVG 55.12% - BEST 49.62% - ER 57.35% - REG	22.92% - AVG 29.30% - BEST 28.53% - ER 23.49% - REG

Indonesia	Louvain	Markov	Spectral
PHASH	32.53% - AVG 17.90% - BEST 32.02% - ER 30.12% - REG	N/A - AVG N/A - BEST N/A - ER N/A - REG	31.07% - AVG 31.81% - BEST 31.61% - ER 28.45% - REG
MOBILE	46.04% - AVG 58.43% - BEST 60.06% - ER 65.11% - REG	19.30% - AVG 64.71% - BEST 53.42% - ER 46.85% - REG	22.55% - AVG 32.10% - BEST 32.39% - ER 35.55% - REG
VGG	34.73% - AVG 64.61% - BEST 77.05% - ER 66.92% - REG	23.72% - AVG 55.69% - BEST 52.95% - ER 73.46% - REG	30.31% - AVG 50.03% - BEST 49.44% - ER 50.07% - REG
SURF	42.67% - AVG 42.36% - BEST 46.08% - ER 41.71% - REG	60.94% - AVG 58.58% - BEST 54.73% - ER 18.18% - REG	40.70% - AVG 39.38% - BEST 44.66% - ER 39.08% - REG
SURF_PHASH	36.81% - AVG 71.95% - BEST 58.89% - ER 67.48% - REG	45.93% - AVG 81.91% - BEST 86.19% - ER 82.02% - REG	32.39% - AVG 44.16% - BEST 03.99% - ER 21.26% - REG
SURF_MOBILE	25.96% - AVG 66.99% - BEST 62.78% - ER 67.01% - REG	48.19% - AVG 93.81% - BEST 88.02% - ER 86.05% - REG	45.76% - AVG 32.26% - BEST 44.13% - ER 18.98% - REG
SURF_VGG	21.19% - AVG 38.49% - BEST 38.75% - ER 43.92% - REG	45.94% - AVG 56.61% - BEST 57.95% - ER 77.68% - REG	31.28% - AVG 21.23% - BEST 25.36% - ER 31.76% - REG

Ukraine	Louvain	Markov	Spectral
PHASH	16.98% - AVG 23.47% - BEST 23.60% - ER 25.19% - REG	N/A - AVG N/A - BEST N/A - ER N/A - REG	18.73% - AVG 19.36% - BEST 21.93% - ER 23.22% - REG
MOBILE	49.76% - AVG 66.43% - BEST 61.99% - ER 73.04% - REG	32.43% - AVG 67.68% - BEST 65.37% - ER 56.56% - REG	23.66% - AVG 55.47% - BEST 47.97% - ER 41.21% - REG
VGG	50.10% - AVG 56.08% - BEST 47.45% - ER 48.96% - REG	25.91% - AVG 63.24% - BEST 51.01% - ER 49.96% - REG	24.68% - AVG 39.51% - BEST 45.26% - ER 35.25% - REG
SURF	32.86% - AVG 32.86% - BEST 32.29% - ER 38.61% - REG	19.51% - AVG 57.36% - BEST 49.56% - ER 47.15% - REG	33.68% - AVG 27.11% - BEST 20.64% - ER 25.35% - REG
SURF_PHASH	14.39% - AVG 38.35% - BEST 29.97% - ER 37.83% - REG	39.89% - AVG 65.60% - BEST 62.26% - ER 58.62% - REG	21.78% - AVG 20.28% - BEST 05.43% - ER 10.09% - REG
SURF_MOBILE	17.35% - AVG 71.55% - BEST 54.51% - ER 66.05% - REG	44.18% - AVG 13.61% - BEST 79.91% - ER 75.77% - REG	28.89% - AVG 37.12% - BEST 59.02% - ER 14.52% - REG
SURF_VGG	31.91% - AVG 56.52% - BEST 46.76% - ER 69.95% - REG	45.82% - AVG 77.42% - BEST 75.35% - ER 75.15% - REG	30.74% - AVG 06.81% - BEST 19.88% - ER 31.81% - REG

D. Cluster Structures.

Additional information about the structure of the clusters is provided in this section.

D.1. Cluster Statistics.

Reddit	Louvain	Markov	Spectral
PHASH	244 - AVG	10586 - AVG	150 - AVG
	257 - BEST	10586 - BEST	150 - BEST
	244 - ER	10586 - ER	150 - ER
	256 - REG	10586 - REG	150 - REG
MOBILE	164 - AVG	355 - AVG	150 - AVG
	257 - BEST	394 - BEST	150 - BEST
	238 - ER	393 - ER	150 - ER
	128 - REG	161 - REG	150 - REG
VGG	127 - AVG	809 - AVG	150 - AVG
	255 - BEST	425 - BEST	150 - BEST
	236 - ER	424 - ER	150 - ER
	256 - REG	425 - REG	150 - REG
SURF	28 - AVG	760 - AVG	150 - AVG
	28 - BEST	760 - BEST	150 - BEST
	28 - ER	760 - ER	150 - ER
	28 - REG	760 - REG	150 - REG
SURF_PHASH	158 - AVG	4827 - AVG	150 - AVG
	537 - BEST	4263 - BEST	150 - BEST
	408 - ER	4261 - ER	150 - ER
	535 - REG	4260 - REG	150 - REG
SURF_MOBILE	203 - AVG	5059 - AVG	150 - AVG
	397 - BEST	4705 - BEST	150 - BEST
	326 - ER	4707 - ER	150 - ER
	396 - REG	4704 - REG	150 - REG
SURF_VGG	173 - AVG	4388 - AVG	150 - AVG
	394 - BEST	3905 - BEST	150 - BEST
	319 - ER	3904 - ER	150 - ER
	391 - REG	3905 - REG	150 - REG

Table 4. The number of clusters produced from each of the 52 combinations on the Reddit data set. The number that correlates with the combination that achieved the top accuracy score on the Imposter-Host task is underlined.

Indonesia	Louvain	Markov	Spectral
PHASH	256 - AVG 257 - BEST 256 - ER 256 - REG	44612 - AVG 44612 - BEST 44612 - ER 44612 - REG	150 - AVG 150 - BEST 144 - ER 147 - REG
MOBILE	159 - AVG 256 - BEST 254 - ER 256 - REG	2264 - AVG 3187 - BEST 3187 - ER 3186 - REG	150 - AVG 150 - BEST 150 - ER 150 - REG
VGG	154 - AVG 257 - BEST 254 - ER 256 - REG	3157 - AVG 1609 - BEST 1590 - ER 1607 - REG	150 - AVG 148 - BEST 144 - ER 149 - REG
SURF	69 - AVG 72 - BEST 68 - ER 73 - REG	3103 - AVG 3136 - BEST 3103 - ER 3103 - REG	150 - AVG 150 - BEST 150 - ER 150 - REG
SURF_PHASH	197 - AVG 1456 - BEST 846 - ER 1609 - REG	16197 - AVG 13659 - BEST 13620 - ER 13648 - REG	150 - AVG 146 - BEST 147 - ER 150 - REG
SURF_MOBILE	183 - AVG 1597 - BEST 846 - ER 1609 - REG	17531 - AVG 14670 - BEST 14639 - ER 14668 - REG	150 - AVG 149 - BEST 146 - ER 150 - REG
SURF_VGG	154 - AVG 2000 - BEST 1150 - ER 2008 - REG	15280 - AVG 11712 - BEST 11687 - ER 11703 - REG	150 - AVG 149 - BEST 150 - ER 150 - REG

Table 5. The number of clusters produced from each of the 52 combinations on the Indonesia data set.

