SUPPLEMENTARY MATERIAL

C PSEUDO-CODE OF MDS AND MBPM

We provide the pseudo-code for MDS and MBPM in Alg. 6 and Alg. 7, respectively.

Algorithm 6: MDS

Input: (1) $G = (V, E)$: an input graph
(2) $p : E \to [0, 1]$: activation probabilities
(3) $S \subseteq V$: a seed set
(4) b: an edge-removal budget
(5) h: the number of propagation hops
Output: $\mathcal{E} \subset E$: a set of edges chosen to be removed

$\mathtt{prob}(v) \leftarrow \pi(v; G, p, S)$ ▷ MC simulation
$A \leftarrow$ the adjacent matrix of G
$\mathtt{rsa} \leftarrow \sum_{0 \le i \le h} A^i \mathbf{1}$
for $i = 1, 2, \ldots, b$ **do**
  $s_e \leftarrow (1 - \mathtt{prob}(u)) \cdot (\mathtt{rsa}(v) + \mathtt{rsa}(u)), \forall (v, u) \in E$
  $e \leftarrow \arg\max_{e \in E} s_e$
  $\mathtt{rsa} \leftarrow \mathtt{update\_rsa}(G, p, e, \mathtt{rsa})$
  $\mathtt{prob} \leftarrow \mathtt{update\_prob}(G, p, e, \mathtt{prob})$
  $E \leftarrow E \setminus \{e\}; \mathcal{E} \leftarrow \mathcal{E} \cup \{e\}$
return $\mathcal{E}$
Function $\mathtt{update\_prob}(G, p, e = (s, t), \mathtt{prob})$:
  $\Delta\mathtt{prob}(v) \leftarrow 0, \forall v \in V$
  $\Delta\mathtt{prob}(t) \leftarrow \frac{p(s,t) \cdot \mathtt{prob}(s) \cdot (1 - \mathtt{prob}(t))}{1 - p(s,t) \cdot \mathtt{prob}(s)}$
  $U \leftarrow \{t\}; V \leftarrow \{t\}$
  while $U \neq \emptyset$ **do**
    $U' \leftarrow \emptyset$
    foreach $(u, v) \in E$ *such that* $u \in U, v \in V$ **do**
      $\Delta\mathtt{prob}(v) \leftarrow \Delta\mathtt{prob}(v) + \frac{p(u,v) \cdot \Delta\mathtt{prob}(u) \cdot (1 - \mathtt{prob}(v))}{1 - p(u,v) \cdot \mathtt{prob}(u)}$
      if $v \notin V$ **then**
        $U' \leftarrow U' \cup \{v\}; V \leftarrow V \cup \{v\}$
    $U \leftarrow U'$
  $\mathtt{prob}(v) \leftarrow \mathtt{prob}(v) - \Delta\mathtt{prob}(v), \forall v \in V$
  return prob
Function $\mathtt{update\_rsa}(G, p, e = (s, t), \mathtt{rsa})$:
  $\Delta\mathtt{rsa}(t) \leftarrow p(s, t) \cdot \mathtt{rsa}(t)$
  $U \leftarrow \{s\}; V \leftarrow \{s\}$
  while $U \neq \emptyset$ **do**
    $U' \leftarrow \emptyset$
    foreach $(v, u) \in E$ *such that* $v \in V, u \in U$ **do**
      $\Delta\mathtt{rsa}(v) \leftarrow \Delta\mathtt{rsa}(v) + p(v, u) \cdot \Delta\mathtt{rsa}(u)$
      if $v \notin V$ **then**
        $U' \leftarrow U' \cup \{v\}; V \leftarrow V \cup \{v\}$
    $U \leftarrow U'$
  $\mathtt{rsa}(v) \leftarrow \mathtt{rsa}(v) - \Delta\mathtt{rsa}(v), \forall v \in V$
  return rsa

D ADDITIONAL EXPERIMENTAL RESULTS

D.1 Q1. Performances

We report the effectiveness (the reduced ratio of influence) and the running time, along with the standard deviations, for each

Algorithm 7: Modified BPM

Input: (1) $G = (V, E)$: an input graph
(2) $p : E \to [0, 1]$: activation probabilities
(3) $S \subseteq V$: a seed set
(4) b: an edge-removal budget
(5) d: the number of samplings
Output: $\mathcal{E} \subset E$: a set of edges chosen to be removed

