Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Applied Soft Computing 113 (2021) 107947

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

An evolvable adversarial network with gradient penalty for COVID-19 infection segmentation

Juanjuan He [a,b], Qi Zhu [a,b], Kai Zhang [a,b], Piaoyao Yu [a,b], Jinshan Tang [c,*]

[a] College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China
[b] Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan, China
[c] Department of Health Administration and Policy George Mason University, Fairfax, VA, 22030, USA

ARTICLE INFO

Article history:
Received 31 May 2021
Received in revised form 20 August 2021
Accepted 22 September 2021
Available online 12 October 2021

Keywords:
COVID-19
Infection segmentation
Wasserstein generative adversarial network
Gradient penalty
Evolutionary algorithm

ABSTRACT

COVID-19 infection segmentation has essential applications in determining the severity of a COVID-19 patient and can provide a necessary basis for doctors to adopt a treatment scheme. However, in clinical applications, infection segmentation is performed by human beings, which is time-consuming and generally introduces bias. In this paper, we developed a novel evolvable adversarial framework for COVID-19 infection segmentation. Three generator networks compose an evolutionary population to accommodate the current discriminator, i.e., generator networks evolved with different mutations instead of the single adversarial objective to provide sufficient gradient feedback. Compared with the existing work that enforces a Lipschitz constraint by weight clipping, which may lead to gradient exploding or vanishing, the proposed model also incorporates the gradient penalty into the network, penalizing the discriminator's gradient norm input. Experiments on several COVID-19 CT scan datasets verified that the proposed method achieved superior effectiveness and stability for COVID-19 infection segmentation.

1. Introduction

COVID-19 spreads globally, with more than 200 million confirmed cases in 223 countries and over 4 million deaths. AI techniques have attracted significant attention in the fight against COVID-19. For example, Fang et al. presented a novel SLAM algorithm using RGB and depth images to reduce cross-infection risk between doctors and patients. This method could also raise hospital operation efficiency [1]. Dash et al. presented a new audio feature called C-19CC, which can diagnose the initial condition without visiting a hospital [2].

Medical imaging technologies play a vital role in fighting COVID-19. These technologies can be used to diagnose COVID-19 and evaluate the treatment of COVID-19 patients. Wang et al. offered CHP-Net to differentiate and localize COVID-19 from community-acquired pneumonia [3]. CHP-Net can also extract more features from chest X-ray radiographs than other ConvNet. One crucial application to use medical imaging is to segment the COVID-19 infections to assess the severity of the patients through medical images, which can aid doctors in the treatment. However, in clinical applications, infection segmentation is performed by human beings, which is time-consuming and generally introduces bias. In this paper, we focus on automatic segmentation of COVID-19 infections.

Recently, various methods have been proposed for the segmentation of COVID-19 infection. For example, Oulefki et al. proposed an efficient Kapur entropy-based multilevel thresholding unsupervised network, minimizing the over-segmented regions [4]. Shan et al. presented a VB-Net, which incorporated the attention mechanism to capture rich contextual relationships [5]. Zhou et al. included an attention mechanism with a U-Net architecture, which can re-weight the feature spatially for better feature representations [6]. To tackle the low contrast between COVID-19 infections and normal tissues, Inf-Net used a parallel partial decoder to aggregate the high-level features and generated a global map to solve the implicit reverse attention [7]. Chen et al. used aggregated residual transformations and soft attention mechanism to improve the model's capability [8]. Yu et al. proposed a lightweight deep learning model (MiniSeg) for COVID-19 segmentation [9]. MiniSeg had 83K parameters and was not easy to cause overfitting issues. It also had high computational efficiency and was convenient for practical deployment.

Methods based on GANs were developed for COVID-19 infection segmentation. Xu et al. presented a weakly supervised lesion framework (GASNet) by embedding the generative adversarial training process into the segmentation network [10]. GASNet was supervised by chest CT scans without voxel-level annotations,

* Corresponding author.
E-mail address: jtang25@gmu.edu (J. Tang).

https://doi.org/10.1016/j.asoc.2021.107947
1568-4946/

and the generator was used to segment the lesions in a COVID-19 CT image. Attention U-Net-based GAN was developed for lung segmentation on COVID-19 X-ray images [11]. Although the research in [11] was designed not specifically for COVID-19 infection segmentation, it can easily modify the approach for COVID-19 and other diseases. To reduce the impact of domain shift between the real and synthetic data, Chen et al. introduced a conditional GAN to make the embedding distribution closer [12].

Although these prior works have achieved remarkable progress, GANs tend to be challenging to be trained and even suffer from model collapse [13]. Besides, network architecture and hyperparameters setting will significantly affect the GAN-generated samples' quality and lead to gradient vanishing issues and doesn't not yield good results [14,15]. Weight clipping is one of the most comprehensive strategies for the 1-Lipschitz constraint. It enforces parameters in a given range (between $-c$ and c). However, this behavior might make most of the weights equal to $-c$ or c, which weakens the fitting ability of GANs and commonly results in exploding or vanishing gradients [16]. Moreover, many recent efforts on GANs are contributed to handle the training difficulties by developing various adversarial training objectives [13]. Since each objective has its own and downsides [17], no single one is "the best" for all conceivable types of metrics, i.e., the trade-off between objectives varies from problem to problem.

This paper proposes an evolvable GAN framework for automatic COVID-19 infection segmentation. The proposed framework employs three adversarial objective functions as mutation operators to optimize the generator network for generating high-quality samples. Generators' population evolves in each iteration, which attempts to cut down the distance between the generated and actual distribution. Moreover, we adopt gradient penalty instead of weight clipping to satisfy the Lipschitz continuity condition to achieve steady. The Wasserstein distance is also incorporated to replace the Jensen–Shannon divergence (JS divergence) commonly used in GANs. By this means, the impact of hyperparameters is alleviated, leading to better generative performance. Besides, we also designed a fitness function according to the discriminator for evaluating the performance of the evolved generator network. The best offspring is preserved as the next parent for evolution. Extensive experiments on four public COVID-19 CT scan datasets demonstrate that our method can improve stability and generative performance. In a nutshell, our main contributions and innovations in this paper are summarized as follows:

- Inspired by the evolutionary algorithm in deep learning, we utilized three different mutation operators to update the generator for automatic COVID-19 infection segmentation, which can overcome the limitation of a single adversarial training objective.
- To alleviate the gradient vanishing caused by weight clipping and JS divergence, we introduced gradient penalty and adopted Wasserstein distance in our network to satisfy 1-Lipschitz constraint.
- We present an evolvable adversarial framework for automatic COVID-19 infection segmentation in CT images. Compared with some previous methods, the segmentation results of our proposed method have fewer mis-segmented regions and more accurate boundaries, especially in the subtle infection regions. Numerical experience also indicates that our approach tends to be more stable and efficient in six widely adopted metrics.

2. Related works

In this section, we first give a short introduction to GANs. Then the concept of evolutionary algorithms and some works integrated with neural networks are briefly summarized.

2.1. Generative Adversarial Networks (GANs)

GANs contain a generator and a discriminator, which offers an excellent framework for training deep generative models. In the training process, deepfake samples are created by the generator to deceive the discriminator while the discriminator goes out of its way to distinguish ground truth in the training datasets from these fake samples [18]. They progress alternately until the generator wins the adversarial game, i.e., the generator network can synthesize examples so that the discriminator network cannot make a better decision than randomly guessing. GANs have been widely applied in image processing, such as image generation [19], photo editing [20], image-to-image translation [21], and video prediction [22].

GANs and their variants are prevalent deep learning models for automatic medical image segmentation. GANs consider both local and global contextual relations between pixels in images and thus can improve the ability to directly enforce the learning of multiscale spatial constraints. They provide practical ways to deal with complex medical images [23]. For example, U-net-GAN combines a GAN strategy to train a deep learning network to segment multiple organs on chest CT images [24]. Spine-GAN connected GAN, LSTM, and atrous autoencoder in an integrated end-to-end framework to segment multiple spinal structures [25]. RescueNet used unpaired adversarial training to segment the whole tumor, followed by core and enhance regions on brain MRI scans [26].

