

# Modeling Imaged Welding Process Dynamic Behaviors Using Generative Adversarial Network (GAN) for a New Foundation to Monitor Weld Penetration Using Deep Learning

Edison Mucllari<sup>1</sup>, Yue Cao<sup>2</sup>, Qiang Ye<sup>1</sup>, YuMing Zhang<sup>2\*</sup>

1: Department of Mathematics, University of Kentucky, Lexington, KY 40506, USA

2: Department of Electrical and Computer Engineering and Institute for Sustainable Manufacturing, University of Kentucky, Lexington, KY 40506, USA

\* yuming.zhang@uky.edu

**Abstract:** This paper aims to model the dynamic behaviors of welding process that can be observed by humans and monitored by imaging sensors. This is fundamental in deciding if such dynamic behaviors can be used as raw data, that contains sufficient relevant information, to monitor what occurs underneath the workpiece determining the weld integrity. Challenges arise as featurization of these observations proves difficult, and the governing laws for the underlying process remain largely unknown. Leveraging the ability of Generative Adversarial Networks (GANs) to generate intricate phenomena from simple inputs by automatically approximating unknown underlying laws, we propose using a GAN to predict the dynamic behaviors of welding processes. In this framework, dynamic behaviors serve as the output, while the trained GAN functions as the model for the underlying process, generating the observed behaviors from the inputs. The success of predictions depends not only on the GAN's capability to approximate the process but also on the sufficiency of the inputs provided. With this consideration, we model an exceptionally complex process that correlates weld joint penetration with observed phenomena to determine if the underlying process involves critical inputs beyond penetration alone. Our findings reveal the necessity of welding currents for accurate predictions by the GAN model. This implies that the combined information from human-observed/imaged phenomena and welding currents provides a more comprehensive basis for monitoring weld penetration, contributing to the advancement of robotizing welding processes with the necessary intelligence.

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

## 1. Introduction

Per the International Federation of Robotics, “50 percent of all the world’s robots are used for welding. Specifically, 33 percent are employed for spot welding, 16 percent are doing arc welding, and 1 percent are performing some other type of welding operation.” [1] Arc welding, as the most widely used welding process, is thus still far from being fully robotized. This is largely due to the lack of generalizable approaches for robots to adapt to the process, similarly to human welders, to assure the quality of the produced welds.

Human welders produce quality welds through real-time adaptive adjustment of welding parameters per observed welding process dynamic behaviors while following the established welding procedure. The focus of their real-time adaptive adjustment is on the weld fusion/weld penetration which plays the most critical role in assuring the weld integrity. Robotized welding process can follow the established welding procedure better than human welders and should outperform over human welders if effective adaptive adjustment can also be made to assure the weld penetration. However, monitoring weld penetration is a classical challenge as it occurs underneath the workpiece, making it not directly observable.

Due to its crucial role in robotizing welding, this challenge has continuously attracted numerous researchers from around the world, testing various different ideas since the 1970s including pool oscillation [2, 3] that correlates the weld penetration to oscillation frequency, infrared sensing [4] that utilizes the difference in emissivity among solid and liquid to detect the weld pool boundary, and weld sag [5] that

increases with the penetration. Image processing and other algorithms have been used to calculate features from sensed phenomena and advanced techniques including neural networks [6], neuro-fuzzy systems [7, 8, 9], and support vector machines [12] have been used to map the calculated features to the penetration.

The advancements in deep learning (DL) techniques provide a new opportunity. Previous efforts attempted to propose features, develop algorithms for extracting and correcting these features, and then relate them to penetration labels using models. They ranged from simpler regression models to more complex models, including neural networks, with the features as input and penetration as the output. They all involved representing high-dimensional data/images using lower-dimensional features, and these representations may not be adequate, often requiring iterative efforts for improvement. Each modification in the features necessitates the development of new algorithms for extraction, making the process time-consuming, and the success uncertain and generally not generalizable. DL, which offers the potential for automatically optimizing features without the need for extensive human iteration, provides an opportunity toward generalization.

