Do We Need a New Foundation to Use Deep Learning to Monitor Weld Penetration?

Edison Mucllari, Rui Yu, Yue Cao, Qiang Ye, YuMing Zhang, *Fellow, IEEE*

***Abstract* – Deep learning has been successfully used to automate the modeling process that trains a network/model from a given experimental dataset to calculate the output directly using high-dimensional complex raw data. However, the trained network is an inverse of the welding process (forward process) that produces the welding phenomena/measured raw data as the output with the penetration as the input of the forward process. Now the question is in addition to the current state of the weld penetration to be estimated if the forward process also has other inputs to determine its output. If it has, then the inverse model has to be constructed accordingly. This will call for a new foundation for deep learning-based monitoring of penetration. This letter proposed a novel innovative generative adversarial network (GAN) with GRU (Gated Recurrent Unit) in the generator, i.e., GRU-GAN, to model the extremely complex forward process to generate the observed topside welding image (output of the forward process) from the backside images (as comprehensive quantification of weld penetration). It is found that the produced topside welding image is not only determined by the current backside image but also by its history. A new foundation thus must be established to guide deep learning-based monitoring of weld penetration. The prediction model/network as an inverse model must be in compliance with the forward process that includes the history of the state of the weld penetration as its input.**

Keywords: weld, weld penetration, deep learning, GAN, GRU

I. Introduction

This letter aims at answering a basic question related to monitoring the weld penetration using a deep learning approach. As the weld penetration currently can only be assured by skilled human welders, the question is fundamental in robotizing/automating welding processes for challenging applications. Analysis of literature suggests that almost all, if not all, of efforts in the area share a similarity: using the measured welding phenomena as the raw information source/sources to correlate to the weld penetration [1]. Our question is if this has been based on a solid foundation and if its foundation has flaws. Answering this question will guide new directions, or call for establishing new and more solid foundations to solve this long-standing technical challenge, possibly in various ways.

Welding joins two members of materials together by melting their facing edges and weld penetration quantifies this melting, typically either by the weld pool depth (penetration depth) for incomplete/partial penetration or by the backside bead width for complete/full penetration (Fig. 1). As the penetration is unobservable occurring underneath, almost all existing methods, if not all of them, are to find promising measurable information sources and correlate their "right features" to the penetration. Various measurables have been proposed as promising raw information sources [1] including pool oscillation [2], infrared images [3], acoustic signals [4], weld pool [5], weld pool geometrical appearance [6], 3D weld pool surface [7]. While conventional methods have been based on features hand-crafted from the raw information sources resulting in an un-automated, trial-and-error process consisting of a number of separate sub-processes, deep learning provides a revolutionary solution by optimizing the features to allow automating the entire process. Since the occurrence of the first peer reviewed publication [8] in 2019, there have been approximately 100 records in Web of Science in Deep Learning AND weld AND penetration by the end of year 2022 [1].

Deep learning revolutionized the area of weld penetration monitoring by providing a way to directly link the raw information/sources to the weld penetration so that their relationship/mapping can be obtained through an automated process. This revolution is due to the development of deep learning but our efforts in this area have been in the application domain. They used raw information sources from previous physics/mechanisms-based studies as represented in [2-7]. Improved methods to measure raw information sources may have been used and major contributions have been limited to use different deep learning networks to accommodate different raw information sources/multiple information sources [9-15], as inputs of the networks whose outputs are the weld penetration state.

This letter will hypothesize that using raw information to directly link the penetration is not flawless and test this hypothesis to call the need for new foundations. Section II will provide and analyze the background to propose the hypothesis. Section III will outline how the hypothesis will be tested and Section IV is devoted to experiments in particular for what are the actual dataset. Section V proposes a generative adversarial network (GAN) to model the forward

Manuscript received: February XX, 2023; Revised: XX XX, 2023; Accepted: XX XX, 2023.

This paper was recommended for publication by Editor AAA AAA upon evaluation of the Associate Editor and Reviewers' comments.

Edison Mucllari, Rui Yu, Yue Cao, Qiang Ye, and YuMing Zhang are with the University of Kentucky, Lexington, KY 40506. (corresponding author: phone: 859-323-3262; e-mail: yuming.zhang@uky.edu). Edison Mucllari and Qing Ye are also with the Department of Mathematics, and Rui Yu, Yue Cao and YuMing Zhang are also with the Department of Electrical and Computer Engineering and Institute for Sustainable Manufacturing, University of Kentucky, Lexington, KY 40506.

This work is partially supported by the National Science Foundation under Grants No. 2024614 and DMS-2208314, the Institute for Sustainable Manufacturing, Department of Electrical and Computer Engineering, and Department of Mathematics at the University of Kentucky.

Digital Object Identifier (DOI): see top of this page.

process that generates the topside weld pool image as the "observed welding phenomena". In Section VI Results, Discussion and Improvement, we show that the current state of the weld penetration is not sufficient in producing the topside image, but the sufficiency is improved by including the history of the penetration state. We also proposed an improved network to model the forward process by better using the history. Conclusions are summarized in Section VII.

