Shallow Water Equation Simulation's Super-resolution Using GAN

ANONYMOUS AUTHOR(S)

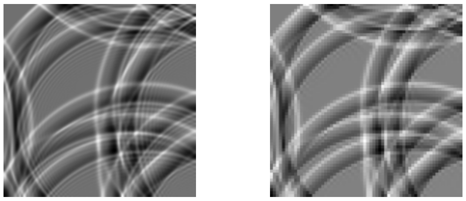
In this paper, we follow [Xie et al. 2018]'s idea, trying to extend the generative adversarial learning into physics problems' super-resolution task. We extend the GAN based super-resolution for shallow water equation simulation and show that the original [Xie et al. 2018]'s method cannot handle this 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). 2019. Shallow Water Equation Simulation's Super-resolution Using GAN. 1, 1 (January 2019), 7 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 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.



(a) hi-res ground truth (b) low-res down sampling

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.

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 channel for density field, two optional channels for velocity fields) and its high resolution smoke density counterparts y. Firstly, they run numerical simulation with high resolution grid and get the high 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 optional 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 PROBLEM STATEMENT

Shallow water equation (SWE) is a PDE which generates a set of height field over a flat domain to approximate the motion of fluid (wave, wake etc.). Different from the 2D or 3D Navier-Stokes equation driven smoke simulation, the SWE gives simulation for water in a domain Ω with a 2.5D status. In shallow water equation simulation, under the assumption that the vertical velocity component is constant, the water will be modeled as a varying height field $h(t) : \Omega \rightarrow \mathbb{R}$, and velocity field $u(t) : \Omega \rightarrow \mathbb{R}^2$ as time passes. As a consequence, the result of a SWE with specified boundary condition $\mathcal{B}$ can be viewed a sequence of 3-channel (one for height h, two for velocity u) images Y. We use $Y_*^k, * \in \{h, u\}$ to indicate the 2D "image" of channel $*$ at kth frame, where the Y_0 is the height channels and the Y_1 is the u_x velocity component channel and the Y_2 is the u_y velocity component channel.

Giving a high resolution result Y, the down-sampled low resolution counterpart is denoted as X. The problem to be addressed in this paper is to learn a function $\mathcal{F}$ that estimates the Y_0 from

XXXX-XXXX/2019/1-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

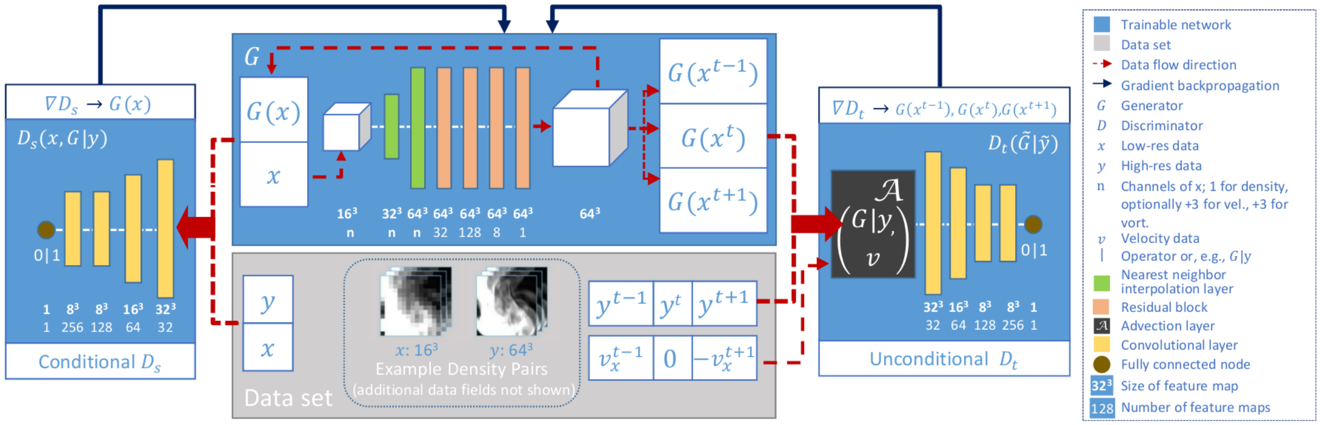


Fig. 2. **The tempoGAN structure's overview** The three neural networks (blue boxes) are trained in conjunction. The data flow between them is highlighted by the red and black arrows.