Ukraine	Louvain	Markov	Spectral
PHASH	252 - AVG 257 - BEST 252 - ER 256 - REG	16433 - AVG 16433 - BEST 16433 - ER 16433 - REG	150 - AVG 150 - BEST 148 - ER 147 - REG
MOBILE	162 - AVG 256 - BEST 257 - ER 256 - REG	416 - AVG 511 - BEST 501 - ER 510 - REG	150 - AVG 149 - BEST 148 - ER 150 - REG
VGG	138 - AVG 252 - BEST 238 - ER 256 - REG	1068 - AVG 437 - BEST 437 - ER 436 - REG	150 - AVG 150 - BEST 150 - ER 150 - REG
SURF	17 - AVG 18 - BEST 17 - ER 21 - REG	2169 - AVG 2169 - BEST 2169 - ER 2169 - REG	149 - AVG 148 - BEST 150 - ER 148 - REG
SURF_PHASH	94 - AVG 1203 - BEST 694 - ER 1202 - REG	8398 - AVG 5453 - BEST 5449 - ER 5451 - REG	150 - AVG 148 - BEST 148 - ER 150 - REG
SURF_MOBILE	97 - AVG 1282 - BEST 658 - ER 1286 - REG	9084 - AVG 5730 - BEST 5722 - ER 5727 - REG	150 - AVG 149 - BEST 147 - ER 150 - REG
SURF_VGG	98 - AVG 1191 - BEST 645 - ER 1189 - REG	8351 - AVG 5523 - BEST 5511 - ER 5520 - REG	150 - AVG 146 - BEST 148 - ER 150 - REG

Table 6. The number of clusters produced from each of the 52 combinations on the Ukraine data set.

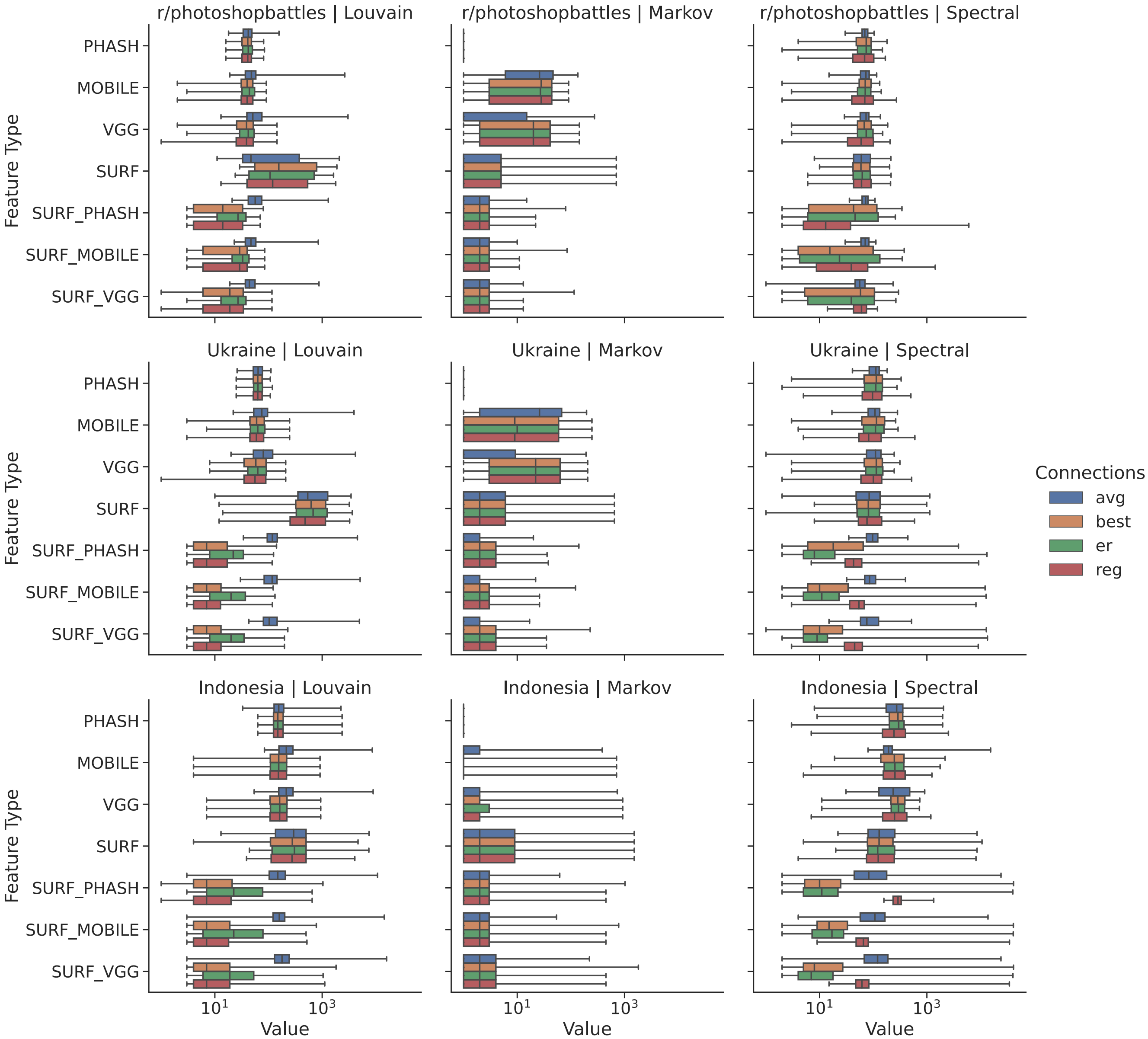

Figure 1. Box plots of the distribution of cluster sizes for each data set and each combination of feature type, clustering algorithm, and connection type. Note that the x-axis has a logarithmic scale.

D.2. Cluster Image Distributions.

As the motif mining pipeline is intended to aid human observers, we believe the distribution of images amongst the clusters is of the utmost importance. Fig. 1 shows box and whisker plots for all of the possible combinations. While the Markov clustering algorithm delivers the highest accuracy scores on the Imposter-Host test, it is important to realize that the majority of the clusters are of size 1, or in other words useless to analysts. The highest realized accuracy score was SURF_MOBILE-BEST-MARKOV on the Indonesian data set. However, the second quartile for the image distribution was at 2 images per cluster and the third quartile is only 3 images per cluster. Out of these clusters only 63.38% were of a size larger than 1, and only 20.59% contained more than 3 images (i.e., valid for the Imposter-Host task). From Fig. 1 we can see that this trend holds for almost all possible combinations when Markov clustering is used. It is for this reason that we recommend Louvain clustering be used with the combined global-local features. In contrast to the Markov statistics, SURF_MOBILE-BEST-LOUVAIN, on the Indonesian data set, has a second quartile at 7 images and the third quartile is 19 images. Additionally 100% of the clusters have more than 1 image per cluster and 78.46% have more than 3 images. Per

Fig. 1 this trend holds similar for all combinations and on all three data sets.

If one were to look at just Fig. 1 they might come to the conclusion that Spectral clustering achieves a similar distribution to Louvain clustering and may wonder why the authors recommend Louvain clustering over Spectral clustering. It is for this reason that the whiskers are important. The maximum cluster size for SURF_MOBILE-BEST-SPECTRAL is 40,909 images. The data set contains 44,612 images. With 40,909 images in a single cluster this means that 91.69% of the images are essentially unsorted. We consider this case unhelpful to human reviewers, in much the same way as Markov clustering puts thousands of images into their own individual clusters. It is for these reasons that we believe Louvain clustering is the best of the three methods tested for motif mining.

E. Graph Structures.

Additional information about the graph structure of the clusters is provided in this section.

