1 Proposed Framework

1.1 Base RL-based Policy Support System

- **Agent:** The agent represents the policymaker. Our goal is to train it to be an intelligent policymaker with the ability to select the optimal NPIs policy under different environmental states.
- **Environment:** The SIR model for epidemic simulation.
- **Observation:** We set 1 week as the interval between each decision point. In order to match the reality, we assume that the virus parameters cannot be observed by the agent. The observation contains 6 features, including the number of the population of the S, I, and R, the increasing number of the I and R since the last decision point, and the last selected action.
- **Action:** We use control effects instead of specific NPIs as the action directly, set the upper limit to 0.9, the lower limit to 0, and give a graduation value of 0.1, which means we have 10 actions in total. With the strictest action, the parameter β will be reduced by 90%.
- **Reward:** The reward function is the key to guiding the direction of RL. In this study, we subdivide the reward into two sub-rewards. The first one represents the effect of public health and is reflected in the number of weekly new cases. The second one represents the action cost, which will get a high value under strict action. The total reward is completed by *Eq. S1*. And we set $w_1 = 10, w_2 = 1, and\ I_{st} = 10$.

$$\begin{aligned} \text{Reward} &= w_1 \cdot R_1 + w_2 \cdot R_2 \\ R_1 &= Sigmoid\left(\frac{I_{st} - I_{new}(t)}{I_{st}}\right) \\ R_2 &= -u(t) \end{aligned} \tag{S1}$$

- **Discount Factor**: The discount factor $\gamma' \in [0,1]$, defines how valuable short-term rewards are compared to the long-term reward. In this case, we set this parameter as 0.99.

According to the Bellman equation, the Q-function $Q^{\pi}: s \times a \rightarrow r$ defines the expected future discounted reward for taking action a in state s and then following policy π thereafter, as shown in *Eq. S2* [1].

$$\mathrm{Q}_{t+1}(\mathrm{s}_t, \mathrm{a}_t) \leftarrow \mathrm{Q}_t(\mathrm{s}_t, \mathrm{a}_t) + \alpha_t \left[\mathrm{r}_{t+1} + \gamma \max_{\mathrm{a}'} \mathrm{Q}_t(\mathrm{s}_{t+1}, \mathrm{a}') - \mathrm{Q}_t(\mathrm{s}_t, \mathrm{a}_t)\right] \tag{S2}$$

Given the infinite state space in our case, we select the deep Q-network (DQN) algorithm. DQN is a value-based RL method and uses an artificial neuron network to optimize the action-value function so that similar states can get similar output actions [2, 3].

2 Experiments

2.1 SIR Model and Baseline RL Model

Our study seeks to test the designed RL-based policy support system using data from the Chinese Mainland in the context of the seasonal influenza epidemic.

In particular, we utilize the SIR model to simulate the progression of the influenza epidemic, categorizing individuals into susceptible (S), infected/infectious (I), and removed (R) groups [4-6]. Note that, we integrated the recovered and death cases into the removed category, assuming that individuals who have recovered from the virus have gained immunity and are not susceptible to future infection. *Fig. S1(a)* illustrates the inter-group relations, with β and γ denoting the exposure rate, infection rate, and removal rate, respectively. The efficacy of NPIs on disease control is represented by the symbol u. Moreover, similar to previous studies using the SIR model [5, 7], random disturbances have been incorporated into the simulation to introduce the complexity of real-world scenarios. In *Fig. S1(a) – Eq. 1*, $P_{random}^{S\to I}$ represents a 10% probability of increasing or decreasing by a maximum of 5 infectious cases between two sequential statuses, S and I, to simulate accidental infections. Using random numbers $m_1 \sim m_3$ with a uniform distribution, the hyper-parameters above have been adjusted (as per *Fig. S1(a) – Eq. 2*) to simulate fluctuations in transmission and variations in policy implementations.

Meanwhile, we establish the initial RL Model based on previous research [7, 8]. As shown in *Fig. S1(b)*, we have presented a straightforward overview of our baseline RL model. The agent acts as a representative of the policymaker. In order to simulate the environment of the influenza epidemic, we implemented the SIR model. We utilized the efficacy of control as the action, restricted by upper and lower bounds of 0.9 and 0, respectively, with a step size of 0.1, resulting in 10 total actions. Apparently, the strictest action results in a reduction of the parameter β by 90%. Observations occur at weekly intervals with 6 attributes, including population numbers within the susceptible (S), infected/infectious (I), and removed (R) populations, the increase in the infected/infectious (I) and removed (R) populations since the last decision round, and the most recent action selected. In terms of reward, the total reward comprises two sub-rewards (as per *Fig S1(a) – Eq. 3*, $w_1 = 10, w_2 = 1, and\ I_{st} = 5$, which means the acceptable number of weekly new infected cases): the status of public health reflected by the sigmoid function of the standardized number of new patients exceeding expectations (R_1) and the action cost that receives a high value under strict action (R_2). The discount factor, $\gamma' \in [0,1]$, delineates the relative importance of short-term rewards in comparison to long-term rewards. In our study, we set the value of this parameter to 0.99, and this value has been widely used in prior studies [7, 9].