X. Because we only concern the high resolution height field data, we ignore the Y_1 and Y_2 and use Y to express the high resolution height field channel Y_0 directly in the following and keep the X to be X_0 which uses only the height field channel or $X_{0,1,2}$ which uses the height field channel and the two velocity channels together as generator G's input. In other words, we would like to find

$$\mathcal{F}(X) \approx Y, \tag{1}$$

where the ≈ indicates a properly defined Loss function (see below). It should be noticed that the above requirement is not equivalent to frame-by-frame estimation, i.e. we are not asking for a function

$$\mathcal{F}(X^k) \approx Y^k, \tag{2}$$

which cannot use temporal information in sequence X.

2.1 Loss Functions

Firstly, we follow [Xie et al. 2018]'s idea to test its ability for shallow water equation simulation's super-resolution task. In their work, the network structure shown in Fig. 2 is used. We replace the density field with height field named, and keep all the left the same.

In the following loss definition, $x, y, G(x)$ indicate a single frame in X and Y and the faked result to approximate the y respectively, and $\widetilde{X}$ and $\widetilde{Y}$ and $\widetilde{G}(\widetilde{X})$ are consecutive three frames in X and Y and the faked results to approximate the $\widetilde{Y}$ respectively. It is easy to see that the first loss

$$Loss_{D_s}(D_s, G) = -E_m[\log D_s(x,y)] - E_n[\log(1 - D_s(x, G(x)))] \tag{3}$$

only uses and measures the spatial information to update the spatial discriminator D_s. The second loss

$$Loss_{D_t}(D_t, G) = -E_m[\log D_t(\widetilde{Y})] - E_n[\log(1 - D_t(\widetilde{G}(\widetilde{X})))] \tag{4}$$

uses and measures temporal information of three consecutive frames to update the temporal discriminator D_t. The last loss is defined as

$$\begin{aligned} Loss_G(D_s, D_t, G) = &-E_n[\log D_s(x, G(x))] - \lambda_{D_t} E_n[\log D_t(\widetilde{G}(\widetilde{X}))] \\ &+ E_{n,j}\lambda_f^j ||F^j(G(x)) - F^j(y)||_2^2 + \lambda_{L_1} E_n ||G(x) - y||_1. \end{aligned} \tag{5}$$

to update the generator G.

We denote the group of above three loss with parameter $\lambda_{D_t} > 0$ as $\widetilde{L}$, which measures both spatial and temporal loss. A new group of above three loss is defined as the composition of L and for L, $Loss_G$ with parameter $\lambda_{D_t} = 0$ which only measures spatial loss, and further we need not to update the temporal discriminator with the temporal discriminator's loss Eq. (4). The corresponding result for an input x is written as $\widetilde{L}(x)$ and $L(x)$ respectively.

2.2 Data Set

Currently, to simplify the problem, we solve shallow water equation with standard MAC finite difference discretization with fixed high resolution grid, time interval, gravity, fluid density, uniform depth 5, and only left the boundary condition $\mathcal{B}$ to control the variety of the data set.

