Understanding Election Rigging Methods[1]

Doh Kyung-Goo

This article is intended for those who are skeptical about whether it's even possible to manipulate election results. Just as understanding the risks allows for better countermeasures, a thorough grasp of election rigging methods can raise awareness about electoral fraud.

Contents

1. Introduction

When analyzing past election results in South Korea, traces of manipulation that seem artificial are often revealed. However, when I tell my acquaintances, "The election results seem abnormal. I suspect election fraud," they respond by saying, "How could that happen with the National Election Commission in place?" — reluctant to believe it. They counter with comments like, "There's no mention of anything like that in the broadcast or newspapers," or they add, "The people are all watching carefully, so is election fraud even possible? Especially on a nationwide scale?" resulting in negative reactions. I wondered if there was a way to help these acquaintances understand that election results can be systematically manipulated through careful planning. As I pondered this, I devised my own logic for planning and executing election fraud, and organized my thoughts into writing.

Wikipedia defines an election as "a formal group decision-making process whereby a population chooses an individual or multiple individuals to hold public office." For an election—fundamental to democracy—to be conducted fairly and justly while respecting the five principles of universality, equality, directness, secrecy, and freedom, ***the election system must ensure integrity***. To that end, the system must meet the following four conditions:

1. ***One person, one vote*** (no multiple votes)
2. ***Guarantee of secret ballot*** (no one but the voter can know the choice made)
3. ***Transparent tallying*** (counting without omission or alteration)
4. ***Verifiable results*** (evidence available to confirm the vote count)

In paper-based elections, satisfying all these conditions requires strict management of voting, ballot handling, and counting procedures. Let's walk through a typical election process step by step.

- During voting, the voter arrives at the polling station, verifies their identity with an authorized ID, receives a secret ballot paper, and records their vote. Their participation is then recorded in the voter list.
  → If the number of recorded voters matches the number of ballots counted, condition 1 (*one person, one vote*) is fulfilled.
- The voter marks their choice in a private booth, then folds the ballot to conceal the contents and deposits it into a sealed ballot box.
  → Since the ballot contains no voter information, and unless the voter secretly takes a picture of the marked ballot, the choice remains only in the voter's memory and cannot be proven to others—satisfying condition 2 (*guarantee of secret ballot*).
- Once voting ends, the ballot box is sealed and stored securely. During vote counting, election officials and observers open the box in front of everyone, sort, and tally the ballots, and then announce the results.
  → If the sealed box remains intact and the process is transparent with observers confirming no irregularities, condition 3 (*transparent tallying*) is maintained.
- After counting, the ballots are returned to the ballot box, resealed, and stored safely for a legally mandated period.
  → If the sealed ballot box remains intact and can be reopened for a recount if necessary, then condition 4 (*verifiable results*) is met.

Following these procedures properly ensures the election appears fair and legitimate. The core guarantee of election integrity is that all stakeholders—candidates and voters—can verify all results at every stage. Even if objections arise, the results can be reconfirmed through recounts, reducing distrust.

However, any human-created system tends to have vulnerabilities that could allow for artificially manipulated results by obscuring procedures. Is the election system an exception? Is it reasonable to unconditionally trust election officials to manage and count votes honestly? Could a group of political forces, struggling to gain support through normal means, collude with election officials to commit fraud unnoticed? Since elections have a profound impact on citizens' lives, we must proactively identify and prepare for vulnerabilities that could enable manipulation, including detailed examination of possible avenues for artificial interference.

This article explores various methods of election manipulation that minimize detectable traces while maximizing the likelihood of success from the perspective of the fraud planner, demonstrating that systematic planning and execution of election fraud is feasible. The outline is as follows:

- Section 2 discusses the necessary conditions for successful election rigging and basic vote manipulation methods.
- Section 3 explores how rigging can be planned and prepared.
- Sections 4 & 5 deal in detail with the methods and procedures for two types of rigging: ballot box manipulation prior to counting and ballot count manipulation during counting.
- Section 6 introduces a double manipulation approach that combines both methods, aiming to eliminate traces left by individual techniques.
- Finally, Section 7 concludes that the most secure voting method is ***immediate on-site tallying after voting ends***, especially since election environments can be manipulated to facilitate undetectable results.

2. Fundamentals of Election Rigging

Election rigging refers to the act of interfering with the voting and counting processes of an election to artificially reverse or alter the results in favor of a specific candidate. In a paper-based election where votes are cast at polling stations, the winner is determined by the number of votes each candidate receives. Therefore, the only way to artificially overturn the results is through manipulation of the vote count.

2-1. Three Necessary Conditions for Successful Election Rigging

Since election rigging is a serious crime akin to attempting to overthrow the state, perpetrators strive to carry out such acts as covertly as possible. To successfully manipulate an election, the following three conditions must be met:

1. ***Secrecy in Execution*** : The manipulation must not be revealed to voters or observers.
2. ***Trace Removal*** : No trace of the manipulation should be left behind.
3. ***Complete Manipulation*** : Even if a recount occurs, the manipulated results must remain consistent with the intended outcome.

These three conditions are collectively referred to as the ***Three Necessary Conditions for Successful Election Rigging***.

2-2. Methods of Election Rigging

Before diving into the details, let us define some key terminology:

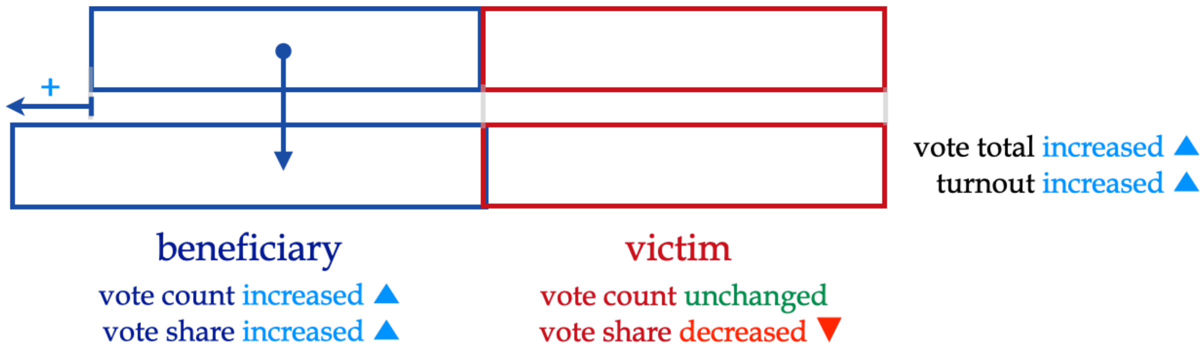
- *Voter*: An individual entitled to participate in the election.
- *Vote Total*: The number of voters who actually voted.
- *Turnout*: The ratio of the vote total to the total eligible voting population.
- *Candidate's Vote Count*: The total number of votes received by a candidate.
- *Candidate's Vote Share*: The ratio of a candidate's votes to the vote total.
- *Beneficiary candidate*: The candidate who benefits from the rigging.
- *Victim candidate*: The candidate who loses due to manipulation.

Vote count manipulation can occur via three techniques:

- ***Add-vote manipulation*** : Increase the beneficiary's vote count by a necessary amount.
- ***Swap-vote manipulation*** : Decrease the victim's vote count and increase the beneficiary's by the same amount.
- ***Discard-vote manipulation*** : Decrease the victim's vote count by a necessary amount.

Let's examine each method with respect to maximizing the "Three Necessary Conditions for Successful Election Rigging" in a paper-based voting environment, where votes are cast on paper ballots, collected in ballot boxes, and counted after voting ends.

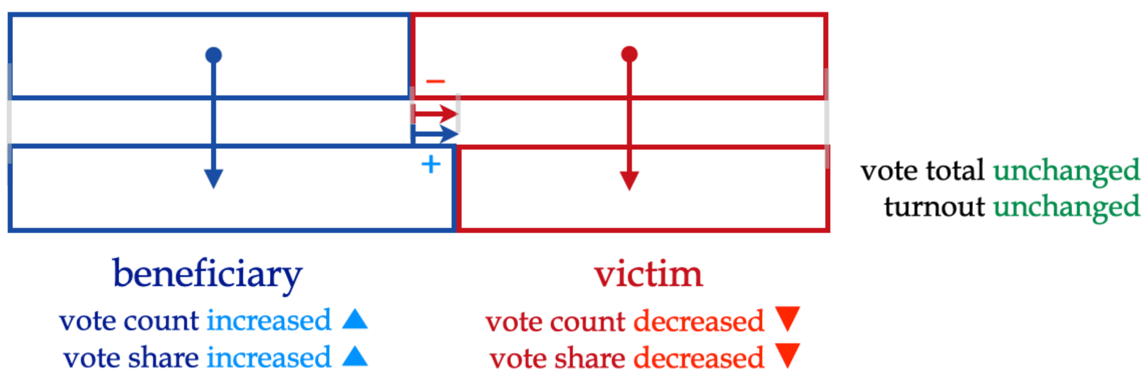
Add-vote manipulation



This involves increasing the beneficiary candidate's vote count by a specified amount. As a result, the candidate's vote share rises proportionally, and the total vote count increases, causing turnout to rise as well. The victim's vote count remains unchanged, but because the total votes increased, their vote share decreases.

To satisfy the *Complete Manipulation* condition, forged ballots (referred to as "ghost ballots") marked for the beneficiary must be created and inserted into the ballot box. These ghost ballots represent votes that do not genuinely exist. The number of forged ballots must match the voting records, which are falsely recorded as legitimate votes by voters who did not actually participate. However, if a genuine voter later verifies and discovers they supposedly voted when they did not, the manipulation could be exposed—failing the *Trace Removal* condition. Additionally, maintaining secrecy around the issuance and insertion of ghost ballots presents significant challenges.

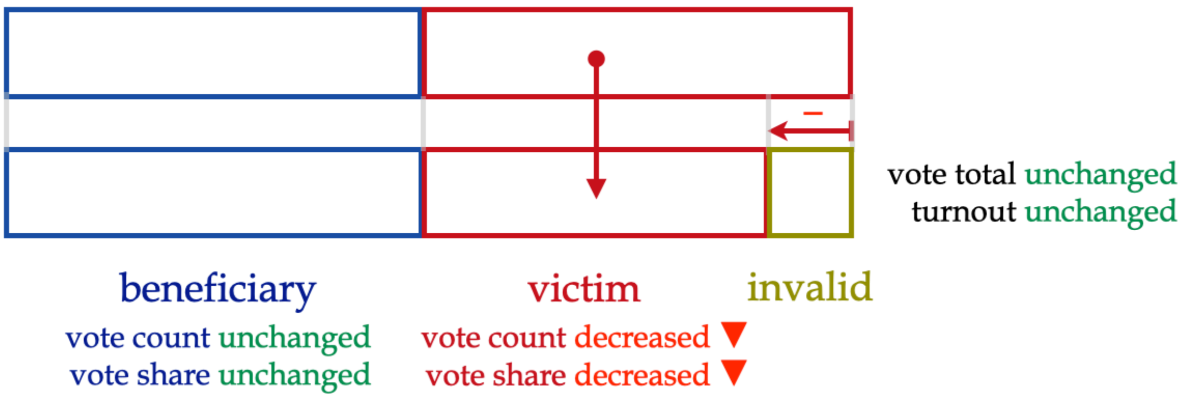
Swap-vote manipulation



This involves removing genuine votes for the victim and replacing them with forged ballots (referred to as "fake ballots") for the beneficiary, keeping the total vote count unchanged. The beneficiary's vote count increases while the victim's decreases, and overall turnout remains stable.

The main challenge in performing vote swap manipulation is fulfilling the *Secrecy in Execution* condition. It requires selecting genuine ballots for the victim, then discreetly removing, destroying, and replacing them with fake ballots for the beneficiary—all without leaving traces. This process demands significant manpower and careful protection to prevent detection. If genuine ballots are improperly handled and leave traces, it becomes impossible to fully meet the T*race Removal* condition.

Discard-vote manipulation



If the votes for the victim candidate are decreased while keeping the votes for the beneficiary candidate unchanged, the total number of votes and voter turnout will also decrease accordingly. However, since a reduction in the total votes and turnout after voting has concluded is logically impossible, this would leave traces of manipulation. Therefore, the only way to manipulate votes while maintaining the total vote count and turnout is to treat the reduced votes for the victim candidate as invalid votes. As a result, the victim's vote count and vote share decrease by the amount of invalid votes, while the beneficiary's vote count and share remain unchanged; this relative advantage favors the beneficiary.

To legitimately invalidate ballots marked for the victim candidate, one could either damage the ballots enough to render them invalid or selectively discard them and replace them with forged invalid ballots. The primary obstacle to executing such a vote invalidation manipulation is similar to the secrecy requirements of vote swapping methods. Securing damaged ballots, and then damaging, discarding, and inserting them, requires a substantial amount of manpower and time. Additionally, the process of discarding genuine ballots and creating and inserting damaged ballots could be exposed. There is also a possibility that some genuine ballots that are improperly damaged or discarded could leave detectable traces, making the manipulation easier to uncover.