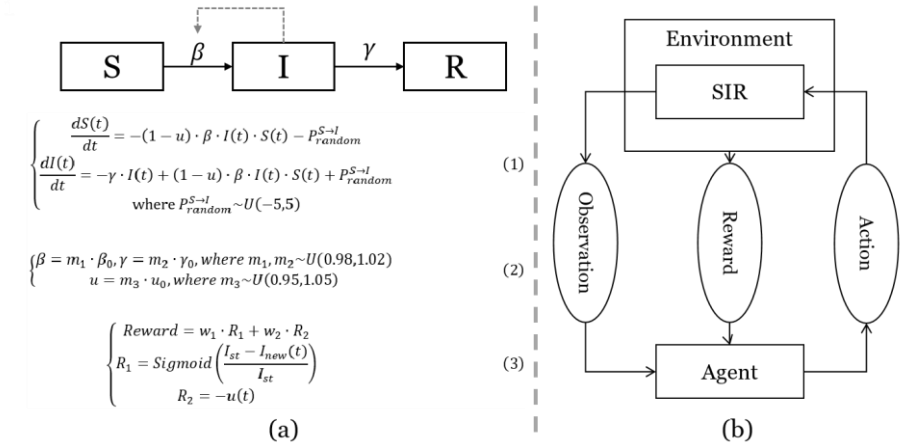

Fig. S1. SIR Model and Baseline RL Model

Due to the infinite state space involved in our research, we implemented the deep Q-network (DQN) algorithm, a value-based reinforcement learning method, that leverages an artificial neural network to optimize the action-value function, ensuring that comparable states result in similar action outputs [2, 3]. We trained our baseline RL model with the epidemic data in both the northern provinces and southern provinces of China. The parameters are shown in *Table S1*. To simulate using the SIR model, we use the first-order Euler method to estimate the influenza parameters for infection rate and removal rate. It can be seen that during the phase of COVID-19, the effective reproduction number (R_e) of the disease is lower than in the other two phases. This result is consistent with previous studies, which have shown that the R_e of influenza is between 1.1 and 1.6, with a median of 1.28 [10].

Table S1. Estimated Parameters

Location	**Northern Provinces**			**Southern Provinces**		
Phase	**Pre-COVID**	**COVID-19**	**Post-COVID**	**Pre-COVID**	**COVID-19**	**Post-COVID**
β	1.235e-9	1.518e-9	1.352e-9	9.105e-10	1.032e-9	9.890e-10
γ	0.436	0.786	0.571	0.557	0.783	0.654
R_e	1.621	1.105	1.355	1.368	1.104	1.266
Duration (week)	1-52	53-104	105-200	1-52	53-104	105-200
Initial Population	S=572,260,881, I=4,438, and R=4,497, December 29, 2019 (the 52nd week)			S=837,496,381, I=4,324, and R=8,203, December 29, 2019 (the 52nd week)		

2.2 Component of Comparison and Interpretation and Results of the Base Model

The four policies we use as benchmarks are designed as follows.

- **Double thresholds policy (DTP)**: DTP refers to taking the strictest action when the population of I exceeds the first threshold $H_1 = 100$, taking the action $u = 0$ when the population of I is less than the second threshold $H_2 = 5$, and taking the moderate action $u = 0.5$ when the population between the 2 thresholds.
- **Complete intervention policy (CIP)**: CIP refers to taking the strictest action at each step, whose control effect is 0.9.
- **Random intervention policy (RIP)**: RIP refers to taking arbitrary action at each step.
- **Non-intervention policy (NIP)**: NIP refers to making the epidemic develop free and taking the action $u = 0$ at each step.

The test results for the northern provinces scenario can be found in the main text, and the results for the southern provinces scenario are shown in *Fig. S2*. The DQN policy controlled the simulated influenza epidemic of southern provinces and kept the number of infectious below 5. The proposed RL approach did not display optimal performance across all multi-rewards, it demonstrated significant advantages in terms of total rewards.