We noticed that although each channel is similar to a sequence of common gray images, or the 2D smoke data. However, in our data, the height field (and velocities) can be in arbitrary range instead of $[0,1]$ (or $[0,255]$). For instance, it can be in $[0,0.5]$ or even $[-500,1500]$. To alleviate this issue, we limited the boundary condition $\mathcal{B}$ as a flat height field and zero velocity, but only lift three random picked points from its rest height 5 to 15 at the first frame of the sequence. With such a setting of boundary condition, the high resolution height $h(t)$ is usually in range of $[4.5, 15]$ and the magnitude of velocity $|u(t)|$ is usually in range $[0.15, 3.2]$

After getting the high resolution results, we do a gaussian blur with fixed sigma parameter and a nearest neighbor interpolation

abb.	meaning
Hi	hi-res ground truth
Lo	low-res input
$\widetilde{L}$	spatial and temporal loss
L	spatial only loss

Table 1. Abbreviation

with a fixed factor 4 to get the low resolution data. The down sampling operator is denoted as $\mathcal{D}$.

3 CHALLENGES AND EXPERIMENTAL RESULTS

We made a series experiments and found that it is indeed a very challenge problem and cannot be easily solved although the data set has been greatly simplified.

3.1 Parameter Setting

In the following figures, we label each result using the words in Tab. 1.

We made the following experiments:

(1) Feature related
- Using velocity component or not, this also loss related (v).

(2) Loss related:
- Tuning the detail enhancement coefficients which means tuning some GAN's hyper-parameters like the loss weight or learning rate or iteration number tuning for generator and discriminator (d).
- Using temporal loss or not (t).
- Adding gradient loss or not (g).

(3) Net related
- Using different activation function (RELU or LRELU).
- Using different DOFs of the net (N or M layers).

(4) Data set related
- Sequence or single frame training set (s or f).
- Testing on different input.

So there are 7 independent settings. To avoid make all the combination of the above, we organize the experiments as follows:

- We first show that the state-of-the-art method ("+v+d+t-g+RELU+N+s") does not work regardless how the hyper-parameters are tuned. So, in the following experiments,we fix these hyper-parameters to be consistent which means "-d".
- Then, on experiments "±v-d-t-g+RELU+N+f" we show that using a single frame as the training set, velocity component is helpful. So we always toggle the velocity component on in the following experiments which means "+v".
- We also confirmed that temporal loss is useful on experiments "+v-d±t-g+RELU+N+f".
- We show that "+v-d-t-g+RELU+N+f" is better than "+v-d-t-g+RELU+M+f", so we always use N-layers the following experiments.
- We show that +g is not better than -g in "+v-d+t±g+RELU+N+f".
- Experiments "+v-d-t-g+(RELU|LRELU)+N+f" show than leaky relu (LRELU) is better than RELU. So we choose to use LRELU instead of using RELU in [Xie et al. 2018].

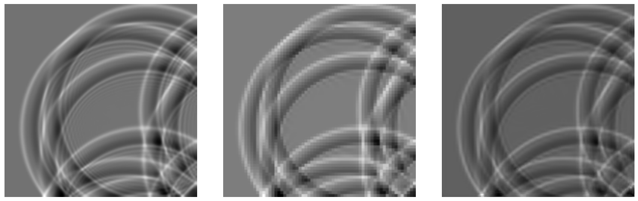
In summary, we have the table:

name	Better choice	Experiments	Fig.
vel (v)	+	"±v-d-t-g+RELU+N+f"	5
detail (d)	-	"+v+d+t-g+RELU+N+s"	3, 4
tempo (t)	+	"+v-d±t-g+RELU+N+f"	8
gradient (g)	-	"+v-d+t±g+RELU+N+f"	11
act. func.	LRELU	"+v-d-t-g+(RELU|LRELU)+N+f"	10
layers	N	"+v-d-t-g+RELU+(N|M)+f"	9

Table 2. Variants

Details of the above experiments can be found below:

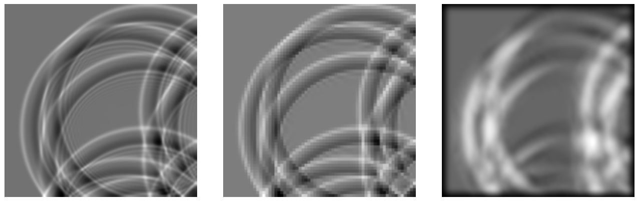
(1).**The results using [Xie et al. 2018]'s training strategy**
We train the GAN with such five sequences which have 120 frames with [Xie et al. 2018]'s training method which uses mini-batch to optimize the GAN and test GAN on another non-trained sequence. We still get blurry output no matter how I do loss weight or learning rate or iteration number tuning for generator and discriminator, using full Eqs. (3) to (5) for training. Mainly like this in Fig. 3:



(a) Hi y (b) Lo $x = \mathcal{D}(y)$ (c) $\widetilde{L}(x)$

Fig. 3. As the [Xie et al. 2018]'s training method, it tends to generate blur results.