However, there are four significant problems in the existing GAN models: non-convergence, mode collapse, diminished gradient, and high sensitivity to the hyperparameter selections [15]. LSGAN utilizes the least square loss function for the discriminators and can converge faster than most GANs. It partly avoids mode collapse but not assigns a high cost to generate well-performance samples [27]. E-GAN employs multiple objectives as mutation operations and evolves the generators' populations [28]. E-GAN can integrate the advantages of different training objectives and select the best offspring to generate better-performing samples under such circumstances. WGAN minimizes an efficient approximation of Wasserstein distance instead of JS divergence in the classic GAN without the requirement of maintaining a careful balance between the generator and the discriminator during training [13]. WGAN also copes with mode collapse but sometimes still generates low-quality samples or fails to converge. WGAN-GP finds that most of these problems are due to weight clipping, leading to exploding or vanishing gradients [16]. WGAN-GP enforces Lipschitz constraint by working with gradient penalty and achieve high-quality generations.

2.2. Evolutionary algorithms

Inspired by biological evolution, evolutionary algorithms have succeeded in many computing tasks, including optimization, modeling, and design [29,30]. Evolutionary algorithms often perform well-approximating solutions to almost all types of problems with high robustness. Besides, the advantages of evolutionary algorithms also include self-organization, self-adaptation, and self-learning. Therefore, they can effectively tackle intricate problems and have broader applicability than traditional optimization methods [31].

Recently, many problems in deep learning were solved using evolutionary algorithms. For example, multi-node evolutionary neural networks adopt an evolutionary algorithm to optimize the hyperparameters for automating network selection on computational clusters [32]. Evolutionary algorithms are also used to optimize deep learning architectures and extended to optimize the topology, components, and hyperparameters [33]. In [34], the

performance of deep learning networks was improved by evolving a population of autoencoders, i.e., learning multiple autoencoder features, evaluating them based on their reconstruction quality, and generating new individuals by adopting mutation operators.

3. Methods

The proposed medical image segmentation framework is described in Fig. 1. Unlike classic GAN, which has a generator and a discriminator, a population generator network evolves with different mutations instead of the single adversarial objective. In this case, we exploit the advantages and suppress the shortcomings of different GAN objectives by optimizing the generator network to get high-quality samples and more stable performance. Each generator is a U-Net variant. The encoding part consists of 4 × 4 convolutional layers with stride 2, batch normalization layers, activation layers (leaky ReLU), and corresponding feature maps. The decoding part includes the image resize layers with factor 2, 3 × 3 convolutional layers with stride 1, batch normalization layers, and activation layers (ReLU). The encoding setup of the discriminator is the same as the generator.

Roughly speaking, the original CT images are input into the generator initially, and the discriminator will train the output and ground truth with gradient penalty. The distance between the generated and actual distribution is evaluated at the following stage, which provides fitness scores for offspring selection in the generator evolution process. Then, we evolve a population of the generators for the best one from three different mutation operators. Fitness scores evaluate the offspring networks so that the best-performing offspring is selected for the next iteration.

We utilize three mutations to generate the offspring generator networks. The first mutation operator, known as L_1 mutation, is appeared as follows:

$$\mathcal{M} = \mathbb{E}_{x\sim P_g, x'\sim P_r}\left[\left\|D\left(x\right) - D(x')\right\|_1\right] \tag{1}$$

The L_1 mutation has strong robustness and is not sensitive to abnormal samples. If there are several abnormal samples during training, the L_1 mutation would not adjust to fit the individual abnormal sample, which is more stable than other mutations. But in the later stage of training, the loss of L_1 mutation will fluctuate around the stable value, and it is difficult to converge to higher accuracy.

The second mutation function, named conditional mutation, is as follows:

$$\mathcal{M} = \mathbb{E}_{x\sim P_g, x'\sim P_r}\left[\log(1 - D(x|x'))\right] \tag{2}$$

The conditional mutation adopts conditions to supervise the generator. The condition can be any auxiliary information, such as class labels or data from other modalities. In this work, we use ground truth as a condition to lead the distribution of generated images more similar to actual distribution.

The Non-saturating mutation, which penalizes the generator to deceive the discriminator, has good convergence property, i.e.:

$$\mathcal{M} = -\mathbb{E}_{x\sim P_g}\left[logD(x)\right] \tag{3}$$

The Non-saturating mutation would not saturate. In the initial stage of training, when the discriminator can easily distinguish between true and false samples, the generator will face the problem of gradient vanishing. This mutation will converge faster at the beginning of training, but the optimization objects are easily affected, leading to gradient instability.

Through the above three mutation operators, we will obtain three different offspring generator networks after each iteration. Since each mutation has its own and downsides, no single one is "the best" for all conceivable types of metrics. Thus, we utilize the evaluation function to evaluate the offspring's fitness scores and select the best as the parent network in the next iteration.

We replace JS divergence commonly used in GANs with Wasserstein distance, which measures the minimum consumption under optimal path planning when one distribution is moved to another [35]. Wasserstein distance is defined as:

$$W\left(P_1, P_2\right) = \inf_{\gamma\sim\prod(P_1,P_2)} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x - y\|\right] \tag{4}$$

where P_1 and P_2 represent two distributions. x and y represent samples. $x \sim P_1$ means x is from distribution P_1. $\mathbb{E}$ represents expectation calculator. $\gamma\left(x, y\right)$ stands for joint distributions, which indicates how much "mass" must be transported from x to y to transform the distributions P_1 into the distribution P_2. $\prod\left(P_1, P_2\right)$ denotes the set of all $\gamma(x, y)$ whose marginals are P_1 and P_2 respectively.

When Wasserstein distance is applied to GAN, it is defined as:

$$W\left(P_r, P_g\right) = (1/K)\sup_{\|D\|_L\le K} \mathbb{E}_{x\sim P_r}\left[D(x)\right] - \mathbb{E}_{\tilde{x}\sim P_g}\left[D(\tilde{x})\right] \tag{5}$$

where P_r is the distribution of the real samples, and P_g is the distribution of the generated samples. D is the discriminator network. $D\left(x\right)$ is the output of the D with sample x and $\|D\|_L \le K$ is the function f, which enforces a K-Lipschitz constraint. It means that we could take the maximum distribution distance between the two exceptions when the function D meets the K-Lipschitz constraint. The loss function of the discriminator network is computed as follows:

$$L = \mathbb{E}_{x\sim P_r}\left[D_\omega(x)\right] - \mathbb{E}_{\tilde{x}\sim P_g}\left[D(\tilde{x})\right] \tag{6}$$

where L is Wasserstein distance. D_ω is the discriminator network with a set of parameters ω. The discriminator network needs to optimize L to decrease the Wasserstein distance.

When the support of P_r and P_g is a low-dimensional manifold in a high-dimensional space, the overlap between P_r and P_g is always 0. In this scenario, the JS divergence will be constantly equal to $log2$, leading to gradient vanishing. But Wasserstein distance can continuously provide sufficient gradients no matter whether the two distributions overlap or not. This excellent feature of Wasserstein distance can solve vanishing gradient problems and make the network more stable.

Different from the commonly used weight clipping, we adopt gradient penalty in discriminator to satisfy the 1-Lipschitz constraint, and the final loss function is as follows:

$$L = \mathbb{E}_{x\sim P_r}\left[D(x)\right] - \mathbb{E}_{\tilde{x}\sim P_g}\left[D\left(\tilde{x}\right)\right] + \lambda\mathbb{E}_{\hat{x}\sim P_{\hat{x}}}\left[\left(\|\nabla_{\hat{x}}D\left(\hat{x}\right)\|_2 - 1\right)^2\right] \tag{7}$$

where λ is the gradient penalty coefficient. $\hat{x}$ is a random sample from $P_{\hat{x}}$. $P_{\hat{x}}$ is uniform distribution along the straight lines between the pairs of points sampled from the actual and generator distribution [16]. To satisfy the 1-Lipschitz constraint, the gradient penalty directly constrains the gradient norm of the discriminator's output concerning its input.