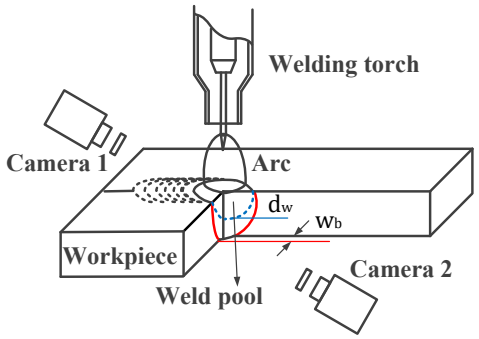

Figure 1: Illustration of weld penetration. Blue dash: weld pool of incomplete penetration with depth d_w; red solid: weld pool of complete penetration with backside bead width w_b.

II. Background and Hypothesis

Fig. 1 can be used as an example to illustrate the weld penetration. (More comprehensive introduction on weld penetration topic is presented in [16].) The autogenous gas tungsten arc welding (GTAW) in the figure is for illustration purpose and other processes can be used. The weld pool with red lines as the boundary shows complete joint penetration (also referred to as full penetration) with a backside weld bead width w_b. If the weld pool has the blue dashes as the boundary, the workpiece is not completely melted to the bottom surface forming an incomplete joint penetration (also referred to as partial penetration) being characterized by d_w. w_b and d_w are critical for the respective penetration case: being less than designed/desired results in unacceptable defect and being larger than designed/desired results in increases in materials properties degradation, heat input, residual stress and distortion [16]. The problem of penetration monitoring to be solved by research community is to use sensors, referred to as top sensors, that can be carried by the torch to obtain process measurables to estimate w_b or d_w, rather than using sensors that may directly view them. We use the state of penetration x for either w_b or d_w depending on the application.

Top sensors obtain process feedback or observed phenomena Ξ (which may be complex) which are believed to have sufficient information to determine the penetration x per previous physics/mechanism-based studies. Existing efforts train a model $x = f(\Xi)$. Earlier and most efforts in deep learning fit $x(k) = f(\Xi(k))$ and some fit $x(k) = f(\Xi(j)'s)$ $(j \le k)$ [15]. This letter questions the soundness of the foundation for $x(k) = f(\Xi(k))$ based approaches.

First of all, Ξ are behaviors of the welding process. As welding process is complex, there has been no single study to prove theoretically that particular Ξ being used may have sufficient information to determine the penetration. The "proofs" reported are based on experimental data and are not real. Second, an estimation model $x(k) = f(\Xi(k))$ may be considered an inverse of the process model $\Xi(k) = g(x(k))$. The estimation model $x(k) = f(\Xi(k))$ may be reasonable only if the process model $\Xi(k) = g(x(k))$ is reasonable.

The process model is more fundamental than an estimation model in catching the mechanism and essence. We question the process model $\Xi(k) = g(x(k))$ from two angles: (1) if there are other factors η in addition to the penetration state x that also affect the welding process and thus affect part of its observation Ξ; (2) even if there are no other factors such as in a condition that is close to the nominal one, it is a question if $\Xi(k)$ is only determined by the current state $x(k)$ of the penetration without being affected by its history $x(j)(j < k)$. A definite answer to either question will justify the need for a new foundation. As such, this letter focuses, and can just focus, on the second question although the first question also deserves attention (but it is more application dependent while the second question has more general implications independently from applications).

In a process model $\Xi(k) = g(x(k))$, x is the cause and Ξ is the consequence. This is a causal process where x occurs first and can be considered the input of a system with Ξ as its output. It is apparent that this system must be dynamic as there is apparently a transition time from the penetration to affect the process behaviors. It is likely that this transition is not a simple, ideal delay but a gradual process. As such, a penetration state affects the process behaviors not just in one instant but in a period. This suggests that $\Xi(k) = g(x(j)'s)(j \le k)$, or $\Xi(k) = g(X(k))$ where $X(k) \in R^n$ is formed by $x(j)$ $(k - n \le j \le k)$, must be in general more reasonable than $\Xi(k) = g(x(k))$. This is the hypothesis, that calls for a new foundation for deep learning-based monitoring of weld penetration, to be tested in this letter.

III. Proposed Method of Test

While there may be other ways to test the hypothesis concerned, this letter will test it by building networks that use x to generate Ξ to see if the generated Ξ, denoted as $\hat{\Xi}$, can well match with the measured Ξ. That is, the network will function as the mapping g from x to Ξ as a model of the underlying physical welding process with the specific concern for its partial behaviors represented by Ξ. Theoretically, if the network structure is right (determined by network type and the number of adjustable parameters) and the training dataset is large and diverse enough, we will be able to catch the deterministic correlation from x to Ξ. The modeling errors (training and test) reflect either model structure insufficiency, inaccuracy of the input (x) in determining Ξ and the effect from other uncontrolled factors but not the dataset's diversity that affects the error in application of the trained network. The average error as reflected by the variance will reflect the former two. If we have the same datasets and similar model structure to compare $\Xi(k) = g(x(k))$ with $\Xi(k) = g(X(k))$, we will be able to see if $x(k)$ alone is sufficient and if there is a need to include some history of x, i.e., $x(j)(j < k)$.

