VII. RESULTS

After constructing the S-Block algorithm, we tested the performance of the algorithm and conducted comparisons with other existing algorithms. During the experimentation phase, we worked under the following assumptions.

Propagation Probability

In our experimental setup, we consider the simultaneous propagation of two types of information (true news and fake news) within the network. Since the use of fixed propagation probability does not conform to the real-world propagation situation, we use the normal distribution to simulate the propagation probability [26]. Additionally, considering the higher appeal of fake news, resulting in its faster spread, we assume a slightly higher propagation probability of fake news(P_F) compared to true news (P_T).

The Number of Fake News Seed Nodes

In order to evaluate the performance of the algorithm across various scenarios with different numbers of fake news seed nodes, we conducted four experiments on each dataset while keeping other conditions controlled. The number of fake news seed nodes(S_F) was set to 4, 6, 8 and 10 in each experiment respectively.

The Number of True News Seed Nodes

Most existing algorithms use the method of manually setting the number of true news seed nodes, but such experimental settings may cause an unreasonable allocation of resources. In contrast, our algorithm employs dynamic programming to determine an optimal number of true news seed nodes to block the spread of fake news effectively. To ensure the consistency of the comparison between algorithms, all algorithms used the same number of true news seed nodes as the S-block algorithm for comparison. This approach allows for a more robust evaluation of the algorithms' performance.

Evaluation Metrics

To evaluate the performance of the algorithms, we use the following two indicators:

$$Ratio = \frac{N_T}{N_T + N_F}$$

$Ratio$ measures the spreading ability of true news seed nodes, which is obtained by

calculating the ratio of the number of true news nodes in the network to all activated nodes after the propagation ends. A higher $Ratio$ indicates that the seed nodes selected by the algorithm have a higher influence in the network.

$$Percentage\ saved = 1 - \frac{N_{F\ with\ T}}{N_{F\ without\ T}}$$

$Percentage\ saved$ measures the combating capability of the S_T against fake news. It is determined by comparing the number of nodes infected by the fake news when only fake news is propagating in the network ($N_{F\ without\ T}$), to the number of nodes infected by fake news after introducing S_T into the network for concurrent propagation ($N_{F\ with\ T}$). $Percentage\ saved$ is then obtained by calculating $1 - \frac{N_{F\ with\ T}}{N_{F\ without\ T}}$. Its core idea is to examine how many nodes that would have been infected by the fake news are saved (their state remaining "inactive" or "true" at the end of propagation) when S_T are introduced into the network to counteract the fake news, in comparison to the scenario where only fake news is propagating. A higher $Percentage\ saved$ shows that the seed nodes selected by the algorithm have a better effect of curbing the spread of fake news. Since a single experimental result is not sufficient to ensure the stability of the algorithm and the reliability of the results, in our experiments, we calculated the average performance across 20 propagation simulations to validate the accuracy of the algorithm. Furthermore, we examined the variations in the two evaluation metrics ($Ratio$ and $Percentage\ saved$) as the number of propagation iterations increased. This was done to validate the correlation between the algorithm performance and the number of propagation iterations.

Dataset

In the early stage of the project, we conducted experiments on small-scale networks through network generation algorithms such as planted_partition_graph, gnm_random_graph. After that, we placed the final experiments on large-scale network datasets from SNAP, specifically selecting three undirected graph datasets (ego-Facebook, ca-GrQc, lastfm-asia) and two directed graph datasets (wiki-vote, email-Eu-Core) to show the experimental results.

Baseline Algorithms

To compare the performance of the algorithm in each data set, we selected some existing algorithms to compare with the S-Block algorithm:

$Random$ is a heuristic algorithm randomly selects a corresponding number of nodes in

the network as seed nodes for true news propagation. The selection of seed nodes and the algorithm's performance exhibit a significant level of stochasticity. By comparing our algorithm with this approach, we can validate the effectiveness of our algorithm relative to the random selection strategy.

$Out-Degree\ Centrality$ is a heuristic algorithm chooses seed nodes for true news propagation by selecting the number of nodes with the highest out-degree in the network. Its effectiveness heavily relies on the characteristics of the network structure, particularly tends to perform well in networks with substantial variations in node out-degrees or concentrated propagation probability distributions. Out-degree centrality was chosen as a basis for the comparative algorithm as it is a commonly used measurement in network analysis.

$Betweenness\ Centrality$ is a heuristic algorithm that represents a measure of the shortest path to a node in a network. This algorithm inspires the design of the S-Block algorithm,it is used to compare with the S-block algorithm.