$\mathcal{E} \leftarrow \emptyset$
for $i = 1, 2, \ldots, b$ **do**
  $s_e \leftarrow 0, n_e \leftarrow 0 \forall e \in E$
  for $j = 1, 2, \ldots, b$ **do**
    $l_e \sim \mathrm{Bernoulli}(p(e)), \forall e \in E$
    $E' \leftarrow \{e \in E : l_e = 1\}$
    $G' \leftarrow (V, E')$
    $n_S \leftarrow$ the number of nodes reachable from S on G'
    foreach $e \in E$ *such that* $l_e = 0$ **do**
      $s_e \leftarrow s_e + n_S; n_e \leftarrow n_e + 1$
  foreach $e \in E$ **do**
    if $n_e \geq 1$ **then**
      $s_e \leftarrow s_e / n_e$
    else
      $s_e \leftarrow \infty$
  $e \leftarrow \arg\min_e s_e$
  $E \leftarrow E \setminus \{e\}; \mathcal{E} \leftarrow \mathcal{E} \cup \{e\}$
return $\mathcal{E}$

method, for each $b \in \{3, 5, 7, 10\}$, in Fig. 6 and Table 6. The standard deviations are computed over different seed sets and the values can be high since the seed sets can be highly different. Overall, it shows a fair trade-off between time and the reduction ratio except for RumorGuard-I.

D.2 Time for training and running GNN

We report the training time for a GCN in Table 7 and compare the running time of GNN with MC simulation in Table 8. Overall, the running time used by a GCN is at least 100 times shorter than the time taken by the MC simulation.

D.3 Budget scalability

We conduct experiments with all the considered methods with the budget b increasing from 1 to 10. In Table 9, for each method and each dataset, we report the minimum value of b with which the method runs out of time (i.e., takes more than one hour on a single seed set), where "." indicates that the method does not run out of time even with $b = 10$. The methods whose outputs do not depend on the seed set are not included. Among the variants of RumorGuard, only RumorGuard-G fails on the largest dataset WC. Among the baseline methods, MBPM has the lowest budget scalability, and Greedy does not scale well on large graphs.

Table 6: The effectiveness (the reduced ratio of influence) of each method with the standard deviations, with budget $b \in \{3, 5, 7, 10\}$. O.O.T denotes out-of-time, i.e., the method does not terminate within one hour on a single seed set in the corresponding setting.

method	$b = 3$			$b = 5$		
	WC	CL	ET	WC	CL	ET
Random	0.0016 (0.0214)	-0.0133 (0.1105)	0.0010 (0.0182)	-0.0045 (0.0216)	0.0010 (0.1188)	0.0081 (0.0158))
OdC	0.0022 (0.0225)	0.0063 (0.1077)	0.0123 (0.0207)	0.0117 (0.0238)	0.0352 (0.1534)	0.0119 (0.0407)
BC	-0.0006 (0.0226)	0.0006 (0.0860)	-0.0011 (0.0208)	0.0019 (0.0220)	-0.0057 (0.0992)	-0.0028 (0.0206)
PR	-0.0017 (0.0175)	-0.0036 (0.1204)	-0.0002 (0.0187)	-0.0017 (0.0201)	-0.0010 (0.1077)	-0.0124 (0.0182)
BPM	**O.O.T**	0.0159 (0.1221)	0.0387 (0.0471)	**O.O.T**	-0.0021 (0.1074)	0.0477 (0.0527)
KED	-0.0006 (0.0174)	0.0109 (0.1162)	-0.0015 (0.0180)	0.0029 (0.0178)	0.0097 (0.1317)	-0.0029 (0.0556)
MDS	0.0098 (0.0254)	-0.0004 (0.1085)	0.0298 (0.0366)	0.0173 (0.0229)	0.0202 (0.1341)	0.0368 (0.0421)
MBPM-100	0.0258 (0.1166)	0.0228 (0.1402)	-0.0010 (0.0187)	0.0198 (0.1173)	0.0032 (0.1289)	0.0140 (0.0844)
MBPM-1000	0.1253 (0.2175)	0.0026 (0.1236)	0.1249 (0.2049)	**O.O.T**	0.0107 (0.0936)	0.1710 (0.2209)
MBPM-10000	**O.O.T**	-0.0020 (0.1371)	0.3900 (0.2020)	**O.O.T**	0.0024 (0.1364)	0.4414 (0.2072)
MBPM-100000	**O.O.T**	**O.O.T**	0.4461 (0.1771)	**O.O.T**	**O.O.T**	**O.O.T**
Greedy-10	0.0059 (0.0481)	-0.0182 (0.0976)	0.0244 (0.0995)	0.0062 (0.0459)	-0.0052 (0.1040)	0.0236 (0.1011)
Greedy-100	**O.O.T**	-0.0052 (0.1144)	0.1560 (0.2363)	**O.O.T**	0.0052 (0.1374)	0.1635 (0.2356)
RIS-0.6	0.2280 (0.1720)	0.3813 (0.1941)	0.2937 (0.1968)	**O.O.T**	0.5045 (0.2064)	0.3641 (0.2102)
RIS-0.4	**O.O.T**	0.4428 (0.1960)	0.3202 (0.2078)	**O.O.T**	0.5227 (0.2136)	0.4156 (0.2163)
RIS-0.2	**O.O.T**	0.4456 (0.1893)	0.3688 (0.2100)	**O.O.T**	0.5591 (0.2217)	0.4513 (0.2212)
RumorGuard-G	0.3609 (0.1796)	0.5240 (0.1876)	0.4675 (0.1855)	0.4311 (0.1726)	0.6547 (0.1877)	0.5613 (0.1828)
RumorGuard-O	0.3474 (0.1814)	0.5137 (0.1889)	0.4485 (0.1766)	0.4081 (0.1760)	0.6404 (0.2074)	0.5389 (0.1861)
RumorGuard-I	0.3261 (0.1728)	0.4783 (0.2062)	0.3920 (0.1869)	0.3747 (0.1717)	0.6149 (0.1877)	0.4703 (0.1780)