Components, Edges	Reddit	Ukr	Indo
PHASH	256C, 38068E	256C, 58537E	256C, 908593E
MOBILE	256C, 38523E	256C, 58565E	256C, 202405E
VGG	256C, 41440E	256C, 63952E	256C, 193731E
SURF	1C, 161253E	1C, 197858E	14C, 475000E
SURF_PHASH	412C, 24877E	935C, 35938E	1085C, 209128E
SURF_MOBILE	336C, 21728E	1112C, 32887E	1237C, 203859E
SURF_VGG	324C, 18837E	984C, 33389E	1372C, 213988E

Table 7. The number of components and edges the generated graph contained for each feature type for each data set. Of particular interest is each global feature resulting in 256 components (due to the number of FAISS centroids), SURF features producing 1, 1, and 14 components (due to their locality and diversity of query results), and the combined features resulting in a relatively high number of components implying the discovery of 'sub-structures' of similar images within the already calculated FAISS centroids.

E.1. Centroid and Tag Number Experiments.

Components, Edges	128 Centroids	256 Centroids	512 Centroids	1024 Centroids
PHASH	128C, 50406E	256C, 38068E	512C, 25491E	1024C, 19200E
MOBILE	128C, 59138E	256C, 38523E	512C, 26390E	1024C, 19937E
VGG	128C, 68027E	256C, 41440E	512C, 27070E	1024C, 19652E
SURF	1C, 158000E	1C, 161253E	1C, 159750E	1C, 157600E
SURF_PHASH	233C, 24753E	412C, 24877E	733C, 24086E	1257C, 22406E
SURF_MOBILE	205C, 22353E	336C, 21728E	599C, 21466E	1116C, 20432E
SURF_VGG	200C, 19667E	324C, 18837E	588C, 19008E	1083C, 18039E

Table 8. The number of components and edges the resulting graphs had when the index was created with 128, 256, 512, and 1024 centroids. This shows that regardless of the number of centroids chosen all the global features accomplish is exposing the pre-existing centroid space from the OPQ index.

Components, Edges	8 Length Tag	16 Length Tag	32 Length Tag	64 Length Tag
SURF_PHASH	259C, 27575E	412C, 24877E	N/A	N/A
SURF_MOBILE	362C, 20789E	336C, 21728E	633C, 18447E	577C, 18706E
SURF_VGG	340C, 18777E	324C, 18837E	558C, 16040E	533C, 16189E

Table 9. How the length of the global tag affects the number of components and edges in the resulting graph. The fact that PHASH features have a length of 16 was the primary driver of that length being used. One can see however that increasing the tag almost doubles the number of components between 16 and 32. If the goal is a larger number of discrete clusters this might be a worthwhile change.

F. Supplemental Qualitative and Quantitative Results

Additional visual results showing graphs and images, as well as additional Impostor-Host plots for various experiments are shown in this section.

F.1. Meme Motifs

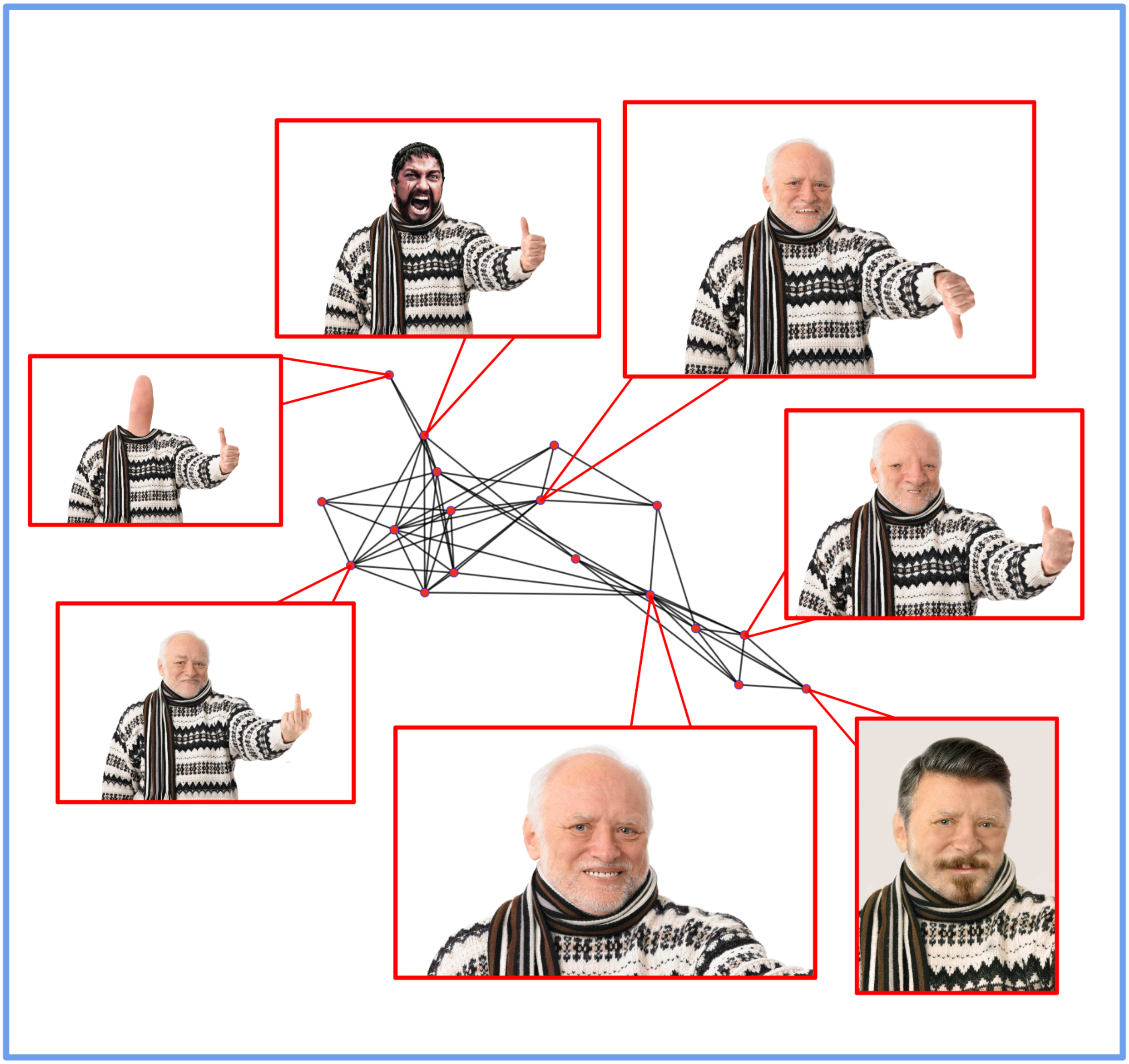

Figure 2. An example of a Reddit motif demonstrating the kind of visual remixing present in the data set.