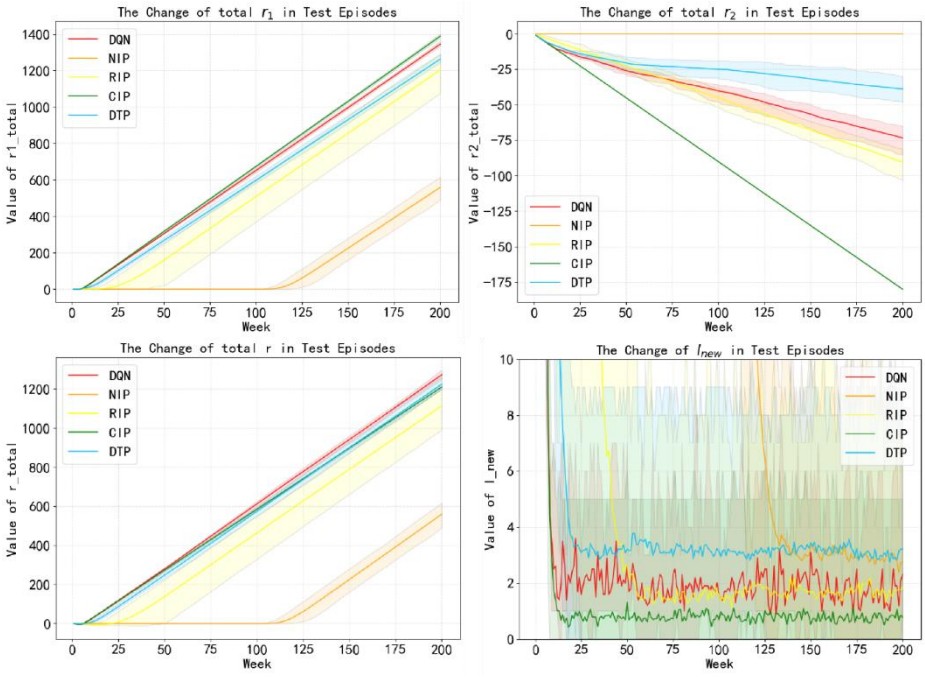

Fig. S2. Performance of Different NPIs Policies in Southern Provinces Scenario

As shown in *Fig. S3*, the DQN policy implemented more stringent action in the initial stages of the epidemic and gradually relaxed measures in the medium and long term. This is consistent with the scenario tested in the northern provinces in the main text. This indicates that the DQN policy developed by the proposed RL-based policy support system can be applied to different scenarios.

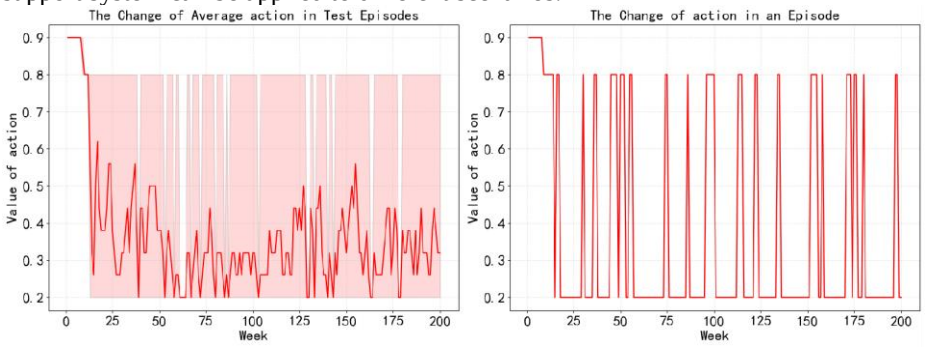

Fig. S3. The Actions of DQN in Southern Provinces Scenario

2.3 Component of Virus Mutation

As shown in *Fig. S4*, we build an internal SIR simulation component for Agent. We mark the internal model as $\mathrm{SIR}_{\mathrm{Agent}}$, and mark the external model as $\mathrm{SIR}_{\mathrm{Env}}$. While simulating, both two models can receive the same initial population of 3 categories (S, I, and R) and the same initial transmission parameters. When artificial mutation happens, $\mathrm{SIR}_{\mathrm{Env}}$ can get the changed parameters directly, while $\mathrm{SIR}_{\mathrm{Agent}}$ will estimate the new parameters with 10 steps' population data of the 3 categories. We provide these settings because virus mutations cannot be detected immediately, and medical testing of virus parameters requires a certain time cost [11, 12]. At the beginning of the epidemic, due to the unpredictability of the strains, the detection efficiency of pathogens could not be ensured [13]. Our settings can ensure the delay of obtaining disease parameters, which is consistent with the real-world situation.