When the discriminator network is trained to the optimal situation, the discriminator network can be expressed as:

$$D^*\left(x\right) = \frac{P_r\left(x\right)}{P_r\left(x\right) + P_g\left(x\right)} \tag{8}$$

We input the generated samples into the discriminator network in the evaluation part, considering samples' quality. Then, the offspring network's fitness can be obtained. The closer the fitness scores are to 1, the closer the offspring network's image is to the real distribution. The fitness function is shown as follows:

$$\mathcal{F} = \mathbb{E}_{x\sim P_g}\left[D(x)\right] \tag{9}$$

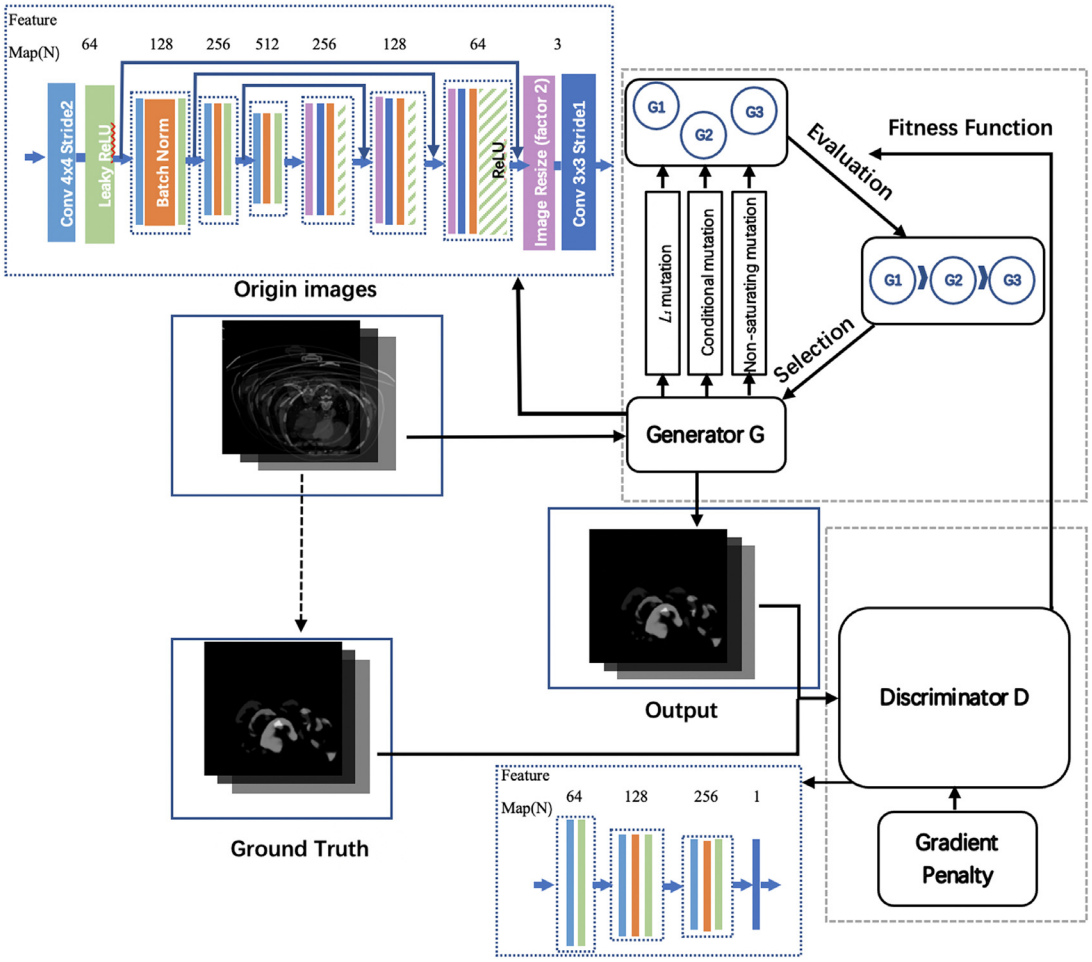

Fig. 1. The framework of our proposed evolvable adversarial network for medical image segmentation.

The offspring network with the highest fitness score is selected for the next iteration, while other networks are eliminated. Therefore, the generator network will be optimized according to different mutation functions to adapt to the continuously updated discriminator network. Compared with the original GAN, this evolvable framework can integrate the advantages of different adversarial training objectives and is more stable.

In most situations, the fitness of the generated offspring can be judged by two properties of the generated samples: (1) the quality and (2) the diversity. The mode collapse issue in GANs optimization may lead to poor diversity of generated samples. But as a fully supervised network, our study only needs to judge the fitness of the generated offspring by the quality of the generated samples.

In the training process, considering the initial discriminator's parameters ω_0 and the generators' parameters $\{\mu_0^1, \mu_0^2, \ldots, \mu_0^{n_p}\}$, where n_p is the number of the parent generators (In this work, $n_p = 1$).

In each iteration, the discriminator was updated for n_D steps firstly (In this work, $n_D = 2$. The batch size $m = 8$). Meanwhile, we sample a batch of label $\{x^{(i)}\}_{i=1}^{m} \sim P_r$, a batch of output $\{\tilde{x}^{(i)}\}_{i=1}^{m} \sim P_g$ and a random number $\epsilon \sim U[0, 1]$, where P_r is the distribution of the real samples, P_g is the distribution of the generated samples. $\hat{x}$ is computed as follows:

$$\hat{x} \leftarrow \epsilon x + (1 - \epsilon)\tilde{x} \tag{10}$$

The discriminator's parameters were updated based on $x^{(i)}, \tilde{x}^{(i)}$, $\hat{x}$ and gradient penalty, the update step is shown as follow:

$$g_\omega \leftarrow \nabla_\omega \left[\frac{1}{m}\sum_{i=1}^{m} D_\omega\left(x^{(i)}\right) - \frac{1}{m}\sum_{i=1}^{m} D_\omega\left(\hat{x}^{(i)}\right) + \lambda \frac{1}{m}\sum_{i=1}^{m} \left(\| \nabla_{\hat{x}} D_\omega\left(\hat{x}\right) \|_2 - 1 \right)^2 \right] \tag{11}$$

$$\omega \leftarrow Adam(g_\omega, \omega, \alpha, \beta_1, \beta_2) \tag{12}$$

where D_ω is the discriminator, λ is the hyper-parameter of gradient penalty, α, β_1, β_2 are Adam hyper-parameters (default $\alpha = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.999$, $\lambda = 0.1$).

Then, each parent generator was optimized by n_m mutation operators (In this work, $n_m = 3$). We sample a batch of input $\{x^{(i)}\}_{i=1}^{m} \sim P_{CT}$, where P_{CT} is the distribution of the COVID-19 CT samples. The generator's parameters are updated as follows:

$$g_{\mu^{j,l}} \leftarrow \nabla_{\mu^j} \mathcal{M}_G^l \left(\{x^{(i)}\}_{i=1}^{m}, \mu^j \right) \tag{13}$$

$$\mu_{child}^{j,l} \leftarrow Adam(g_{\mu^{j,l}}, \mu^j, \alpha, \beta_1, \beta_2) \tag{14}$$

where $\mathcal{M}_G^l$ is the lth mutation operator, and $\mu^{j,l}$ denotes the parameter of the jth parent generator updated by the lth mutation operator. In this way, we can obtain a batch of children generators with the parameter $\mu_{child}^{j,l}$.