Summary of Manipulation Methods

The effects of different manipulation techniques on vote count, vote share, vote total, and turnout are summarized in Table 1.

Table 1. Thee relationship between the change in the number of votes by manipulation method, and the vote count, vote share, vote total, and turnout

Manipulation Method	Beneficiary		Victim		Vote Total	Turnout
	Vote Count	Vote Share	Vote Count	Vote Share		
Add-vote	▲	▲	=	▼	▲	▲
Swap-vote	▲	▲	▼	▼	=	=
Discard-vote	=	=	▼	▼	=	=

▲ : increased, ▼ : decreased, = : unchanged

2-3. Types of Election Manipulation

Based on the timing of execution, election manipulation is classified into two categories:

- ***Ballot Box Manipulation*** - Conducted before the vote tally begins. Although the complexity varies by method, all three manipulation techniques—add-vote, swap-vote, and discard-vote—are theoretically possible.
- ***Counting Manipulation*** - Performed during the vote counting process. The add-vote method is impossible to execute discreetly once turnout is announced, but swap-vote and discard-vote methods are feasible during the count.

3. Preparation for Election Manipulation

Having understood the methods for manipulating vote counts, let us now explore how to prepare for election rigging. The most critical aspect of preparation is estimating the scale of manipulation required. The step-by-step process for this estimation is as follows:

A. Investigate the expected turnout and each candidate's support share; based on these, predict each candidate's expected number of votes.
B. Designate the beneficiary and victim candidates.
C. Set the target vote share for the beneficiary candidate.
D. Choose the manipulation method.
E. Using the predicted vote counts, calculate the necessary scale of manipulation (e.g., the number of forged ballots) to achieve the target result.

Let's examine each step in detail.

3-1. Estimating the Expected Vote Count of the Target Candidate

The predicted support share and turnout are essential for determining whether manipulation is necessary and for estimating its scale. The expected votes for a candidate are calculated as:

Expected vote count = Voter total × Expected turnout × Support share

Using these figures, setting a target vote share for the beneficiary candidate allows for an initial estimate of the required manipulation — specifically, the number of forged ballots needed. Since these calculations rely on the predicted total votes and support share, accurate estimations of both are crucial to ensure successful manipulation.

Polling surveys are the most common method to estimate voter support without actual voting. When conducted objectively according to statistical standards, they can reliably predict support share. Manipulators would base their calculations on such objective poll results to maximize success.

However, publicly revealing actual poll support share before the election could leave traces, undermining the *Trace Removal* condition. Sudden discrepancies between manipulated results and polls might arouse suspicion. To avoid this, manipulators could conduct separate, deliberately skewed surveys that mimic the support share they intend to create, and then reveal these to match their rigging plan. Technically, skewing poll results by selectively sampling favorable voters is not particularly difficult.

3-2. Estimating the Scale of Manipulation

Is there a systematic way to calculate the minimum number of forged ballots required? Yes. By using the predicted vote counts for each candidate, we can estimate the necessary scale of manipulation.

Professor Maeng Ju-Seong's Manipulation Formula

After analyzing vote count data from the 2020 South Korean general election, Professor Maeng Ju-Seong identified signs of artificial manipulation. He found a correlation among candidates' vote counts, the required manipulation scale, and the number of forged ballots needed. Accordingly, he proposed formulas for each manipulation method:

$$\text{Add-vote} : K = R \times V_2 - V_1$$

$$\text{Swap-vote} : K = \frac{R \times V_2 - V_1}{R + 1}$$

$$\text{Discard-vote} : K = V_2 - \frac{V_1}{R}$$

where

V_1 : votes for the beneficiary candidate

V_2 : votes for the victim candidate

R : correction coefficient

K : the minimum number of forged ballots required

Using these formulas, a simple computer program can compute the minimum number of forged ballots needed and the resulting manipulated vote share for the beneficiary, given the votes for each candidate and the correction coefficient. The following Python code snippets implement each manipulation method:

```
# Input: V1 = votes for the beneficiary candidate,
#        V2 = votes for the victim candidate,
#         R = correction coefficient
# Output: K = the minimum number of forged ballots needed,
#       VR1 = the resulting vote percentage for the beneficiary

# Add-vote manipulation
def add(V1, V2, R):
    K = round(R * V2 - V1)
    VR1 = (V1 + K) / (V1 + V2 + K)
    return K, VR1

# Swap-vote manipulation
def swap(V1, V2, R):
    K = round((R * V2 - V1) / (R + 1))
    VR1 = (V1 + K) / (V1 + V2)
    return K, VR1

# Discard-vote manipulation
def discard(V1, V2, R):
    K = round(V2 - V1 / R)
    VR1 = V1 / (V1 + V2 - K)
    return K, VR1
```

Let's understand the principle of the manipulation formula through a hypothetical electoral district with 10,000 voters and two candidates. Suppose the expected turnout is 60%, the expected support share for the beneficiary candidate is 48%, and for the victim candidate is 52%. The expected vote counts for the two candidates are calculated as follows:

$$V_1 = 10{,}000 \times 0.6 \times 0.48 = 2{,}880$$
$$V_2 = 10{,}000 \times 0.6 \times 0.52 = 3{,}120$$

Using these expected votes and a correction coefficient R ranging from 1.0 to 2.0, we run the program to determine the required number of forged ballots K and resulting manipulated vote shares. The results are summarized in Table 2.

When the correction coefficient R is set to 1.0 and K is calculated using the manipulation formulas:

- Add-vote : $K = 240$ → The beneficiary's votes increase from 2,880 to 3,120, resulting in a tie with the opponent.
- Swap-vote : $K = 120$ → The beneficiary's votes become 3,000 (2,880 + 120), and the victim's decrease to 3,000 (3,120 - 120), thus tying the election.
- Discard-vote: $K = 240$ → The victim's votes decrease from 3,120 to 2,880, again resulting in a tie.

Table 2. A hypothetical election case showing relationships between R and K as well as between R and manipulated vote share for each method

Voters = 10,000 Turnout = 60% Vote Total = 6,000				Vote Count		Benefi-ciary	$V_1 = 2{,}880$		Victim	$V_2 = 3{,}120$	
	Add-vote			Swap-vote			Discard-vote				
	$K = R \times V_2 - V_1$			$K = \frac{R \times V_2 - V_1}{R + 1}$			$K = V_2 - \frac{V_1}{R}$				
R	K	Manipulated Vote Count		K	Manipulated Vote Count		K	Manipulated Vote Count		Manipulated Vote Share	
		V_1^+	V_2^+		V_1^+	V_2^+		V_1^+	V_2^+	Benefi-ciary	Victim
1.0	240	3,120	3,120	120	3,000	3,000	240	2,880	2,880	**50%**	**50%**
1.1	552	3,432	3,120	263	3,143	2,857	502	2,880	2,618	**52.4%**	**47.6%**
1.2	864	3,744	3,120	393	3,273	2,727	720	2,880	2,400	**54.6%**	**45.5%**
1.3	1,176	4,056	3,120	511	3,391	2,609	905	2,880	2,215	**56.5%**	**43.5%**
1.4	1,488	4,368	3,120	620	3,500	2,500	1,063	2,880	2,057	**58.3%**	**41.7%**
1.5	1,800	4,680	3,120	720	3,600	2,400	1,200	2,880	1,920	**60%**	**40%**
…	∧	∧	≡	∧	∧	∨	∧	≡	∨	∧	∨
2.0	3,360	6,240	3,120	1,120	4,000	2,000	1,680	2,880	1,440	**66.7%**	**33.3%**

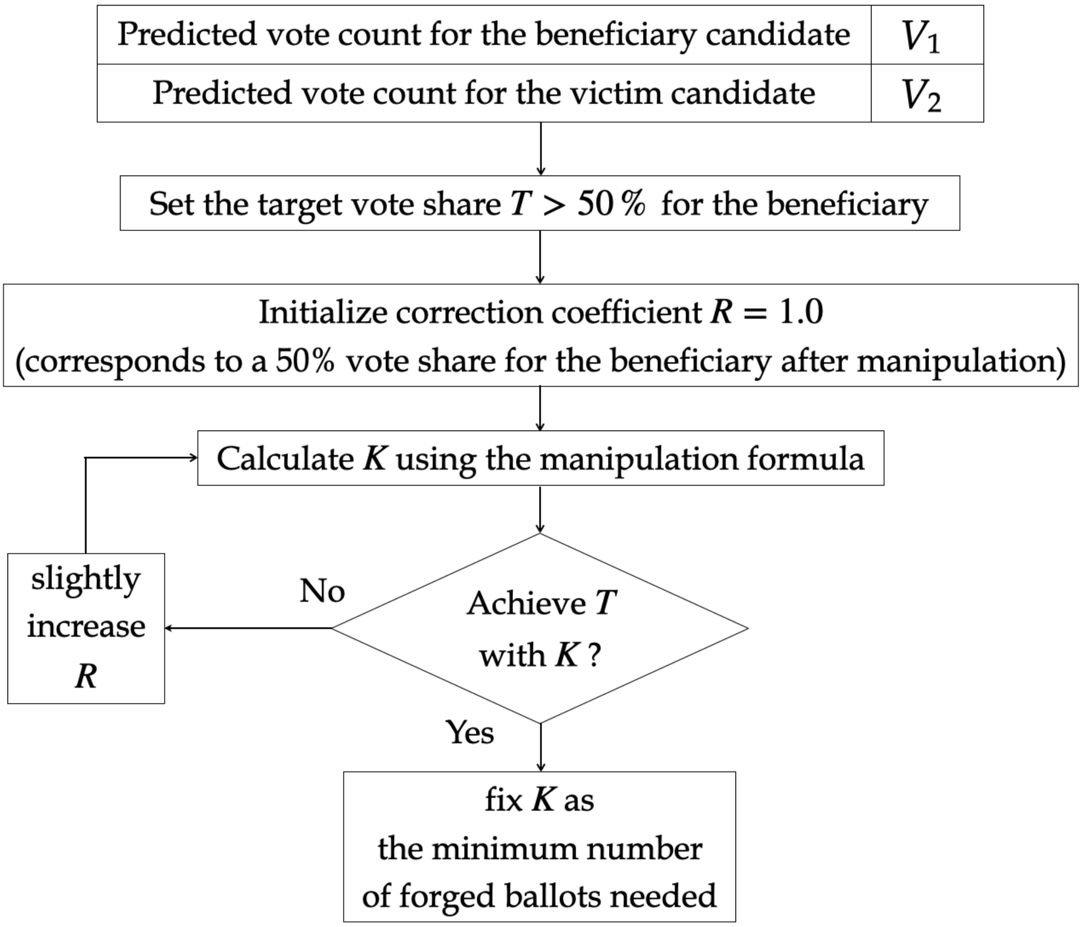
∧ : increasing, ≡ : no change, ∨ : decreasing

As the correction coefficient R increases beyond 1.0, the required number of forged ballots K also rises, and the advantage for the beneficiary candidate widens. When $R = 1.5$, the relative vote share after manipulation shifts to 60% for the beneficiary and 40% for the victim; at $R = 2.0$, it becomes approximately 66.7% vs. 33.3%.

Key point: Regardless of the initial vote counts V_1 and V_2, the relationship between R and the resulting post-manipulation vote share remains consistent across all manipulation methods. This means that even if the initial votes change, the link between R and the resulting vote share is always predictable. Leveraging this, one can develop an automatic program that calculates the minimum number of forged ballots needed to reach a target vote share for the beneficiary, based on the desired manipulation rate.

Automatic Calculation of the Minimum Number of Ballots Needed for Manipulation

Given the predicted votes for the beneficiary candidate V_1, the victim candidate V_2, and the target vote share T for the beneficiary, we can automatically compute the minimum number of forged ballots K needed using the manipulation formulas introduced earlier. The flow chart for the algorithm logic is as follows:



The process begins with $R = 1.0$, the value corresponding to no manipulation beyond equal vote. Then, gradually increase R, calculating the required forged ballots K each time, until the desired support rate T is reached. The value of K at that point represents the minimal number of forged ballots necessary to attain the target vote share.

This algorithm can be used for any manipulation method, since the only difference lies in the manipulation formula applied. To verify this, observe the implementation of the following program code, which takes the vote counts for both candidates, the target support share `T` for the beneficiary, and the manipulation formula `M` to calculate the minimum forged ballots:

```python
# Input: V1 = vote count for the beneficiary candidate,
#        V2 = vote count for the victim candidate,
#         T = target support share for the beneficiary,
#         M = manipulation formula, one of {add, swap, discard}
# Output: k = the minimum number of forged ballots
def rig_the_vote(V1, V2, T, M):
    r = 1.0                # Start with correction coefficient r = 1.0
    k, s = M(V1, V2, r)   # Calculate initial k and manipulated share s
    while s < T:               # Repeat until the target T is achieved
        r += 0.0001                 # Slightly increase r
        k, s = M(V1, V2, r)        # Recalculate k and s with new r
    return k
```

In the code, comments beginning with # are explanatory notes and are ignored by the computer during execution.