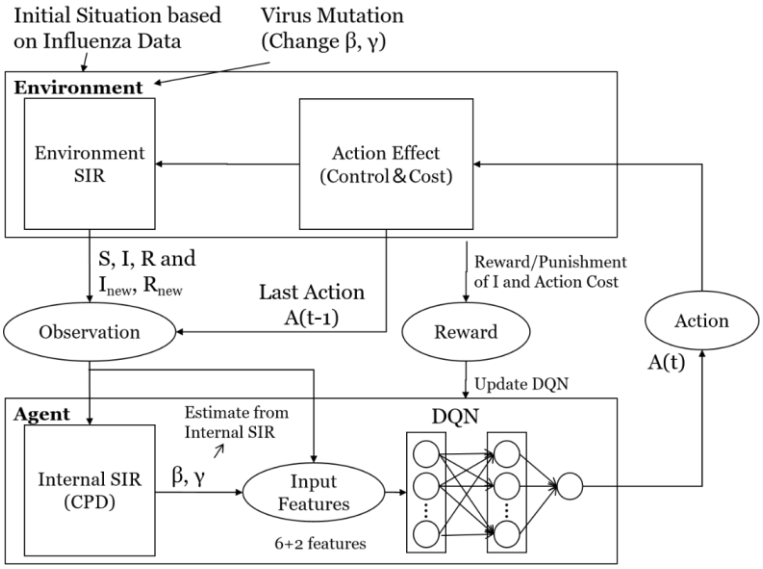

Fig. S4. CPD-augmented RL Framework

In order to determine the location of mutation points and predict the reproductive coefficient of mutant strains, we referred to the research of [14] and applied the random martingale process to achieve the CPD algorithm (Shown in *Algorithm S1*). We set 2 mutations during an episode. The parameters and the date of mutations are shown in *Table S1*.

Algorithm 1 Change Point Detection of Martingale Process

Set SIR_{env}, SIR_{agent} as environment and internal models
Define Estimate_parameters(t_0, t_1) as the method to estimate the β, γ from week t_0 to week t_1
Initial $n \leftarrow 1, T_{n-1} \leftarrow 0$
Initial λ as the threhold
Initial β, γ as the parameters of SIR_{agent}
while True **do**
  $\hat{i}_t \leftarrow SIR_{agent}$, and $i_t,\ S_t,\ I_t \leftarrow SIR_{env}$
  $Z_t \leftarrow \frac{|\hat{i}_t - i_t|}{\sqrt{S_t I_t}}, t \geq T_{n-1}$
  $p_t \leftarrow \frac{\sum_{u=T_{n-1}}^{t} \mathbf{1}_{z_u \geq z_t}}{t - T_{n-1}}$
  $M_t \leftarrow \prod_{u=T_{n-1}}^{t} \frac{s}{1-\exp(-s)} \exp(-s p_u)$
  if $M_t > \lambda$ **then**
    $n+ = 1$
    $T_{n-1} = t$
  end if
  if $t < T_{n-1} + 10$ **then**
    $\beta, \gamma \leftarrow$ Estimate_parameters (T_{n-1}, t)
  end if
  $t+ = 1$
end while

Algorithm S1. Change Point Detection of Martingale Process

Fig. S5 and *Fig. S6* show the performance of DQN and DQN-CPD in northern and southern provinces scenarios. The vertical dashed line represents the point at which the simulated influenza viral mutation occurred.

In *Fig. S5*, it can be seen that the DQN-CPD policy has been implementing the strictest action before the first viral mutation occurred (the 52nd week), which can accurately identify the presence of mutations and make action changes. Although this incurs a greater action cost, compared to the DQN policy, it can better decrease the number of infectious in the simulated epidemic.