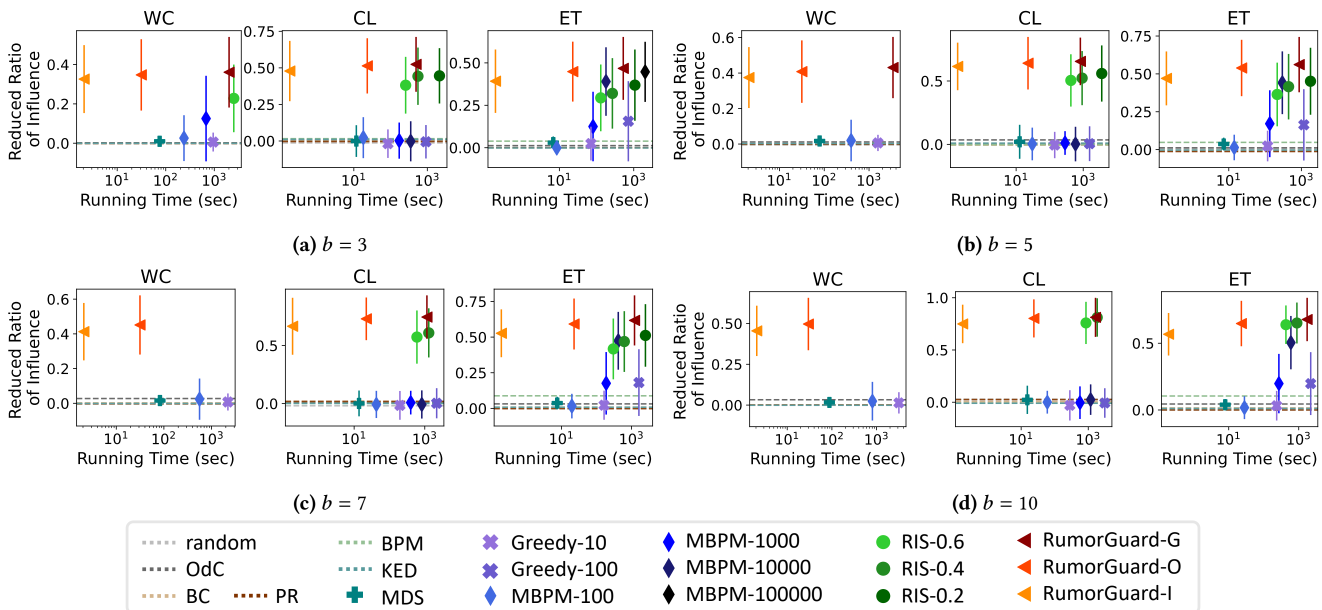
method	$b = 7$			$b = 10$		
	WC	CL	ET	WC	CL	ET
Random	-0.0011 (0.0187)	-0.0187 (0.1244)	0.0014 (0.0210)	0.0002 (0.0201)	-0.0072 (0.1150)	0.0005 (0.0182)
OdC	0.0275 (0.0330)	0.0106 (0.1120)	0.0315 (0.0439)	0.0327 (0.0338)	0.0263 (0.1294)	0.0448 (0.0494)
BC	0.0027 (0.0194)	0.0196 (0.1118)	0.0090 (0.0557)	0.0020 (0.0234)	0.0054 (0.1013)	0.0158 (0.0534)
PR	-0.0035 (0.0225)	0.0187 (0.1071)	-0.0020 (0.0203)	0.0006 (0.0239)	0.0257 (0.1377)	0.0002 (0.0218)
BPM	**O.O.T**	0.0016 (0.0980)	0.0882 (0.1145)	**O.O.T**	-0.0085 (0.1030)	0.1047 (0.1432)
KED	-0.0010 (0.0158)	0.0017 (0.1316)	0.0079 (0.0569)	-0.0012 (0.0197)	-0.0100 (0.1319)	0.0113 (0.0579)
MDS	0.0160 (0.0244)	0.0015 (0.1112)	0.0372 (0.0442)	0.0163 (0.0258)	0.0230 (0.1354)	0.0396 (0.0424)
MBPM-100	0.0239 (0.1188)	-0.0071 (0.1172)	0.0160 (0.0862)	0.0238 (0.1185)	0.0003 (0.1092)	0.0176 (0.0860)
MBPM-1000	**O.O.T**	0.0084 (0.1020)	0.1759 (0.2193)	**O.O.T**	-0.0043 (0.1561)	0.1978 (0.2225)
MBPM-10000	**O.O.T**	-0.0074 (0.1214)	0.4741 (0.2028)	**O.O.T**	0.0248 (0.1461)	0.5046 (0.2000)
MBPM-100000	**O.O.T**	**O.O.T**	**O.O.T**	**O.O.T**	**O.O.T**	**O.O.T**
Greedy-10	0.0076 (0.0493)	-0.0152 (0.1241)	0.0224 (0.1024)	0.0136 (0.0654)	-0.0300 (0.1453)	0.0321 (0.1111)
Greedy-100	**O.O.T**	0.0030 (0.1299)	0.1809 (0.2340)	**O.O.T**	-0.0074 (0.1414)	0.1977 (0.2358)
RIS-0.6	**O.O.T**	0.5732 (0.2292)	0.4172 (0.2134)	**O.O.T**	0.7590 (0.2020)	0.6390 (0.1443)
RIS-0.4	**O.O.T**	0.6093 (0.2117)	0.4688 (0.2131)	**O.O.T**	0.8113 (0.1836)	0.6517 (0.1534)
RIS-0.2	**O.O.T**	**O.O.T**	0.5116 (0.2195)	**O.O.T**	**O.O.T**	**O.O.T**
RumorGuard-G	**O.O.T**	0.7455 (0.1871)	0.6171 (0.1757)	**O.O.T**	0.8124 (0.1870)	0.6780 (0.1625)
RumorGuard-O	0.4513 (0.1705)	0.7303 (0.1838)	0.5918 (0.1787)	0.4967 (0.1607)	0.8023 (0.1824)	0.6477 (0.1702)
RumorGuard-I	0.4127 (0.1654)	0.6663 (0.2454)	0.5268 (0.1679)	0.4548 (0.1543)	0.7493 (0.1843)	0.5674 (0.1594)