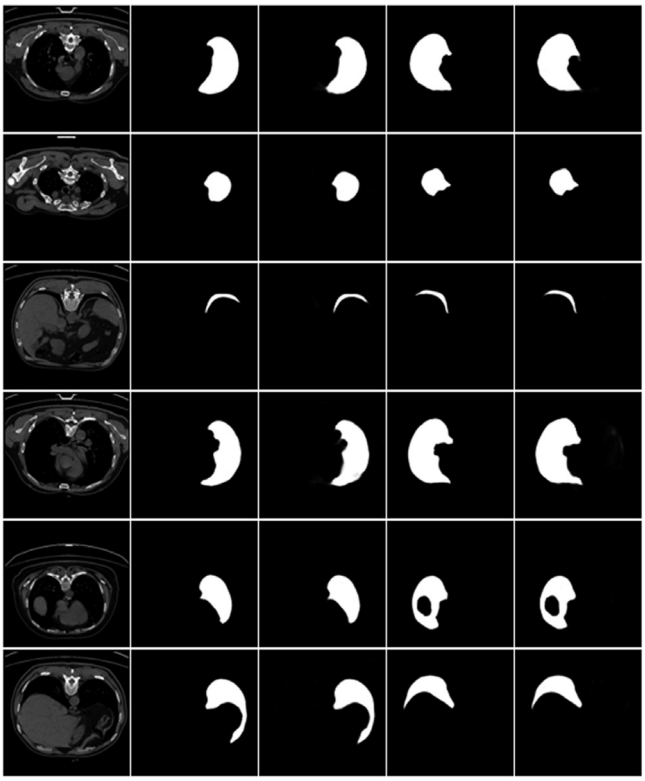

Fig. 2. Visual comparison of COVID-19 infection regions segmentation results on the COVID-19 CT scan dataset.

Finally, we use the fitness function to evaluate the fitness score of each children generator and sort them in descending order.

$$\left\{\mathcal{F}^{j_1,l_1}, \mathcal{F}^{j_2,l_2}, \ldots\right\} \leftarrow sort\left(\left\{\mathcal{F}^{j,l}\right\}\right) \tag{15}$$

The n_p children generators with higher fitness scores will be selected as the parent generators for the next iteration.

$$\left\{\mu^1, \mu^2, \ldots, \mu^{n_p}\right\} \leftarrow \left\{\mu_{child}^{j_1,l_1}, \mu_{child}^{j_2,l_2}, \ldots, \mu_{child}^{j_{np},l_{np}}\right\} \tag{16}$$

4. Experiments

4.1. Datasets

This section evaluated the proposed method using four public COVID-19 CT datasets, including COVID-19 CT scan dataset, COVID-19-1110 dataset, COVID-19-9 dataset, and MS COVID-19 dataset. We also conducted ablation studies to verify the effectiveness of each contribution in our framework on the COVID-19 CT scan datasets.

COVID-19 CT scan dataset [36]. The COVID-19 CT scan dataset consisted of 20 annotated COVID-19 chest CT volumes. Each CT volume was finally verified by senior radiologists with more than ten years of experience. The volumes of each CT scan dataset subject had a resolution of 512 × 512 with slices about 176 by mean (200 by median). We cropped each subject into a sub-volume of 480 × 480 × 160 to remove the black border regions while keeping the entire lung regions.

COVID-19-1110 dataset [37]. The COVID-19-1110 dataset consisted of 1110 COVID-19 CT studies. The dataset was provided by medical hospitals in Moscow, Russia. A small subset of studies (50 pcs.) was annotated by the experts of the Research and Practical Clinical Center for Diagnostics and Telemedicine Technologies of the Moscow Health Care Department.

COVID-19-9 dataset [38]. The COVID-19-9 dataset consisted of 9 axial COVID-19 volumetric CTs from Radiopaedia. This dataset includes both positive and negative slices (373 out of 829 slices have been evaluated by a radiologist as positive and segmented).

MS COVID-19 dataset [39]. The MS COVID-19 dataset consisted of 100 axial CT images from more than 40 patients. All the CT images were collected by the Italian Society of Medical and Interventional Radiology. The CT images were segmented by a radiologist using three labels: ground-glass opacity (GGO), consolidation, and pleural effusion.

4.2. Experimental setup and metrics

To verify the performance of the proposed method, we compared it with three state-of-the-art COVID-19 segmentation methods, including Inf-Net [7], MiniSeg [9], and the U-Net [40]. In the training process, the batch size was 8, the learning rate was 0.0002, and the number of offspring selected in each iteration was 1. All experiments were performed on a machine with GTX 2060 GPU, Intel Core i5-9400F CPU, and a 16G RAM equipped with PyTorch. Each slice was resized to 256 × 256 as the input. Then, we randomly selected 80% of each subject for the training set, 10% for the validation set, and 10% for the test set.

We used six widely adopted metrics, including the Dice similarity coefficient (Dice), Intersection over Union (IoU), Sensitivity (Sen), Specificity (Spec), Mean Absolute Error (MAE), and Structure Measure (SM). The formula of Dice, IoU, Sen, and Spec are shown as follows:

$$Dice = \frac{2 * TP}{FN + 2 * TP + FP} \tag{17}$$

$$IoU = \frac{TP}{FN + TP + FP} \tag{18}$$

$$Sen = \frac{TP}{TP + FN} \tag{19}$$

$$Spec = \frac{TN}{TN + FP} \tag{20}$$

where TP, TN, FP and FN refer to true positive, true negative, false-positive and false-negative pixels of the output images and the ground truth.

MAE measures the pixel-wise error between the output images and the ground truth, which is defined as:

$$MAE = \frac{1}{w \times h} \sum_{x}^{w} \sum_{y}^{h} |S(x, y) - G(x, y)| \tag{21}$$

where w and h are the width and height of the output image S and the ground truth G, and (x, y) donates the coordinate of each pixel in S and G.

Structure Measure measures the structural similarity between a prediction map and the ground truth mask, which is consistent with the human visual system and defined as [41]

$$SM = (1 - \alpha) * S_o(S, G) + \alpha * S_r(S, G) \tag{22}$$

where α (default $\alpha = 0.5$) is a balance factor between the object-aware similarity S_o and the region-aware similarity S_r.

4.3. Comparison experiments and ablation studies

***(1) Left and Right Lung Segmentation Results*:** In the first experiment, we segmented left and right lungs on the COVID-19 CT scan dataset and compared them with the ground truth. As shown in Fig. 2, the proposed method can accurately segment the

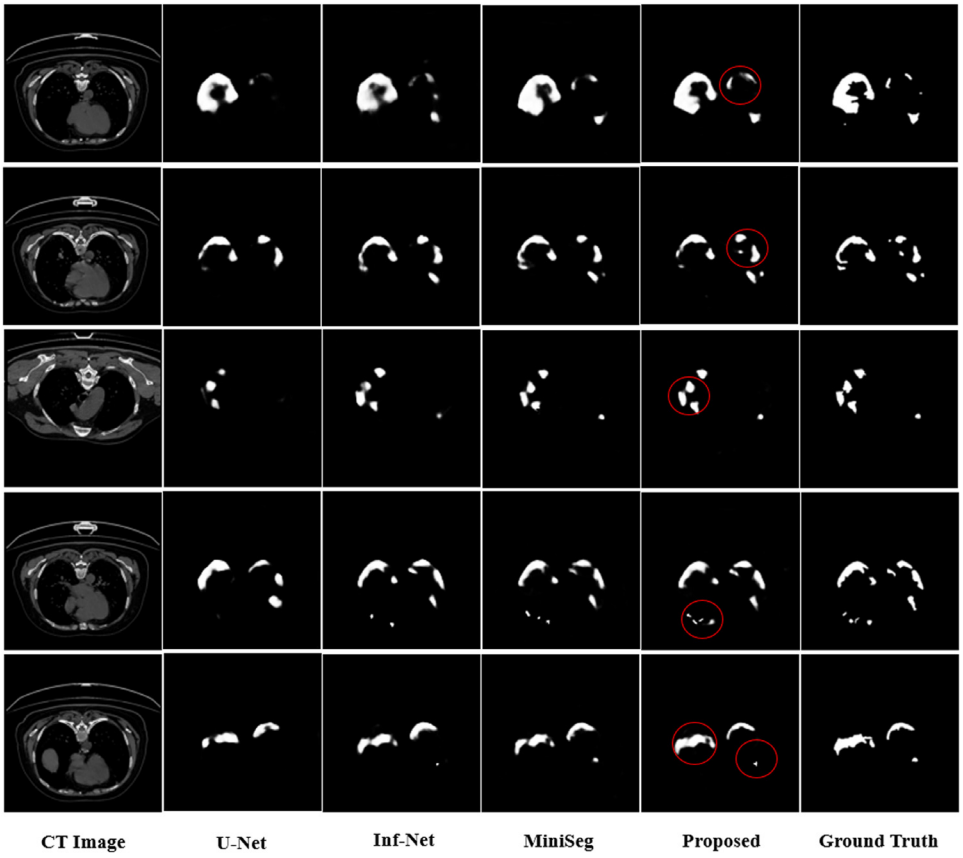

Fig. 3. Visual comparison of lung region segmentation results on the COVID-19 CT scan dataset.

corresponding lung area for various CT images, both left and right lung.