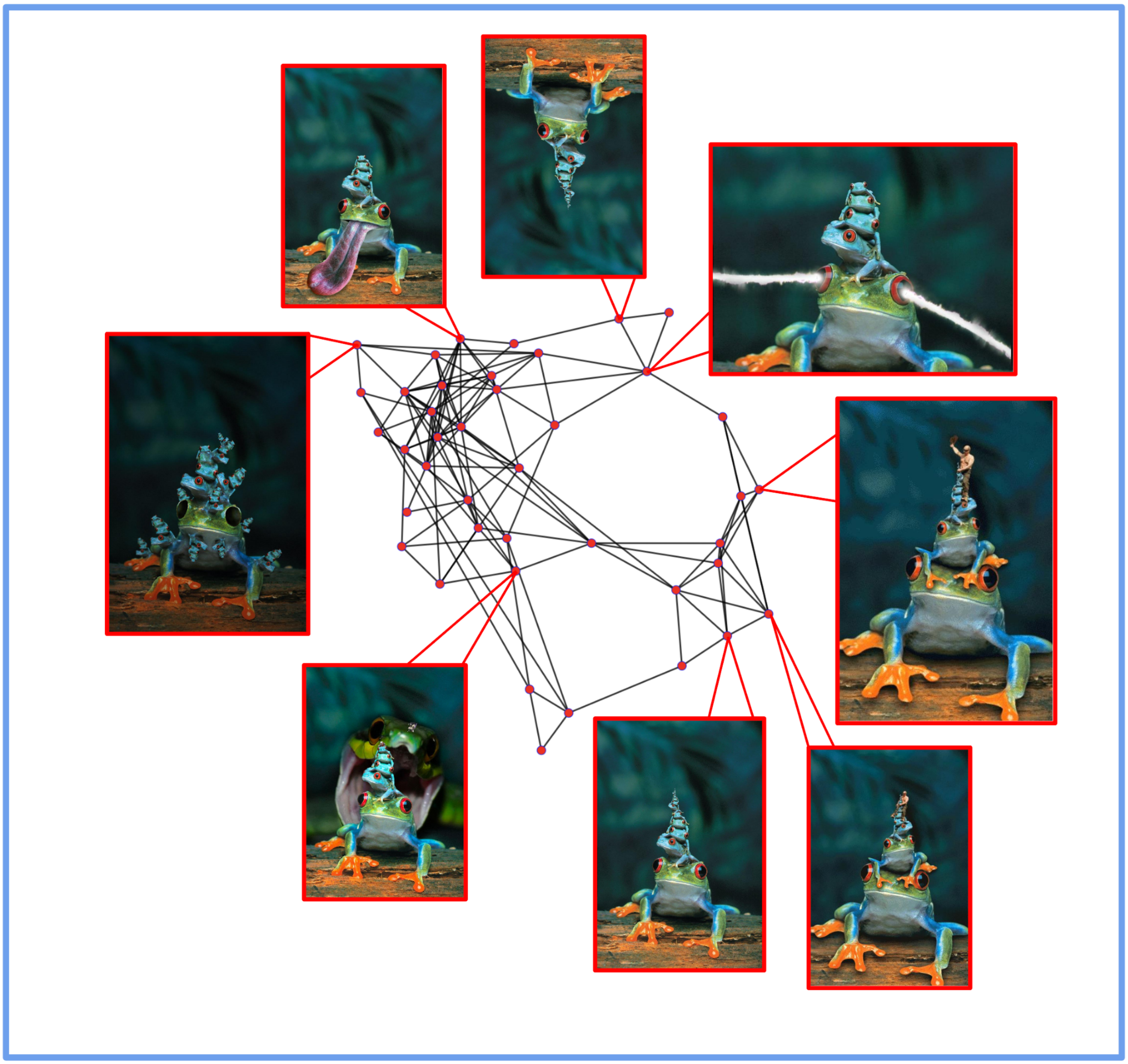

Figure 3. A motif containing remixes of a stack of tree frogs. This graph shows the usefulness of the global feature information, as all the images look very similar globally and matching in this case benefits from the combined feature type.

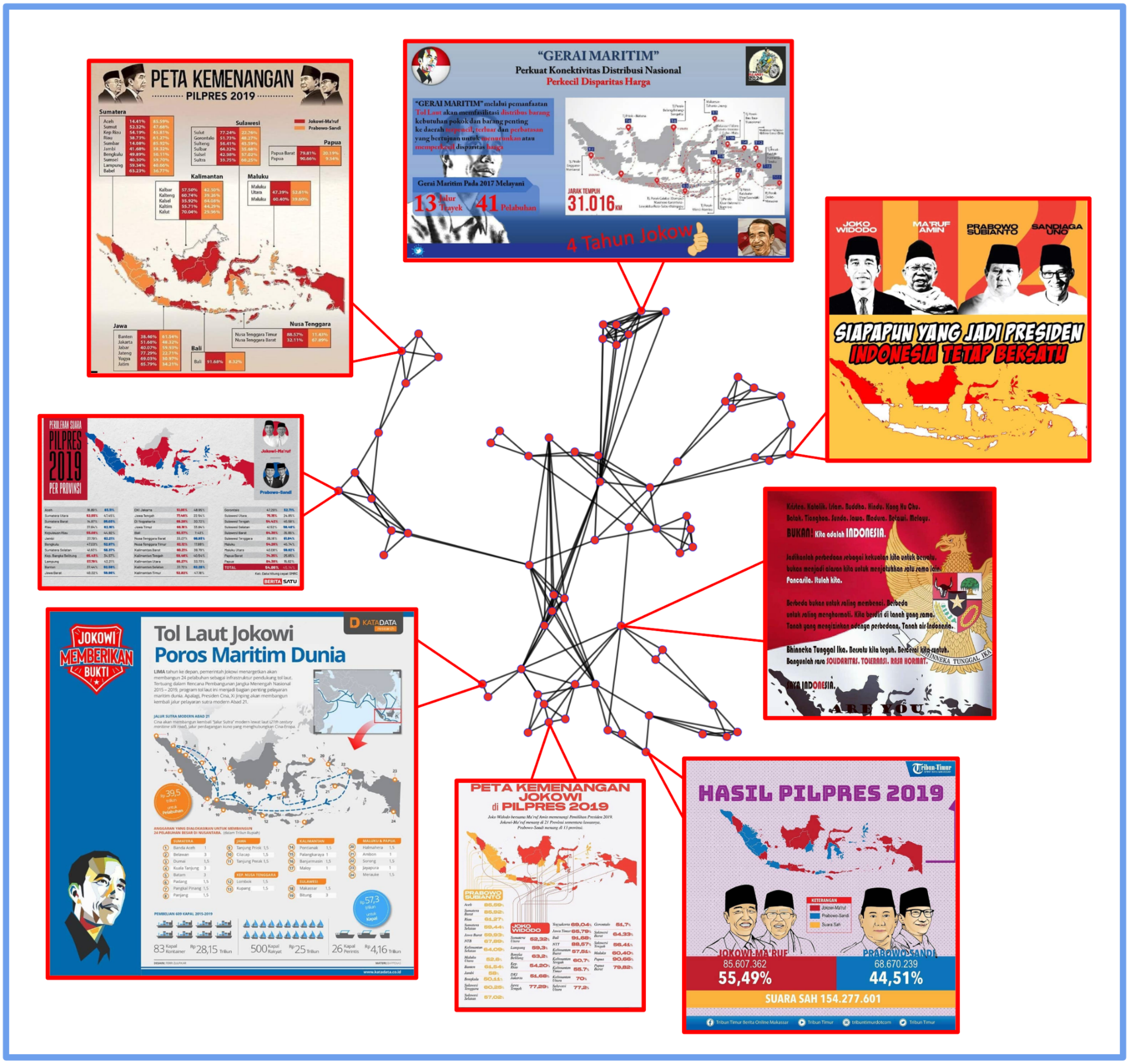

Figure 4. An example of the local features being used to create a graph from the Indonesian data set. Each image, while globally very different, contains at least part of a map of Indonesia. The local features are able to find the shared map portions in each of the images and group them together.

Figure 5. A collection of presidential campaign ads from the Indonesian election in 2019. The same base image is used throughout but is remixed in various contexts. This kind of campaign ad remixing was common in the data set for both candidates.