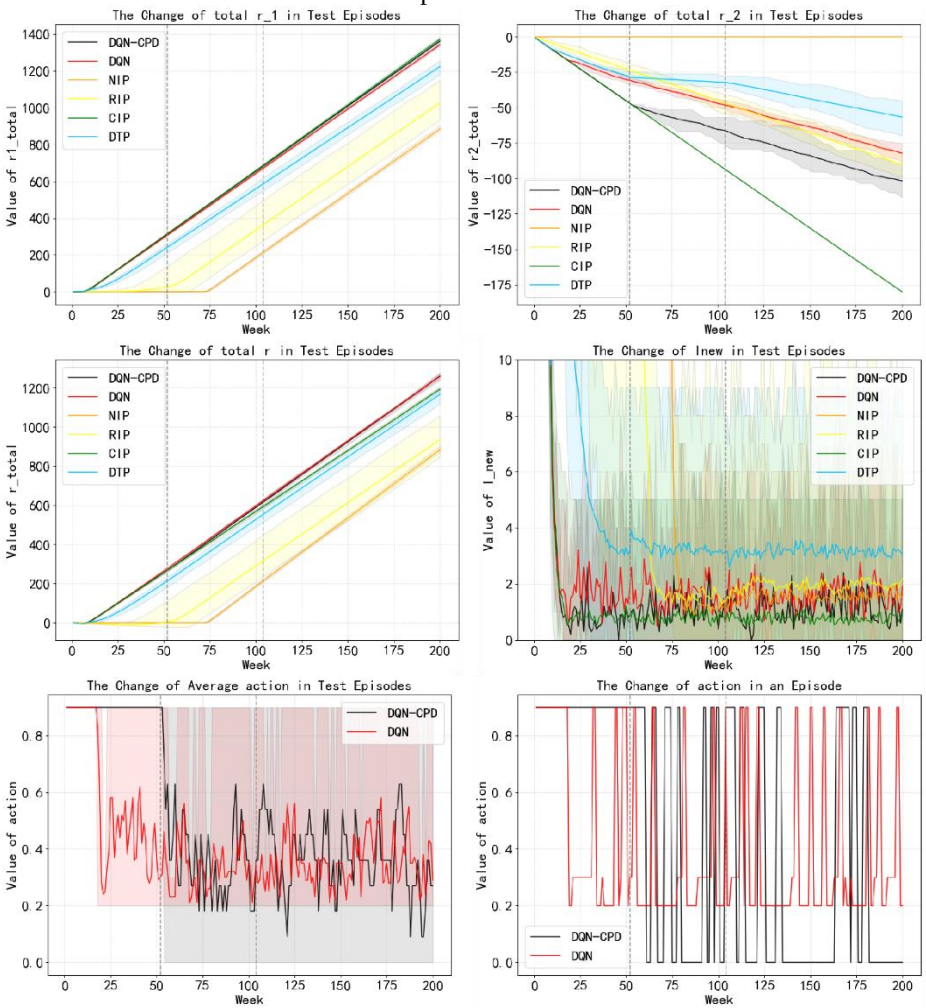

Fig. S5. Performance of DQN-CPD Policy in Northern Provinces Scenario

In *Fig. S6*, it can be seen that the DQN-CPD policy can still clearly identify viral mutations and take more appropriate and relaxed actions to achieve epidemic control effects consistent with the CPD policy.

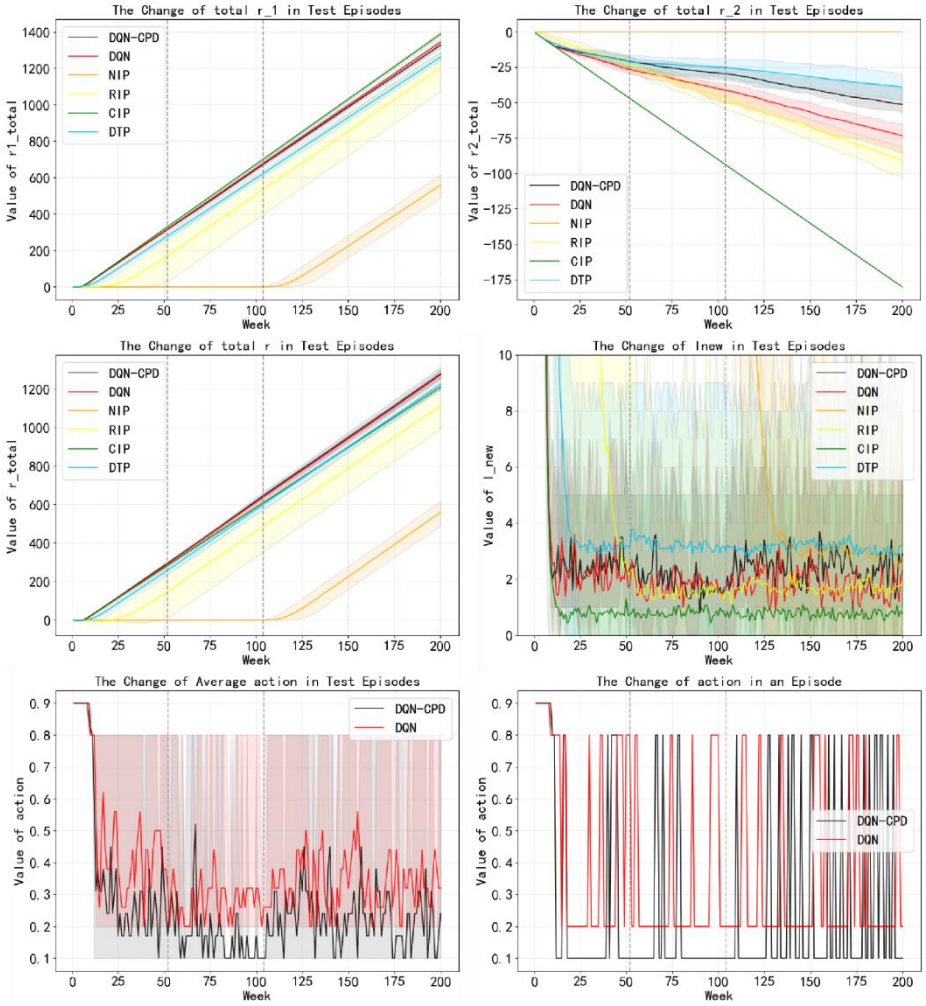

Fig. S6. Performance of DQN-CPD Policy in Southern Provinces Scenario

Besides, no matter in which scenarios, the DQN-CPD policy can achieve better total rewards, approximately 0.2% higher than the DQN policy. Through these results, we prove the significance of considering virus mutation, which may perform better in real situations with kinds of influenza virus.