$PageRank\ Centrality$ is a heuristic algorithm operates under the assumption that a node becomes more important when it receives recommendations from other highly influential nodes. In simpler terms, nodes with a higher number of incoming links or larger cumulative probability are more likely to possess influence. The PageRank centrality measure provides a way to rank nodes in a network based on their relative importance, therefore, it is chosen as our comparative algorithm.

$Largest\ Infectees$ is a heuristic algorithm that selects nodes according to their potential influence, and it also limits the range of candidate set nodes by excluding nodes close to the seed nodes of fake news according to [29]. This algorithm takes significantly longer time than centrality-based algorithms. However, much faster than traditional greedy-based algorithms. This is a basic block algorithm idea, which is one of the sources of ideas for the S-Block. Therefore, it is chosen as the comparison algorithm.

We conducted a total of 20 sets of experiments on 4 different fake news seed node numbers for 5 different networks and made a statistical summary of the experimental results.

Results in Undirected Graphs

According to Table III, the first 4 experiments are carried out on ca-GrQc, except when the number of fake news seed nodes is 4, our algorithm can always select the true news seed nodes which are less than the number of fake news seed nodes to block the spread of

fake news. On ca-GrQc, the random seed selection strategy performs poorly, which means this network is worth experimenting with. With the increase in the number of fake news nodes, the ratio of our algorithm has declined, but the two indicators of ratio and percentage saved are both in the top two, and the percentage saved is always above 50%, which proves that the blocking strategy of the algorithm is effective. Observing the Figure 14, 15, 16, 9, the ratio and percentage saved of our algorithm change with the number of iterations, it can be seen that before the fifth iteration, both indicators have an upward trend, then the ratio gradually decreases while the percentage saved becomes stabilized.

TABLE III: Experiment Results for ca-GrQc Dataset

ca-GrQc								
Fake seed	**True seed**		**Algorithm**					
			Random	**Outdegree**	**Betweenness**	**Pagerank**	**Largest Infectees**	**S-Block (our)**
4	**5**	**Ratio**	0.02	0.42	0.26	0.25	**0.60**	0.55
		Percentage Saved	0.01	0.51	0.36	0.36	0.63	**0.65**
		Execution Time(s)	0.00	0.00	49.24	0.10	145.18	3220.45
6	**5**	**Ratio**	0.01	0.34	0.20	0.23	0.48	**0.52**
		Percentage Saved	0.00	0.55	0.29	0.36	0.57	**0.71**
		Execution Time(s)	0.00	0.00	43.63	0.09	216.74	1322.46
8	**4**	**Ratio**	0.01	0.34	0.13	0.19	0.37	**0.40**
		Percentage Saved	0.01	0.45	0.25	0.29	0.51	**0.55**
		Execution Time(s)	0.00	0.00	51.97	0.02	296.98	2266.74
10	**5**	**Ratio**	0.17	0.28	0.19	0.23	**0.40**	0.38
		Percentage Saved	0.26	0.47	0.25	0.30	**0.56**	0.55
		Execution Time(s)	0.00	0.00	58.95	0.02	389.76	3738.36

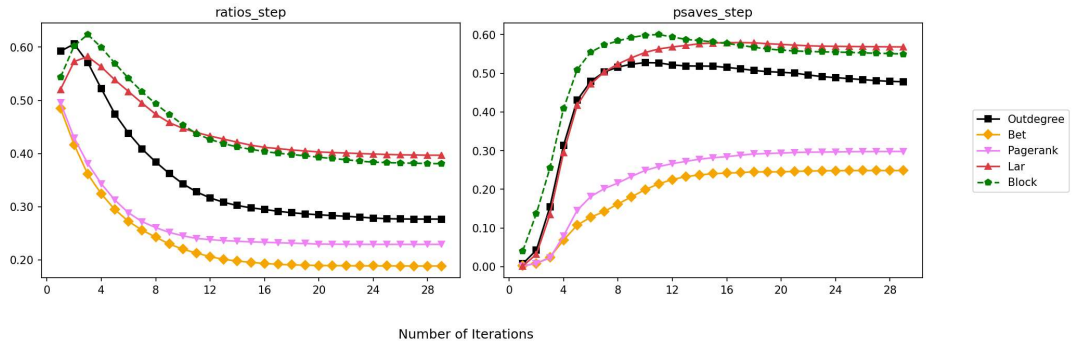

Fig. 9: Number of iterations in ca-GrQc false seed=10

According to Table IV, the experiment conducted on lastfm-asia was reasonable since the random strategy is underperformed. On this network, the number of true news seed nodes selected by the S-Block is always less than that of fake news seed nodes. Except

when the number of fake news seed nodes is 4, the result of the out-degree centrality algorithm is slightly better than the S-Block, our algorithm performs optimally in the rest of the experiments. However, as the number of fake news seed nodes increases, the running time of the S-Block also continuously increases. Observing the Figure 17, 18, 19, 10, on lastfm-asia, the ratio and percentage saved shows the similar trends as ca-GrQc when the number of iterations increase.