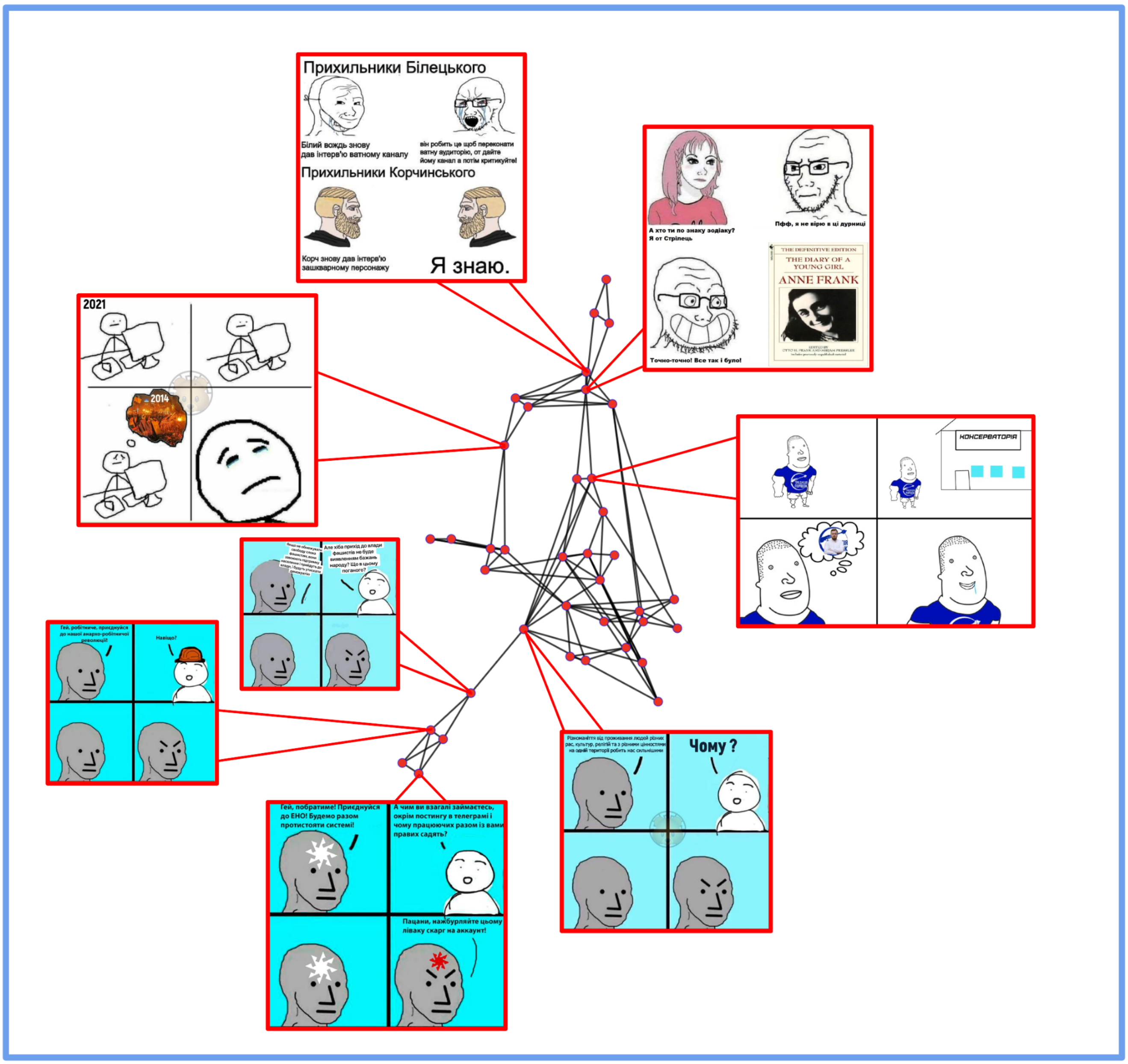

Figure 6. A graph of Ukrainian memes. While there is a bit of variation in visual appearance, all of the memes share the same four panel structure and a subgroup of them share the same genre on top of which various topics are remixed.

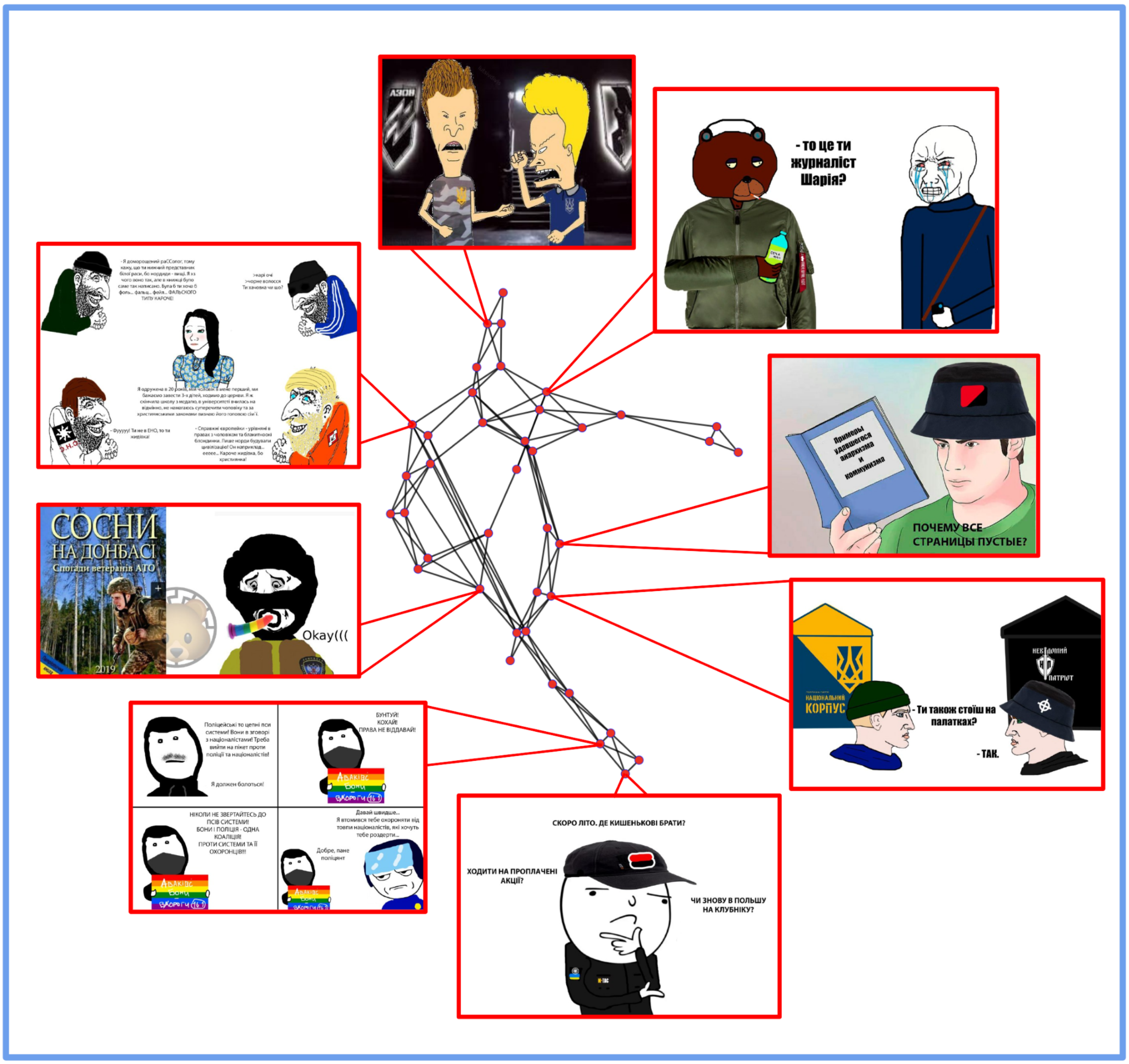

Figure 7. An example of a graph which human observers may consider interesting but we would consider a failure from an algorithmic perspective. While human observers might be interested in exploring the online meme space on Telegram, visually the images in this graph do not have much in common. While the implemented motif mining pipeline is good, it is far from perfect and not every graph contains a recognizable motif from a computer vision stand-point.