(2).**The results using negative layer loss weight with [Xie et al. 2018]'s training strategy**
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 (1)'s training configuration, it shows that the training process may diverge easily,seeing Fig. 4, using full Eqs. (3) to (5) and training the GAN with such five sequences with a mini batch which size is 16 :

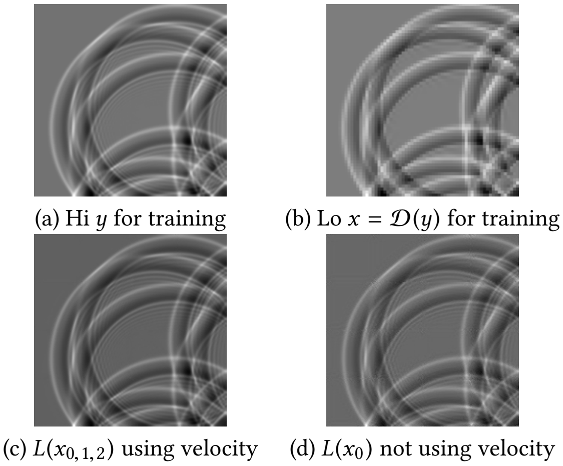


(a) Hi y (b) Lo $x = \mathcal{D}(y)$ (c) $\widetilde{L}(x)$

Fig. 4. The negative layer loss weight with the [Xie et al. 2018]'s training method, it leads the GAN to diverge as shown in (a)

(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 a different gradient direction that cannot approximate the global training set gradient direction well. 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).**The influence of whether using two velocity channels as input**
Under this condition, we can see that if the input low resolution data includes velocity, it can give out more continuous faked data, using L spatial only loss for training. Fig. 5:



(a) Hi y for training (b) Lo $x = \mathcal{D}(y)$ for training

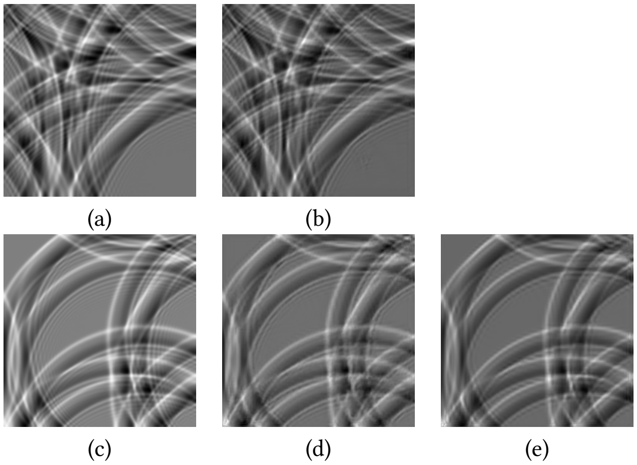
(c) $L(x_{0,1,2})$ using velocity (d) $L(x_0)$ not using velocity

Fig. 5. It shows that using velocity field leads more continuous and smooth results after the same amount of iterations

(3.2).**The influence of the selected frame as training set**
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 L spatial only loss for training. Fig. 6:

(3.3).**The CNN cannot learning the feature's orientation**
The convolution cannot learn the feature's orientation which means that the learned feature dose not meet the rotation invariant. Fig. 7:

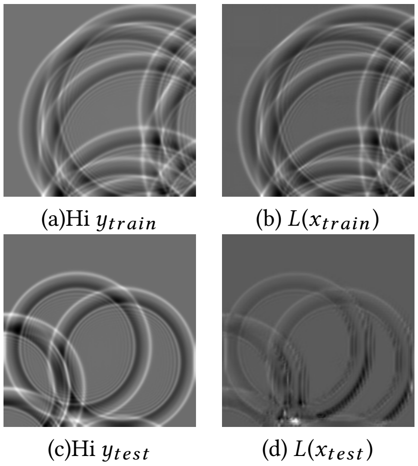
This can be solved by rotating the training data or use a rotation invariant CNN for generator. Maybe it can be a contribution, because of the high symmetry ring nature of SWE simulation, the rotation invariant should be satisfied plausibly.



(a) (b)

(c) (d) (e)

Fig. 6. **The GAN can learn thin details in this training configuration** (a) Hi y_{train} (b) $L(x_{train})$ (c)Hi y_{test} (d) $L(x_{test})$ in this training setting. In this training setting,the tested frame generated by generator showing tiny but detailed feature and (e) $L(x_{test})$ in [Xie et al. 2018]'s training setting. In the [Xie et al. 2018]'s training setting, the tested frame generated by generator showing blur artifacts with no detailed feature.



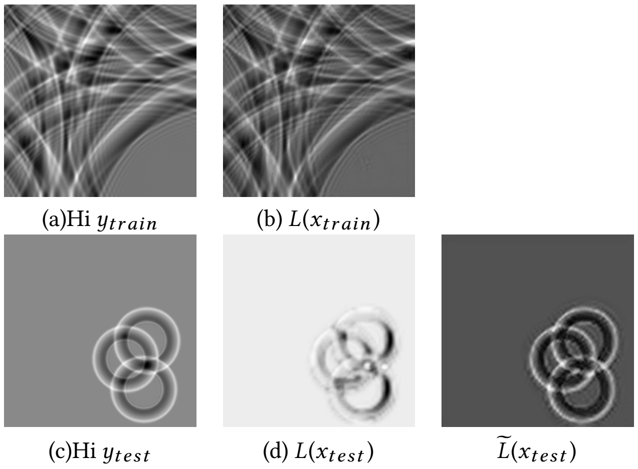
(a)Hi y_{train} (b) $L(x_{train})$

(c)Hi y_{test} (d) $L(x_{test})$

Fig. 7. **The CNN does not satisfy rotation invariant, so the learned feature is orientation dependent** as shown in (b) and (d) where (b) The trained frame generated by generator, showing very similar results as ground truth, but the details' orientation are mainly pointing to the upper left but (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.

(3.4).**The learned feature in this setting is limited and the influence of whether using the temporal discriminator**
This setting leads that this GAN cannot 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, seeing Fig.8(d), but if we use Eqs. (3) to (5) for this training frame and its two adjacent frames, the testing data's quality can be improved obviously, seeing Fig.8(e):

(3.5).**The influence of the trained DOF of the spatial discriminator**
We show that when we use only one frame's density and velocity

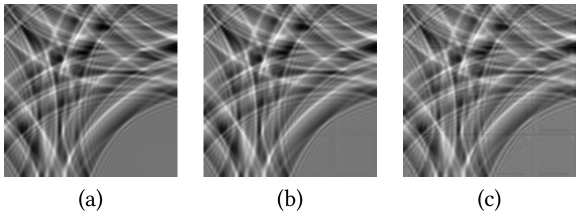


(a)Hi y_{train} (b) $L(x_{train})$

(c)Hi y_{test} (d) $L(x_{test})$ $\widetilde{L}(x_{test})$

Fig. 8. If the tested frame does not have the similar feature as the trained frame, the generated results are bad, but if we train the GAN using temporal discriminator for this training frame and its two adjacent frames, the testing data's quality can be improved obviously.