TABLE IV: Experiment Results for lastfm-asia dataset

lastfm-asia								
Fake seed	**True seed**		**Algorithm**					
			Random	**Outdegree**	**Betweenness**	**Pagerank**	**Largest Infectees**	**S-Block (our)**
4	**3**	**Ratio**	0.00	**0.47**	0.42	0.32	0.33	0.44
		Percentage Saved	0.00	**0.61**	0.55	0.41	0.43	0.54
		Execution Time(s)	0.00	0.00	175.76	0.04	561.69	3516.33
6	**3**	**Ratio**	0.16	0.47	0.37	0.30	0.25	**0.51**
		Percentage Saved	0.15	0.51	0.50	0.39	0.41	**0.58**
		Execution Time(s)	0.00	0.00	299.58	0.16	1285.23	6659.78
8	**3**	**Ratio**	0.14	0.40	0.40	0.26	0.26	**0.49**
		Percentage Saved	0.22	0.51	0.52	0.40	0.30	**0.58**
		Execution Time(s)	0.00	0.00	218.49	0.05	927.80	10990.92
10	**3**	**Ratio**	0.03	0.41	0.39	0.26	0.13	**0.43**
		Percentage Saved	0.05	**0.57**	0.52	0.40	0.14	**0.57**
		Execution Time(s)	0.00	0.00	354.05	0.24	2564.59	80915.02

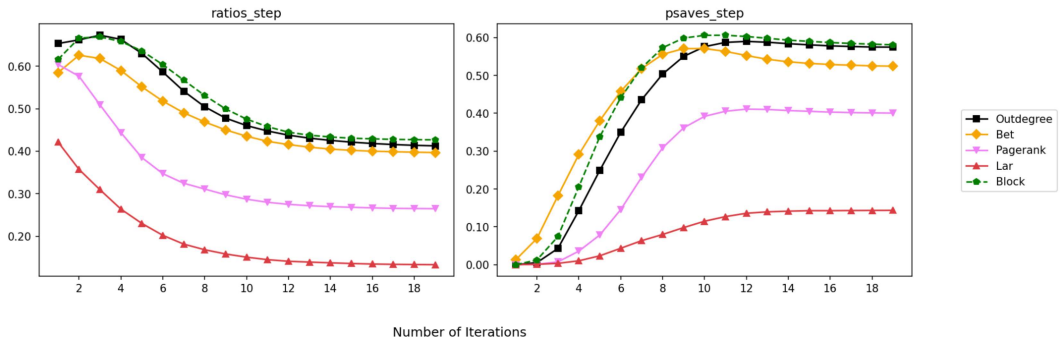

Fig. 10: Number of iterations in lastfm-asia false seed=10

According to Table V, when the experiment is carried out on ego-Facebook, all other algorithms except our method performs poorly. As the number of fake news nodes increases, the number of true nodes selected by the S-Block remains at 2. However, the ratio and percentage saved indicators show an upward trend and are the best among all algorithms, far superior to other algorithms. Furthermore, the running time of our algorithm also increases

as the number of fake news nodes increases. Observing the Figure 20, 21, 22, 11, different from the trend of ratio and percentage saved changing with the number of iterations on the other two undirected graphs, on ego-Facebook, the ratio of our algorithm decreases with the increase of the number of iterations at the beginning, but gradually reverse the direction and tends to be stable; percentage saved has been showing an upward trend, much higher than other algorithms in the later iterations.

TABLE V: Experiment Results for ego-Facebook dataset

ego-Facebook								
Fake seed	**True seed**		**Algorithm**					
			Random	**Outdegree**	**Betweenness**	**Pagerank**	**Largest Infectees**	**S-Block (our)**
4	**2**	**Ratio**	0.20	0.20	0.20	0.14	0.19	**0.21**
		Percentage Saved	0.09	**0.28**	**0.28**	0.22	0.26	**0.28**
		Execution Time(s)	0.00	0.00	77.8	0.24	317.44	1359.03
6	**2**	**Ratio**	0.01	0.16	0.16	0.13	0.12	**0.18**
		Percentage Saved	0.00	**0.35**	**0.35**	0.28	0.22	**0.35**
		Execution Time(s)	0.00	0.00	74.21	0.22	464.53	1827.25
8	**2**	**Ratio**	0.03	0.18	0.18	0.16	0.17	**0.44**
		Percentage Saved	0.03	0.28	0.28	0.20	0.22	**0.42**
		Execution Time(s)	0.00	0.00	93.68	0.32	740.79	28021.5
10	**2**	**Ratio**	0.01	0.19	0.19	0.15	0.16	**0.45**
		Percentage Saved	0.00	0.26	0.26	0.27	0.26	**0.48**
		Execution Time(s)	0.00	0.00	70.22	0.26	798.39	42892.28