In this program, the increment of the correction coefficient r is set to 0.0001 based on the scale of the hypothetical election. The smaller step size improves the precision of the calculation. However, considering that the goal is simply to find the minimal number of forged ballots, a larger step might be more appropriate depending on the size and specifics of the election scenario.

Simulation of Manipulation Scale Calculation

Voters	10,000		Beneficiary	Victim
Expected Turnout	60%	Expected Vote Share	48%	52%
Expected Vote Total	6,000	Expected Vote Count	2,880	3,120

Using the previously discussed hypothetical electoral district above, we set the beneficiary candidate's manipulation target vote share to 52%, and run the program for each manipulation method to calculate the required scale of manipulation as follows:

```
>>> rig_the_vote(2880, 3120, 0.52, add)
500
>>> rig_the_vote(2880, 3120, 0.52, swap)
240
>>> rig_the_vote(2880, 3120, 0.52, discard)
462
>>>
```

The results indicate that the minimum number of forged ballots needed is 500 for add-vote manipulation, 240 for swap-vote manipulation, and 462 for discard-vote manipulation.

Assuming votes were cast according to the expected support shares in this hypothetical district, the outcome after manipulation using the calculated number of forged ballots for each method is summarized in Table 3.

Table 3. Changes in votes, vote ratio, total votes, and voter turnout before and after manipulation

Manipulation Target	52%	Beneficiary		Victim		Invalid		Voters	10,000
		Vote Count	Vote Share	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turn-out
Before		2,880	48%	3,120	52%			6,000	60%
Add-vote	500	3,380	52%	3,120	48%			6,500	65%
Swap-vote	240	3,120	52%	2,880	48%			6,000	60%
Discard-vote	462	2,880	48%	2,658	44.3%	462	7.7%	6,000	60%
Ralative Vote Share			52%		48%				

Let's examine each manipulation method to understand how these outcomes were achieved:

- Add-vote manipulation : The beneficiary's vote count increases by 500, raising their vote share to 52%. The victim's vote count stays the same, but their vote share drops to 48%. Since 500 ghost ballots have been added, vote total is higher, and turnout increases to 65%.
- Swap-vote manipulation : The beneficiary's vote count increases by 240, reaching 3,120 votes, with vote share at 52%. The victim's vote count decreases by 240, down to 2,880 votes, vote share at 48%. Vote total and turnout remain unchanged because the added votes for the beneficiary are exactly offset by the decreased votes for the victim.
- Drop-vote manipulation : By removing 462 votes from the victim and converting them into invalid votes, the victim's vote share drops to 44.3%. The proportion of invalid votes becomes 7.7%. The beneficiary's support share remains at 48%, but the victim's drops, changing the relative ranking. Now, excluding invalid votes, the vote share of the two candidates becomes 52%: 48%. Thus, the target vote share for the beneficiary candidate is successfully achieved. Like the swap method, this approach also relies on nullifying votes for the victim into invalid ballots, keeping vote total and turnout unchanged.

Traces of Manipulation

In the case of add-vote manipulation, as the target vote share increases, so does the voter turnout. If the target vote share for the beneficiary candidate is set too high, the resulting turnout can exceed 100%. In the hypothetical example, if the target vote share is set at 69%, the minimum number of ghost ballots is calculated as:

```
>>> rig_the_vote(2880, 3120, 0.69, add)
    4065
>>>
```

Since the original number of voters is 6,000, adding 4,065 ghost ballots would make the total votes 10,065, which exceeds the actual number of registered voters, 10,000. This clearly leaves obvious traces of manipulation, as the number of votes cannot surpass the number of actual voters.

3-3. Steps for Preparing Election Rigging

Let's follow the steps to prepare for election manipulation using the hypothetical election case we discussed earlier.

A. Predicting the Expected Turnout and Support Share for the Target Candidates
- An election district with 10,000 voters
- Expected voter turnout : 60%.
- Expected support share : Candidate 1 at 48%, Candidate 2 at 52%

B. Calculating the Expected Vote Counts for the Target Candidates
- Candidate 1 : 10,000 × 0.6 × 0.48 = 2,880 votes
- Candidate 2 : 10,000 × 0.6 × 0.52 = 3,120 votes

C. Setting the Manipulation Target
- Designate the Candidate 1 who is expected to lose as the *beneficiary candidate.*
- Designate the Candidate 2 who is expected to win as the *victim candidate.*
- Set the Candidate 1's manipulation target at 52%.

D. Propagating Distorted Support Shares
- Publicize manipulated support shares that align with the set manipulation target (e.g., showing Candidate 1 with 52% support).

E. Choosing the Manipulation Method
- Choose add-vote method, with a correction coefficient increase precision set at 0.0001.

F. Calculating the Number of Ghost Ballots

- Using the expected vote counts for the beneficiary (2,880), the victim (3,120), and the manipulation target of 52%, run the program to calculate the minimum required ghost ballots:

```
>>> rig_the_vote(2880, 3120, 0.52, add)
500
```

- The number of ghost ballots to issue for the beneficiary candidate = 500

If another manipulation method (e.g., swap-vote or discard-vote) is chosen instead in step E, only the calculation program would differ; the subsequent steps remain the same.

3-4. Relationship Between Vote Count Prediction Accuracy and Success Probability of Election Manipulation

The expected vote counts for each candidate are predicted based on the anticipated turnout and support shares for each candidate.

The expected turnout can be estimated by referring to past election records and utilizing various relevant data, such as age-group interest levels and voting participation trends. Table 4 illustrates how the difference between the predicted and actual turnout affects the manipulation outcome in the same hypothetical election case.

Table 4. The variation of vote share after add-vote manipulation according to the difference between estimated and actual turnout in the hypothetical election case

Actual Turnout	Before Manipulation						After 500 Add-vote Manipulation					
	Vote Total	Beneficiary			Victim		Total Votes	Beneficiary			Victim	
		Vote Count	Vote Share		Vote Share	Vote Count		Vote Count	Vote Share		Vote Share	Vote Count
30%	3,000	1,440	48%	<	52%	1,560	3,500	1,940	55%	>	45%	1,560
40%	4,000	1,920	48%	<	52%	2,080	4,500	2,420	54%	>	46%	2,080
50%	5,000	2,400	48%	<	52%	2,600	5,500	2,900	53%	>	47%	2,600
60%	6,000	2,880	48%	<	52%	3,120	6,500	3,380	52%	>	48%	3,120
70%	7,000	3,360	48%	<	52%	3,640	7,500	3,860	51%	>	49%	3,640
80%	8,000	3,840	48%	<	52%	4,160	8,500	4,340	51%	>	49%	4,160
90%	9,000	4,320	48%	<	52%	4,680	9,500	4,820	51%	>	49%	4,680
95%	9,500	4,560	48%	<	52%	4,940	10,000	5,060	51%	>	49%	4,940

Compared to the difference between the predicted and actual turnout, the variation in support share after manipulation is relatively small. Even if the actual turnout drops to half the expected 30% — resulting in a slight increase of about +3% for the beneficiary's support share — or if the turnout is as high as 95%, pushing the post-manipulation support share nearly 100%, the final support still falls short of the target by only about -1%. This small margin is enough to overturn the ranking. These results demonstrate that the impact of discrepancies between predicted and actual turnout on the success of manipulation is minimal.

On the other hand, the accuracy of the support share prediction (the support share of each candidate) is directly related to the success or failure of manipulation. If the actual support share of each candidate closely matches the predicted support share, manipulation is likely to succeed. However, if the actual support share deviates beyond a certain threshold, manipulation can fail. Table 5 illustrates how the outcome of manipulation varies when the actual support shares for the beneficiary and victim differ more than the originally predicted 48% to 52% in the same hypothetical scenario.

Table 5. The change of manipulation results according to the difference between the predicted and actual support rates in the same hypothetical election case

Before Manipulation						After 500 Add-vote Manipulation					
Vote Total	Beneficiary			Victim		Vote Total	Beneficiary			Victim	
	Actual Vote Count	Actual Vote Share		Actual Vote Share	Actual Vote Count		Actual Vote Count	Actual Vote Share		Actual Vote Share	Actual Vote Count
6,000	2,880	48%	<	52%	3,120	6,500	3,380	52%	>	48%	3,120
	2,820	47%	<	53%	3,180		3,320	51.1%	>	48.9%	3,180
	2,750	45.8%	<	54.2%	3,250		3,250	50%	=	50%	3,250
	2,700	45%	<	55%	3,300		3,200	49.2%	<	50.8%	3,300
	2,640	44%	<	56%	3,360		3,140	48.3%	<	51.7%	3,360

Specifically, if the beneficiary candidate's actual support share drops more than 2.2 percentage points below the expected 48% — that is, falling to 45.8% or lower — even adding 500 ghost ballots for manipulation will not be enough to overturn the candidate's victory. This phenomenon similarly applies also to swap-vote and discard-vote manipulations.

4. Ballot Box Manipulation

Ballot box manipulation is a type of election tampering carried out before the vote counting begins. How can we maximize the fulfillment of the "Three Necessary Conditions for Successful Election Rigging" when performing ballot box manipulation? Most importantly, regardless of the complexity of execution, it is crucial to allocate sufficient time for the process.

In the environment where votes are cast and counted on the same day, with voters, candidates, and observers overseeing the entire process, manipulating the ballot box becomes effectively impossible. Therefore, securing enough time is an essential requirement for successful ballot box manipulation.

A clever solution to ensure ample time is to adopt **early (absentee) voting**, which allows voters to cast ballots before the official election day. By doing so, manipulation can be performed during the early voting period—when there is enough time—rather than on election day itself, when the environment is highly monitored. Early voting involves keeping the ballot boxes in secure, sealed locations for an extended period, providing sufficient time to carry out manipulations quietly, often with only a few people, thus avoiding detection by voters, candidates, or observers.

If the claim of enhancing convenience and increasing turnout is used to promote early voting, gaining public support for its necessity may not be difficult.

4-1. Obstacles and Challenges in Early Ballot Box Manipulation

Simulation of Early Ballot Box Manipulation

Let's examine what happens if only the early ballot box is manipulated, leaving the election day ballot box untouched, using the same hypothetical scenario. Suppose the early voting turnout is 20%, the election day voting turnout is 40%, and votes are cast at support shares of 48% for the beneficiary and 52% for the victim. After votes are cast, the status of the early and election day ballot boxes would be as shown in Table 6.

Assuming we determine the minimum forged ballots needed to achieve the manipulation goal based on the total expected votes from both early and election day voting, we then perform manipulation solely on the early ballot box. If the manipulation goal for the beneficiary is set at 52%, the outcomes of each manipulation method are summarized in Tables 7a, 7b, and 7c.

Table 6. Expected election results in the case of split voting between early and election day, based on the hypothetical district

Expected Election Result	Beneficiary		Victim		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Early	960	48%	1,040	52%	2,000	20%
Election Day	1,920	48%	2,080	52%	4,000	40%
Total	2,880	48%	3,120	52%	6,000	60%

Table 7a. Early Ballot Box Add-vote Manipulation (ghost ballots = 500)

Ballot Box	Beneficiary		Victim		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Early	1,460	58%	1,040	42%	2,500	25%
Election Day	1,920	48%	2,080	52%	4,000	40%
Total	3,380	52%	3,120	48%	6,500	65%

Table 7b. Early Ballot Box Swap-vote Manipulation (fake ballots = 240)

Ballot Box	Beneficiary		Victim		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Early	1,200	60%	800	40%	2,000	20%
Election Day	1,920	48%	2,080	52%	4,000	40%
Total	3,120	52%	2,880	48%	6,000	60%

Table 7c. Early Ballot Box Discard-vote Manipulation (damaged ballots = 462)

Ballot Box	Beneficiary		Victim		Invalid		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Early	960	48%	578	28.9%	462	23.1%	2,000	20%
Election Day	1,920	48%	2,080	52%	0	0%	4,000	40%
Total	2,880	48%	2,658	44.7%	462	7.7%	6,000	60%
		52%		48%				

Analysis of Obstacles in Early Ballot Box Manipulation

If the votes follow the predicted pattern, all numerical targets are easily achieved. But what about satisfying the *Three Necessary Conditions for Successful Election Rigging*?

1. First, the condition of *Complete Manipulation*—which requires that the same election result is maintained even after a recount—is easily met. Since the manipulation of the ballot box is finalized in accordance with the target outcome before the tallying begins, any subsequent honest counting should produce the same result.