The authors [11] recently underscored the transformative impact of DL and elucidated the reasons behind its potential systematically. DL was first reported for monitoring weld penetration in 2019 [12], and the number of records in Web of Science has grown rapidly since. Regrettably, the new efforts still have continued to follow the past pattern centering on application specifics. The crucial aspects in taking transformative DL to desired generalizability, identified by the authors to be the sufficiency of the raw data/model inputs [11], have been largely overlooked.

What is particularly missing is in-depth understanding that human-observable phenomena, which serve as inputs to most deep learning models for predicting the weld penetration, exhibit dynamic relationships with the weld penetration. While application/process specifics as centered in existing studies are considered available, the lack of such in-depth knowledge/understanding critical to generalizability is not easily recognized! Authors' recent study [13] has taken the first step in gaining an in-depth understanding of how the observations ( $\Xi$ ) correlate with and are determined by the events occurring beneath the workpiece ( $x$ ) which represents the weld penetration. It revealed that the human-observed welding process dynamic behaviors are much better modeled by  $\Xi(k) = P(x(k), x(k-1), \dots)$  than by  $\Xi(k) = P(x(k))$ . This implies that adding the penetration history  $x(k-1), \dots$ , into the model input improves the adequacy of the raw information. However, the human-observed phenomena are still not fully accurately predicted [13]. As such, the understanding of the human-observed behaviors is still not complete.

In seeking for more complete understanding, we look at an illustrative example, the oscillation of the weld pool which represents the initial phenomena studied for weld penetration monitoring [2, 3]. The oscillation behavior observed is not solely determined by the geometry of the weld pool representing the penetration but *also by the arc pressure  $u$  stimulating the pool to oscillate*. While this example serves as illustration, it has been undoubtedly demonstrated that the DL models of our concern generally must be based on  $\Xi(k) = P(x(k), x(k-1), \dots; u(k), u(k-1), \dots)$ , encompassing the stimuli and dynamics.

In this paper, we use a GAN to map the proposed inputs to the unprocessed/un-featurized observations from the gas tungsten arc welding process, the primary arc process used for precise joining where the penetration must be assured. We will reveal that adding welding currents into the inputs of the underlying model as approximated by the trained GAN significantly improved the prediction accuracy, thus the modeling of the underlying process. The result is fundamental in designing a generalizable deep learning model to predict the weld penetration from observed welding process phenomena.

## 2. Underlying Process and Modeling Approach

### 2.1 Underlying Process

The underlying welding, i.e., gas tungsten arc welding (GTAW), process can be simply illustrated first by Fig. 1. The welding torch imposes the arc on the upper surface of the workpiece to process (heat and melt) the workpiece. This processing produces complex phenomena, and we use two cameras to view the relevant phenomena. Camera 1 views the phenomena  $\Xi$  human welders observe during welding by centering at the

weld pool generated by the arc on the upper surface of the workpiece. The resultant image is referred to as the top image  $I_t = f(\Xi)$  where the observed objects include the weld pool and its surrounding region, the tungsten electrode which emits the electrons that flow to the workpiece to form the arc current  $I$ , and the arc established between the tungsten and workpiece. Camera 2 views the opposite area underneath the workpiece on the lower surface and the resultant image is referred to as the bottom image  $I_b$ . In the GTAW experiments conducted to generate the dataset, the welding speed is fixed at 2mm/s, and the current  $I$  varies between 80 A to 105 A randomly every 2 seconds, producing top and bottom images, and corresponding image pairs, of various types. The cameras capture 60 images per second, establishing a dataset with 16,200 data/image pairs after removing the arc ignition phases in each experiment. Fig. 2 shows a series of  $[I(k), I_b(k)]$  image pairs sampled from a 5-second period at 1 second per pair. As the variation that causes the various image pairs is due to the random change in the welding current, this series is representative of various types of images and image pairs.