In most deep learning applications, a complex/high dimension signal such as Ξ is used as the input of a network

which outputs a low-dimension quantity such as x. A variant of convolutional neural network (CNN) is the typical choice for such applications. For this application to test our hypothesis, we need generating the high dimension phenomena Ξ, which is a kind of image from the weld pool, from the lower dimension input x as shown in Fig. 2 which illustrates the proposed network to test the hypothesis. In particular in Fig. 2, $I(x(k))$ and $I(X(k))$ are the raw information to calculate respective x and can provide more comprehensive information on the weld penetration state than the simplified quantity x (w_b or d_w). The dimensions of the extracted features from $I(x(k))$ and $I(X(k))$, i.e., $\varphi(x(k))$ and $\varphi(X(k))$, are the same, i.e., $D\left(\varphi(x(k))\right) = D(\varphi(X(k))$. The Generator Network generates $\Xi(k)$ from $\varphi(x(k))$ or $\varphi(X(k))$. Its structure MUST be the same despite the input, either $\varphi(x(k))$ or $\varphi(X(k))$, for a fair comparison but the network will be trained separately. The structures for the Featuring Networks for the two kinds of inputs, $I(x(k))$ and $I(X(k))$, cannot be exactly the same but should be similar. The Featuring Networks will also be trained separately. As will be discussed below, $I(x(k))$ and $I(X(k))$ will be images captured from the backside of the workpieces during welding. The Featuring Network thus should be a variant of CNN. The Generator Network should be a GAN. The details of the network design will be discussed later.

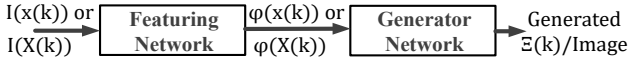

Figure 2: Proposed network structure to test the hypothesis.

IV. Experiments and Dataset

Gas tungsten arc welding (GTAW) is the primary process used to produce welds with assured complete penetration required in most critical applications. In this letter, an experimental system at the University of Kentucky Welding Research Laboratory is used to conduct GTAW experiments [15]. This experiment system has two cameras (Fig. 1) with Camera 1 to view the weld pool as Ξ (topside image) and Camera 2 to view the backside of the work piece as $I(x)$ (backside image) from which x can be calculated as in [15]. The backside image provides complete raw information for the penetration status and x as defined in this letter is just a simplified measure. Using the backside image as the input to represent the weld penetration to test the hypothesis is thus more accurate and appropriate. Fig. 3 shows a pair of topside and backside images captured at the same time. The topside camera (Camera 1) is an Xiris camera of XVC-1100, a high dynamic range camera designed to view melt pools despite the presence of strong radiation from arc, laser, metal vapors, etc. The back-side camera (Camera 2) is a regular CCD because of the absence of strong radiation. The cameras capture the images simultaneously from both sides of the workpieces at 60 Hz.

To generate a meaningful dataset, four experiments have been conducted using real time randomly varying welding current and welding speed, the two most important welding parameters determining the weld penetration, to generate a diverse set of highly dynamic weld penetration x/back-side image $I(x)$ and topside images (process phenomena Ξ). In the experiments, the welding current varied from 70A to 130A and speed from 1.2mm/s to 2mm/s. With them being changed each 2 seconds randomly, conditions were created to generate highly dynamic and "extremely difficult to predict" topside images/welding behaviors. The captured images from all the experiments allow us to obtain dataset $(I(x(k)), \Xi(k))$'s and $(I(X(k), \Xi(k))$'s. 4,760 pairs can be collected from each experiment for both datasets to have the size for datasets as 4,760*4=19,040. After removing the data from the beginning of each experiment, the size of the datasets being actually used is 18,567. The data is then split into training and testing, where 80% corresponds to the training data and 20% is for the test.

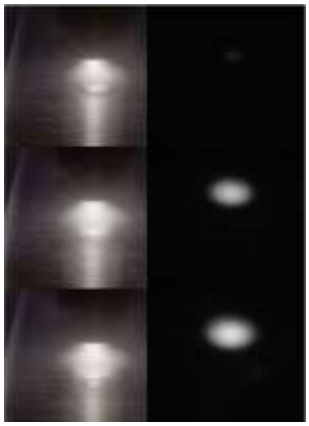
Figure 3: Paired topside (left) and backside (right) images.

V. Welding Generative Adversarial Networks

This work models the forward/welding process by Generative Adversarial Networks (GANs) [17]. Since their introduction in 2014, GANs have seen a wide range of applications. The proposed neural network incorporates the usage of a minimax type architecture in between a generative model (generator) and a discriminative model (discriminator). GANs take as an input a vector z from a random distribution and the task is to approximate the distribution of the real topside images by using backpropagation to train the generator and discriminator. The discriminator takes inputs from both the real images and the generated images, referred to as fake images, and it is trained to distinguish real images from the fake ones. The stronger the discriminator is, the better the generator has to become to be able to fool the discriminator. There has been tremendous work in improving the generator and discriminator of GANs due to the training instability. For image generation, Deep Convolutional Generative Adversarial Networks (DCGANs) [18] successfully introduced convolutions into GANs and they provide some architectural constraints and also the most effective hyperparameters they retrieve from their experiments to ensure a more stable training.