This result shows that considering virus mutations will help the system perform better. It also confirms that it is necessary for us to consider the gap in the general system design and narrow the gap, taking it as one of the design perspectives.

2.4 Component of Multi-rewards

The values of r_2 we used are 1, 3, 5, 7, and 9. And when r_2 increases to 10 or above, the system will take $u = 0$ due to the large proportion of action costs. Therefore, the paper only shows the results when r_2 is greater than or equal to 9.

In *Fig. S7*, the black line represents the DQN-CPD, and the red line represents DQN, we can see that with the increase of w_2, the 2 systems obviously take significantly more relaxed actions, and instead, both 2 systems will take the action $u = 0$ when w_2 rise to 9 in the northern provinces scenario. In addition, the larger the w_2, the worse the performance of epidemic control, which is reflected in the control of weekly new infected.

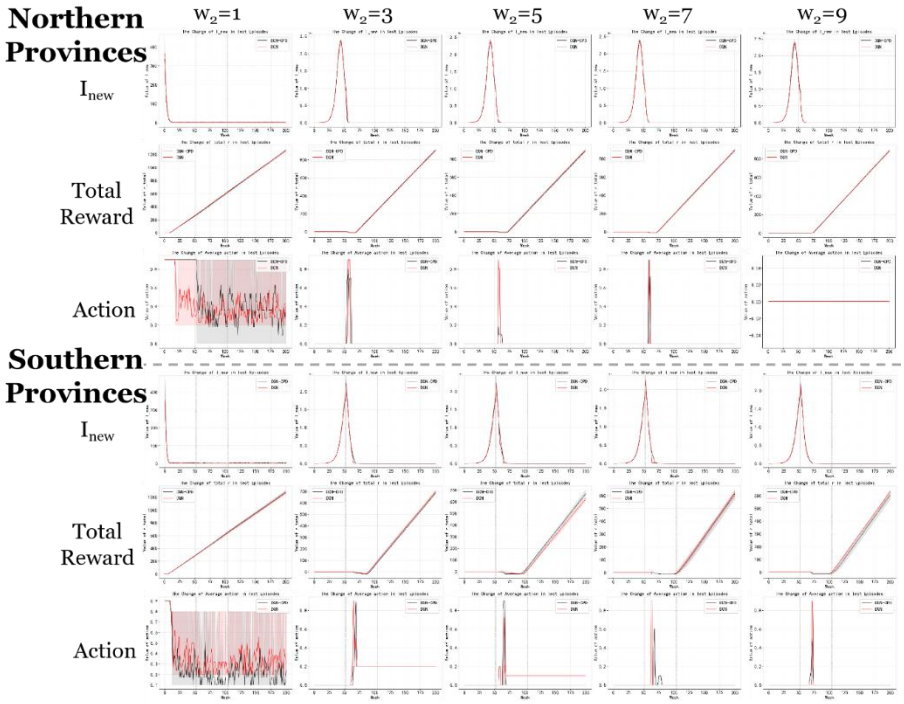

Fig. S7. Performance with Various Weights

2.5 Component of Guided Rewards

We summarize the previous research on RL-based policy support systems in epidemic control and conclude 2 kinds of AI immoral behaviors, threshold fluctuation (**TF**) and algorithm cheating (**AC**). We find that the 2 immoral behaviors are related to reward function design. We record the reward settings in 4 research with immoral behaviors and summarize them in *Table S2.*