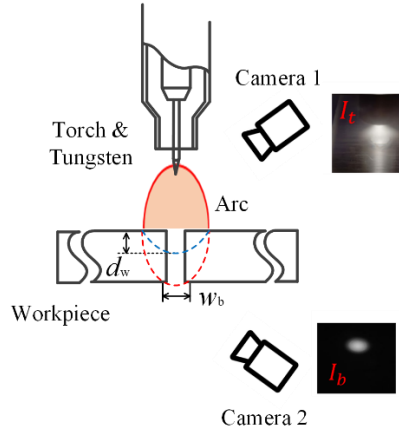


Figure 1: Illustration of Underlying Process.

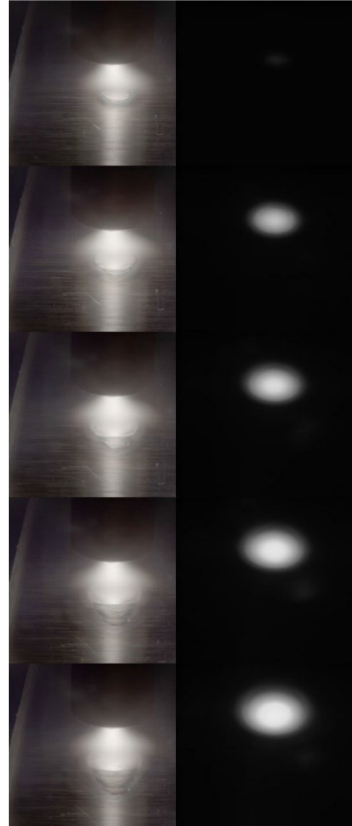


Figure 2: Five-second sequence of image pairs illustrating images of various types.

The top image  $I_t$  captures human observations  $\Xi$  based on which human welders adjust their operations. We have mentioned that their adjustment is to maintain the weld penetration  $x$  at the desired state  $x^*$ . The weld penetration  $x$  is either measured by the depth of the weld pool  $d_w$  [11] (Fig. 1) if the workpiece is not completely melted in the thickness direction or the backside bead width of the weld pool  $w_b$  [11] (Fig. 1) if the workpiece is completely penetrated through the entire thickness of the workpiece. The bottom image  $I_b$  can be used to determine  $w_b$  directly. It can also provide the most direct measurement to estimate  $d_w$  [11]. The bottom image  $I_b$  thus corresponds to the weld penetration  $x$ , i.e.,  $I_b = g(x)$ , the human welder tries to maintain during their operation. In this study, we use the bottom image  $I_b$  as the full label of weld penetration, while  $d_w$  or  $w_b$  can be considered simplified measures. Our fundamental question is whether the human-observed behavior (top image  $I_t$ ) is purely decided by the weld penetration (bottom image  $I_b$ ). If not, the foundation for using the top image  $I_t$  to estimate the penetration  $I_b$  would not be perfectly solid, or the raw information is not sufficient [11].

The above question defines a “virtual underlying process” that produces the top image  $I_t = f(\Xi)$  from the bottom image  $I_b = g(x)$ . This is an extremely complex process that has not been studied. The complexity is due to the complexity of the actual physical processes involved. To elaborate, the arc imposes both heat flux (complex distribution) and forces, including surface forces (arc pressure, surface tension) and body forces (electromagnetic force, convection), on the workpiece surface and underneath. The forces coupled with the welding conditions dominate how the heat imposed by the arc are transported within the workpiece to melt it and form the penetration which can be observed on the lower surface. After the weld pool and weld penetration change, the behaviors observed on the upper surface must also alter accordingly. In this way, the complete process is from the upper to the lower and then from the lower to the upper and our focus in this study is from the lower to the upper.