2. Second, the obstacle related to *Trace Removal*—which requires leaving no evidence of tampering—varies depending on the manipulation method. Let's review each:
   - Add-vote: To execute this, one must identify non-voting individuals and transform them into ghost voters to increase the vote count. However, accurately identifying all non-voters beforehand is extremely difficult. Moreover, gradually increasing the number of ghost voters during the voting period is necessary because real-time recording and public disclosure of cumulative votes could reveal unnatural increases, leaving traces. A more challenging obstacle is making ghost voters, who are not physically present at the polling station, sign the voter register—a practice that is virtually impossible without detection.
   - Swap-vote: Removing ballots for the victim candidate entails the physical destruction or removal of votes without leaving traces and without exposing the process to observers.
   - Discard-vote: Making the support for the victim candidate appear as invalid votes results in an excess of invalid ballots compared to the actual election day, which leaves obvious traces. To conceal this, comparable amounts of invalid votes must be created and inserted into the election day ballot box—an operation that is risky and complicated. Furthermore, manipulating and damaging ballots through various methods increases the risk of leaving detectable traces.
   - The most critical common challenge across all three methods is the phenomenon where support shares for early votes and election day votes

contradict each other. For example, in the hypothetical scenario mentioned earlier, the manipulated results in Table 8 show that:

- On election day, the victim candidate wins by 4 percentage points.
- However, in early votes, the beneficiary candidate leads by as much as 16-20 percentage points, depending on the method.

Such a large discrepancy in support shares over just a few days among the same population is statistically impossible according to the law of large numbers, thus leaving unmistakable traces of manipulation.

Table 8. Summary of the vote rate contradiction phenomenon

Manipulation Method	Early Vote Share			Election Day Vote Share		
	Beneficiary		Victim	Beneficiary		Victim
Add-vote	58%	>	42%	48%	<	52%
Swap-vote	60%	>	40%	48%	<	52%
Discard-vote	48%	>	28.9%	48%	<	52%

3. Third, the *Secrecy in Execution* condition—the need to prevent revealing manipulation—is mainly maintained during the manipulation process. Key preparations include:
   - Ensuring enough time to carry out actions quietly without rushing.
   - Securing isolated locations away from outside contact.
   - Developing covert methods to bring ballots in secretly.
   - Using procedures to unseal and reseal ballot boxes without appearing suspicious.
   - Implementing measures to prevent involved persons from leaking details.

   Any lapses in these areas increase the risk of detection. Careful planning and strict secrecy are essential to avoid exposure.

When using early voting to gain extra manipulation time, different outcomes are observed depending on the manipulation technique and the fulfillment of the three conditions. The summarized problems associated with early ballot box manipulation are listed in Table 9.

Table 9: Obstacles in early vote manipulation

Item	Add-vote	Swap-vote	Discard-vote
Turnout	Must increase votes without leaving traces that non-voters apparently voted	No change	No change
Complexity of manipulation	Simplest among the three: just insert ghost ballots into the box	Burden of labor and time to dispose of genuine ballots safely and discreetly	• Same as swap-vote, difficulty in damaging genuine ballots • Large discrepancies between early and election day invalid votes
Common	Large discrepancies often appear between early and election day votes, revealing tampering		

Overcoming Obstacles in Early Ballot Box Manipulation

Let's examine how to overcome these obstacles during manipulation.

- The most significant obstacle common to all methods is the *conflicting results between early and election day votes*. This issue naturally arises when only the early ballot box is manipulated. Assuming that manipulation of the election day ballot box at the same scale is impossible, failing to meet the *Trace Removal* condition remains a major challenge.
- The discard-vote manipulation, which involves a stark difference in invalid vote shares between early and election day ballots, faces the same issue. To eliminate traces, the same scale of discard-vote manipulation would need to be performed on the election day ballot box. Therefore, to compensate for vulnerabilities in *Trace Removal*, one must also conduct discard-vote manipulations on election day, despite the increased difficulty of covert operations.
- The shared challenge of *Trace Removal* and *Secrecy in Execution* in swap-vote and discard-vote manipulations can be somewhat alleviated if well-trained accomplices systematically and flawlessly carry out the operations without leaving traces.

- Another approach involves *completely replacing the real ballot box with a forged one*. A pre-made fake ballot box, designed to match the manipulation target with counterfeit ballots, can be swapped after early voting ends. This provides ample time for clandestine replacement. Although this increases the burden of *Trace Removal* and *Secrecy in Execution*, it offers the advantage of perfectly achieving the target support share, regardless of prediction accuracy.
- The major obstacle unique to add-vote manipulation—*the ghost voter problem*—can be partly addressed through electoral system reform and information technology. We will explore this approach in detail in the next section.

4-2. Strategies and Methods for Issuing Ghost Ballots

A major obstacle in add-vote manipulation is the selection of ghost voters. In elections with only same-day voting, overcoming this challenge is nearly impossible because voters must appear in person, verify their identity with ID, and sign the voter list. However, if the early voting system is well-organized using information technology to facilitate manipulation, it becomes possible to create ghost voters discreetly. Let's examine how.

Diversification of the Early Voting System

Expand early voting opportunities so that voters can cast ballots freely anywhere in the country or abroad, not just within their local district. For domestic voters, provide options such as in-district early voting (voting at a designated polling station within their district) and out-of-district early voting (voting at any precinct nationwide), to improve convenience. To implement this, all early voting sites nationwide should be equipped to handle both in-district and out-of-district early votes simultaneously. For voters residing abroad and not registered for overseas voting, further diversify early voting options through methods like residence overseas voting or overseas resident voting, thereby broadening participation. In in-district early voting, votes are stored in designated local facilities until counting; for out-of-district or overseas early votes, ballots are mailed to the voter's district and stored separately until the count. This diverse early voting system creates opportunities to insert ghost ballots.

Methods for Issuing Ghost Ballots

While creating anonymous ghost ballots (unsigned ballots marked for the beneficiary candidate) is relatively straightforward, transforming non-voting individuals into ghost voters and recording them poses a significant challenge. Additionally, issuing all ghost ballots at once can cause a sudden spike in total votes, leaving obvious traces of manipulation. Let's explore how to address these two core issues.

- **Issuance of Early Ballots via Online Ballot Issuing Machine**

  When a voter verifies their identity, signs the voter registry at a designated polling station, and receives their ballot, there is no possibility of issuing a ghost ballot. However, if the early voting system is improperly organized, this possibility could arise. Early voting should be possible both within and outside the jurisdiction.

  For in-jurisdiction voting, voter registration lists can be made available, as the target audience is limited to local voters. However, for out-of-jurisdiction voting—where the target audience is all limited to local voters—it is practically impossible to provide a registry list, thus requiring alternative solutions.

  The only feasible solution appears to be establishing a nationwide "*Unified Voter Registry*" electronic system and developing an "*Online Ballot Issuing Machine*" to be installed at early voting stations across the country. Using internet infrastructure, voter verification and ballot issuance could be centrally managed online. This would allow voters to visit any early voting station nationwide, verify their identity electronically via the machine, and receive their early ballot.

  In this system, early voting stations would primarily handle the installation and management of the ballot issuing machines, while the central authority would be responsible for maintaining and managing all early voting records.

  However, this setup introduces the risk of issuing ghost ballots without detection at the voting site itself. (In reality, to ensure the integrity of the ballot issuing machine, early voter lists should be recorded with signatures at the voting site, allowing later comparison with the central unified voter registry. Yet, such procedures would leave traces of tampering and would thus never be implemented.)

  Furthermore, under the pretext of simplifying and standardizing early voting, in-jurisdiction early voting could also be encouraged to utilize the online ballot issuance system from the central platform. This would eliminate the need to keep the local voter registry at early voting sites, which in turn could facilitate the clandestine issuance of ghost ballots from the central system during in-jurisdiction early voting, without detection on-site.

- **When and how to select voters for ghost voting?**
  Since voters' intent to vote can change at any time before the polls close, arbitrarily selecting ghost voters risks detection. It is unthinkable that someone who hesitated or was about to vote would suddenly discover they have already voted. Therefore, the only feasible approach is to pre-collect a list of voters who are certain not to appear at the polls and prepare the necessary ghost ballots in advance. Possible targets include overseas residents who did not file early ballots or individuals physically unable to vote.

- **How to insert ghost ballots into the ballot box?**
  Since ballots can be issued online at any early voting site for both in-district and out-of-district voting, ghost ballots can be pre-produced at a separate location. These ballots can then be collected into district-specific boxes and secretly inserted into the official ballot boxes just before counting.

- **How to inflate ghost voter numbers during voting?**
  Because the total number of voters is publicly updated at fixed intervals during the voting period, issuing many ghost ballots simultaneously online could cause an unnatural spike, easily detectable as tampering. To avoid this, ghost ballot issuance must be carefully timed during the early voting period. A practical tactic is to issue ghost ballots at a fixed ratio compared to real ballots: for example, when the number of genuine ballots is E and the number of ghost ballots is K, calculate a correction factor:
  $$n = \lfloor E \div K \rfloor$$
  Here the symbol $\lfloor \_ \rfloor$ means to discard the decimal places and keep only the integer. Every time the number of genuine ballots issued reaches a multiple of n, an additional ghost ballot is issued. For instance, if $E = 100$ and $K = 30$, then:
  $$n = \lfloor E \div K \rfloor = \lfloor 100 \div 30 \rfloor = \lfloor 3.333... \rfloor = 3$$
  Every time 3 real early ballots are issued, 1 additional ghost ballot can be issued. For every 3 people who cast an early vote, the number of ghost early voters increases by 1.

  Let us examine how ghost ballots are issued on the early voting day when 100 voters vote exactly as expected, and the correction multiplier is set to 3. Since voters' choices are secret and cannot be known, real ballots are marked with an 'x', while ghost ballots are marked with a '1'. If we list the issued ballots in order, 50 per row, the sequence looks like this:

```
xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xx
x1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1
xxx1xxx1xxx1xxx1xxx1xxxxxxxxxx
```

Including ghost ballots, a total of 130 ballots are issued in this sequence. Because 100 divided by 30 leaves a remainder of 10, the ghost ballot issuance goal is prematurely reached. The remaining 10 ballots are filled with 'x' in the subsequent part.

If actual early voters match or exceed the expected 100, reaching the ghost ballot issuance goal is not a problem. However, if the number of early voters is significantly less—say fewer than 90—the remaining ghost ballots must be issued all at once right before voting ends. For example, if only 80 voters vote early instead of 100, the issuing sequence might look like this:

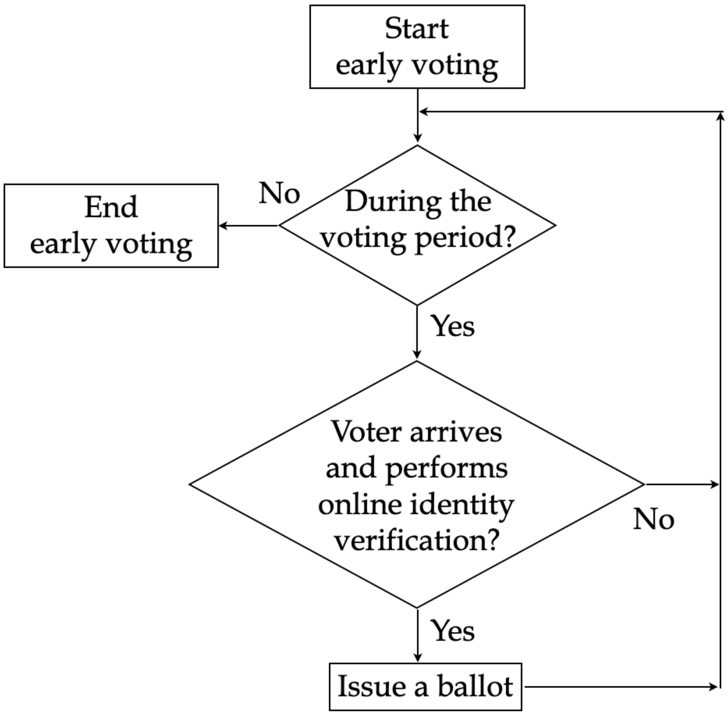
```
xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xx
x1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1xxx1
xxx1xx1111
```

In this case, four ghost ballots (which had not yet been issued) are forced to be issued all at once at the end. Since many voters vote in a short period at the very end, this also leaves a clear trace of manipulation.

To address this, the issuance rate of ghost ballots must be dynamically adjusted based on the real-time voting rate. If the current hourly voting rate falls below the expected rate, the correction multiplier value is temporarily decreased. When the hourly rate recovers, the multiplier resets to its original value. By controlling the issuance rate in this manner, ghost ballots can be issued more evenly over time, helping achieve the target before voting ends without leaving obvious traces.