Table S2. Reward Function in Selected Research with Immoral Behaviors

Immoral Behavior	Paper	Reward Settings	reference
Threshold Fluctuation (TF)	1	$ICU_{error}(t) = ICU_{actual}(t) - ICU_{threshold}$ $reward_1(t) = \begin{cases} 0 \ if \ R(t) = R(t)_{high} \\ -\alpha_1 \ if \ R(t) \neq R(t)_{high} \end{cases}$ $reward_2(t) = \begin{cases} 0 & if \ ICU_{error}(t) \leq \alpha_2 \\ -\frac{\alpha_1}{\alpha_2} \cdot ICU_{error}(t) \ if \ ICU_{error}(t) > \alpha_2 \end{cases}$ $reward_{total}(t) = rewaqrd_1(t) + \alpha_3 \cdot reward_2(t)$	[15]
	2	$reward = \begin{cases} r_{health} + r_{economic} \ if \ r_{health} \ and \ r_{economic} > 0 \\ 0 \ otherwise \end{cases}$ $r_{health} = \begin{cases} (TI - I_{new})/TI \ if \ I_{new} < TI \ and \ D < TD \\ 0 \ otherwise \end{cases}$ $r_{economic} = 1 - \frac{\sum wi \cdot Ai}{\sum wi}$	[9]
	3	$reward(t) = E_t \cdot e^{-r \cdot A_t} - s \cdot D_t$ $E_t = \frac{Current \ Economy}{Total \ Population \cdot M_t}$ $D_t = \frac{Cumulative \ Death}{Total \ Population}$ $A_t = \frac{Active \ Case}{Total \ Population} \cdot 100$	[16]
	4	$reward_1(k+1) = \begin{cases} -1 \ if \ I_s((k+1)T) > H \\ 0 \ otherwise \end{cases}$ $reward_2(k+1) = \begin{cases} \frac{e((k+1)T) - e(kT)}{e(kT)} \ if \ e((k+1)T) < e(kT) \\ 0 \ if \ e((k+1)T) \geq e(kT) \end{cases}$ $reward_3(k+1) = \begin{cases} +1.3 \ if \ c_{a_k} = very \ low \ cost \\ +1.2 \ if \ c_{a_k} = low \ cost \\ +1 \ if \ c_{a_k} = medium \ cost \\ -1 \ if \ c_{a_k} = high \ cost \end{cases}$ $reward(k+1) = reward_2(k+1) + reward_2(k+1) + \beta_w reward_2(k+1)$	[8]
Algorithm Cheating (AC)	5	Same as 4	[8]

In order to better prove the real existence of **TF** and **AC**, we simulate the reward function design in these researches and verify our summary of reward design errors. The reward we designed for TF is shown in *Eq. S4* and for AC in *Eq. S5*.

$$R_1 = \begin{cases} 1, & if \ \ I_{new} < 10 \\ 0.5, & if \ \ 10 \le I_{new} < 25 \\ 0.25, & if \ \ 25 \le I_{new} < 50 \\ 0.1, & if \ \ 50 \le I_{new} < 100 \\ 0, & if \ \ I_{new} \ge 100 \end{cases} \tag{S4}$$

$$R_1 = \begin{cases} 0, if I_{new}(t) > I_{new}(t-1) \\ 1, if I_{new}(t) < I_{new}(t-1) \end{cases} \tag{S5}$$

The results of the reward function for TF are shown in *Fig. S8*, which clearly shows an obvious phenomenon of threshold fluctuation. In this setting, although the policy developed by RL can effectively control the epidemic, on the one hand, this setting requires appropriate threshold selection, which is not an easy task in real-world situations; on the other hand, it can be observed that due to the reward orientation of the RL system, the policies it provides will exhibit significant fluctuations near the threshold, making it difficult to strictly control the number of weekly new infectious below the given threshold 5.

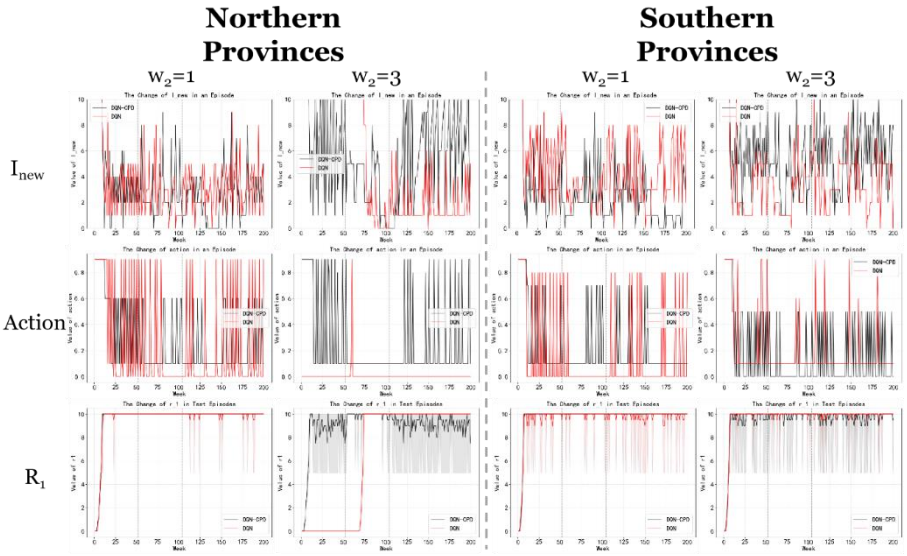

Fig. S8. The Performance of RL-based Policies with TF

The results of the reward function for AC are shown in *Fig. S9*, which demonstrates the existence of AI cheating behavior. The performance under this reward function is normal, but in the southern province scenario, when w2=3, abnormal phenomena can be observed. The DQN policy gains positive rewards through the "short loosening, long tightening" policy, which leads to a significantly worse control effect of the simulated influenza epidemic. The number of weekly new infections is much higher than in the original reward setting, which contradicts the original intention.

