Shallow Water Equation Simulation's Super-resolution Using GAN

ANONYMOUS AUTHOR(S)

In this paper, we follow [Xie et al. 2018]'s idea, trying to extend the generative adversarial learning into physics problems' super-resolution task. We extend the GAN based super-resolution for shallow water equation simulation and show that the original [Xie et al. 2018]'s method cannot handle this problem well, mainly because the training strategy, the neural network's trained degree of freedom(DOF) limitation and the discriminator loss are not optimal for shallow water equation simulation's super-resolution task. And we try to modify these three aspects to improve the GAN's ability for SWE simulation's super-resolution by considering the character of SWE simulation under some specified starting value condition.

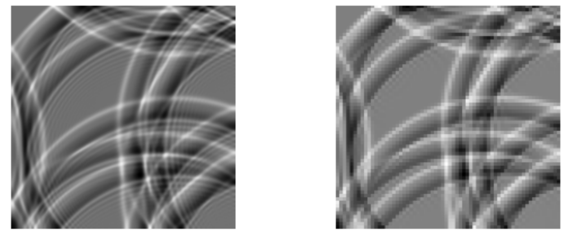
Additional Key Words and Phrases: physics-based deep learning, generative models, computer animation, fluid simulation

ACM Reference Format:
Anonymous Author(s). 2018. Shallow Water Equation Simulation's Super-resolution Using GAN. 1, 1 (December 2018), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the computer vision and machine learning field, the generative models have highly success for generating new images. But how to extend these generative models to other fields is not investigated sufficiently. In [Xie et al. 2018], they extend the GAN into the navier-stokes equation driven smoke density field's super resolution task and show that the modified GAN has some ability to preserve the temporal coherence of the generated physics sequence data. Following this, we investigate the ability of GAN for another physics process–shallow water equation simulation's super-resolution and show that the GAN's competency boundary to complete plausible mapping of low resolution data and high resolution counterparts. Different from smoke's super-resolution whose spatial continuity is not obvious, the SWE simulation sequence has more symmetric and visible ring cycle details in Fig. 1(a) which is difficult to recover. And the wave's strong interference phenomenon leads the difference between frames huge but the smoke simulation sequence does not have such obvious interference like SWE simulation. So how to recover such highly detailed but tiny feature of high resolution data from its low resolution counterparts is a challenge in super-resolution task, seeing in Fig. 1.

In [Xie et al. 2018], they use the 2D or 3D smoke simulation results as data to train or test with a modified GAN network. For example, in 2D, the input data is pairs of low resolution smoke data x (one density field, two unnecessary velocity fields) and its high resolution smoke density counterparts y. Firstly, they run numerical simulation with high resolution grid and get the high



(a) (b)

Fig. 1. **Ring cycle details of high resolution SWE simulation and its low resolution counterpart** (a) A frame of the high resolution SWE simulation results shows many symmetric and spatial continuous ring cycle details which is difficult for super-resolution because it is so tiny. (b) This frame's low resolution down sampled data. When showing the low-resolution input data, we always employ nearest neighbor up-sampling, in order to not make the input look unnecessarily bad.

resolution results(density field and velocity field) and do a gaussian blur with fixed sigma and next do a nearest neighbor interpolation to get its low resolution counterparts. The GAN's generator G will get input of low resolution x(one density field and two velocity fields) and give out high resolution fake density field $G(x)$ and a spatial discriminator D_s will judge whether it is real data y by numerical simulation or generator's faked results $G(x)$ and a tempo discriminator D_t will judge continuous three frames are real data $\widetilde{Y}$ or generator's faked results $\widetilde{G}(\widetilde{X})$ where the $\widetilde{X}$ represents the continuous three frames of low resolution data. This is the work of [Xie et al. 2018] mainly. In addition, they process the raw data with some rotation transformation to do a data augmentation. The modified GAN structure can be shown as below in Fig. 2.