As such, the behaviors  $\Xi$  observed on the upper surface in the top image  $I_t$  are determined by the arc (input of the welding process) and the penetration (result of the welding process). The bottom image  $I_b$  (weld penetration  $x$ ) itself should be insufficient in determining the top image  $I_t$ . It is the coupling of the arc and penetration that determines the behaviors of the phenomena  $\Xi$  observed in the top image  $I_t$ . A model that can accurately produce the top image  $I_t$  requires input raw information from the arc and penetration. This is a fundamental improvement in our understanding as it extends our foundation from  $f(I_t, I_b) = 0$  to  $f(I_t, I_b, a) = 0$  where  $I_t, I_b, a$  stand for the top image, bottom image and arc. As a key in the deep learning is to assure that the sufficient raw information has been provided to the deep learning model, this extension promises for a next level of deep learning-based monitoring of weld penetration which is critical in robotizing welding processes.

We note that most of the deep learning-based monitoring techniques for weld penetration are based on  $f(I_t, I_b) = 0$  [11]. There are a few studies that include the inputs from the welding process [14, 15]. However, the addition is not justified per discussion of the underlying physical process, similarly as in many studies that use multiple information sources.

In our above discussion, we have not mentioned the dynamics of the underlying process which is equally important. That is, is the current top image  $I_t(k)$  determined by the current bottom image  $I_b(k)$  or by its history  $I_b(k), I_b(k-1), \dots$ ? This is another fundamental question if the behaviors observed by human are dynamic. In our previous study [13], we have used a GAN to generate  $I_t(k)$  from  $I_b(k)$  and from  $I_b(k), I_b(k-1), \dots$  to see if the use of the history improves the sufficiency of the raw information. We found that the sufficiency was significantly improved. However, large errors in the prediction of the top images were still observed. As will be shown in this paper, introducing the welding current, denoted as  $u$ , to present the input of the welding process in general, eliminates all errors human may observe.

## 2.2 Generative Modeling

We note that the majority of studies on deep-learning based monitoring of weld penetration use observable complex phenomena  $\Xi$  (high-dimensional data) as inputs to predict low-dimensional labels. High-dimensional data contains relevant and irrelevant features that may both be extracted to fit the labels.

Validation data may also contain the same irrelevant features and using the validation result to measure the model quality is a compromise as there are no other better measures. However, in generative modeling, low-dimensional data is used to predict complex high-dimensional data. If successfully modeled, the fundamental role of the simple input in determining the complex output can be better assured. In this extent, we propose to study through generative modeling and use a GAN to generate complex top image  $I_t$  from simple bottom image  $I_b$  and welding process input  $u$ .

Generative Adversarial Networks (GANs) [16] were first introduced in 2014. There has been a remarkable progress in further improving the GAN [16] architecture as well as successfully integrating the model in various applications. The GAN framework incorporates a minimax game between a generative model (generator) and a discriminator model (discriminator). In a nutshell, GANs [16] are generative models that learn the data distribution by generating data samples. For conditional generation tasks, conditional GANs [17] are usually employed where the condition inputs are built into GAN's generator and discriminator.

Deep Convolutional Generative Adversarial Networks (DCGANs) [6] incorporate convolutions into the GANs [16] architecture to work with image data. They provide important and helpful information to ensure a more stable training while using only convolution layers (no linear layers). They have been shown to be capable of generating high quality images.

### 3. GRU-GAN Model

GANs can be trained to generate complex distributions from simple distributions. This implies that GAN are capable of modeling unknown complex mappings. In the training process, the discriminator does not try to discriminate/match between a generative image from/with a particular real image but distinguish all generated images from real images. Our work is based on conditional GANs [17]. The conditional part of our GANs [16] corresponds to the bottom images  $I_b(k)$ 's and the welding current waveform  $u(k)$ 's. For each real top image  $I_t(k)$ , the conditional GAN tries to generate a similar one  $\hat{I}_t(k)$  using specific conditions as presented by the corresponding bottom images and welding currents. The conditional GAN matches between real and generated top images.