***(2) Comparison with Existing Methods and Ablation Study*:** We compared the proposed method with Inf-Net, MiniSeg, U-Net on the COVID-19 CT scan dataset. As shown in Fig. 3, although all these methods' results were roughly the same in segmentation boundaries, the proposed method can segment the infected area of COVID-19 with some more details (marked with red circles) than other methods.

We also conducted ablation studies to verify the effectiveness of each contribution in the proposed method. We utilized the six metrics defined above to perform quantitative comparisons and compared the quantitative results of the proposed method under different settings: Proposed-1 (proposed method without mutation), Proposed-2 (proposed method without Gradient Penalty), Proposed-3 (proposed method without Gradient Penalty and mutation). We performed experiments using the proposed method under different settings and other techniques, including Inf-Net, MiniSeg, and U-Net, on COVID-19 infection region segmentation and lung region segmentation. The results are in Tables 1 and 2, respectively. In Table 1, the proposed method outperformed Inf-Net and U-Net in Dice, IoU, Sen, Spec, and MAE, SM by a large margin. We attributed this improvement to GANs and mutations, which considered both local and global contextual relations between pixels in images and combined the advantages of different adversarial training objectives. As a network designed for accurate and efficient COVID-19 segmentation with limited training data, MiniSeg also performed better than Inf-Net and U-Net, but the proposed method still outperformed MiniSeg with a 0.47% improvement in terms of Dice and 1.84% improvement in terms of Sen. The performance on SM also improved by 1.54%.

As the baseline method, Proposed-3 refers to GAN based on U-Net. Compared with U-Net, Proposed-3 boosted performance with 4.48% improvement in Dice and 5.64% improvement in IoU. Due to the mutations, Proposed-2 outperformed Proposed-3 with 1.8% improvement in Dice and 2.38% improvement in IoU. Proposed-1 also beat Proposed-3 with 0.54% improvement in Dice by employing gradient penalty instead of weight clipping. Although Proposed-3 uses weight clipping, its shortcomings do not show up obviously because of its good parameter settings. For the same reason, the progress of Proposed-1 is limited despite using gradient penalty. The results of Proposed-2 are better than Proposed-3 and Proposed-1 due to taking advantage of mutation. In the following experiments, we also discuss the parameter c of weight clipping. It can be observed that when gradient penalty and mutations were utilized together, compared with Proposed-2, the proposed method can further boost the performance with 0.6% improvement in terms of Dice and 2.57% improvement in terms of Sen. As shown in Table 2, we can also find that the proposed method had similar progress on lung region segmentation task.

Table 1
Quantitative results of COVID-19 infection regions on the COVID-19 CT scan dataset. The best results are shown in bold fonts.

	Methods	Dice	IoU	Sen	Spec	MAE	SM
COVID-19 CT scans	Proposed	**0.7853**	**0.6465**	**0.8615**	0.9987	**0.0024**	**0.8525**
	Proposed-1	0.7667	0.6217	0.8164	0.9953	0.0077	0.8243
	Proposed-2	0.7793	0.6384	0.8358	0.9973	0.0039	0.8352
	Proposed-3	0.7613	0.6146	0.8041	0.9935	0.0106	0.8217
	MiniSeg	0.7806	0.6402	0.8431	**0.9991**	0.0035	0.8371
	Inf-Net	0.7664	0.6213	0.8157	0.9943	0.0052	0.8296
	U-Net	0.7165	0.5582	0.7361	0.9861	0.0227	0.7947

***(3) Convergence Analysis*:** To verify the effectiveness of each contribution in the proposed methods, we compared the convergence

Table 2
Quantitative results of lung regions on the COVID-19 CT scan dataset. The best results are shown in bold fonts.

	Methods	Dice	IoU	Sen	Spec	MAE	SM
COVID-19 CT scans (Lung Masks)	Proposed	**0.9734**	**0.9482**	**0.9612**	**0.9783**	0.0005	**0.9861**
	Proposed-1	0.9556	0.9149	0.9431	0.9615	0.0008	0.9786
	Proposed-2	0.9607	0.9244	0.9487	0.9668	0.0007	0.9798
	Proposed-3	0.9462	0.8979	0.9326	0.9571	0.0012	0.9713
	MiniSeg	0.9616	0.9261	0.9485	0.9725	**0.0004**	0.9827
	Inf-Net	0.9534	0.9109	0.9397	0.9618	0.0008	0.9773
	U-Net	0.9226	0.8563	0.9177	0.9512	0.0015	0.9499

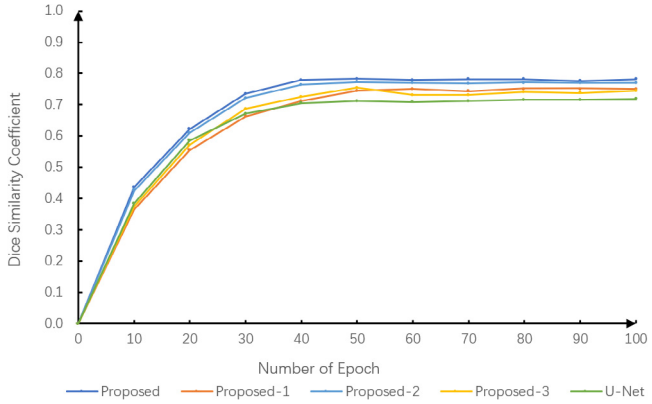

Fig. 4. Training process of COVID-19 infection regions segmentation on the COVID-19 CT scan dataset.

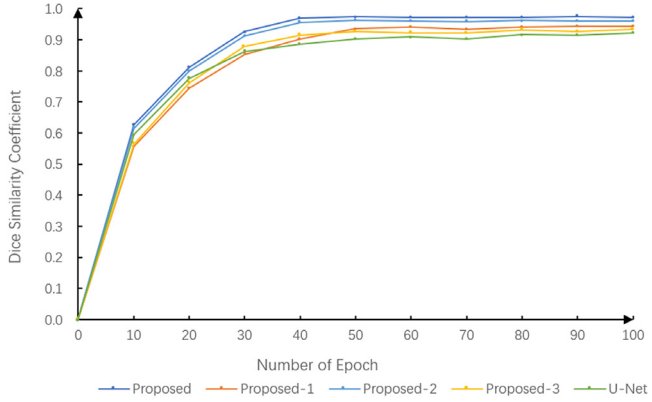

Fig. 5. Training process of lung regions segmentation on the COVID-19 CT scan dataset.

of the proposed method, Proposed-1, Proposed-2, Proposed-3, and U-Net. As shown in Figs. 4 and 5, the proposed method and Proposed-2 converged faster than the other models in the initial training stage. We owed this improvement to the Non-saturating mutation. We can also find that the proposed method obtained better results than others, and we presume the credit should be given to mutations and gradient penalty.

***(4) Stability Analysis*:** To further illustrate the significance of penalty gradient in our evolvable adversarial network, we compared the performance of the proposed method, Proposed-1, Proposed-2, and Proposed-3, with different parameter ranges $[-c, c]$. The evaluation criterion was the Dice coefficient after training 100 Epochs. As shown in Fig. 6, when the value of c was too large or too small, the Dice results obtained by Proposed-2 and Proposed-3 were significantly reduced, while the results obtained by our method and Proposed-1 were more stable than others. We suppose that should be attributed to gradient penalty. In addition, Proposed-2 still achieved better performance than

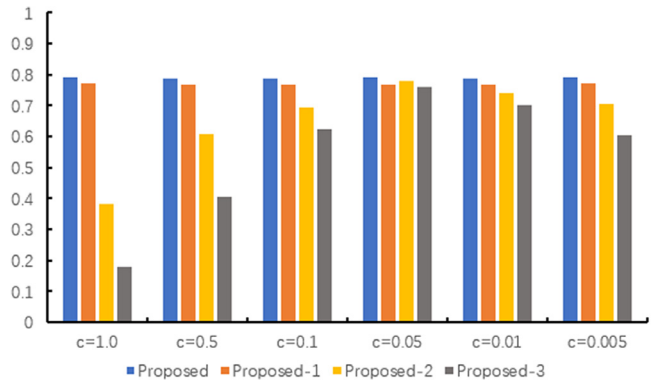

Fig. 6. Dice results after 100 training Epochs with different weights for weight clipping on the COVID-19 CT scan dataset.