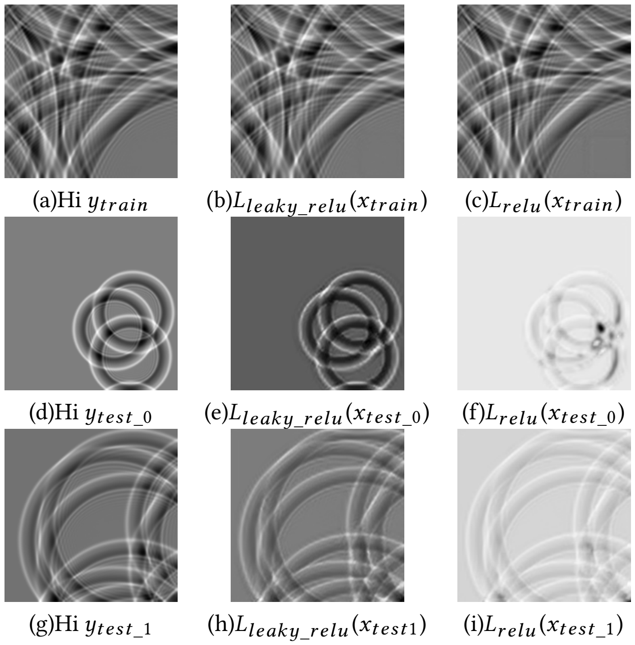
data as generator's input and do not use the tempoGAN's tempo discriminator and only use the spatial discriminator, if the spatial discriminator does not include enough trained parameter DOF, the result will be only suboptimal. Here we call the original number of discriminator's training DOF as N, and we decrease the size of the last convolution layer to be the half of N and this will lead the final full connect layer's trained DOF to be half of the the N and we call this new number of trained DOF for the spatial discriminator as M. Fig. 9:



(a) (b) (c)

Fig. 9. **If we decrease the DOF of spatial discriminator, the result will be worse after the same amounts of iterations** (a) Hi y_{train} (b) $L_{large}(x_{train})$. The trained frame generated by generator if we use the original spatial discriminator as [Xie et al. 2018].(c) $L_{small}(x_{train})$. 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).**The influence of whether using relu or leaky relu for generator's activation function**
Our experiment's results also show that compared to using relu activation function for generator in [Xie et al. 2018], 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 L spatial only loss for training. Fig. 10:



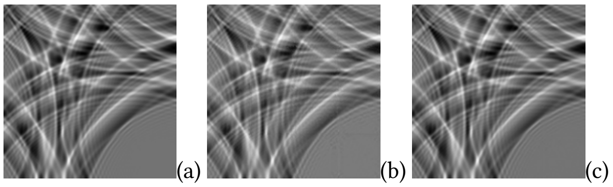
(a)Hi y_{train} (b)$L_{leaky\_relu}(x_{train})$ (c)$L_{relu}(x_{train})$

(d)Hi y_{test_0} (e)$L_{leaky\_relu}(x_{test\_0})$ (f)$L_{relu}(x_{test\_0})$

(g)Hi y_{test_1} (h)$L_{leaky\_relu}(x_{test1})$ (i)$L_{relu}(x_{test\_1})$

Fig. 10. **Compared to using relu activation function for generator, it is better to use leaky relu activation function for generator** For (b) and (c), the difference is not obvious, but when we test it, the relu's results are worse obviously. Comparing the (e)and(f) or comparing (h)and(i), it shows the relu's generated results are worse than using leaky relu.

(3.7).**The influence of adding a spatial gradient penalty to generator's loss**
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. (5) and train the GAN with these new Eqs. (3) to (5). But unfortunately, the results with using spatial gradient penalty is blurrier than the results without using spatial gradient penalty. Fig. 11:



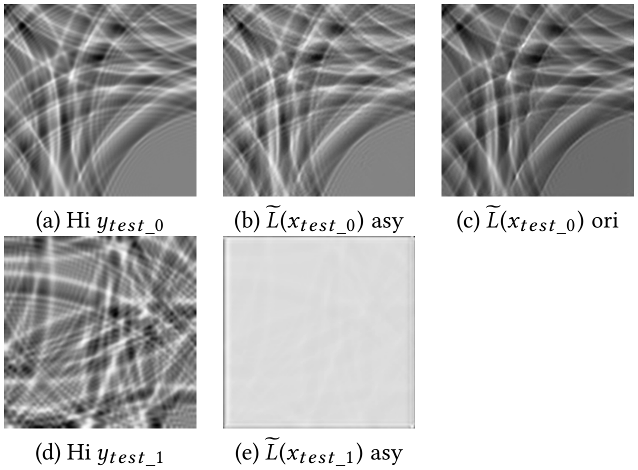
(a) (b) (c)

Fig. 11. (a) Hi y_{train} (b) $\widetilde{L_{grad}}(x_{train})$. The trained frame generated by generator with using gradient penalty,showing blurrier results than that without using gradient penalty. (c)$\widetilde{L}(x_{train})$. The trained frame generated by generator without using gradient penalty.

3.2 Asynchronous training strategy for details

So the current best combination is "+v-d+t-g+LRELU+N". But even on this setting, using sequence data as the [Xie et al. 2018]'s strategy (each time we choose a mini-batch from the training sets to optimize the GAN), the results are still blurry.

Inspired by our result trained by one frame which always use the global training set's gradient to optimize the GAN, We find that for SWE's super resolution task, if we want to capture sequence data's diverse but tiny features as much as possible, we need to optimize the GAN using the global training set's gradient for each time. But because of the limitation of the GPU memory currently, we use the gradient to optimize the GAN asynchronously which means that we separate the training set into some disjoint mini-batches and we compute the gradient for per mini-batch and accumulate these gradients into the global gradient to optimize the GAN once in one optimization iteration. This is useful for generating these tiny features and preventing the blur artifacts with [Xie et al. 2018]'s training strategy. see Fig. 12's (a),(b) and (c). We call this strategy as "asynchronously training".



(a) Hi y_{test_0}

(b) $\widetilde{L}(x_{test\_0})$ asy

(c) $\widetilde{L}(x_{test\_0})$ ori

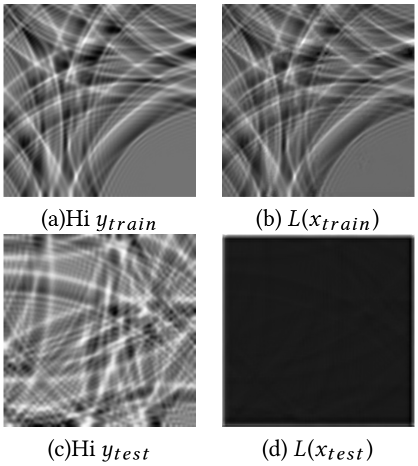
(d) Hi y_{test_1}

(e) $\widetilde{L}(x_{test\_1})$ asy

Fig. 12. The asynchronous training method can generate more features that original method as (b) and(c) shows.But the trained GAN still only suit for the similar starting height field condition as shown in (d) 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] and (e) The tested frame generated by generator is unusable.

Even with the asynchronously training strategy for better detail recovery, there are still problems about generality.

- **Rotation**: Although this asynchronous training strategy can solve the blur artifacts, which means that we can get more tiny features on training set's super resolution that cannot be accomplished by the [Xie et al. 2018]'s original mini-batch training strategy, we still have some other difficulty to apply the GAN to a new SWE sequence's super resolution because the CNN is not rotation invariant. If the training set's sampled frames are such different with the testing set, the testing set's results that is not captured by training set will still be blurrier than the part that is captured by training set. So we need to sample the training set uniformly: one method is to rotate the raw training set $90 * i(i = 1, 2, 3)$ degrees to augment the training sets that can capture the possible orientation of testing data at the most extent; another method is simpler than the first that we give the initial boundary condition central symmetrically and generate the training sets. With this uniform sampling, we get a suitable training set and then train the GAN with the asynchronous strategy, we can get much better results on the testing set's super resolution task.
- **Height and boundary conditions**: No matter the (3)'s training method or the (1)'s training method, it will only suit for the similar height field bounded by [4.5,15]. It cannot 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 using L spatial only loss for training. see Fig. 13. For more, we extend the (3)'s training method into the asynchronous training method for sequence training not a single frame, but when we modify the testing set's boundary condition like Fig. 13 and apply the GAN for it, we still have bad results. see Fig. 12's (d) and (e). So the no-bounded height field of SWE sequence is a challenge for super resolution.