The conditional GAN we develop predicts the top image  $I_t(k)$  from the bottom images  $I_b(k-j)$ 's and the welding current  $u(k-j)$ 's as shown in Fig. 3. We follow a similar approach in [13] that generates  $I_t(k)$  from  $I_b(k-j)$ 's where a GRU model is combined with a GAN model. The main novelty is to incorporate the current information as a condition input which reflects an improvement in understanding of the fundamentals of the underlying process analyzed above. To generate  $I_t(k)$ , we use  $n$  bottom images  $I_b(k+1-j)$  ( $j = 1, \dots, n$ ), denoted as  $\mathbf{I}_b(k)$ ; and for each  $I_b(k+1-j)$  used,  $m$  welding currents  $u(k+1-j+(1-p))$  ( $p = 1, \dots, m$ ) forming a current waveform are added. The currents added form a  $m \times n$  matrix filled with  $u(l)$  ( $l = k, \dots, k+2-n-m$ ), denoted as  $\mathbf{U}(k)$ .

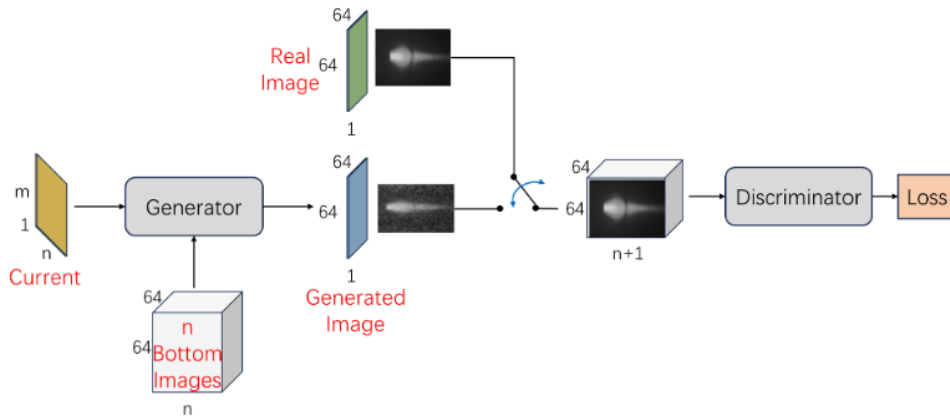


Figure 3: Proposed conditional GRU-GAN.

#### 3.1 GRU-GAN Discriminator

Illustrated in Fig. 4 is the architecture of the discriminator. It receives an input tensor of dimensions  $64 \times 64 \times (n+1)$ , formed by concatenating  $n$  consecutive bottom images  $I_b(k)$  and  $I_t(k)$ , representing either the real image  $I_t(k)$  or the generated image  $\hat{I}_t(k)$ . The discriminator architecture consists of four successive convolution layers with channel configurations of 16, 32, 64, and 128, respectively. The convolution layers utilize a kernel size of 4, a stride of 2, and zero padding, except for the final layer. Batch normalization [19] and leaky ReLU with a coefficient of 0.2 are applied between convolution layers. The last layer of the discriminator employs the sigmoid activation function. Refer to Figure 4 for an illustration of this architecture.

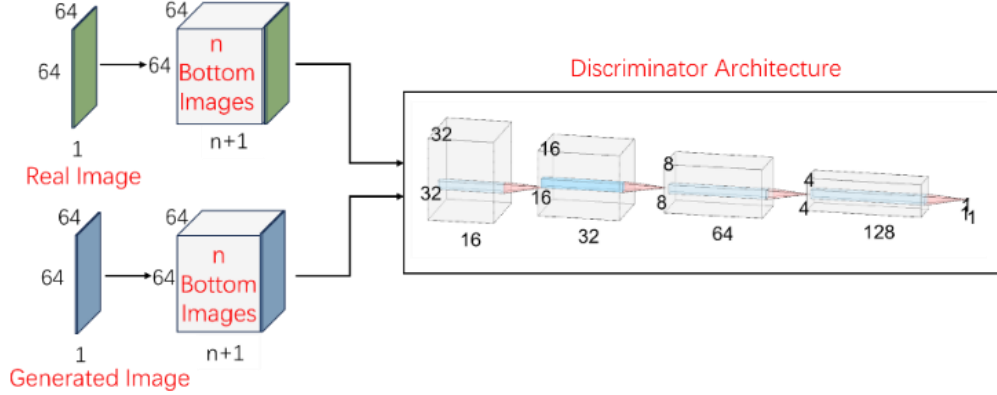


Figure 4: Discriminator.