We have integrated several features and hyperparameters of DCGAN into our model. These hyperparameters are as follows: Instead of using pooling layers as it is usually suggested in CNN models, in our architecture, we implement strided convolutions for the discriminator and fractional-

strided convolutions for the generator as in DCGANs. Our model consists of only convolution layers, since we avoid the fully-connected layers, as mentioned in [18]. Applying batch normalization in every layer improves the performance and the corresponding activation functions for the generator and discriminator are ReLU and Leaky ReLU (with 0.2), respectively. Furthermore, the output of the generator uses tanh activation and the output of the discriminator uses a sigmoid activation function. All weights are initialized from a Normal Distribution with mean 0 and standard deviation 0.02. The learning rate is set to $3 * 10{-}4$ with Adam optimizer [19] and the model is trained for 200 epochs.

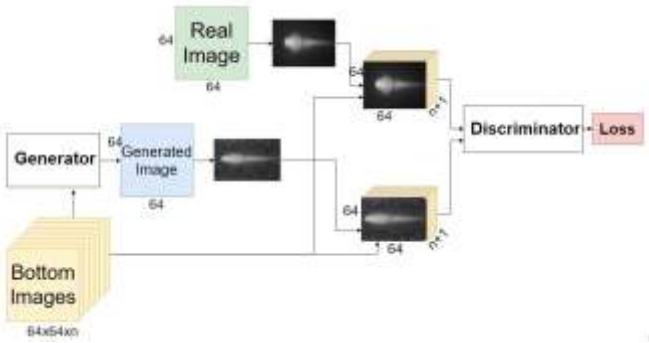

Figure 4: Welding GAN. n: the number of bottom images used.

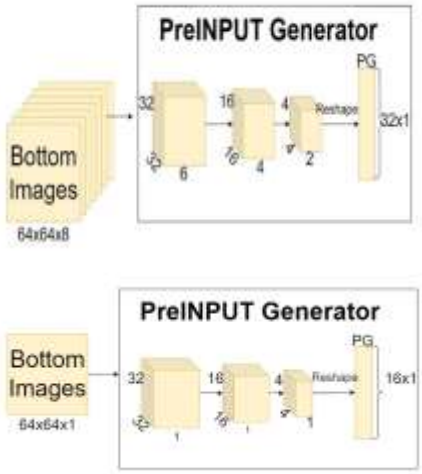

Figure 5: Pre-input generator architecture. Upper: Using eight bottom images; lower: Using a single bottom image

The proposed GAN model for this work is illustrated in Fig. 4. We require our model to have two properties. First, we need the generator to be able to generate images with a similar probability distribution as the top images by incorporating the bottom/backside ones. Second, the generated image should be as close as possible to the corresponding real topside image. We propose using a number (n) of most recent consecutive bottom images, i.e., $X(k) \in R^n$, to generate the topside image $\Xi(k)$. Thus, we incorporate the usage of Conditional Generative Adversarial Nets [20]. Using a similar idea as in Conditional GANs, we perform conditioning by feeding $X(k)$, as well as the vector z, which comes from a random distribution. Since we need our generated image $\hat{\Xi}(k)$ and $\Xi(k)$ to be as close as possible, the implemented generator loss function considers the generator loss and the loss due to $\| \hat{\Xi}(k) - \Xi(k) \|$. Our proposed model has more of a semi-supervised learning architecture, since we introduce $\| \hat{\Xi}(k) - \Xi(k) \|$ based loss function as well to enforce the newly generated images to be similar to the experimental images.

As has been mentioned, we condition the model based on n consecutive bottom images. The hyperparameter n affects the capability of our proposed model and there should be an optimal n^* where our model performs the best. However, to test the hypothesis, any n that shows a significant improvement in the model accuracy would be sufficient. An optimal n is thus not necessary and searching for such optimal n^* dilutes the focus on the new concept. As such, we use n=8 to count for a dynamic period which is slightly over 0.1 second (8/60=0.13 second). As such, we need a pre-input generator as illustrated in Fig. 5. First, we stack all eight bottom images to derive a single tensor with dimension 64x64x8 (where 8 corresponds to the number of channels) (Fig. 5(a)). We apply a three-layer Convolution Neural Network (CNN) to extract more features from the concatenated images before we feed it into the GAN generator. The kernel size and the stride for the first two layers are the same and they are set to 4 and 2. The third CNN layer has a kernel size of 6 and a stride of 4. Moreover, the padding size is 0 for all layers. The output of the CNN model is then reshaped into a 32×1 (16×1) vector as part of the generator input. The pre-input generator for using one backside image is illustrated in Fig. 5(b).

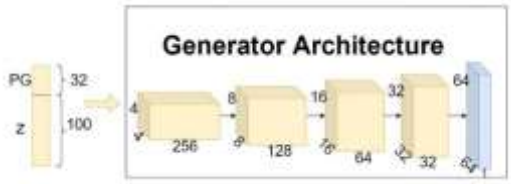

Figure 6: Generator architecture. PG: output vector from the pre-input generator.