F.2. Connection Type Plots

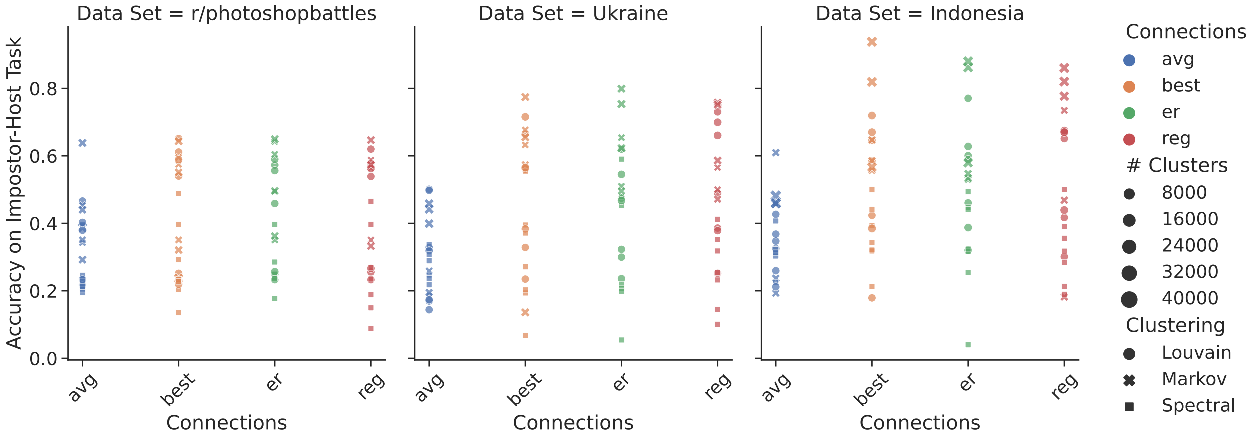

Figure 8. The accuracy scores of the Imposter-Host test across the three data sets for each connection type. Each of the three clustering methods is noted with a different shape. The size of each marker is proportional to the number of clusters.

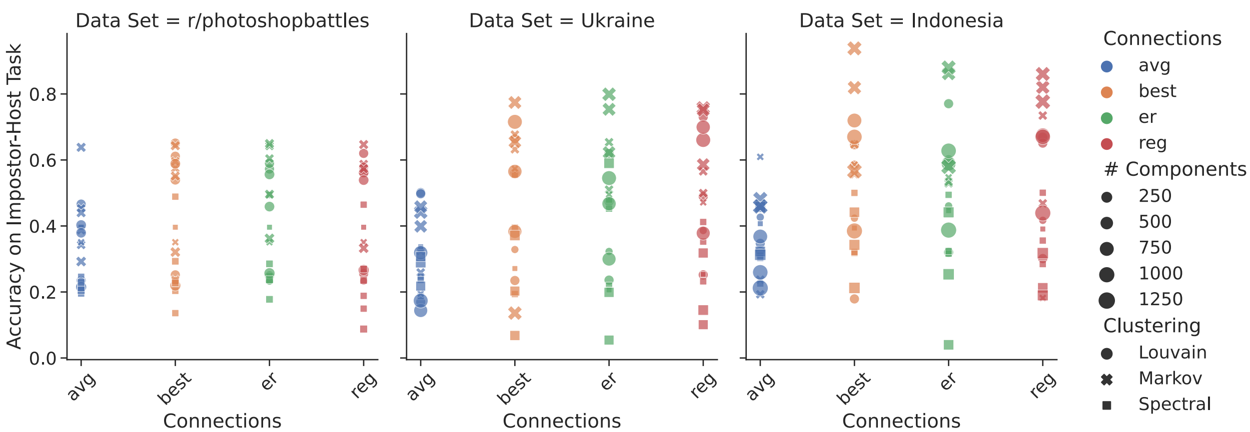

Figure 9. The accuracy scores of the Imposter-Host test across the three data sets for each connection type. Each of the three clustering methods is noted with a different shape. The size of each marker is proportional to the number of components in the graph.

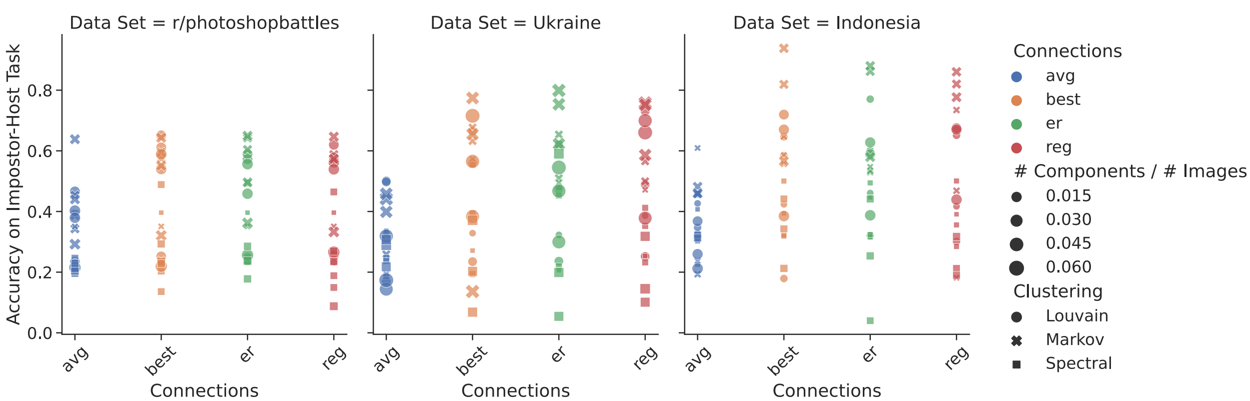

Figure 10. The accuracy scores of the Imposter-Host test across the three data sets for each connection type. Each of the three clustering methods is noted with a different shape. The size of each marker is proportional to the ratio of the number of components in the graph to the number of images in the dataset.

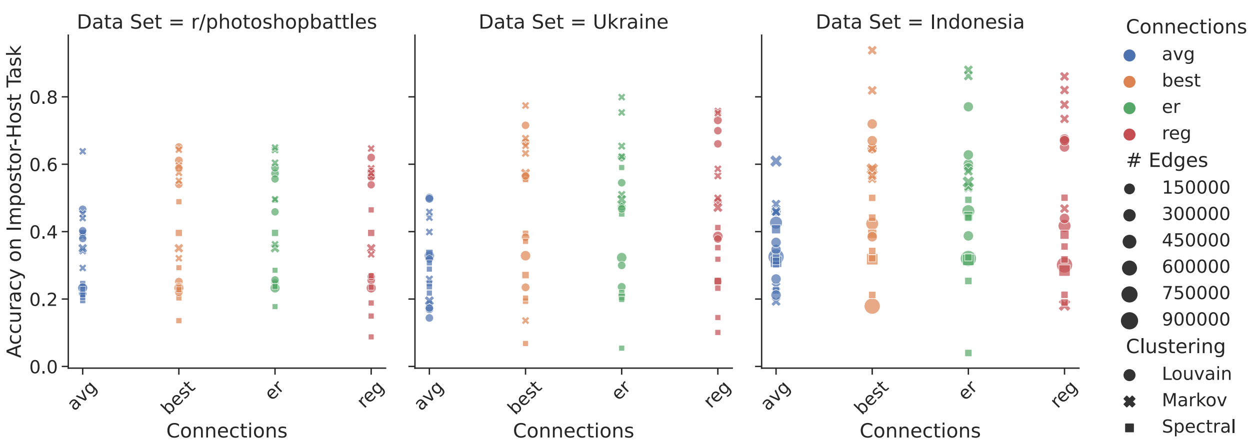

Figure 11. The accuracy scores of the Imposter-Host test across the three data sets for each connection type. Each of the three clustering methods is noted with a different shape. The size of each marker is proportional to the number of edges in the graph.