### 3.2 CNN Embedding

As anticipated, the arrangement of the bottom images holds significance in determining the corresponding top image. To address this, we treat the input as a sequence and incorporate Gated Recurrent Units (GRU) [20]—a type of recurrent neural network—into the generator. Other notable recurrent models include LSTM [21], NC-GRU [22], GORU [23], among others. In our proposed conditional GAN (conditional GRU-GAN), the conditioning part comprises the bottom images  $I_b(k)$  and the welding current matrix  $U(k)$ . Additionally, we conducted experiments with the NC-GRU [22] architecture. Given that the condition input sequence involves images, we integrate a CNN embedding into our main architecture, where each bottom image undergoes convolutional neural network (CNN) embedding to extract essential features before being fed into the GRU. The CNN Embedding layer comprises three convolution layers with channel configurations set to 4, 8, and 16, respectively. All convolution layers use a kernel size of 4, a stride of 2, and a zero-padding size of 1. Batch normalization is applied between each convolution layer, and the output is reshaped into a 1024-dimensional vector at the final layer. Refer to Figure 5 for a more detailed visual representation of the CNN Embedding.

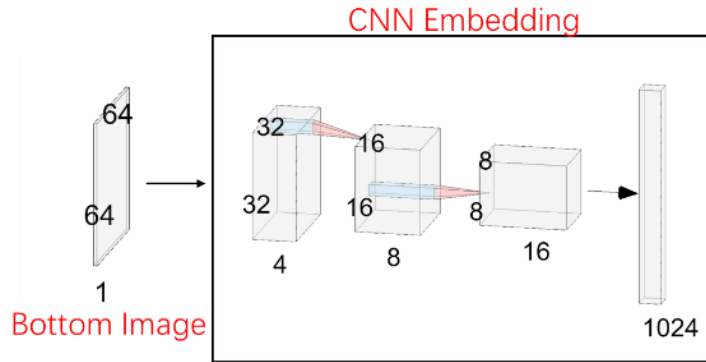


Figure 5: CNN Embedding

### 3.3 GRU-GAN Generator with Welding Current

As mentioned before, a substantial difference from the work of [13] is the welding current information we incorporate as part of the input of the GRU generator. Figure 6 shows the GRU generator to

produce  $I_t(k)$ . Every GRU cell thus takes  $I_b(k)$  but the welding current as well. Since there is only one current value for each bottom image  $I_b$  and the current demonstrates the effect through waveform, we form a current vector using  $m$  most recent current values  $u(k-j), \dots, u(k-j-m+1)$ . Thus, due to the incorporation of the current information, during the first iteration of the training process, we first predict  $I_t(k)$  at  $k = n + m$ . When  $n = 60$  and  $m = 30$ , the first generated image will be  $I_t(90)$ .

The current vector passes through two layers of 1D convolutions (CNN 1D). The kernel size for both layers is set to 3 and the numbers of channels are 3 and 6, respectively. The initial hidden state of the GRU is initialized using a Gaussian random vector and is considered as the  $z$  vector referenced in the GAN architecture, where the Generative Adversarial Network is trained to approximate the distribution of the top images that we need to generate. The dimension of the  $z$  vector is set to 256. The GRU output (i.e. the last hidden state  $h(k)$ ) is then reshaped into  $16 \times 16 \times 1$  and this image is passed through two deconvolution layers to obtain an image of dimension  $64 \times 64 \times 1$ . This image is the generated image of our model.