Table 3
Quantitative results of COVID-19 infection regions on the COVID-19-1110 dataset. The best results are shown in bold fonts.

	Methods	Dice	IoU	Sen	Spec	MAE	SM
COVID-19-1110	Proposed	**0.6683**	**0.5018**	0.8018	0.9761	**0.0105**	**0.7729**
	MiniSeg	0.6491	0.4804	**0.8113**	0.9789	0.0129	0.7526
	Inf-Net	0.6275	0.4572	0.7883	0.9777	0.0238	0.7129
	U-Net	0.5934	0.4219	0.7178	0.9691	0.0477	0.6747

Proposed-3 due to the mutations. When $c = 0.05$, Proposed-2 and Proposed-3 had the best performance, but at this time parameter c was only a rough estimate, and a series of experiments were required to find the optimal global solution of parameter c, which was also one of the reasons why the network with weight clipping training was difficult. In consequence, the proposed method was not affected by parameter range limitation during training and achieved better stable performance.

***(5) Results on COVID-19-1110 dataset*:** We also compared the proposed method with those obtained by Inf-Net, MiniSeg, U-Net on the COVID-19-1110 dataset. As shown in Fig. 7, although U-Net can roughly segment the COVID-19 infection regions, the performance was not promising. We can find that the segmentation results of the proposed method outperformed other methods and were closer to the ground truth. Specifically, compared with other methods, the results of the proposed method had fewer mis-segmentation regions and more accurate boundaries, especially in the subtle infection regions.

The quantitative results are shown in Table 3. The proposed method outperformed U-Net with 7.49% improvement in Dice and outperformed MiniSeg with 2.14% improvement in IoU. Compared with Inf-Net, SM was improved from 71.29% to 77.29%. The proposed method also achieved better performance in most evaluation metrics.

In addition, it is evident that the results in Table 1 are better than those in Table 3 in terms of all indicators. We assumed that this was due to the different infection degrees of patients between these two datasets. Comparing the CT images in the COVID-19 CT scan dataset and the COVID-19-1110 dataset, most of the CT images in the latter dataset have smaller infected areas than those in the former one. Thus, we infer that the larger degree of the infected area, the better the segmentation results.

***(6) Results on the COVID-19-9 dataset*:** To further verify our contributions on COVID-19 CT segmentation, we compared the results on another newer dataset—the COVID-19-9 dataset. Fig. 8 shows the qualitative results of the proposed method and the other three methods. By visually checking the segmentation results in Fig. 8, we can find that all the methods can get relatively good results for large and clear infections. However, due to the

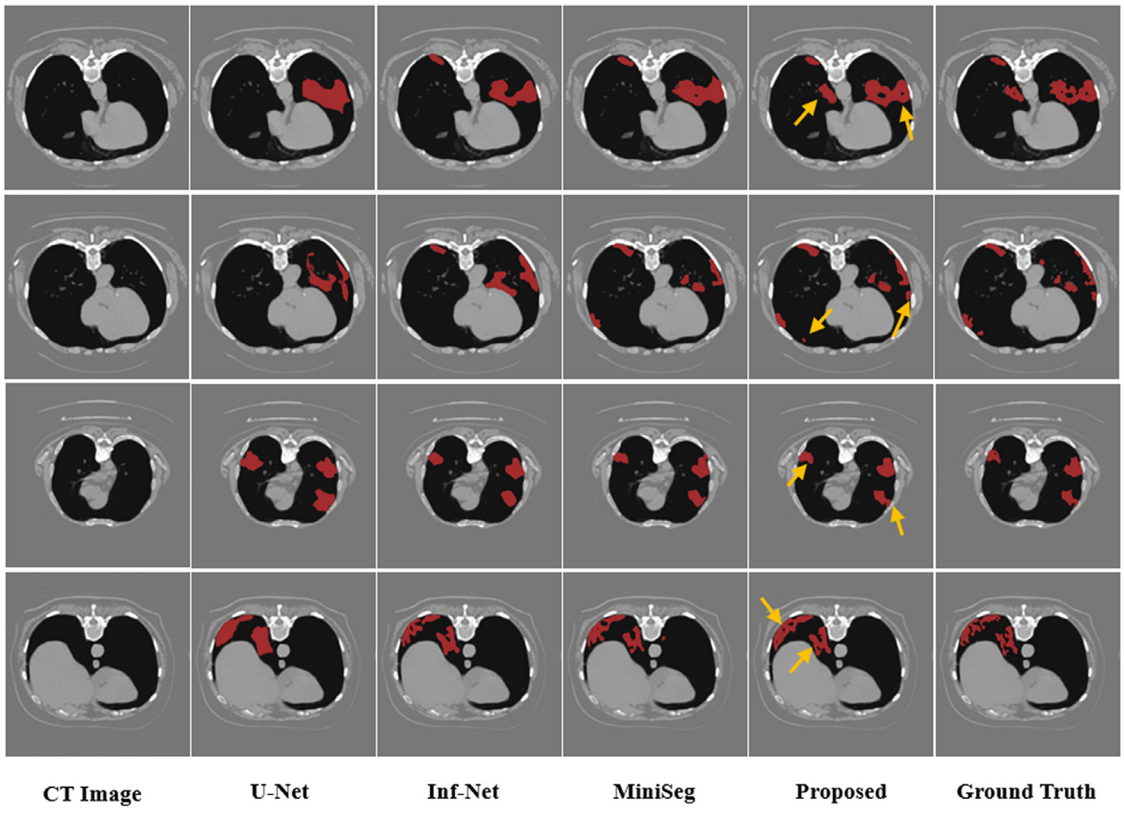

Fig. 7. Visual comparison of COVID-19 infection regions segmentation results on the COVID-19-1110 dataset, where the red labels indicate COVID-19 infection regions.