The outputted 32×1 (16×1) vector will be concatenated with a randomly distributed 100×1 vector z together to form the input of the generator that is illustrated in Fig. 6. In our case, the GAN uses eight backside images to generate a top image. These eight images are considered model input in conventional definition but in GAN, including its advanced Conditional GAN, they are referred to as the condition and form GAN's input together with a random vector z as the noise [21]. The importance of the random vector z as an input to GAN is well acknowledged [21]; that is, without z, GAN may still learn from the condition but it would produce deterministic outputs so that it would fail to match any distribution other than a delta function [22]. In different GAN architectures there are different ways to introduce random noise. In our work, we use Gaussian vector z. The generator has five layers where strided convolutions as in DCGANs [18] are applied to increase the dimension to 64x64x1. The channel dimension for each layer is 256, 128, 64, 32 and 1. Since the image dimension coming from the real data is 64x64x1, we need to have the same dimension for the generated images as well. Batch normalization is applied in every CNN layer except for the input and output.

For the discriminator, we also use the same bottom images as the condition as adapted from Conditional GANs [20]. We

feed the same eight bottom images into the input discriminator by concatenating it with the real top image into a single tensor of dimension 64x64x9. The same is applied to the generated images when fed into the discriminator. The discriminator CNN model has four hidden layers and an output with a single element. The kernel size for all convolutional layers is 4 and the stride is 2, except for that the last one has a kernel size of 6 and a stride of 4. Similar to the generator, batch normalization is implemented in every layer, except for the input and the output. The number of channels for each convolution is 9, 16, 32, 64, 128 and 1. Figure 7 illustrates the proposed discriminator.

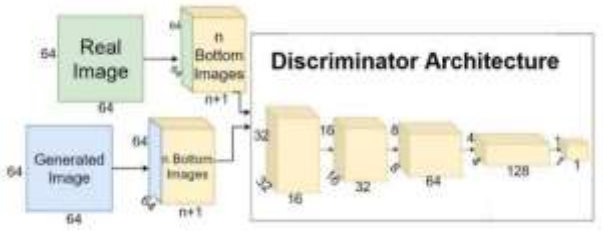

Figure 7: Discriminator architecture: n: # of consecutive bottom images.

Based on our experiments, Wasserstein loss, introduced at Wasserstein GAN [23], is the best loss function that resulted in better generated images when the generator architecture is as given in Figure 6. We implement the Wasserstein loss function for the discriminator as in [24]. The generator loss function is modified to incorporate the loss in between the generated image and the corresponding real top image. After conducting many experiments comparing the real and fake images using pixel to pixel mean squared error, mean absolute error, etc., we conclude that Structural Similarity Index Measurement (SSIM) [25] produced better results, since it considers the luminance, contrast, and the structure in between both images. The SSIM loss function is given by:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (1)$$

where, μ_x, μ_y, σ_x, σ_y denote the mean and standard deviation of both images x and y, $C_1 = (K_1L)^2$ and $C_2 = (K_2L)^2$ with $K_1, K_2 \ll 1$ and L = *max value*. C_1 and C_2 are constants and they are included to avoid instability when $\mu_x + \mu_y$ and $\sigma_x + \sigma_y$ are very close to zero. As stated in [25], for image quality assessment, it is suggested to apply the SSIM function locally rather than globally into the entire image. We compare the real and generated images using the SSIM index locally, by considering an 11 x 11 square window and the corresponding mean and standard deviation are computed within the local window. Then, we apply the mean of every SSIM index to evaluate the entire image quality.

The discriminator and generator are trained by alternate maximization and minimization of the loss functions $\mathcal{L}_D$ *and* $\mathcal{L}_G$ *respectively* given as below:

$$\mathcal{L}_D = E_{x\sim p_{data}(x)}[D(x)] - E_{z\sim p_z}\big[D\big(G(z)\big)\big] \quad (2)$$

$$\mathcal{L}_G = 0.2 * \big(-E_{z\sim p_z}\big[D\big(G(z)\big)\big]\big) + \\ +0.8 * \big(1 - SSIM(generated, real)\big) \quad (3)$$

where p_{data} is the topside image probability distribution and p_z is the Gaussian distribution. Another loss function that resulted in better image generation in our work is the Hinge loss [26]. The generator loss does not change but the discriminator loss is given as below:

$$\mathcal{L}_D = E_{x\sim p_{data}(X)}\big[\min\big(0,-1+D(x,y)\big)\big] - \\ -E_{z\sim p_z}[\min(0,-1-D(G(z),y)] \quad (4)$$

VI. Results and Discussion

After performing many experiments, we obtained results as shown in Figure 8. The images on the right column are the generated topside images ($\hat{\Xi}(k/X(k))$ using eight consecutive bottom images and those in the left column are generated ($\hat{\Xi}(k/x(k))$) by using only one bottom image. When comparing the results with the real topside images (middle), right ones generated by eight bottom images are significantly closer. This implies one bottom image does not have the needed information to generate the topside image (Ξ) but 8 images have. To quantitatively assess their difference, we use the SSIM that allows us to focus on the luminance, contrast and structure [25] among the comparative images. The SSIM scores on the test data are given in Table 2. As the best SSIM score is 1, using eight consecutive backside images improved significantly over from using just one backside image. We also record the SSIM loss for every epoch for both GAN models and the results are shown in Figure 9. We can distinguish the improvement of using eight images over using one image in every epoch. (SSIM loss is just 1-SSIM score). To further show such improvement, we also present the RMSE loss, as the average of all images' pixel to pixel differences between generated images and their respective counterparts, in Table 1. Based on our results, we can see the improvement obtained from using eight bottom images. Thus, the conditional GAN architecture, where the condition is based on eight consecutive bottom images, better generates the top images. The hypothesis in question is thus tested and a new foundation is needed to construct the inverse model for the monitoring of the weld penetration.