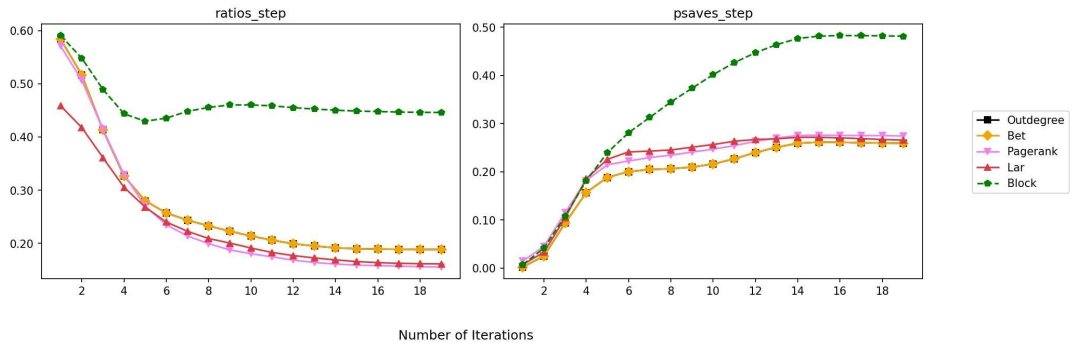

Fig. 11: Number of iterations in ego-Facebook false seed=10

Through experiments and comparisons on different undirected graphs, the performance of the S-Block is stable, which proves the advantages of the algorithm on undirected graphs, especially as the number of fake news nodes increases, the advantages of our algorithm become more obvious.

Results in Directed Graphs

According to Table VI, the S-block algorithm consistently selects fewer true news seed nodes compared to the number of fake news seed nodes on the email-Eu-core network. The running time of our algorithm slightly longer compared to other algorithms, and the performance is not ideal although there are some variations among the comparative algorithms. The Figure 8 shows the disparity between out-degree and in-degree of nodes in this network, indicating that certain nodes exhibit significant differences in degree, consequently impact the effectiveness of our algorithm. From Figure 23, 24, 25, 12, the ratio of our method shows a decrease trend initially and then decrease, while the percentage saved start with a increase trend and then reaches a stable state.

TABLE VI: Experiment Results for email-Eu-core dataset

email-Eu-core								
Fake seed	**True seed**		**Algorithm**					
			Random	**Outdegree**	**Betweenness**	**Pagerank**	**Largest Infectees**	**S-Block (our)**
4	**3**	**Ratio**	0.01	0.41	**0.48**	0.38	0.37	0.40
		Percentage Saved	0.00	**0.49**	0.47	**0.49**	0.43	0.39
		Execution Time(s)	0.00	0.00	3.18	0.02	9.16	12.46
6	**4**	**Ratio**	0.03	**0.40**	0.34	0.34	0.28	0.31
		Percentage Saved	0.01	**0.46**	0.41	0.45	0.36	0.33
		Execution Time(s)	0.00	0.00	4.02	0.02	15.09	39.14
8	**3**	**Ratio**	0.02	**0.31**	0.25	0.28	0.25	0.28
		Percentage Saved	0.03	0.36	**0.38**	0.36	0.30	0.33
		Execution Time(s)	0.00	0.00	3.33	0.02	16.39	36.17
10	**3**	**Ratio**	0.09	**0.34**	0.26	0.25	0.16	0.29
		Percentage Saved	0.13	0.35	**0.38**	0.37	0.23	0.37
		Execution Time(s)	0.00	0.00	3.86	0.03	23.7	54.15