1.1 Related Works

1.2 Contributions

2 LOSS FUNCTIONS

3 EXPERIMENTAL RESULTS

Firstly, we follow [Xie et al. 2018]'s idea to test its ability for shallow water equation simulation's super-resolution task.

Different from the 2D or 3D Navier-Stokes equation driven smoke simulation, the SWE gives simulation for water with a 2.5D status. In shallow water equation simulation, under the assumption that the vertical velocity component v_z is constant, the water will be modeled as a varying height field h as time passes. So the simulation data can be viewed as a height field time sequence and two velocity field time sequences for v_x and v_y. These data is the same as the original 2D smoke data in the form but essentially the height field can change without starting value limitation. It can be in [5,15] or [50,150] or even [500,1500] but the density field data is still bounded in [0,1]. This leads that the GAN network can not learn the arbitrary

XXXX-XXXX/2018/12-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

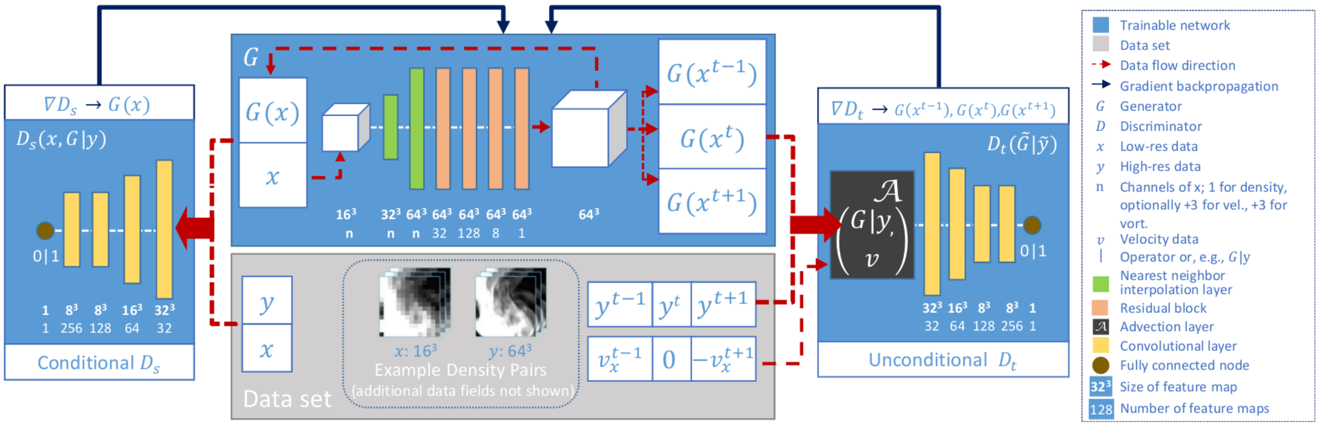


Fig. 2. **The tempoGAN structure's overview** The three neural networks (blue boxes) are trained in conjunction. The data flow between them is highlighted by the red and black arrows.

bounded value's height field's super resolution mapping or it is very difficult to learn! So we now simplify this question with these assumption as below and do our experiments and follow the Fig. 2 as our main super resolution task structure by replacing the density field with height field named as x for one frame of low resolution data (one height field and two velocity fields) and y for one frame of high resolution data (one height field) and $\widetilde{X}$ and $\widetilde{Y}$ for three frames of data respectively as well. So the loss functions can be summarized as:

$$Loss_{D_t}(D_t, G) = -E_m[logD_t(\widetilde{Y})] - E_n[log(1 - D_t(\widetilde{G}(\widetilde{X})))] \quad (1)$$

$$Loss_{D_s}(D_s, G) = -E_m[logD_s(x, y)] - E_n[log(1 - D_s(x, G(x)))] \quad (2)$$

$$Loss_G(D_s, D_t, G) = -E_n[logD_s(x, G(x))] - \lambda_{D_t} E_n[logD_t(\widetilde{G}(\widetilde{X}))]$$
$$+E_{n,j}\lambda_f^j ||F^j(G(x)) - F^j(y)||_2^2 + \lambda_{L_1} E_n||G(x) - y||_1 \quad (3)$$