Table 1: RMSE loss on the generated images

Number of Backside Consecutive Images (n)	RMSE loss
Eight Images (n=8)	0.105
One Image (n=1)	0.140

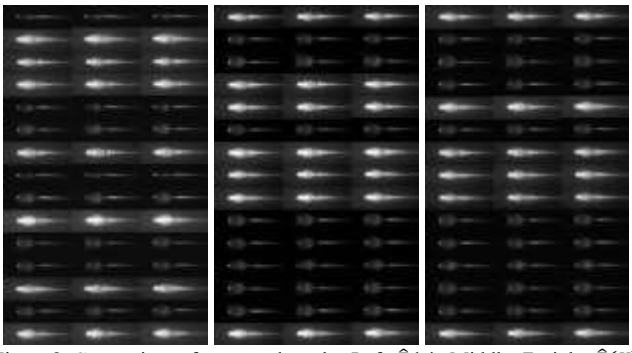

Figure 8: Comparison of generated results. Left: $\hat{\Xi}(x)$; Middle: Ξ; right: $\hat{\Xi}(X)$ with $n = 8$

Table 2: SSIM score on the generated images

Number of Backside Consecutive Images (n)	SSIM score
Eight Images (n=8)	0.906
One Image (n=1)	0.799

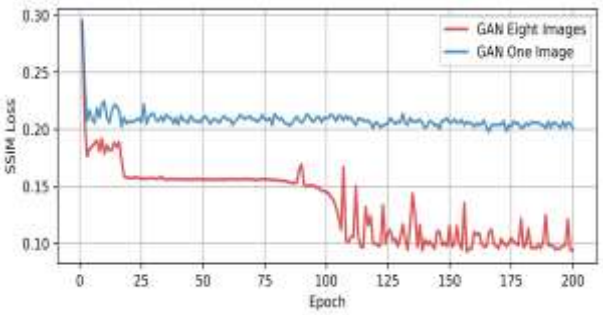

Figure 9: SSIM loss of GAN using eight images and one image

VII. Improvement

While we have tested the hypothesis through comparison of results among using $n = 1$ and $n = 8$, we note that the forward process may be better modeled as the SSIM score is not one yet. To this end, we propose using a Recurrent Neural Network (RNN) to model the dynamics of the input sequence of images. Specifically, we use the Gated Recurrent Unit (GRU).

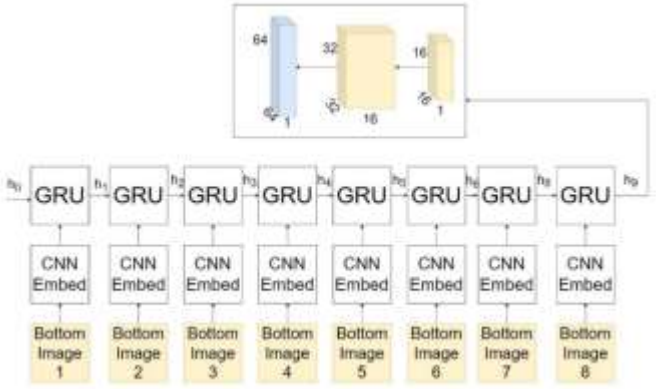

Figure 10: GRU generator architecture.

Generator Structure: Since we are working with consecutive images, it would be expected that a sequential architecture may help the GAN to gain more information, that already exists in the bottom consecutive images. Gated Recurrent Units (GRUs) [23] is one of the most popular sequential/language models with many practical applications. There are works in combining sequence models with GANs [19] to obtain better text generation as shown in [17]. In our case, we need to generate images similar to top ones and the order of bottom images is of utmost significance in improving the generated images. We expect a correlation in between the bottom images and considering the input as a sequence of images, allowing our model to get a better understanding regarding the welding process and thus generating more accurate images. As such, we propose to use the GRU Generator illustrated in Fig. 10 with more details to follow. In this proposed new sequential GAN architecture, referred to as GRU-GAN, we only change the generator and we use the same discriminator as used earlier.

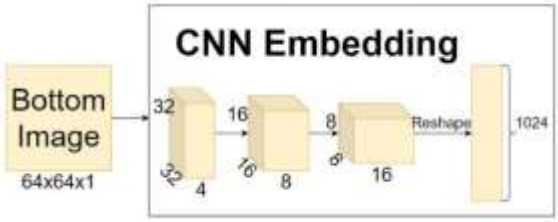

Figure 11: CNN Embedding architecture for every input image.