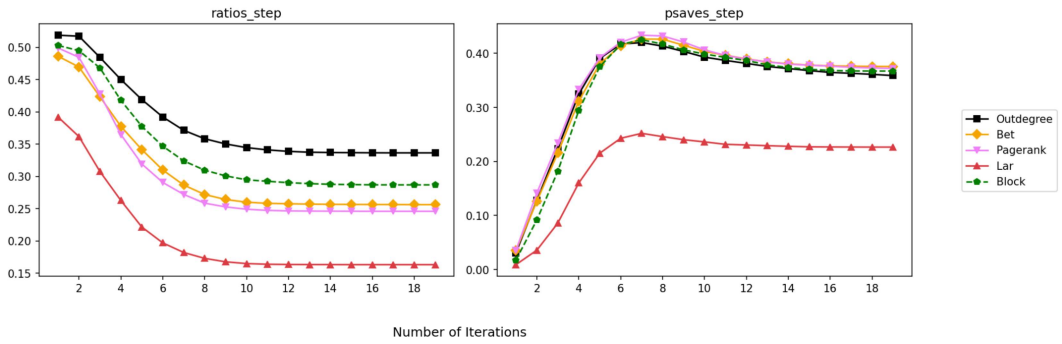

Fig. 12: Number of iterations in email-Eu-Core false seed=10

According to Table VII, the performance of the out-degree centrality algorithm is more prominent when the number of fake news nodes is small on Wiki-vote. But when the number of fake news nodes increases to 10, the ratio of the S-Block algorithm outperforms the out-degree centrality algorithm. For Figure 8 of the differences in the out-degree and in-degree of the nodes mentioned above, it is found that only a small number of nodes in the Wiki-vote network have a large difference in the out-degree and in-degree, and the distribution is more concentrated than that of email-Eu-core. Therefore, the effect of our algorithm is better than that when running on email-Eu-core. Observing the Figure 26, 27, 28, 13, on Wiki-vote, the ratio and percentage saved of our algorithm change with the number of iterations are similar to that of email-Eu-core.

TABLE VII: Experiment Results for Wiki-vote dataset

Wiki-vote								
Fake seed	**True seed**		**Algorithm**					
			Random	**Outdegree**	**Betweenness**	**Pagerank**	**Largest Infectees**	**S-Block (our)**
4	**2**	**Ratio**	0.01	**0.49**	0.48	0.03	0.37	0.43
		Percentage Saved	0.01	**0.54**	0.46	0.08	0.38	0.47
		Execution Time(s)	0.00	0.00	90.82	0.20	64.97	235.23
6	**2**	**Ratio**	0.16	**0.42**	0.39	0.02	0.33	0.27
		Percentage Saved	0.16	**0.45**	0.43	0.04	0.36	0.32
		Execution Time(s)	0.00	0.00	87.42	0.13	76.7	576.17
8	**2**	**Ratio**	0.00	**0.39**	**0.39**	0.02	0.08	0.34
		Percentage Saved	0.00	**0.46**	0.43	0.05	0.36	0.34
		Execution Time(s)	0.00	0.00	96.13	0.21	83.67	764.22
10	**2**	**Ratio**	0.00	0.40	0.38	0.03	0.16	**0.42**
		Percentage Saved	0.00	**0.45**	0.41	0.05	0.16	0.39
		Execution Time(s)	0.00	0.00	109.4	0.1	112.25	1190.08

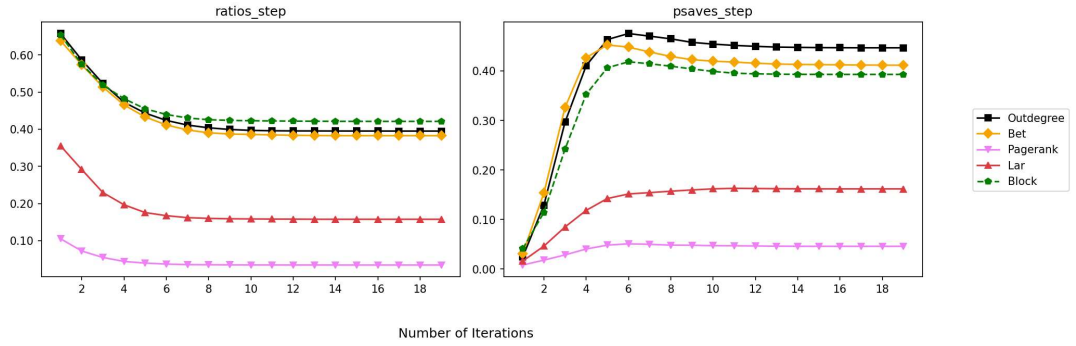

Fig. 13: Number of iterations in wiki-vote false seed=10

Compared to the running time on the undirected networks, our algorithm works faster on directed graphs. While the performance of our algorithm has improved with an increase in the number of fake news seed nodes, it still cannot beat the best performed algorithm in each network.