F.3. Feature Type Plots

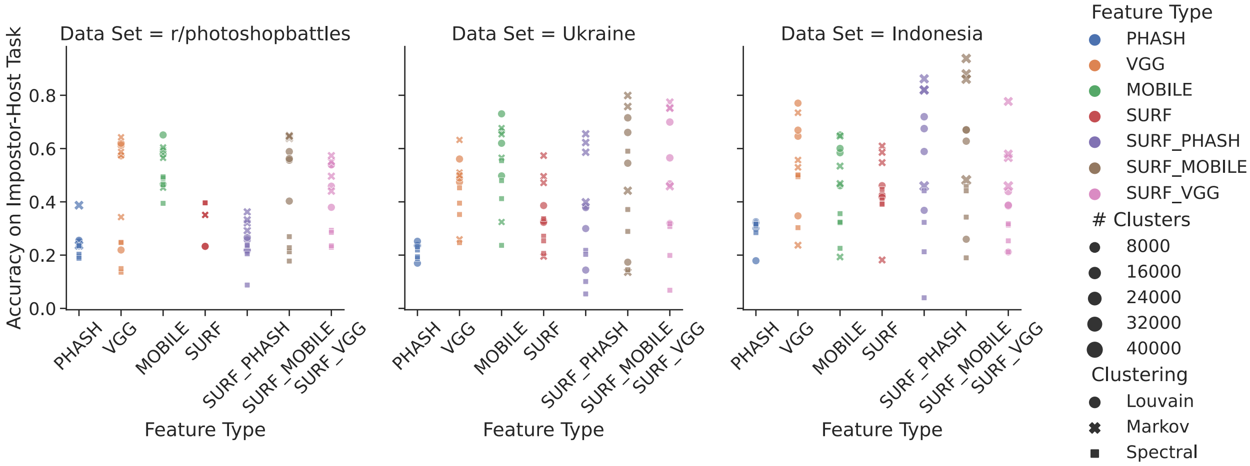

Figure 12. The accuracy scores of the Imposter-Host test across the three data sets for each feature type. Each of the three clustering methods is noted with a different shape. The size of each marker is proportional to the number of clusters.

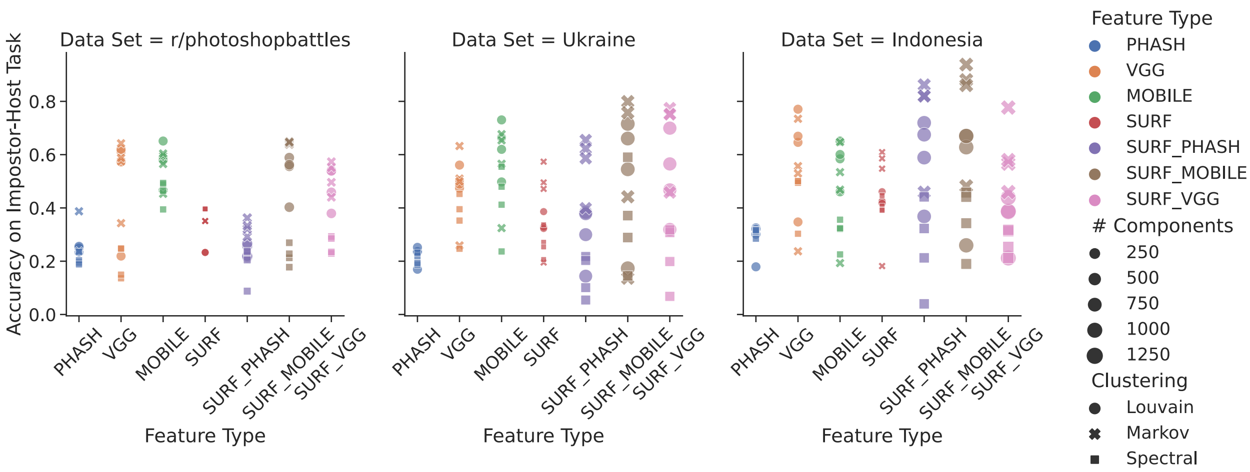

Figure 13. The accuracy scores of the Imposter-Host test across the three data sets for each feature type. Each of the three clustering methods is noted with a different shape. The size of each marker is proportional to the number of components in the graph.

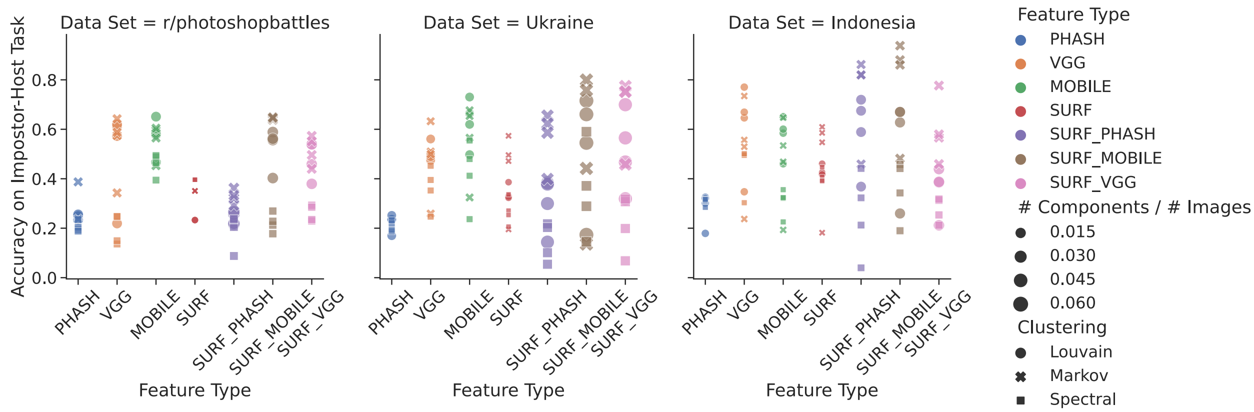

Figure 14. The accuracy scores of the Imposter-Host test across the three data sets for each feature type. Each of the three clustering methods is noted with a different shape. The size of each marker is proportional to the ratio of the number of components in the graph to the number of images in the dataset.

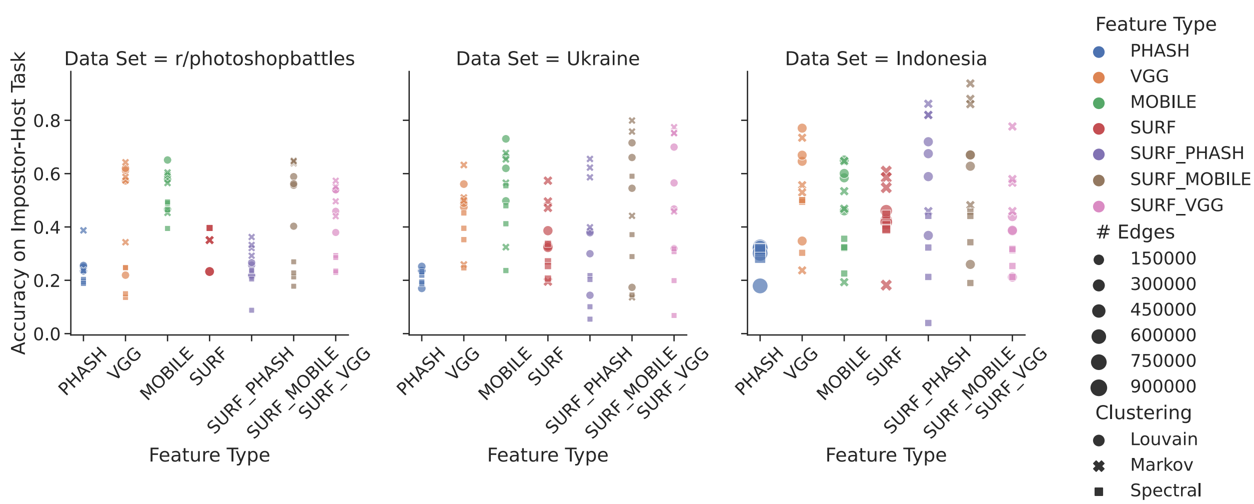

Figure 15. The accuracy scores of the Imposter-Host test across the three data sets for each feature type. Each of the three clustering methods is noted with a different shape. The size of each marker is proportional to the number of edges in the graph.