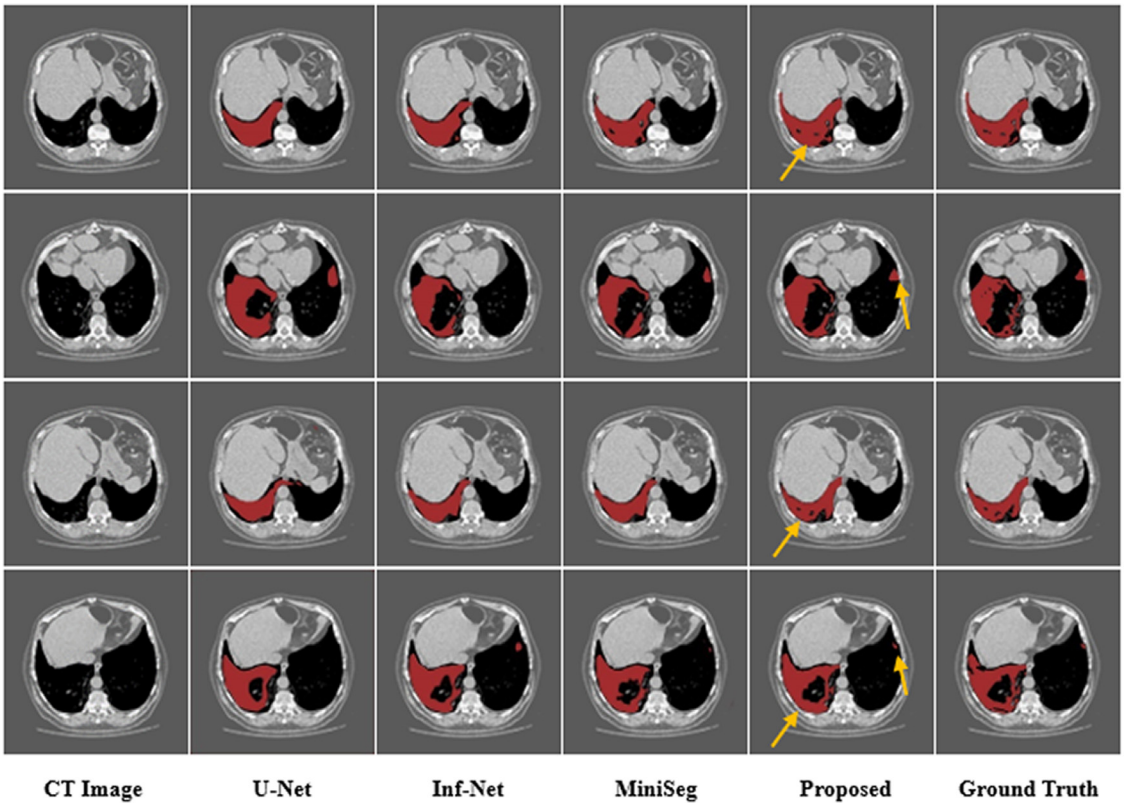

Fig. 8. Visual comparison of COVID-19 infection regions segmentation results on the COVID-19-9 dataset, where the red labels denote COVID-19 infection regions, and the yellow arrows highlight some segmentation details.

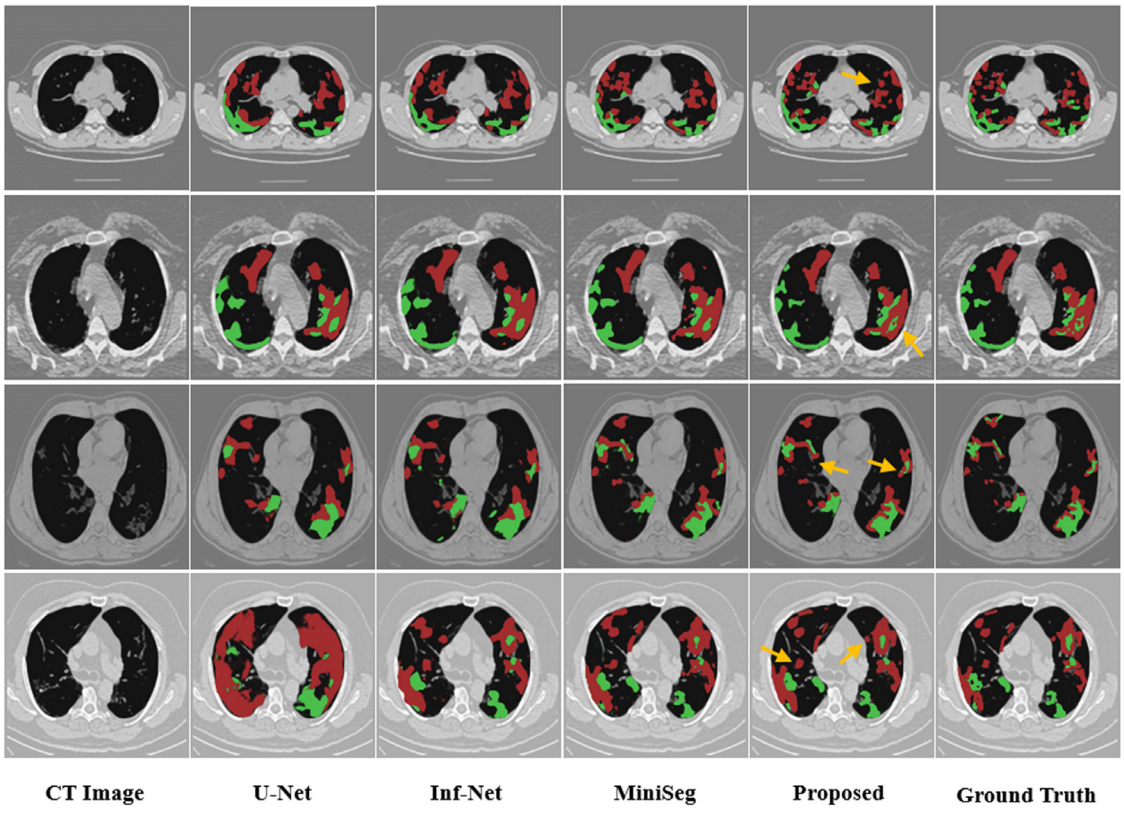

Fig. 9. Visual comparison of multi-class lung infection segmentation results, where the red and green labels indicate the GGO and consolidation, respectively.

Table 4
Quantitative results of COVID-19 infection regions on the COVID-19-9 dataset. The best results are shown in bold fonts.

	Methods	Dice	IoU	Sen	Spec	MAE	SM
MS-9 CT scans	Proposed	**0.7574**	**0.6095**	**0.8591**	0.9819	**0.0338**	**0.8242**
	MiniSeg	0.7353	0.5814	0.8497	0.9791	0.0386	0.7917
	Inf-Net	0.7248	0.5684	0.8417	**0.9857**	0.0429	0.7931
	U-Net	0.6874	0.5237	0.8172	0.9431	0.0673	0.7655

limitation of the single adversarial training objective and gradient vanishing in the other three ways, the proposed method performed better in some details (yellow arrows).

The quantitative results are shown in Table 4. The proposed method outperformed Inf-Net with 3.11% improvement in Structure Measure, which also denoted that the proposed method was more consistent with the human visual.

***(7) Results on MS COVID-19 dataset*:** In clinical settings, in addition to the overall evaluation, quantitative evaluation of different types of lung infections (such as GGO and consolidation) is also very important. Therefore, we extended the proposed method to multi-class lung infection labeling to provide richer information for further diagnosing and treating COVID-19. As shown in Fig. 9, the proposed method can more accurately segment GGO and consolidation infections than other methods. As can be observed, although both the proposed method and MiniSeg achieved promising performance, the proposed method obtained better results on some small lesion areas.

Table 5 shows the quantitative results on the MS COVID-19 dataset. The proposed method achieved the competitive performance on GGO segmentation in Dice, Sen, and MAE. Compared with MiniSeg on GGO segmentation, the proposed method improved the results from 76.93% to 78.21% in Dice. The result in terms of SM was also improved from 83.47% to 84.35%. For more challenging consolidation segmentation,

the proposed method achieved the best performance in Sen and MAE. MiniSeg only outperformed the proposed method with 0.42% improvement in Dice and 0.48% improvement in SM. It is worth noticing that the proposed method also achieved the best performance in terms of Dice, Sen, MAE, and SM on the average of segmentation results, which can further illustrate the effectiveness of the proposed method.

5. Conclusions

Segmentation of the infection lesions from CT volumes is essential for quantitative measurement of disease progression [42, 43]. In this paper, we proposed a new evolvable GAN segmentation framework for automatic COVID-19 infection segmentation. We focused on both the gradient vanishing problems and the limitation of the single adversarial training objective. The fitness function was designed to select the best offspring network

Table 5
Quantitative results of GGO, consolidation, and average on the MS COVID-19 dataset. The best results are shown in bold font.

Methods	Ground-glass opacity				Consolidation				Average			
	Dice	Sen	MAE	SM	Dice	Sen	MAE	SM	Dice	Sen	MAE	SM
Proposed	**0.7821**	**0.8812**	**0.0533**	**0.8435**	0.7417	**0.8534**	**0.0615**	0.8015	**0.7619**	**0.8729**	**0.0574**	**0.8225**
MiniSeg	0.7693	0.8654	0.0588	0.8347	**0.7459**	0.8528	0.0648	**0.8063**	0.7576	0.8591	0.0618	0.8205
Inf-Net	0.7626	0.8695	0.0614	0.8145	0.7222	0.8489	0.0674	0.7983	0.7424	0.8592	0.0644	0.8064
U-Net	0.7505	0.8551	0.0726	0.8094	0.7125	0.8431	0.0798	0.7928	0.7315	0.8491	0.0762	0.8011

after each iteration. We utilized gradient penalty to satisfy the 1-Lipschitz constraint to alleviate the instability issue. Four public COVID-19 CT scan datasets were employed for qualitative and quantitative analysis. Experiments show that our method can improve stability with high quality of segmentation results. Therefore, our work could help clinicians determine whether a patient is infected and accurately segment the infected area.