Before we feed the sequence of bottom images into the GRU model, we apply a CNN Embedding model architecture to extract the most important image features for each image. The CNN Embedding model consists of three convolution layers where kernel size, stride and padding are the same in every layer and they are set to 4, 2, 1, respectively. The number of channels for every convolution layer is 4, 8 and 16 and the input image has only one channel. Based on our experiments, employing the learning rate decay every 40 epochs with coefficient 0.7 and batch normalization [20] in between every layer resulted in a better performance. The CNN Embedding model takes one image of dimension $64 \times 64 \times 1$ at one time step, since we incorporate this model in the same manner as an Embedding layer works in a language model. The output of the CNN Embedding model is reshaped into a vector of dimension 1024 and is viewed as one element of the input sequence for the GRU model. The architecture is illustrated in Figure 11.

GRU Generator: As has been mentioned, we apply the same CNN Embedding model into eight consecutive bottom images and the CNN Embedding output is the input sequence that is fed into the GRU model. In the new generator architecture (GRU Generator), we still consider a Conditional GANs [24] type architecture. Thus, the initial hidden state is initialized using a random distribution and is considered as the z vector that was suggested and used at the previous generator model. The hidden state dimension for the GRU is set to 256. In the GRU architecture, each GRU cell takes (1) the output of its respective CNN Embedding and (2) the output from the previous GRU cell, referred to as hidden state, as its input. Its output is the updated hidden state to be used in the next GRU cell. The output of the GRU structure is the output of the last GRU cell. In our case, this architecture allows the flow of information coming from every time step, (i.e., h_1-h_9). The output is then reshaped into a 16x16x1 image and a series of two fractionally-strided convolutions are applied to convert this high-level representation into a 64x64x1 image, which is the generated image. The applied loss function that has better results is the Hinge function, whose mathematical formulation is given in Section 5.

Improvement Results: As shown in Figure 13, the generated images (right) are difficult to distinguish from the real one (middle). To get a better insight about the difference between employing eight consecutive images and one image in GRU GAN, we can check on Figure 12, where the SSIM loss (between the generated and real top image) is recorded

for every epoch. For a dynamic process, the effect of timing and order are also critical. The GRU model provides a mechanism to discriminate bottom images per timing/order and is thus able to more effectively grasp information due to the sequential architecture. As such, the generated images do not have noticeable differences from the real ones. The resultant SSIM scores on the test dataset given in Table 3 quantifies the improvements. Furthermore, in Table 4, we present the RMSE loss in between the generated image from the GRU-GAN (using eight images and one image) with the experimental top images. Our results on SSIM score, RMSE loss and the graph of the SSIM loss function in Figure 12, all show an improvement from employing eight images and using GRU-GAN. Hence, the GRU-GAN has more accurately modeled the forward process. This improvement is apparently due to an effective use of the sequential information that has been in the used history and this information was not used in our previous model in Section 5.

Table 3: SSIM score on the images using GRU-GAN

Number of Backside Consecutive Images (n)	SSIM score
Eight Images (n=8)	0.925
One Image (n=1)	0.837

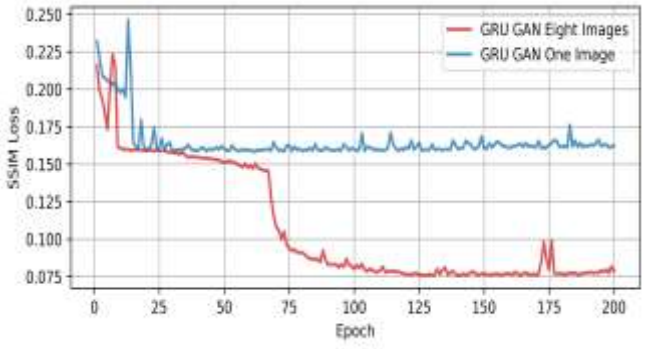

Figure 12: SSIM loss of GRU GAN using eight images and one image

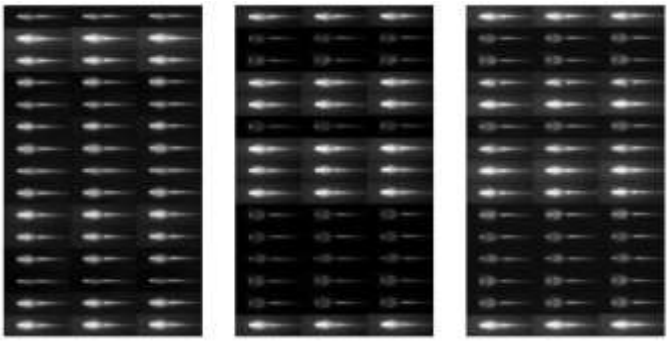

Figure 13: Comparison of generated results with GRU-GAN. Left: $\hat{\Xi}(x)$; Middle: Ξ; right: $\hat{\Xi}(X)$ with $n = 8$

Table 4: RMSE loss on the images using GRU-GAN

Number of Backside Consecutive Images (n)	RMSE loss
Eight Images (n=8)	0.079
One Image (n=1)	0.011

VIII. Conclusions

This letter answered a fundamental question of whether the forward welding process is dynamic, i.e., the influence of the weld pool as partially represented by the state of the weld penetration on the welding process phenomena is dynamic. To this end, we proposed a GAN to model the forward process and found that the current state of the weld penetration (current weld pool) is not sufficient to determine the currently observed welding phenomena. As such, the theoretical foundation on which the current efforts in deep learning-based weld penetration monitoring are based is challenged! A new foundation is thus needed.