G. Telegram Users Used as Sources for Ukraine Dataset

Medvezhatko1488, sashakots, russ_orientalist, white_powder2020, karpatska_sich, NSDviz, dadzibao, olifand_rolands, ASupersharij, BerezaJuice, dark_k, joker_ukr, kryuchoktv, legitimniy, notesdetective, rezident_ua, smolii_ukraine, tayni_deputata, thanksrinat, nationalcorps, nedotorkani, ivkolive, ze_konets, ukrnastup, dubinskypro, ruheight, AleksandrSemchenko, botsmanua, borodatayaba, gistapa, kachuratut, poliakovanton, BeregTime, MaksymZhorin, tradition_and_order, KlymenkoTime, sorosata, tsibulya_ua, Ten_NaPleten, donbasscase, lugansk_inside, sorok40russia, ze_landia, zv_kyiv, moh_zdoh, wargonzo, apleonkov, PiB88, format_W, gribvictoria, maksnazar, sheptoon, dobkinmm, UlejUA, spletnicca, razvedinfo, rus_demiurge, LastBP, zlobniaukr, mig41, catars_is, ukrain1an_news, korchynskiy, ua_stalker, project_solaris, liberaxy, orthodox_news, sooproon_bestiary, tasty_flashbacks, fascio_memes, intolerant_historian, Ironvoter, mem_lozha, knpu_division, kekistandivision, EternalMuscovites, nt_orthodox, intolerant_journalist, AD_i_OR, nazbolukr, odindrugqoom, DeepStateUA, ukrnastup, ep867, legion_of_kuchma, NFafaf, History_Q, vidardivision, avantguardia, ulpra, KARAS_EVGEN, GrantDetector, privatnamemarnya, OstanniyCapitalist, afemina, totalopir, intermariumnc, intolerant_warfighter, ukrmemesmineproblemes, evil_ukraine, national_resistance_ua, propala_gramota, postbased, ukrainianintolerant, korchynskiy, Ukrainianintolerantrezerv, RightLit, selo_divisionS, mayonez_sorosa, ubd_ua, national_corp_kyiv, centuriaua

H. CBIR Output Comparison

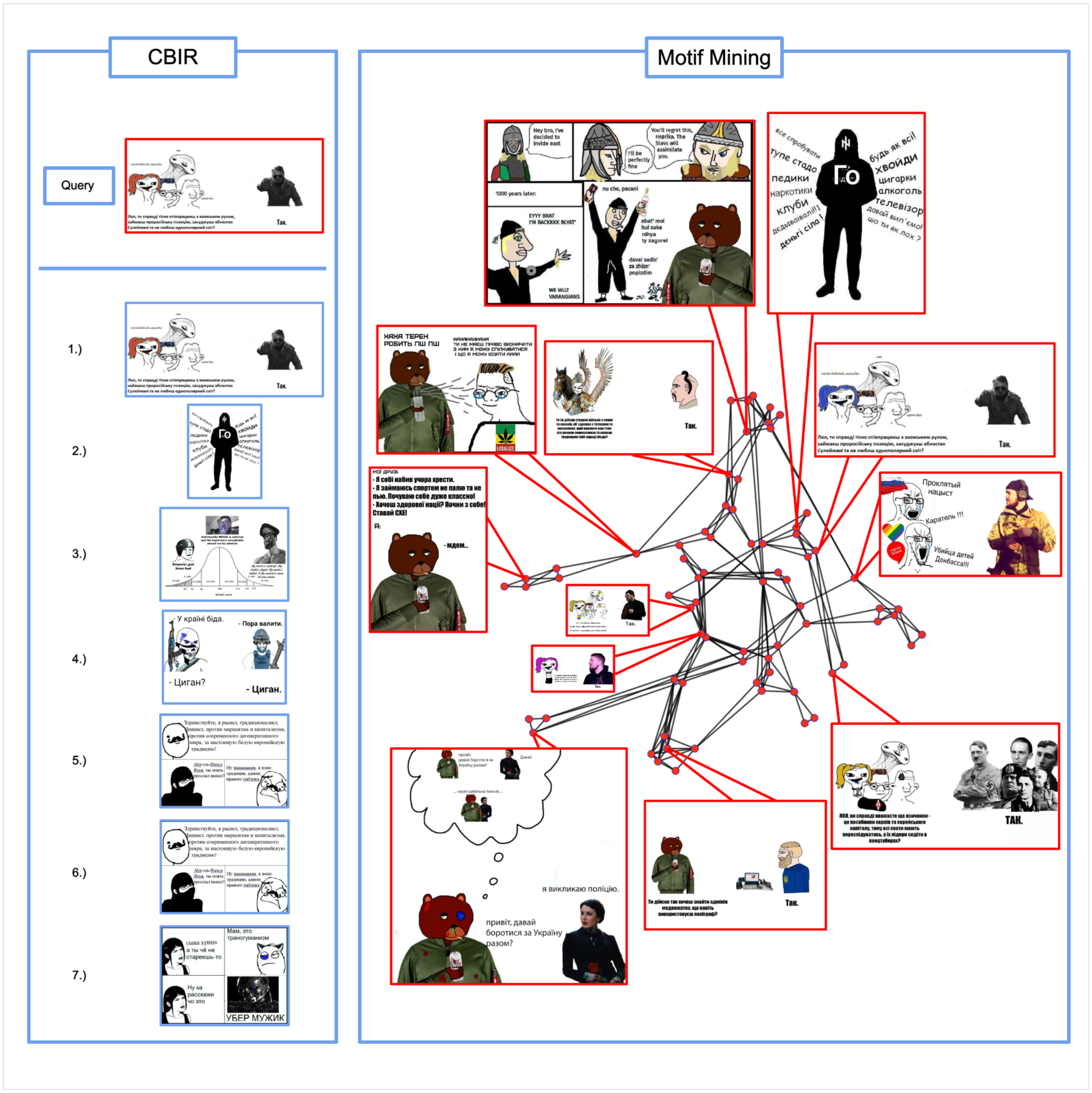

Figure 16. An experimental comparison between the output format that a standard CBIR system [?] returns for a query versus the standard output the motif mining pipeline outlined in this work would give for the same dataset.

In Fig. 16 one can see the difference in output between a CBIR system [?] and the implemented motif mining pipeline. CBIR systems will return a single ranked list for any given query image as can be seen on the left of the Figure. A motif mining pipeline, on the other hand, will return an explorable graph of related images. In order to generate this comparison, an image from the motif graph on the right was chosen as the query image. This figure demonstrates the difference in usefulness of the output of the two systems, especially when the data set is large, unsorted, and unlabelled. Instead of needing to know *a priori* which query images to use, or randomly selecting them, a reviewer interested in exploring the data set can instead use the motif mining system to generate motif graphs from the data set. Notice how the format of the query image is shared stylistically amongst different images in the motif graph and also how the graph explores the relationship between the bear cartoon character that is remixed in the memes on the left of the motif graph with the memes on the right.

I. Normalizing Factor in the Motif Mining Formalization

To elaborate on the normalization factor in the motif mining formalization, the minimalization is simply a representation of the kind of optimization we are performing (trying to find a/the best node-clustering that will optimally accord with human opinion). This normalizing factor can be defined to prefer certain clusterings over others (for example, to prevent clusters from becoming too large or too small), and helps present the problem in its full generality. If one were trying to find the optimal human-evaluated node clustering, this normalizing factor would be set to $\gamma(c, \tilde{c}) = 1$ for all pairs of clusters. We do not carry out this optimization as-written due to (1) the exponential size of the search space and (2) the cost associated with evaluating $H(v_1, \ldots v_k, \tilde{v})$ and its inherently subjective nature. Thus, as stated in the main text, this paper pursues an approximation.