Ballot Issuance Program

The procedure for properly issuing ballots with a ballot issuance machine follows this algorithm:



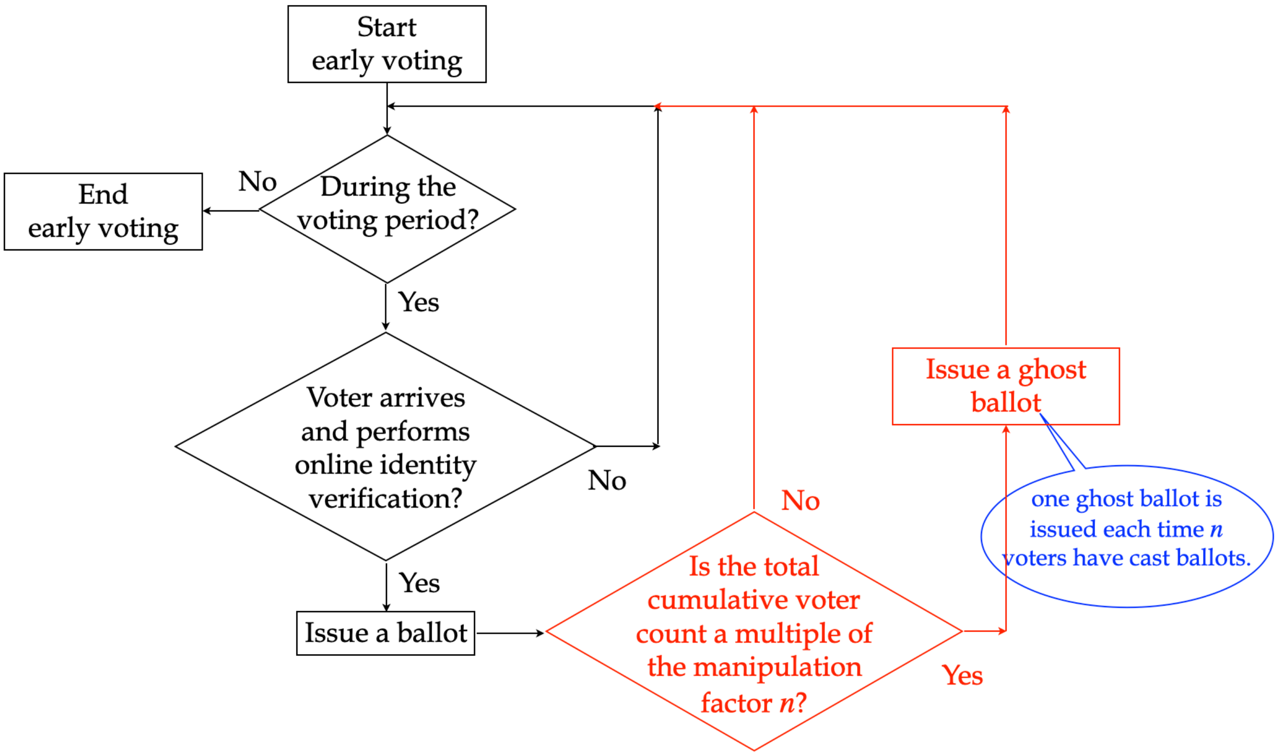
The program code for proper ballot issuance can be simply written as follows:

```
def issue_ballots():
    total = 0             # Initialize voter count to 0
    while on_time():      # During the voting period?
        if check_id():    # Voter arrives and identity is verified online
            total += 1              # Increment ballot count
            issue_a_ballot(total)   # Issue a ballot
    return total          # Return the total number of ballots issued
```

Comments that begin with # explain the corresponding code on the same line.

Ghost Ballot Issuance Program

The procedure for issuing ghost ballots can be implemented by inserting the necessary logic to the normal ballot issuance process. The parts highlighted in red in the following flowchart indicate the additional logic for ghost ballot issuance.



The extended logic allows one ghost ballot to be issued each time n voters have cast ballots. The extended version of the ballot issuance program, with the ghost ballot issuance function added, is shown below. Lines 8 to 11, which were newly inserted, include comments explaining the purpose of the code.

```
1  def issue_ballots(n):
2      total = 0
3      ghost = 0
4      while on_time():
5          if check_id():
6              total += 1
7              issue_a_ballot(total)
8              if total % n == 0:  # Check if total is a multiple of n
9                  total += 1              # Increase total ballot count
10                 issue_a_ballot(total)   # Issue a ghost ballot
11                 ghost += 1.             # Count the ghost ballots issued
12     return total, ghost # Return total ballots issued and ghost ballots count
```

4-3. Summary

Since ballot box manipulation before vote count requires ample time, introducing an early voting system becomes the only viable solution. Under the pretext of enhancing voter convenience and increasing turnout, early voting is expanded to include in-jurisdiction, out-of-jurisdiction, and overseas voting. Then, in the name of improving the efficiency and consistency of early voting operations, centralized online ballot issuing machines are introduced to issue both in-jurisdiction and out-of-jurisdiction ballots from a central authority. This creates an environment where ghost ballots can be issued while satisfying key conditions for *Secrecy in Execution* and *Trace Removal*. Ghost ballots are issued at regular intervals during the early voting period, and ghost voters are selected from a pre-prepared list of individuals identified in advance as unable to participate in the election.

Since pre-ballot box manipulation is completed before vote tallying, the *Complete Manipulation* condition can be easily satisfied. However, the execution must be meticulously planned to maximize secrecy and trace removal. One prominent trace of tampering—discrepancies between support shares in early votes and election day votes—is highly noticeable and problematic.

Another point to consider is that the scale of manipulation is estimated based on predicted support shares and voter turnout, not on actual votes. If the actual support share deviates significantly from predictions, manipulation may fail to reach the target or may fail altogether, especially if support outcomes differ greatly from initial estimates.

5. Ballot Counting Manipulation

Ballot counting manipulation involves tampering during the vote-counting process itself. Once voting has ended, adding votes becomes essentially impossible because the turnout is finalized and announced publicly. Therefore, this section focuses on how to perform swap and discard manipulations while maximally satisfying the *Three Necessary Conditions for Successful Election Rigging*.

5-1. Characteristics and Obstacles of Counting Manipulation

Swap-vote Counting Manipulation

During the count, classifying ballots marked for the victim candidate as votes for the beneficiary candidate—known as swap manipulation—results in the victim's votes disappearing and the beneficiary's votes increasing.

To satisfy the *Triple Necessary Conditions for Successful Election Rigging*, certain obstacles must be overcome:

- Visual manual counts, observed by multiple witnesses, make covert manipulation impossible unless collusion occurs. Even with collusion, doing so in an open environment greatly complicates concealment.
- Even if the manipulation successfully affects the tally, ballots classified as victim votes would remain inside the ballot box, failing the *Complete Manipulation* condition. A manual recount would reveal the tampering, so post-manipulation alterations to the ballot box are necessary.
- Since there is enough time between the initial vote count and the official recount, executing the ballot box replacement while carefully maintaining secrecy can satisfy the *Complete Manipulation* condition.

Discard-vote Counting Manipulation

If, during counting, ballots marked for the victim candidate are classified as invalid (null) votes, the discard manipulation effect is achieved. The main obstacles are similar to those in swap manipulation: maintaining secrecy and avoiding detection.

Approaches to Counting Manipulation

In summary, manual counts observed directly by multiple witnesses make covert manipulation very difficult—unless all observers collude. Since perfect collusion is unlikely, manual counting manipulation—while possible in theory—is practically unfeasible due to the difficulty in maintaining secrecy.

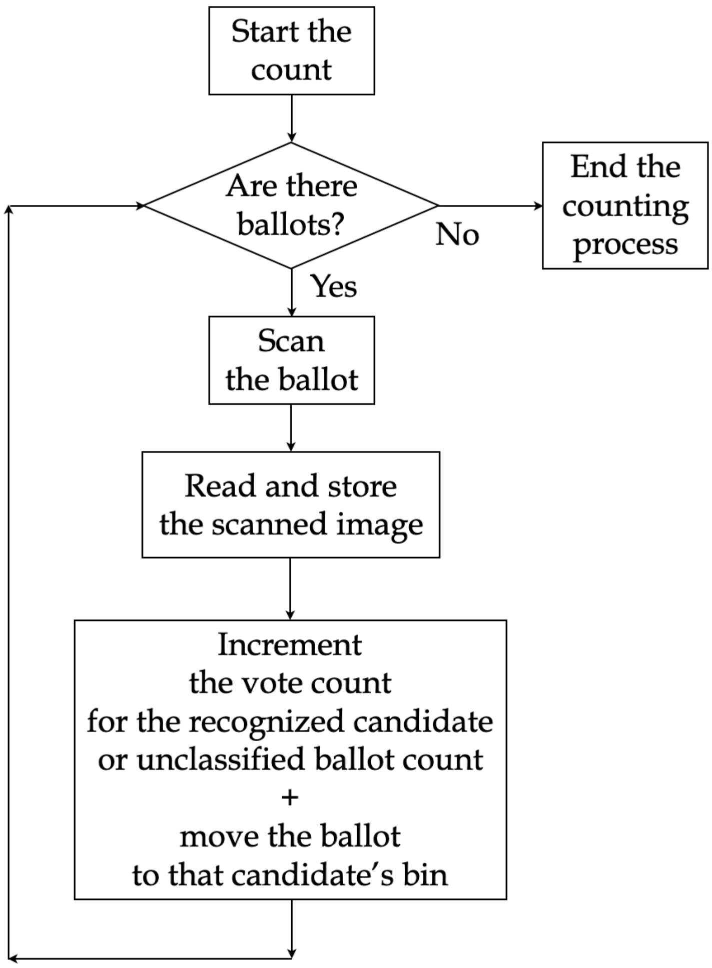
The more feasible approach is to minimize the need for verification and surveillance during counting by automating the process. By utilizing machines, electronics, and information technology, we can develop automatic electronic vote counting machines that perform fast, accurate counting—reducing the risk of detection and enabling subtle manipulation.

Is it convincing to advocate for the necessity of automated electronic vote counting? Given that image recognition and related technologies are highly advanced and widely used in daily life, replacing slow, error-prone manual counts with fast, reliable, and transparent electronic counting devices seems a natural and logical step to improve efficiency. Such devices are likely to be well received, as they promise faster and more accurate vote counts than manual counts by human counters.

In conclusion, replacing manual counts with automated electronic counters, allows the count to be controlled internally by software, which processes votes quickly and efficiently. This makes it more difficult for external observers to verify every aspect, thereby enabling more covert manipulation. Comparing standard counting logic with manipulation logic helps us understand how electronic vote counters could potentially be manipulated.

5-2. Vote Counting Machine

When a bundle of marked ballots is loaded into the system and operated, the electronic vote counting machine scans each ballot, recognizes the marked content, and automatically tallies the votes for each candidate, sending the ballot to the designated candidate's storage bin. The classification algorithm logic for proper operation of this machine is as follows:



The classification process can be understood by following the flowchart from the top. The system scans ballots individually until the entire batch is processed. Each scanned image is interpreted and stored, the candidate's vote tally is incremented accordingly, and the ballot is routed to the appropriate bin. If the system cannot confidently assign a ballot to any candidate due to image ambiguity, it is flagged for manual review. Such ballots are counted as pending and sent to a separate unclassified bin. Final determination is made through manual inspection. Invalid ballots, which lack clearly defined criteria, are also categorized as pending and resolved via manual counting. The following program simulates the operation of the machine:

```python
#     box = storage for scanned ballot images
#   count = record of vote tallies
# ballots = bundle of ballots to be counted
def classify_ballots(box, count, ballots):
    for b in ballots:  # take each ballot from the bundle
        image = scan(b)             # scan the ballot
        detected = detect(image)    # recognize the images
        box[detected].append(image) # store the scanned image
        count[detected] += 1        # increment the candidate's vote count
        # move the ballot to the appropriate classification bin
        # (excluded in the simulation)
    return box, count  # final tally after processing all ballots
```

Reading the comments marked in red on the right side helps understand how the code reflects the described flowchart. Provided there are no recognition errors—such as misclassifying a ballot for one candidate as for another—the machine should operate smoothly, as demonstrated in the simulated experiment below.

In this simulation, 100 ballots mixing 48 votes for Candidate 1 and 52 votes for Candidate 2 are randomly mixed and fed into the machine, assuming a recognition failure rate of approximately 5%. The results are as follows:

```
Number of ballots: 100
12221212121221222222112211122122222121112112112212
21211112111221221112212111121212222122121221111112
-----
Classification results:
   Candidate 1 = 46 votes
   Candidate 2 = 50 votes
   Unclassified = 4 ballots
---
Classified into bin 1:
1111111111111111111111111111111111111111111111
---
Classified into bin 2:
22222222222222222222222222222222222222222222222222
---
Unclassified ballots:
1221
```

Excluding the 4 ballots that were not classified due to recognition failure, the remaining 96 ballots were correctly classified.