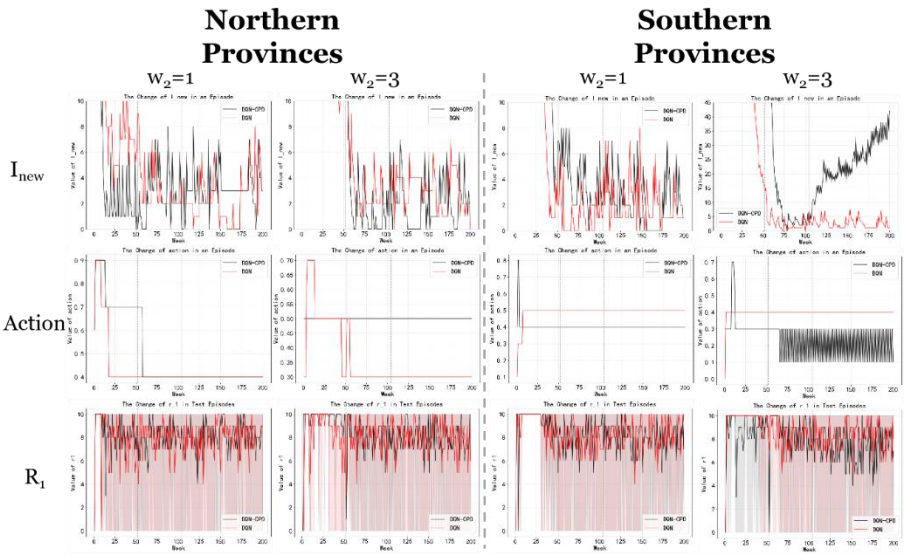

Fig. S9. The Performance of RL-based Policies with AC

References

1. Watkins, C. J. C. H.: Learning From Delayed Rewards. Robotics & Autonomous Systems (1989).
2. Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International conference on machine learning, pp. 1928-1937. PMLR, (2016).
3. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M. J. a. p. a.: Playing atari with deep reinforcement learning in, arXiv, 2013).
4. Godio, A., Pace, F., Vergnano, A.: SEIR Modeling of the Italian Epidemic of SARS-CoV-2 Using Computational Swarm Intelligence. International journal of environmental research and public health 17(10), 1-19 (2020).
5. He, S., Peng, Y., Sun, K.: SEIR modeling of the COVID-19 and its dynamics. Nonlinear Dynamics 101(3), 1667-1680 (2020).
6. Djidjou-Demasse, R., Michalakis, Y., Choisy, M., Sofonea, M. T., Alizon, S.: Optimal COVID-19 epidemic control until vaccine deployment. medRxiv 1-21 (2020).
7. Kompella, V., Capobianco, R., Jong, S., Browne, J., Fox, S. J., Meyers, L. A., Wurman, P. R., Stone, P.: Reinforcement Learning for Optimization of COVID-19 Mitigation policies in, arXiv, 2020).
8. Padmanabhan, R., Meskin, N., Khattab, T., Shraim, M., Al-Hitmi, M.: Reinforcement learning-based decision support system for COVID-19. Biomedical Signal Processing and Control 68(102676 (2021).
9. Chadi, M.-A., Mousannif, H.: A Reinforcement Learning Based Decision Support Tool for Epidemic Control: Validation Study for COVID-19. Applied Artificial Intelligence 36(1), 2031821 (2022).

10. Thompson, R., Wood, J. G., Tempia, S., Muscatello, D. J.: Global variation in early epidemic growth rates and reproduction number of seasonal influenza. International Journal of Infectious Diseases 122(382-388 (2022).
11. Kirkeby, C., Halasa, T., Gussmann, M., Toft, N., Græsbøll, K.: Methods for estimating disease transmission rates: Evaluating the precision of Poisson regression and two novel methods. Scientific Reports 7(9496), 1-11 (2017).
12. Smirnova, A., deCamp, L., Chowell, G.: Forecasting Epidemics Through Nonparametric Estimation of Time-Dependent Transmission Rates Using the SEIR Model. Bulletin of mathematical biology 81(11), 4343-4365 (2019).
13. McIntosh, K.: Coronaviruses: A Comparative Review. In: Current Topics in Microbiology and Immunology / Ergebnisse der Mikrobiologie und Immunitätsforschung, pp. 85-129. Springer Berlin Heidelberg, Berlin, Heidelberg (1974).
14. Perakis, G., Singhvi, D., Skali Lami, O., Thayaparan, L.: COVID-19: A multiwave SIR-based model for learning waves. Production and Operations Management 0(0), 1-19 (2022).
15. Arango, M., Pelov, L.: COVID-19 Pandemic Cyclic Lockdown Optimization Using Reinforcement Learning in, arXiv, 2020).
16. Ohi, A. Q., Mridha, M. F., Monowar, M. M., Hamid, M. A.: Exploring optimal control of epidemic spread using reinforcement learning. Scientific Reports 10(1), 22106 (2020).