(a)Hi y_{train}

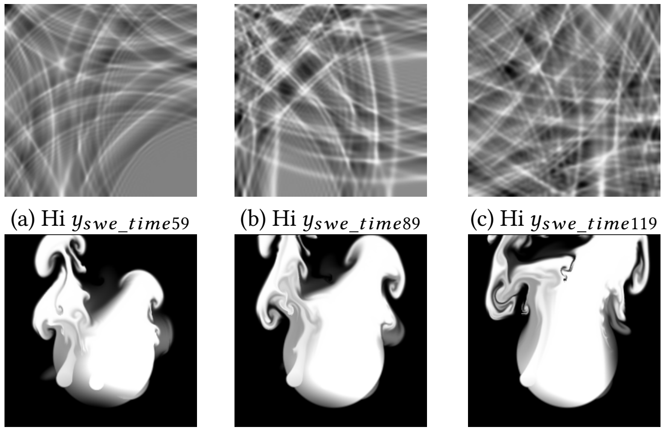
(b) $L(x_{train})$

(c)Hi y_{test}

(d) $L(x_{test})$

Fig. 13. The trained GAN only suit for the similar starting height field condition as shown in (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] and (d) The tested frame generated by generator is unusable.

- **Detail increasing as time going**: Even we assume our training and testing data are in a similar boundary condtion, the change of complexity of the data also bring challenges. Large variance of pattern is common in usual SWE results. Unlike smoke, which usually goes blurred and smoothed, SWE tends to generate more and more detailed vibrations and this vibration can be overlaid because of its interference/reflection phenomenon (see Fig. 14). In a word, as time passes, the information entropy of 2D N-S smoke sequence will not significantly increase but the information entropy of SWE sequence will be too large to be learned. So we think that the feature space becomes infinite and we cannot construct a finite training data set to capture these features. This property brings the difficulty of having good generalization ability.



(a) Hi y_{swe_time59} (b) Hi y_{swe_time89} (c) Hi $y_{swe\_time119}$

(d) Hi y_{smoke_time59} (e) Hi y_{smoke_time89} (f) Hi $y_{smoke\_time119}$

Fig. 14. (a), (b) and (c)show three frames of SWE sequence. (d), (e) and (f)show three frames of 2D-smoke density sequence. As we can see,as time passes, the interference leads more and more different features compared with the front frame. We can prove this interference can lead the numbers of features have exponential growth. But for 2D smoke sequence, the features to be learned are similar no matter for the front frame or the latter frame which are some local curl with no interference! This difference leads a huge challenge for SWE sequence's super resolution. Make an analogy, for a classification problem, give a pure eye image, you can classify it into the correct eye category, but when you overlay such 1000 eye images into one image, what is it and how can you classify it into eye or some other strange category?

4 LIMITATIONS AND FUTURE WORK

Except for that the no-bounded height field and the interference and overlaid details in SWE sequence bring the difficulty of having good generalization ability, for the asynchronous training method,the limitation is from the training time now. Although we can overcome the blur artifacts with the asynchronous training strategy, it need to have much longer training time than the mini batch based training because for one time updating of trained weight, we need to calculate the disjoint mini batch's gradient in sequence and add all once. So in a no parallel manner, the training time cost will be $[training\ set's\ size/mini - batch's\ size]$ times as much as the original training strategy's cost if we want to make the weighting updating times be the same. It can be alleviated by increasing the memory of GPU or computing the per disjoint mini-batch's gradient distributedly and parallelly.

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).