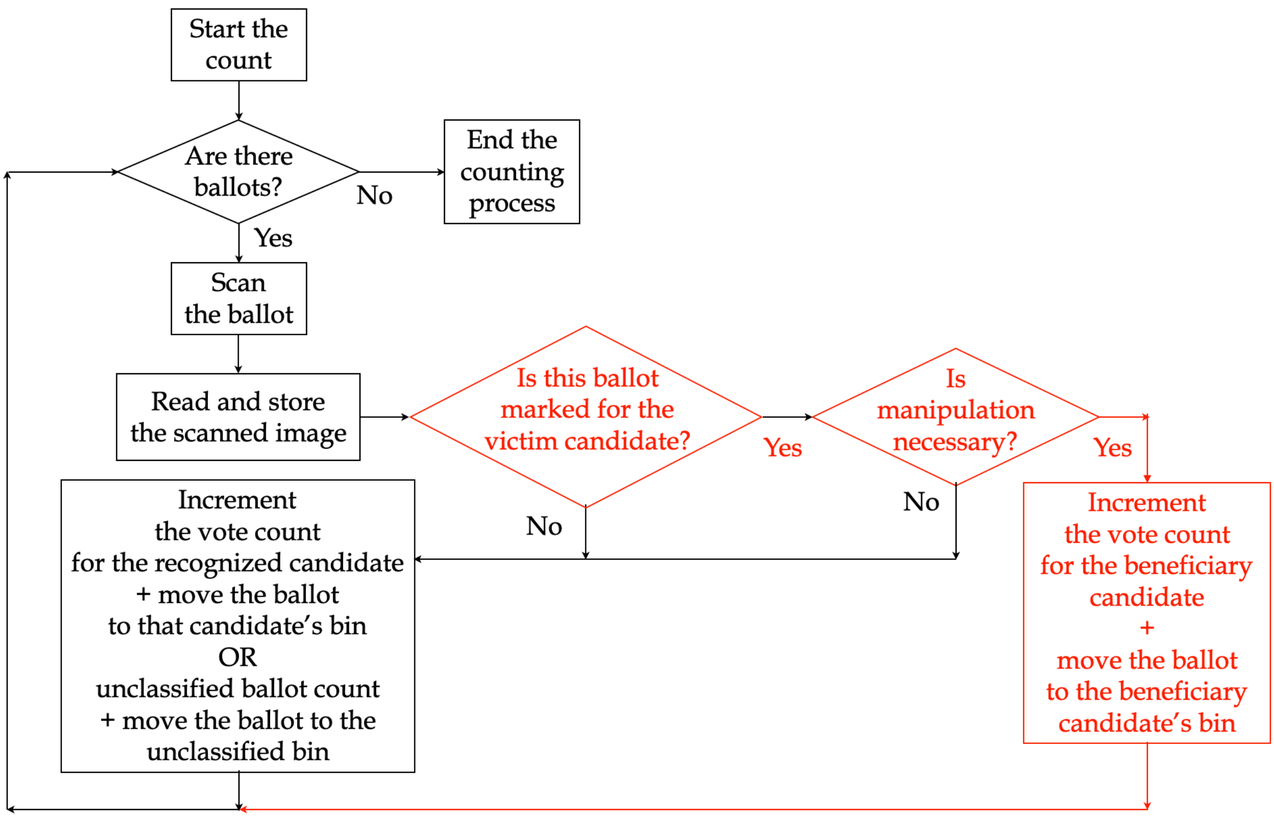
Looking at the core operation of the electronic vote counting machine, the built-in computer operates independently without external communication. It only requires input of superficial information like the scanning area and the number of candidates beforehand. However, for manipulation purposes, additional information is necessary. Let's explore how manipulation could be achieved without significant structural changes to the machine.

5-3. Vote Counting Machine Manipulation

Manipulating the electronic vote counting machine involves deliberately misclassifying ballots, rather than classifying them solely based on the scanner's recognition. In swap-vote manipulation, ballots for the victim candidate are intentionally reclassified as votes for the beneficiary candidate. Conversely, for discard-vote manipulation, votes for the victim candidate are misclassified as invalid or null votes.

Due to ambiguities in judgment criteria and the inherent characteristics of electronic vote counters, the system typically classifies votes as requiring manual re-verification when recognition confidence is low. The final classification of such ballots is performed through visual inspection afterward. As a result, complete ballot invalidation solely via electronic manipulation is practically infeasible. Classifying ballots as needing re-verification often leads to manual inspection, where manipulation risks being detected and thwarted. Therefore, the primary focus shifts to how to systematically perform ballot swapping using an electronic vote counting system.

The following flowchart illustrates how a systematic manipulation process can be integrated into the counting algorithm:



Every time the scanner recognizes a ballot marked for the victim candidate, the system checks whether manipulation is needed. If yes, it reclassifies that ballot as for the beneficiary candidate and moves it accordingly. If no, the ballot remains classified as for the victim candidate.

This manipulation classification algorithm can be used alongside normal classification. If the "Is manipulation necessary?" question always yields "No," the process simply functions as normal—no manipulation occurs, and only the standard classification is performed.

How to set the condition for answering "Yes" or "No"? The core concept involves setting a predetermined frequency—based on the expected vote count for the victim candidate—at which ballots are intentionally misclassified to favor the beneficiary candidate. The swap-vote manipulation frequency N can be determined as follows:

$$\lfloor V \div K \rfloor = N$$

where V is the predicted vote count for the victim candidate and K is the number of ballots to swap.

The system operates as follows: Each time the count of ballots marked for the victim candidate reaches a multiple of N, the algorithm misclassifies the ballot as for the beneficiary candidate. Otherwise, it classifies ballots normally. This systematic approach ensures a consistent manipulation pattern aligned with the target support share.

The following is a program implementation of a simulation that executes the swap-vote manipulation during classification:

```python
# classify_ballots : Electronic ballot counter operation
#   box : storage for scanned ballot images
#   count : storage for candidate vote tallies
#   ballots : batch of ballots to be processed
def classify_ballots(box, count, ballots):
    count_victim = 0   # Initialize victim candidate's vote count to 0
    for b in ballots:
        image = scan(b)
        detected = detect(image)
        if detected == VICTIM:  # Is the vote for the victim candidate?
            if count_victim % N == 0: # Is the count a multiple of N?
                detected = BENEFICIARY  # Manipulate swap classification
            count_victim += 1         # Increment victim's vote count
        box[detected].append(image)
        count[detected] += 1
    return box, count
```

To manipulate classification at regular intervals, three pieces of information are required: the **beneficiary candidate**, the **victim candidate**, and the **manipulation**

frequency. These correspond to the variables `BENEFICIARY`, `VICTIM`, and `N` in the program.

In line 6, a counter is initialized to accumulate the number of votes received by the victim candidate, which is used to determine when the manipulation should occur. The logic in lines 10–13 handles cases where a ballot is interpreted as a vote for the victim. Line 11 checks whether the accumulated count is divisible by `N`. If so, line 12 reassigns the vote to the beneficiary candidate, simulating a manipulated classification.

It is easy to run this manipulation-classification program in **normal mode** (i.e., without manipulation). To disable manipulation, one simply sets the value of `N` to an arbitrarily large number—specifically, a value larger than the total number of ballots to be processed. This ensures the condition in line 11 is never met, and thus, no vote swapping ever occurs.

Let us now simulate the manipulation using the same hypothetical election scenario as before. Suppose the expected number of votes for the victim candidate (Candidate 2) is 3,120, and the number of ballots to be swapped is 240. Then the manipulation frequency N can be calculated as follows:

$$N = \lfloor 3{,}120 \div 240 \rfloor = 13$$

A simulation was conducted using the same setup as the normal operation. A total of 500 ballots were shuffled randomly and divided into five batches of 100 each. The manipulated classification program, with the manipulation frequency set to 13, was then executed batch by batch using the electronic ballot counter. The results of the manipulated count are as follows:

```
* Ballot bundle #1 *
Number of ballots: 100
12122221111111121211212212221222122121221212221212
12111221212121222212212222121122111121121111121222
1 = 48 2 = 52
-----
Classification results:
   Candidate 1 = 47
   Candidate 2 = 48
   Unclassified = 5
---
Classified into bin 1:
12111111111111121111112111111111211111111111111
---
Classified into bin 2:
222222222222222222222222222222222222222222222222
---
Unclassified ballots:
11111
---

*Ballot bundle #2 *
Number of ballots: 100
12111122222222122221221112212221212122221111122221
12212212212112122112122212121121111111222211112222
```

```
1 = 45 2 = 55
-----
Classification results:
   Candidate 1 = 48
   Candidate 2 = 47
   Unclassified = 5
---
Classified into bin 1:
121111111121111111112111111111112111111111111111
---
Classified into bin 2:
22222222222222222222222222222222222222222222222
---
Unclassified ballots:
12222
---

* Ballot bundle #3 *
Number of ballots: 100
11122121212222211112211212212221121111122121122212
21222221211222221111211221111221221111121212221221
1 = 49 2 = 51
-----
Classification results:
   Candidate 1 = 50
   Candidate 2 = 46
   Unclassified = 4
---
Classified into bin 1:
11121111111112111111111111121111111111111211111111
---
Classified into bin 2:
2222222222222222222222222222222222222222222222
---
Unclassified ballots:
1211
---

* Ballot bundle #4 *
Number of ballots: 100
21222111212112211122111111122111212122111211212121
12122122221112112122211211211112112111212222221212
1 = 55 2 = 45
-----
Classification results:
   Candidate 1 = 58
   Candidate 2 = 40
   Unclassified = 2
---
Classified into bin 1:
21111111111111111111211111111111112111111111111111
11111211
---
Classified into bin 2:
2222222222222222222222222222222222222222
---
Unclassified ballots:
12
---

* Ballot bundle #5 *
Number of ballots: 100
12122122221111122221212121222122212222122121212221
22112121121222111112221222212111221221222211111212
1 = 43 2 = 57
-----
```

```
Classification results:
   Candidate 1 = 44
   Candidate 2 = 48
   Unclassified = 8
---
Classified into bin 1:
12111111111111211112111111111112111111111111
---
Classified into bin 2:
222222222222222222222222222222222222222222222222
---
Unclassified ballots:
22221121
---
```

Notice that in the classification results, Candidate 1's ballots are recognized at a consistent ratio of 13:1 as Candidate 1 or 2, illustrating that the manipulation is occurring systematically to maintain the target support share.

Table 10 presents the outcomes of a simulated experiment in which ballot classification was manipulated at a predetermined frequency. A total of 500 ballots were prepared, comprising 240 votes for Candidate 1 (the beneficiary candidate) and 260 votes for Candidate 2 (the victim candidate), corresponding to vote shares of 48% and 52%, respectively. These ballots were processed by the electronic counting machine in five separate batches of 100. Following the manipulation protocol, 247 ballots (49.4%) were allocated to Candidate 1, 229 ballots (45.8%) to Candidate 2, and 24 ballots (4.8%) were designated for re-examination.

Table 10. Results of Simulated Electronic Vote Counting with Classification Manipulation at a Fixed Frequency

500	Before Classification		Manipulated Classification Result			Ballot Switching Count
Bundle	Cand. 1	Cand. 2	Cand. 1	Cand. 2	Unclas-sified	
#1	48	52	47	48	5	4
#2	45	55	48	47	5	4
#3	49	51	50	46	4	4
#4	55	45	58	40	2	4
#5	43	57	44	48	8	4
Total	240	260	247	229	24	20
Vote Share	48%	52%	49.4%	45.8%	4.8%	
Relative Vote Share			51.9%	48.1%		

Excluding the re-examination ballots, the relative vote shares for each candidate were recalculated as follows:

$$247 \div (247 + 229) = 247 \div 476 = 0.519 = 51.9\,\%$$
$$229 \div (247 + 229) = 229 \div 476 = 0.481 = 48.1\,\%$$

These figures confirm that the intended manipulation objective was achieved.

The electronic vote counting machine by design, is expected to function autonomously, primarily scanning and classifying ballots without the need for external communication. This inherent characteristic underpins the feasibility of implementing vote manipulation using only internal instructions specifying the number of ballots to be reclassified between candidates. This feasibility was substantiated through simulation employing both the classification manipulation method and a corresponding software implementation.

Nonetheless, as addressed in Section 3, when the actual vote distribution significantly diverges from the projected distribution on which the manipulation is based, the manipulation may not yield the desired outcome. In contrast to ballot box tampering, which is typically static, manipulation via electronic vote counting can be executed dynamically during the counting process. This dynamic capability introduces the potential for adaptive strategies that respond to discrepancies in real time. The subsequent section explores a manipulation method that ensures the achievement of manipulation objectives even when actual voting outcomes deviate from expectations.

5-4. Target-Guaranteed Vote Counting Machine Manipulation

If the vote counts for each candidate are continuously accumulated in real-time during the vote counting process, it becomes possible to monitor their respective vote shares at any point. Based on this capability, the condition for manipulation—*"Is manipulation necessary?"*—can be defined as:

"Is the current cumulative vote share of the beneficiary candidate below the manipulation target vote share?"