Furthermore, although the degree of infection is different in the CT slices from different angles, the difference between the slices is relatively small if they are from the same CT volume. Therefore, the CT slices from different angles may have a minor impact on the experimental results. In future studies, we will dedicate ourselves to reduce this impact.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Juanjuan He, Qi Zhu, and Piaoyao Yu were supported by the National Natural Science Foundation of China under Grant 61702383 and Kai Zhang was supported by the National Natural Science Foundation of China under Grant 62176191.

References

[1] B. Fang, G. Mei, X. Yuan, et al., Visual SLAM for robot navigation in healthcare facility, Pattern Recognit. 113 (2021) 107822.
[2] T.K. Dash, S. Mishra, G. Panda, et al., Detection of COVID-19 from speech signal using bio-inspired based cepstral features, Pattern Recognit. (2021) 107999.
[3] Z. Wang, Y. Xiao, Y. Li, et al., Automatically discriminating and localizing COVID-19 from community-acquired pneumonia on chest X-rays, Pattern Recognit. 110 (2021) 107613.
[4] A. Oulefki, S. Agaian, T. Trongtirakul, et al., Automatic COVID-19 lung infected region segmentation and measurement using CT-scans images, Pattern Recognit. 114 (2021) 107747.
[5] F. Shan, Y. Gao, J. Wang, Lung infection quantification of COVID-19 in CT images with deep learning, 2020, arXiv preprint arXiv:2003.04655.
[6] T. Zhou, S. Canu, S. Ruan, An automatic COVID-19 CT segmentation based on U-net with attention mechanism, 2020, arXiv preprint arXiv:2004.06673.
[7] D.P. Fan, T. Zhou, G.P. Ji, Inf-net: Automatic covid-19 lung infection segmentation from ct images, IEEE Trans. Med. Imaging 39 (8) (2020) 2626–2637.
[8] X. Chen, L. Yao, Y. Zhang, Residual attention u-net for automated multi-class segmentation of covid-19 chest ct images, 2020, arXiv preprint arXiv:2004.05645.
[9] Y. Qiu, Y. Liu, J. Xu, Miniseg: An extremely minimum network for efficient covid-19 segmentation, 2020, arXiv preprint arXiv:2004.09750.
[10] Z. Xu, Y. Cao, C. Jin, Gasnet: Weakly-supervised framework for COVID-19 lesion segmentation, 2020, arXiv preprint arXiv:2010.09456.
[11] G. Gaál, B. Maga, A. Lukács, Attention u-net based adversarial architectures for chest x-ray lung segmentation, 2020, arXiv preprint arXiv:2003.10304.
[12] H. Chen, Y. Jiang, H. Ko, Domain adaptation based COVID-19 CT lung infections segmentation network, 2020, arXiv preprint arXiv:2011.11242.
[13] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 214–223.
[14] T. Salimans, I. Goodfellow, W. Zaremba, Improved techniques for training gans, Adv. Neural Inf. Process. Syst. 29 (2016) 2234–2242.
[15] M. Arjovsky, L. Bottou, Towards principled methods for training generative adversarial networks, in: Proceedings of the International Conference on Learning Representations, 2017, arXiv preprint arXiv:1701.04862.
[16] I. Gulrajani, F. Ahmed, M. Arjovsky, Improved training of wasserstein gans, in: Advances in Neural Information Processing Systems, 2017, pp. 5767–5777.
[17] K. Wang, C. Gou, Y. Duan, Generative adversarial networks: introduction and outlook, IEEE/CAA J. Autom. Sin. 4 (4) (2017) 588–598.
[18] M.Y. Liu, O. Tuzel, Coupled generative adversarial networks, in: Advances in Neural Information Processing Systems, 2016, pp. 469–477.
[19] J. Bao, D. Chen, F. Wen, CVAE-GAN: fine-grained image generation through asymmetric training, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2745–2754.
[20] H. Wu, S. Zheng, J. Zhang, Gp-gan: Towards realistic high-resolution image blending, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 2487–2495.
[21] J.Y. Zhu, T. Park, P. Isola, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.
[22] X. Liang, L. Lee, W. Dai, Dual motion GAN for future-flow embedded video prediction, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1744–1752.
[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, Generative.adversarial. nets, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[24] X. Dong, Y. Lei, T. Wang, Automatic multiorgan segmentation in thorax CT images using U-net-GAN, Med. Phys. 46 (5) (2019) 2157–2168.
[25] Z. Han, B. Wei, A. Mercado, Spine-GAN: Semantic segmentation of multiple spinal structures, Med. Image Anal. 50 (2018) 23–35.
[26] S. Nema, A. Dudhane, S. Murala, Rescuenet: An unpaired GAN for brain tumor segmentation, Biomed. Signal Process. Control 55 (2020) 101641.
[27] X. Mao, Q. Li, H. Xie, Least squares generative adversarial networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2794–2802.
[28] C. Wang, C. Xu, X. Yao, Evolutionary generative adversarial networks, IEEE Trans. Evol. Comput. 23 (6) (2019) 921–934.
[29] Y.D. Fougerolle, J. Gielis, F. Truchetet, A robust evolutionary algorithm for the recovery of rational gielis curves, Pattern Recognit. 46 (8) (2013) 2078–2091.
[30] Y. Zheng, H. Fu, R. Li, Deep neural network oriented evolutionary parametric eye modeling, Pattern Recognit. (2020) 107755.
[31] K. Deb, A. Anand, D. Joshi, A computationally efficient evolutionary algorithm for real-parameter optimization, Evol. Comput. 10 (4) (2002) 371–395.
[32] S.R. Young, D.C. Rose, T.P. Karnowski, Optimizing deep learning hyper-parameters through an evolutionary algorithm, in: Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments, 2015, pp. 1–5.
[33] R. Miikkulainen, J. Liang, E. Meyerson, Evolving deep neural networks, in: Artificial Intelligence in the Age of Neural Networks and Brain Computing, Academic Press, 2019, pp. 293–312.
[34] S. Lander, Y. Shang, Evoae–a new evolutionary method for training autoencoders for deep learning networks, in: 2015 IEEE 39th Annual Computer Software and Applications Conference, IEEE, 2015, pp. 790–795.
[35] C. Frogner, C. Zhang, H. Mobahi, Learning with a wasserstein loss, Adv. Neural Inf. Process. Syst. 28 (2015) 2053–2061.
[36] M. Jun, G. Cheng, COVID-19 CT lung and infection segmentation, Dataset, v1.0, 2020. https://www.kaggle.com/andrewmvd/covid19-ct-scans.
[37] MosMedData: Chest CT scans with COVID-19 related findings, 2020. https://mosmed.ai/datasets/covid19_1110.
[38] H.B. Jenssen, Covid-19 radiology-data collection and preparation for artificial intelligence, 2020. http://medicalsegmentation.com/covid19.
[39] H.B. Jenssen, Covid-19 radiology-data collection and preparation for artificial intelligence, 2020. https://sirm.org/category/senza-categoria/covid-19/.
[40] O. Ronneberger, P. Fischer, T. Brox, Unet: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer Assisted Intervention, Springer, 2015, pp. 234–241.
[41] D.P. Fan, M.M. Cheng, Y. Liu, et al., Structure-measure: A new way to evaluate foreground maps, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 4548-4557.
[42] Mu Nan, Wang Hongyu, Zhang Yu, Jiang Jingfeng, Tang Jinshan, Progressive global perception and local polishing network for lung infection segmentation of covid-19 ct images, Pattern Recognit. 120 (2021) 108168.
[43] Liu Xiaoming, Yuan Quan, Yaozong Gao, Kelei He, Shuo Wang, Xiao Tang, Jinshan Tang, Dinggang Shen, Weakly supervised segmentation of covid19 infection with scribble annotation on ct images, Pattern Recognit. (2022) 108341.