We also proposed a GRU-GAN to utilize the sequential information in the penetration history to model the forward welding process more accurately and found that the generated images have no noticeable differences with the real ones. The sequential nature in general dynamic systems/processes is thus fundamental in the forward welding process. It is thus also fundamental in establishing the new foundation for deep learning-based monitoring of weld penetration.

References

1. Yu, R, Cao, Y., Chen, H., Qiang, Y, Zhang, Y.M., 2023. "Deep Learning Based Real-Time and In-Situ Monitoring of Weld Penetration: Where we are and what are needed revolutionary solutions?" in review for Journal of Manufacturing Processes
2. Xiao, Y.H. and Den Ouden, G., 1993. Weld pool oscillation during GTA welding of mild steel. Welding Journal, 72, pp.428-s.
3. Chen, W. and Chin, B.A., 1990. Monitoring joint penetration using infrared sensing techniques. Welding Journal, 69(4), pp.181s-185s.
4. Rohe, M., Stoll, B.N., Hildebrand, J., Reimann, J. and Bergmann, J.P., 2021. Detecting Process Anomalies in the GMAW Process by Acoustic Sensing with a Convolutional Neural Network (CNN) for Classification. Journal of Manufacturing and Materials Processing, 5(4), p.135.
5. Kovacevic, R., Zhang, Y.M. and Ruan, S., 1995. Sensing and control of weld pool geometry for automated GTA welding. Journal of Engineering for Industry -Transactions of the ASME, 117(2): 210-222, 1995.
6. Kovacevic, R., Zhang, Y.M. and Li, L., 1996. Monitoring of weld joint penetration based on weld pool geometrical appearance. Welding Journal, 75(10).
7. Liu, Y.K. and Zhang, Y.M., 2015. Supervised learning of human welder behaviors for intelligent robotic welding. IEEE Transactions on Automation Science and Engineering, 14(3), pp.1532-1541.
8. Zhang, Z., Wen, G. and Chen, S., 2019. Weld image deep learning-based on-line defects detection using convolutional neural networks for Al alloy in robotic arc welding. Journal of Manufacturing Processes, 45, pp.208-216.
9. Feng, Y., Chen, Z., Wang, D., Chen, J. and Feng, Z., 2019. DeepWelding: A deep learning enhanced approach to GTAW using multisource sensing images. IEEE Transactions on Industrial Informatics, 16(1), pp.465-474.
10. Wang, Q., Jiao, W. and Zhang, Y., 2020. Deep learning-empowered digital twin for visualized weld joint growth monitoring and penetration control. Journal of Manufacturing Systems, 57, pp.429-439.
11. Cheng, Y., Wang, Q., Jiao, W., Yu, R., Chen, S., Zhang, Y. and Xiao, J., 2020. Detecting dynamic development of weld pool using machine

learning from innovative composite images for adaptive welding. Journal of Manufacturing Processes, 56, pp.908-915.
12. Nomura, K., Fukushima, K., Matsumura, T. and Asai, S., 2021. Burn-through prediction and weld depth estimation by deep learning model monitoring the molten pool in gas metal arc welding with gap fluctuation. Journal of Manufacturing Processes, 61, pp.590-600.
13. Kim, H., Nam, K., Oh, S. and Ki, H., 2021. Deep-learning-based real-time monitoring of full-penetration laser keyhole welding by using the synchronized coaxial observation method. Journal of Manufacturing Processes, 68, pp.1018-1030.
14. Kim, H., Nam, K., Oh, S. and Ki, H., 2021. Deep-learning-based real-time monitoring of full-penetration laser keyhole welding by using the synchronized coaxial observation method. Journal of Manufacturing Processes, 68, pp.1018-1030.
15. Yu, R., Kershaw, J., Wang, P. and Zhang, Y., 2022. How to Accurately Monitor the Weld Penetration From Dynamic Weld Pool Serial Images Using CNN-LSTM Deep Learning Model?. IEEE Robotics and Automation Letters, 7(3), pp.6519-6525.
16. Zhang, Y., Wang, Q. and Liu, Y., 2021. Adaptive intelligent welding manufacturing. Welding Journal, 100(1), pp.63s-83s.
17. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2020. Generative adversarial networks. Communications of the ACM, 63(11), pp.139-144.
18. Radford, A., Metz, L. and Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.
19. Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
20. Mirza, M. and Osindero, S., 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.
21. Wang, X. and Gupta, A., 2016. Generative image modeling using style and structure adversarial networks. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14 (pp. 318-335). Springer International Publishing.
22. Isola, P., Zhu, J.Y., Zhou, T. and Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).
23. Arjovsky, M., Chintala, S. and Bottou, L., 2017, July. Wasserstein generative adversarial networks. In International conference on machine learning (pp. 214-223). PMLR.
24. Miyato, T., Kataoka, T., Koyama, M. and Yoshida, Y., 2018. Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957.
25. Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), pp.600-612.
26. Jae Hyun Lim and Jong Chul Ye. 2017. Geometric GAN. arXiv:1705.02894. Retrieved from https://arxiv.org/abs/1705. 02894.