Whenever a ballot marked for the victim candidate is recognized, this condition is checked using the current cumulative vote share of the beneficiary candidate. If the candidate's vote share falls short of the target, the ballot is reclassified to the beneficiary candidate. In this way, manipulation is carried out only as needed to ensure that the cumulative vote share of the beneficiary candidate consistently meets or exceeds the target. This method guarantees the success of manipulation and is referred to hereafter as ***target-guaranteed vote manipulation***.

In this method, the only information required for the electronic vote counting machine is the manipulating candidate's target vote share. There is no need to pre-estimate the scale of manipulation, nor are projections of turnout or expected vote shares necessary.

Let us examine this in more detail. Before counting begins, a target vote share for the beneficiary candidate is set. As each ballot is processed, the result is added to the cumulative count stored within the machine. This allows the device to access the current vote share at any time. When a ballot marked for the victim candidate is detected, the machine checks whether the beneficiary candidate's current vote share falls below the target. If it does, the ballot is misclassified.

To help illustrate this logic, a simulated electronic vote counting program was created. The original code for normal classification remains unchanged, while additional code implementing the manipulation logic is inserted as follows:

```
 1 # Target-Guaranteed Vote Manipulation Simulation Program
 2 #                   for Electronic Counting Machines
 3 # Inputs: box = storage for scanned ballot images
 4 #       count = record of candidate vote counts
 5 #     ballots = bundle of ballots to be processed
 6 def classify_ballots(box, count, ballots):
 7     global prefix_sum  # stores cumulative vote counts for each candidate
 8     for b in ballots:
 9         image = scan(b)
10         detected = detect(image)
11         if detected == VICTIM:    # Is the ballot for the victim candidate?
12             if percentage_of_vote() < TARGET: # Does not reach the target?
13                 detected = BENEFICIARY          # Swap manipulation
14         box[detected].append(image)
15         count[detected] += 1
16         prefix_sum[detected] += 1 # Increment detected candidate's count
17     return box, count
```

The key information needed for this manipulation is: the **beneficiary candidate**, the **victim candidate**, and the **target vote share**. These correspond to the variables `BENEFICIARY`, `VICTIM`, and `TARGET` in the program. Lines 7 and 16 are additions for accumulating vote counts during classification, and lines 11–13 implement the manipulation logic. When a ballot is detected for the victim, the program calculates the beneficiary candidate’s current vote share using the cumulative data and decides whether to manipulate based on the result. To revert to normal (non-manipulated) mode, simply set the target vote share to 0%. This causes the condition in line 12 to always evaluate as false, and no manipulation is triggered.

This *target-guaranteed* method ensures successful manipulation even if the actual vote distribution differs from expectations. The following simulation demonstrates this, using a scenario where pre-defined manipulation (at fixed intervals) would fail—i.e., the manipulating candidate (Candidate 1) has 45% of the ballots, and the victim (Candidate 2) has 55%. If you examine the classification results carefully, you can see that ballots marked for Candidate 2 were reclassified as Candidate 1's at irregular intervals—only when necessary to achieve the manipulation target.

```
* Ballot bundle #1 *
Number of ballots: 100
12122212211221221121111112222211122112121122212212
11122111222111212212222111211221222112121211212212
1 = 49 2 = 51
-----
Classification results:
   Candidate 1 = 49
   Candidate 2 = 45
   Unclassified = 6
---
Classified into bin 1:
1121211111111111111111111111111111112111111111112
---
Classified into bin 2:
222222222222222222222222222222222222222222222
---
Unclassified ballots:
121211
---

* Ballot bundle #2 *
Number of ballots: 100
22122222112122122212121111222221221222112111122122
12122121211221211112212221112112212121212212222221
1 = 43 2 = 57
-----
Classification results:
   Candidate 1 = 50
   Candidate 2 = 46
   Unclassified = 4
---
Classified into bin 1:
21221111211111112121111111111111111111111111121221
---
Classified into bin 2:
2222222222222222222222222222222222222222222222
---
Unclassified ballots:
2112
---

* Ballot bundle #3 *
Number of ballots: 100
22222111221121212122121221112211221222112212112211
12222112222121221122122221211112121221122122111221
1 = 45 2 = 55
-----
Classification results:
   Candidate 1 = 52
   Candidate 2 = 46
   Unclassified = 2
---
```

```
Classified into bin 1:
22111111111121111111111111112112112111211111111111
11
---
Classified into bin 2:
2222222222222222222222222222222222222222222222
---
Unclassified ballots:
22
---

* Ballot bundle #4 *
Number of ballots: 100
12222121211111112222211212221121211112221122221112
11212212221122222111122222211222112222221112221221
1 = 44 2 = 56
-----
Classification results:
   Candidate 1 = 50
   Candidate 2 = 47
   Unclassified = 3
---
Classified into bin 1:
11111111111111111111111111111122111121121122111121
---
Classified into bin 2:
22222222222222222222222222222222222222222222222
---
Unclassified ballots:
212
-

* Ballot bundle #5 *
Number of ballots: 100
21222111222121111121111212112221121121212212122112
11212222122222112222122122112212222122212221212221
1 = 44 2 = 56
-----
Classification results:
   Candidate 1 = 50
   Candidate 2 = 46
   Unclassified = 4
---
Classified into bin 1:
11111111111111111111111111111121112121112121211221
---
Classified into bin 2:
2222222222222222222222222222222222222222222222
---
Unclassified ballots:
1122
---
```

Let us now examine Table 11. A total of 500 ballots were used, comprising 225 for the beneficiary candidate (Candidate 1, 45%) and 275 for the victim (Candidate 2, 55%), shuffled randomly and counted in batches of 100. As shown in the simulation, even with these unfavorable initial conditions, the manipulation successfully met the target vote share of 52% for Candidate 1—when excluding the ballots marked for rechecking.

Table 11. Simulation Results of Target-Guaranteed Vote Manipulation via Electronic Counting Machines

500	Before Classification		After Manipulated Classification			Ballot Switch Count
Bundle	Cand. 1	Cand. 2	Cand. 1	Cand. 2	Pending Review	
#1	49	51	49	45	6	4
#2	43	57	50	46	4	9
#3	45	55	52	46	2	7
#4	44	56	50	47	3	7
#5	44	56	50	46	4	8
Total	225	275	251	230	19	35
Vote Share	45%	55%	50.2%	46.0%	3.8%	
Relative Vote Share			52.2%	47.8%		

In conclusion, compared to the fixed-frequency manipulation method that predetermines how many ballots to swap, *target-guaranteed vote manipulation* proves to be significantly more effective. It guarantees success regardless of the actual distribution of votes, making it a far superior approach in terms of reliably achieving manipulation goals.

5-5. Summary

Electronic vote counting machines must operate independently, as they only need to scan and classify ballots. Considering this characteristic, the logic for manipulating the count using only minimal necessary information within the machine's internal system was examined.

But what about in a single electoral district that uses multiple electronic vote counting machines? Would they be able to manipulate the results without communication among each other? Since the goal of manipulating the count can be guaranteed for each machine individually, it is inherently possible to achieve the overall district-level manipulation goal even if multiple machines are used separately. However, if the vote support shares of the ballots fed into each machine are significantly opposite, there is a risk that the manipulation could overshoot the target. While multi-machine use doesn't threaten the achievement of the target, it might cause the result to exceed the intended support share. In such cases, a solution is to connect all participating machines to a central system that manages counting results in real-time, and have each machine reference this data during their operation.

This would mean each machine is no longer operating as a completely independent tool but as part of a connected, integrated system, increasing the complexity of management considerably.

Introducing electronic vote counters makes covert classification and swap manipulations relatively easier, provided that tampering is not detected by on-site observers. However, even with extremely fast classification speeds—so fast that manual verification becomes very challenging—there remains a risk that some misclassified ballots could be noticed. Furthermore, observers or election officials cannot be certain that manipulated ballots are not hidden within the classified stacks. For example, in simulation cases, manipulated ballots might be classified as legitimate and placed at the outermost layer of a bundle, and if they end up on the outside, the tampering could be exposed.

Therefore, efforts should be made to minimize the number of manipulation attempts. Despite this, electronic manipulation has a fundamental flaw: when active observation by responsible witnesses or officials is taking place, the success probability remains very low.

Although election results can be manipulated as desired in theory, achieving the conditions of *Complete Manipulation* and *Trace Removal* requires post-counting ballot box manipulation. This might involve opening the ballot box, swapping in fake ballots that look genuine, and resealing it. Alternatively, a separate pre-filled ballot box, containing ballots matching the manipulated results, can be prepared and used to replace the original entirely. The choice depends on the relative difficulty and complexity of covert operation in each case, which should be weighed according to the specific situation.

6. Double Manipulation

Early ballot box manipulation offers the advantage of allowing careful, pre-planned tampering. However, a critical flaw is that the natural contradiction between the results of pre-vote and same-day voting—an unavoidable phenomenon—inevitably leaves traces of manipulation. Moreover, if the actual support share deviates too significantly from estimates, the manipulation may fail. Real-time manipulation during electronic counting, on the other hand, guarantees success in theory, but as the scale of manipulation increases, the likelihood of detection by observers also rises, making covert operation more challenging.

In this section, two methods that combine both pre-vote ballot box manipulation and real-time electronic counting manipulation are examined. These combined approaches aim to compensate for the inherent shortcomings of each method. The objectives of the double manipulation are as follows:

- Goal 1: Avoid leaving traces of contradiction between pre-vote and same-day voting results.
- Goal 2: Overcome deviations of actual support shares from initial estimates.
- Goal 3: Minimize the number of classification adjustments during the real-time electronic count.

Let's analyze whether the two proposed double manipulation methods successfully achieve each of these goals, step by step.

6-1. Overcoming the Discrepancy Between Actual and Estimated Support Rates in Double Manipulation

In ballot box manipulation, the scale of tampering is determined based on predicted turnout and survey support shares. However, if the actual support share deviates significantly from estimates, the manipulation can fail. When the support share differs from the prediction, and the pre-vote ballot box manipulation does not reach the target, the question becomes: how can this be corrected using real-time electronic counting?

The main idea is to compensate for the shortfall left by pre-vote manipulation through adjustments during the electronic count. However, applying the same target support share uniformly across all ballot boxes during counting can cause issues. Due to the stark contrast between results from pre-vote and election-day ballot boxes—stemming from the phenomenon where one aligns with the target and the other overshoots—using a single support target leads to over-manipulation in some cases.

To address this, the desired support shares for pre-vote and election-day ballot boxes must be set separately. Manipulation should then be performed accordingly. This approach presupposes that the distinction between pre-vote and election-day ballots is identifiable.

Procedure for Preparing and Performing Double Manipulation

Preparation:

- **Step 1.** Pre-vote ballot box manipulation: Decide on the manipulation method (e.g., add-vote), select the beneficiary candidate (e.g., Candidate 1), set the target vote share T, and calculate the necessary manipulation scale K.
- **Step 2.** For the real-time count manipulation:
  - Compute the pre-vote target vote share T_{pre} for the pre-vote ballot box:

  $T_{pre} =$ (Estimated beneficiary votes + K) $\div$ (Estimated total pre-votes + K)
  - Set the election-day target vote share T_{day} to match the estimated support share of the beneficiary (e.g., 48%).

Execution:

- **First Manipulation :** Perform the pre-vote ballot box manipulation with scale K.
- **Second Manipulation** : Perform the real-time electronic count manipulation,
  - Set the pre-vote target vote share to T_{pre}.
  - Set the election-day target vote share to T_{day}.

  This approach allows the adjustment scales in each box to compensate for initial prediction errors, ensuring that the overall target vote share is achieved.

Simulation of Double Manipulation

Using the same hypothetical case with turnout 20% pre-vote, 40% election-day, and support shares of 48% and 52%, the initial state of the ballot boxes is:

Ballot Box	Candidate 1		Candidate 2		Voter	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Pre-vote	960	48%	1,040	52%	2,000	20%
Election-day	1,920	48%	2,080	52%	4,000	40%
Total	2,880	48%	3,120	52%	6,000	60%

Preparation:

- **Step 1.** For the pre-vote ballot box manipulation: Decide on add-vote manipulation, select Candidate 1 as the beneficiary candidate, set the target vote share T to 52%. Then determine K based on the predicted votes and T:

```
>>> rig_the_vote(2880, 3120, 0.52, add)
500
```

- **Step 2.** For the real-time count manipulation:
  - Compute the pre-vote target share T_{pre} for the pre-vote ballot box as follows:

T_{pre} = (Estimated beneficiary votes + K) ÷ (Estimated total pre-votes + K)

$$= \frac{960 + 500}{2{,}000 + 500} = \frac{1{,}460}{2{,}500} = 58.4\,\%$$

  - The same-day ballot box target share T_{day} is 48 % , the estimated support share of the beneficiary.

Now, the preparation for double manipulation is complete. At this point, let's conduct two simulation examples: one where the election results turn out as expected, and another where they do not, to test the double manipulation process.