We run shallow water equation simulation with fixed high resolution grid , time interval, gravity, fluid density and a fatten height field which height is 5 and we randomly give three points location and lift the starting height from 5 to 15 in these three points' location with a zero starting velocity field and run the simulation by standard MAC finite difference discretization. After getting the high resolution results, we do a gaussian blur with fixed sigma parameter and a nearest neighbor interpolation with a fixed factor 4 to get the low resolution data. These data is located in the folder: data velocity 2.

(1). We train the GAN with such five sequences which have 120 frames with [Xie et al. 2018]'s training method and test GAN on another non-trained sequence. We still get blurry output no matter how I do loss weight or learning rate or iteration number tuning for generator and discriminator, using full Eqs. (1) to (3) for training. Mainly like this in Fig. 3:

(2). In [Xie et al. 2018], they show that when use a negative layer loss weight λ_f^j, it gives sharp boundary feature preserving. But in

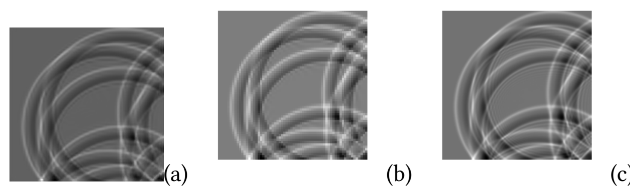


Fig. 3. **As the [Xie et al. 2018]'s training method, it tends to generate blur results** (a) A frame of tested height field which is generated by generator, showing that it loses the tiny but detailed feature. (b) The corresponding low resolution down sample height field input. (c) The corresponding high resolution height field as ground truth.

(1)'s training configuration, it shows that the training process may diverge easily,seeing Fig. 4, using full Eqs. (1) to (3) and training the GAN with such five sequences with a mini batch which size is 16 :

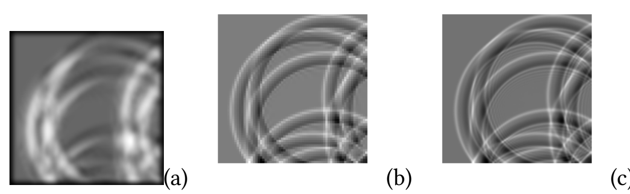


Fig. 4. **The negative layer loss weight with the [Xie et al. 2018]'s training method, it leads the GAN to diverge** (a) A frame of tested height field which is generated by generator after 50000 iterations training, showing such strange result and the large loss value meaning the training process diverged!!! (b) The corresponding low resolution down sample height field input. (c) The corresponding high resolution height field as ground truth.

(3). Under this dilemma, I want to check whether the GAN has no ability to recover tiny features or the training method leads it tends to be blurry because each trained mini batch gives such different gradient direction. So I simplify the train data with using only one frame's height and velocity data(unnecessary, we can train GAN without using velocity data) to test whether the GAN has the ability to recover the high resolution's ring cycle details. So we train the GAN with all training data– only one frame's data every time and do the same amount of iterations as above.

(3.1). Under this condition, we can see that if the input low resolution data includes velocity, it can give out more continuous faked data, using Eqs. (2) and (3) and let the λ_{D_t} equal 0 to abandon the generator's D_t term for training. Fig. 5:

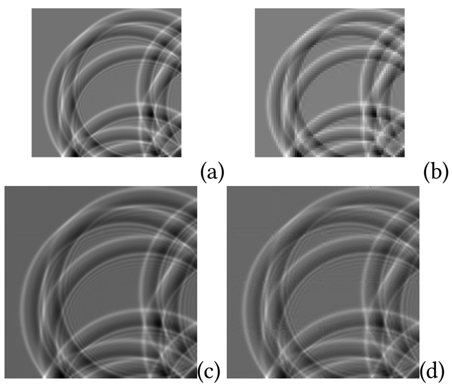


Fig. 5. **It shows that using velocity field leads more continuous and smooth results after the same amount of iterations** (a) The corresponding high resolution height field as ground truth.(b) The corresponding low resolution down sampled height field input. (c) The frame of tested height field generated by generator using velocity field for training, showing continuous and smooth results. (d) The frame of tested height field generated by generator using no velocity field for training, showing obvious blur artifacts.

(3.2). This frame data will highly decide what feature to be learned. And it shows that the GAN can learn the tiny details and when you test the GAN with another frame including the similar details, it will be recovered, using Eqs. (2) and (3) and let the λ_{D_t} equal 0 to abandon the generator's D_t term for training. Fig. 6:

(3.3). Further more, the convolution can not learn the feature's orientation which means that the learned feature dose not meet the rotation invariant, using Eqs. (2) and (3) and let the λ_{D_t} equal 0 to abandon the generator's D_t term for training. Fig. 7:

This can be solved by rotating the training data or use a rotation invariant CNN for generator. Maybe it can be a contribution, because of the high symmetry ring nature of SWE simulation, the rotation invariant should be satisfied plausibly.

(3.4). This setting leads that this GAN can not learn other feature, so when testing data dose not include the similar feature as the trained frame's data, it will not be recovered correctly, using Eqs. (2) and (3) and let the λ_{D_t} equal 0 to abandon the generator's D_t term ,seeing Fig.8(d), but if we use Eqs. (1) to (3) which means we use the temporal discriminator here for this single frame and

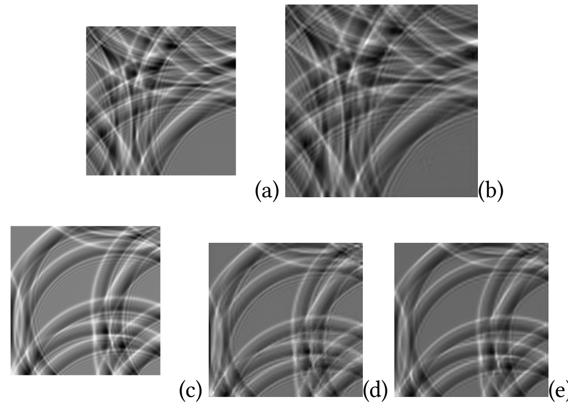


Fig. 6. **The GAN can learn thin details in this training configuration** (a) The corresponding high resolution height field as ground truth to be trained. (b) The trained frame generated by generator, showing very similar results as the high resolution ground truth. (c) Another frame high resolution height field to be tested. (d)In this training setting,the tested frame generated by generator showing tiny but detailed feature. (e) In the [Xie et al. 2018]'s training setting, the tested frame generated by generator showing blur artifacts with no detailed feature, using Eqs. (2) and (3) and let the λ_{D_t} equal 0 to abandon the generator's D_t term as well.

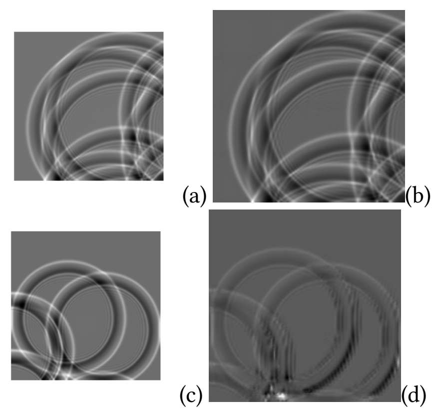


Fig. 7. **The CNN does not satisfy rotation invariant, so the learned feature is orientation dependent** (a) The corresponding high resolution height field as ground truth to be trained. (b)The trained frame generated by generator, showing very similar results as ground truth, but the details' orientaions are mainly pointing to the upper left. (c) Another frame high resolution height field to be tested. (d) The tested frame generated by generator, showing that the details pointing to the upper left are better learned than the details pointing to the lower right.