Table 7: Average training time (in seconds) for training each GCN model.

WC	CL	ET
17,227	7,390	6,818

Table 8: The average running time (in seconds) for computing total influence per seed set when using GCN and MC simulations. MC simulations use 10,000 samplings.

model	WC	CL	ET
GCN	0.0082	0.0056	0.0056
MC simulation	3.7757	0.8276	0.7817



(a) $b = 3$ **(b)** $b = 5$

(c) $b = 7$ **(d)** $b = 10$

Figure 6: The effectiveness (the reduced ratio of influence) and running time of each method, with budget $b \in \{3, 5, 7, 10\}$. The error bars represent one standard deviation. The superiority of RumorGuard is valid with all the values of b.

Table 9: The minimum budget $b \leq 10$ with which each method runs out of time. Each cell with "." implies that the method does not run out of time even with $b = 10$.

methods	WC	CL	ET
MDS	.	.	.
MBPM-10	.	.	.
MBPM-100	.	.	.
MBPM-1000	4	.	.
MBPM-10000	1	.	.
MBPM-100000	1	3	5
Greedy-10	.	.	.
Greedy-100	1	.	.
RIS-0.6	5	.	.
RIS-0.4	2	.	.
RIS-0.2	1	6	10
RumorGuard-G	6	.	.
RumorGuard-O	.	.	.
RumorGuard-I	.	.	.