Execution: Case 1 — When Actual Ballot Boxes Match Expectations

- **First Manipulation** : The pre-vote ballot box manipulation

  Adding 500 ghost ballots into the pre-vote ballot box transforms it as shown below. In fact, manipulation is already successful at this stage because the actual vote share matches the estimate.

Ghost Ballots 500	Candidate 1		Candidate 2		Voter	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Pre-vote	1,460	58.4%	1,040	41.6%	2,500	25%
Election-day	1,920	48%	2,080	52%	4,000	40%
Total	3,380	52%	3,120	48%	6,500	65%

- **Second Manipulation** : Real-time Count Manipulation

  Simulating the scenario where the pre-vote ballot box is manipulated to 58.4 % and the election-day ballot box to 48 % resulted in the following outcomes:

Ballot Box	Candidate 1		Candidate 2		Pending Review		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Pre-vote	1,403	45.0%	978	55.0%	119	4.8%	2,500	25%
Election-day	1,834	45.9%	1,973	49.3%	193	4.8%	4,000	40%
Total	3,237	49.8%	2,951	45.4%	312	4.8%	6,500	65%
	Relative Vote Share							
Pre-vote		58.9%		41.1%				
Election-day		48.2%		51.8%				
Total		52.3%		47.7%				

Since the goal was already achieved in the first manipulation, further second manipulation will rarely trigger classification adjustments.

Execution: Case 2 — When Actual Ballot Boxes Deviate From Expectations

Suppose the actual vote share differs more significantly than expected, with each candidate's share at 45% and 55%, respectively, instead of the predicted 48% and

Ballot Box	Candidate 1		Candidate 2		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Pre-vote	900	45%	1,100	55%	2,000	20%
Election-day	1,800	45%	2,200	55%	4,000	40%
Total	2,700	45%	3,300	55%	6,000	60%

52%. The state is:

- **First Manipulation** : The pre-vote ballot box manipulation
  Adding 500 ghost ballots into the pre-vote box results in:

Ghost Ballots 500	Candidate 1		Candidate 2		Voter	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Pre-vote	1,400	56%	1,100	44%	2,500	25%
Election-day	1,800	45%	2,200	55%	4,000	40%
Total	3,200	49.2%	3,300	50.8%	6,500	65%

Even after the first manipulation, the ranking remains unchanged, so the target isn't achieved.

- **Second Manipulation** : Real-time Count Manipulation

  Repeating the scenario from the previous example—setting the pre-vote count

Ballot Box	Candidate 1		Candidate 2		Pending Review		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Pre-vote	1,388	55.5%	1,001	40%	111	4.4%	2,500	25%
Election-day	1,823	45.6%	1,970	49.3%	207	5.2%	4,000	40%
Total	3,211	49.4%	2,971	45.7%	318	4.9%	6,500	65%
	Relative Vote Share							
Pre-vote		58.1%		41.9%				
Election-day		48.1%		51.9%				
Total		51.9%		48.1%				

to 58.4% and the election-day count to 48%—the results are:

In the second manipulation, the manipulation targets that were not achieved in the first round are attained by separately adjusting the counts of the pre-vote and election-day ballot boxes in real time. As a result, the overall manipulation target of a 52% vote share is achieved via double manipulation.

While it has been confirmed that larger discrepancies in the actual support shares between the two candidates can be overcome, the phenomenon where the results of the pre-vote and same-day votes oppose each other remains unresolved. Since classification of manipulation is only performed when absolutely necessary during count manipulation, Goal 3 can be considered achieved. Next, let's examine the double manipulation method that addresses Goal 1.

6-2. Double Manipulation to Overcome Discrepancies Between Pre-Vote and Election-Day Support Shares

Manipulating only the pre-vote ballot box leaves an undeniable trace of tampering, especially when the support shares from pre-vote and election-day votes contradict each other. This phenomenon can be addressed through double manipulation, combining both pre-vote ballot box tampering and real-time count manipulation. The key difference is that the manipulations of the pre-vote and election-day counts are carried out separately and independently.

Procedure for Preparing and Executing Double Manipulation

Preparation:

- **Step 1** : Pre-vote ballot box manipulation: Decide on the manipulation method (e.g., add-vote), select the beneficiary candidate (e.g., Candidate 1), set the target vote share T, and calculate the necessary manipulation scale K.
  Note that in this process, only the predicted total pre-vote count is used for calculation, not the overall total votes.
- **Step 2** : For the real-time count manipulation: Set the target vote share for both the pre-vote and election-day ballot to be the same T.

Execution:

- **First Manipulation :** Perform the pre-vote ballot box manipulation with scale K.
- **Second Manipulation** : During the real-time electronic count, manipulate the pre-vote and election-day ballot boxes separately to the same target vote share T.

Simulation of Double Manipulation

Using the same hypothetical scenario—expected early vote rate of 20%, election day vote rate of 40%, and support shares of 48% vs. 52%—the initial states of the ballot boxes are:

Ballot Box	Candidate 1		Candidate 2		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Pre-vote	960	48%	1,040	52%	2,000	20%
Election-day	1,920	48%	2,080	52%	4,000	40%
Total	2,880	48%	3,120	52%	6,000	60%

Preparation:

- **Step 1.** For pre-vote ballot box manipulation: Decide on add-vote manipulation, select Candidate 1 as the beneficiary candidate, set the target vote share T to 52%. Then determine K based on the predicted pre-votes and T:

```
>>> rig_the_vote(960, 1040, 0.52, add)
167
```

- **Step 2.** For the real-time count manipulation: Set the target vote share for both the pre-vote and same-day ballot box to be 52%.

Now, the preparation for double manipulation is complete. Again, let's conduct the same simulation examples as before: one where the election results turn out as expected, and another where they do not, to see this approach overcome the weakness witnessed in the previous double manipulation attempt.

Execution: Case 1 — When Actual Ballot Boxes Match Expectations

- **First Manipulation** : The pre-vote ballot box manipulation

  Adding 167 ghost ballots into the pre-vote ballot box transforms it as shown below. Note that only the pre-vote ballot box has been achieved the target vote share of 52% at this stage.

Ghost Ballots 167	Candidate 1		Candidate 2		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Pre-vote	1,127	52%	1,040	48%	2,167	21.7%
Election-day	1,920	48%	2,080	52%	4,000	40%
Total	3,047	49.4%	3,120	50.6%	6,167	61.7%

- **Second Manipulation** : Real-time Count Manipulation
  Separate count manipulations are performed on the pre-vote and election-day ballot boxes, each with the same target vote share of 52%. The results of separate simulated experiments for each are shown in the table below.

Ballot Box	Candidate 1		Candidate 2		Pending Review		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Pre-vote	1,091	50.3%	978	45.1%	98	4.5%	2,167	21.7%
Election-day	1,994	49.9%	1,819	45.5%	187	4.7%	4,000	40%
Total	3,085	50.0%	2,797	45.4%	285	4.6%	6,167	61.7%
	Relative Vote Share							
Pre-vote		52.7%		47.3%				
Election-day		52.3%		47.7%				
Total		52.4%		47.6%				

All three: the pre-vote ballot box, the same-day ballot box, and the total vote tally box, individually achieve the manipulation target vote share T of 52%. However, this time, "*the support rates of pre-vote and election-day votes do not contradict each other!*"

Execution: Case 2 — When Actual Ballot Boxes Deviate From Expectations

Let's conduct a simulation example where the support share discrepancy becomes even larger than initially expected, as in the previous section.

Ballot Box	Candidate 1		Candidate 2		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Pre-vote	900	45%	1,100	55%	2,000	20%
Election-day	1,800	45%	2,200	55%	4,000	40%
Total	2,700	45%	3,300	55%	6,000	60%

- **First Manipulation** : The pre-vote ballot box manipulation
  Inserting 167 ghost ballots into the pre-vote ballot box transforms it as follows.

Ghost Ballots 167	Candidate 1		Candidate 2		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Pre-vote	1,067	49.2%	1,100	50.8%	2,167	21.7%
Election-day	1,800	45%	2,200	55%	4,000	40%
Total	2,867	46.5%	3,300	53.5%	6,167	61.7%

  Because the actual voting results differ from the predictions, the manipulation fails to achieve the target vote share of 52% in the pre-vote ballot box manipulation.

- **Second Manipulation** : Real-time Count Manipulation
  Separate count manipulations are performed on the pre-vote and election-day ballot boxes, each with the same target vote share of 52%. The results of separate simulated experiments for each are shown in the table below.

Ballot Box	Candidate 1		Candidate 2		Pending Review		Voters	10,000
	Vote Count	Vote Share	Vote Count	Vote Share	Vote Count	Vote Share	Vote Total	Turnout
Pre-vote	1,085	50.1%	988	45.6%	94	4.3%	2,167	21.7%
Election-day	1,988	49.7%	1,829	45.7%	183	4.6%	4,000	40%
Total	3,073	49.8%	2,817	45.7%	277	4.5%	6,167	61.7%
	상대 득표율							
Pre-vote		52.3%		47.7%				
Election-day		52.1%		47.9%				
Total		52.2%		47.8%				

Despite the actual support rates deviating significantly from the predictions, the manipulation successfully reaches the target support share of 52%, and importantly, there is no contradiction between the pre-vote and election-day results. "*Killing two birds with one stone!*"

This double manipulation method simultaneously achieves Goals 1 and 2 perfectly. However, its performance regarding Goal 3 has declined compared to another double manipulation approach previously discussed, as it requires more classification adjustments. Since only the shortfall amount is manipulated in the pre-vote ballot box, the total manipulation during the count remains minimal or nearly zero, so it's not problematic. Conversely, during the election-day count, the entire manipulation relies on count adjustments, which increases the manipulation burden. This is a side effect of attempting to achieve Goal 1, resulting in a trade-off that sacrifices Goal 3.

6-3. Summary

This approach demonstrates that both the failure of manipulation due to support rate prediction errors and the visible traces of manipulation—such as support share contradictions between pre-vote and election-day—can be effectively addressed via double manipulation. If we accept that the complete manipulation can be achieved through ballot box post-manipulation after the count, then the remaining critical challenge is covert execution. If covert operation can be executed perfectly, then the election manipulation could be completely concealed in theory.

7. Final Remarks

In democracies where voters attend polling stations and mark their choices on ballots, this report has explored how the voting and counting systems and procedures can be modified to facilitate potential election manipulation. Various methods have been illustrated through a hypothetical example.

Summary

- Election manipulation involves artificially altering vote counts to change the outcome of an election.

- Successful manipulation must satisfy three conditions: (1) *Secrecy in Execution*—undetectability by voters or observers; (2) *Trace Removal*—no evidence of tampering remains; and (3) *Complete Manipulation*—same results are obtained even after recounts. All three conditions must be met.

- The main methods are add-vote, swap-vote, and discard-vote. Each has its strengths and weaknesses. For each, there are a systematic ways to calculate the minimum number of forged ballots needed to reach the target support share.

- The number of forged ballots can only be estimated based on predicted, not actual, votes. Therefore, more accurate support share predictions (e.g., via polls) increase the likelihood of successful manipulation.

- To hide discrepancies between public opinion polls and actual election results, fake polls conducted close to the election can mask traces of manipulation.

- Election manipulation can occur either before tallying (ballot box manipulation) or during counting process (real-time adjustments), categorized as pre-vote and during-count manipulation.

- The most feasible method for significant pre-count manipulation is to introduce early voting and manipulate early ballot boxes. However, this typically leaves a conspicuous discrepancy between early votes and election-day votes—a clear trace of tampering.

- Manipulating only the early ballot box with add-vote methods is relatively easier covertly. The challenge is to pretend non-voters have voted—this can be addressed by installing online ballot issuance systems managed centrally, which can produce ghost ballots without leaving on-site traces.

- Effective classification of ballots enables swap-vote manipulation. Automating this process with electronic vote counting machines allows frequent, controlled, and adaptable manipulations that can be executed discreetly.

- Electronic counting machines increase the success likelihood of manipulation if undetected; however, as the scale of manipulation grows, so does the risk of detection. Therefore, minimizing the scope of manipulation is crucial.

- Inaccurate predictions of support shares and contradictions between early and election-day votes leave detectable traces. A double manipulation approach—combining pre-vote and during-count methods—can address these issues inherently. If the conditions of *Secrecy in Execution* and *Trace Removal* are maintained properly, near-perfect election manipulation, with virtually no traces, could be theoretically achieved.

Conclusion

Among the three necessary conditions for successful election rigging, *Trace Remova*l and *Complete Manipulation* can generally be achieved through well-equipped voting systems and operational procedures. The remaining challenge lies in maintaining strict *Secrecy in Execution*, which depends on trustworthy governance and management.

The only way to truly guarantee fairness and transparency is through an ***immediate, on-site count after vote closure***, allowing verification by both candidates and voters alike. Countries like Taiwan and many European democracies adhere to their traditional systems for this reason, valuing their reliability. Through this discussion, I hope our fellow citizens can recognize this fundamental truth as well.