its adjacent two frames , the testing data's quality can be improved obviously,seeing Fig.8(e):

(3.5). We show that when we use only one frame's density and velocity data as generator's input and do not use the tempoGAN's tempo discriminator and only use the spatial discriminator, if the spatial discriminator does not include enough trained parameter DOF, the result will be only suboptimal, using Eqs. (2) and (3) and let the λ_{D_t} equal 0 to abandon the generator's D_t term. Fig. 9:

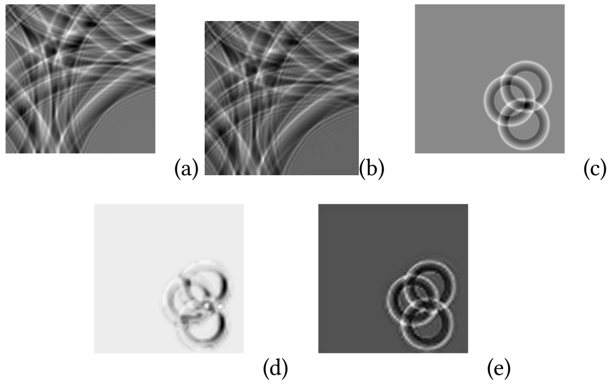


Fig. 8. **If the tested frame does not have the similar feature as the trained frame, the generated results are bad** (a) The corresponding high resolution height field as ground truth to be trained. (b) The trained frame generated by generator, showing very similar results as the high resolution ground truth. (c) Another frame high resolution height field to be tested. (d) The tested frame generated by generator without using temporal discriminator, showing that if the tested frame does not have the similar feature as the trained frame, the generated results are such bad. (e) The tested frame generated by generator with using temporal discriminator, showing that the generated result's quality can be improved obviously.

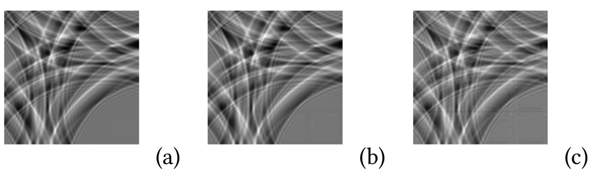


Fig. 9. **If we decrease the DOF of spatial discriminator, the result will be worse after the same amounts of iterations** (a) The corresponding high resolution height field as ground truth to be trained. (b) The trained frame generated by generator if we use the original spatial discriminator as [Xie et al. 2018].(c) The trained frame generated by generator if we slightly decrease the trained parameter DOF of the spatial discriminator, leading the generated result to have stronger block and blur artifacts, showing that increase the GAN's non-linearity may lead to better results.

(3.6). Our experiment's results also show that compared to using relu activation function for generator, it is better to use leaky relu activation function for generator to avoid the sparse feature selection because of the relu activation function. We check this with using Eqs. (2) and (3) and let the λ_{D_t} equal 0 to abandon the generator's D_t term. Fig. 10:

(3.7). For more, we think that for SWE super resolution task, the high resolution version's spatial gradient is important, so we try to penalize the l2 norm of the difference between the generated results' spatial gradient and the ground truth's spatial gradient, mathematically, which means that we add another term $\lambda_{L_2} E_n ||\nabla G(x) - \nabla y||_2^2$ into Eq. (3) and train the GAN with these new Eqs. (1) to (3). But unfortunately, the results with using spatial gradient penalty is blurrier than the results without using spatial gradient penalty. Fig. 11:

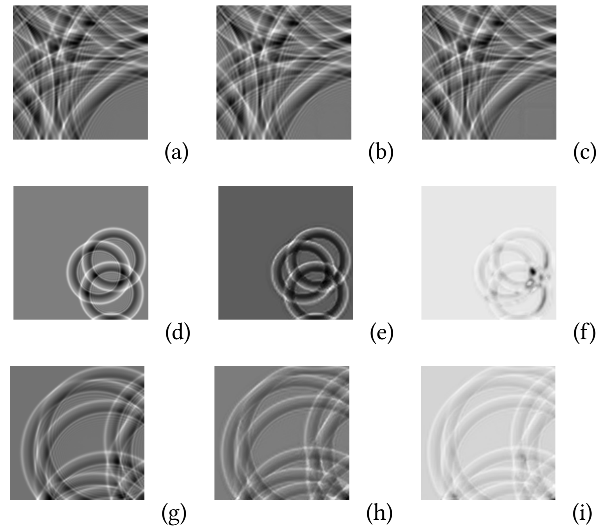


Fig. 10. **Compared to using relu activation function for generator, it is better to use leaky relu activation function for generator** (a) The corresponding high resolution height field as ground truth to be trained. (b) The trained frame generated by generator with using leaky relu as activation function. (c) The trained frame generated by generator with using relu as activation function. For (b) and (c), the difference is not obvious, but when we test it, the relu's results are worse obviously. (d) Another high resolution height field as ground truth to be tested. (e) The tested frame generated by generator with using leaky relu as activation function. (f) The tested frame generated by generator with using relu as activation function. (g) Another high resolution height field as ground truth to be tested. (h) The tested frame generated by generator with using leaky relu as activation function. (i) The tested frame generated by generator with using relu as activation function. Comparing the (e)and(f) or comparing (h)and(i), it shows the relu's generated results are worse than using leaky relu.

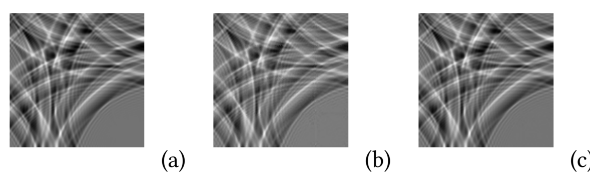


Fig. 11. **The** (a) The corresponding high resolution height field as ground truth to be trained. (b) The trained frame generated by generator with using gradient penalty,showing blurrier results than that without using gradient penalty. (c) The trained frame generated by generator without using gradient penalty.

(4). No matter the (3)'s training method or the (1)'s training method, it will only suit for the similar height field bounded by [5,15]. It can not be used when you change any boundary condition and parameter: gravity, fluid density or using other starting height field such as bounded by [10,20], using Eqs. (2) and (3) and let the λ_{D_t} equal 0 to abandon the generator's D_t term. Fig. 12:

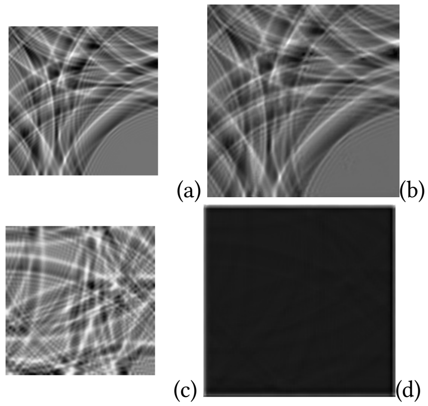


Fig. 12. **The trained GAN only suit for the similar starting height field condition** (a) The corresponding high resolution height field as ground truth to be trained. (b) The trained frame generated by generator, showing very similar results as the high resolution ground truth. (c) The corresponding high resolution height field as ground truth to be tested, but the tested starting height field of this frame is bounded by [10,20]. (d) The tested frame generated by generator is unusable.

4 LIMITATIONS AND FUTURE WORK

REFERENCES

You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. 2018. tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow. *arXiv preprint arXiv:1801.09710* (2018).