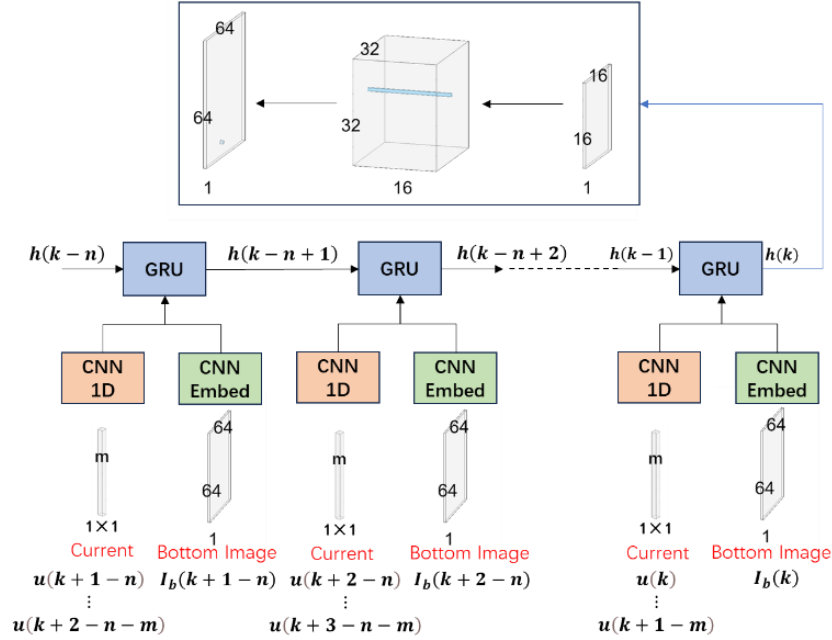


Figure 6: Generator.

To train the model, the GRU loss function is based on the Wasserstein loss [24]:

$$\mathcal{L}_D = E_{x \sim p_{data}(x)}[D(x)] - E_{z \sim p_z}[D(G(z))] \quad (1)$$

To ensure the model is generating a top image  $\hat{I}_t(k)$  matching with  $I_t(k)$ , we add the Structural Similarity Index Measure (SSIM) [25] to the loss. The Structure Similarity Index Measure (SSIM) is defined as

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

where,  $\mu_x, \mu_y, \sigma_x, \sigma_y$  denote the mean and standard deviation of both images  $x := I_t$  and  $y := \hat{I}_t$ ,  $C_1 = (K_1L)^2$  and  $C_2 = (K_2L)^2$  with  $K_1, K_2 \ll 1$  and  $L = \max \text{value}$ . In order to avoid instability when  $\mu_x^2 + \mu_y^2$  and  $\sigma_x^2 + \sigma_y^2$  are very close to zero, we employ  $C_1$  and  $C_2$ , which are constants. SSIM was first introduced in [25] to measure the difference between two images by considering the luminance, contrast and the structure.

Thus, the loss function for training the generator is the following weighted loss:

$$Loss_G = 0.8 * Wass_G + 0.2 * SSIM(Generated Image, Real Top Image) \quad (3)$$

In our training, the learning rate is  $3 * 10^{-4}$  and the model is trained for 100 epochs. An exponential learning rate decay is also applied every 10 epochs with coefficient 0.9.

#### 4. Results and Discussion

The size of our total dataset is 18,567 and we split it into training and testing subsets, with 80% for training and the remaining 20% for testing. We have conducted comparative experiments to see how the ability of the GAN in generating  $\hat{I}_t$  similar or identical to  $I_t$ , as measured by the SSIM score, changes with the condition as shown in Table 1. Assuming each GAN is well trained, the score reflects the sufficiency of the raw information, used as the condition of the GAN, in predicting the top image.

Table 1: **SSIM scores**

| Model # | GAN condition       | SSIM score   |
|---------|---------------------|--------------|
| 1       | $n = 1, m = 0$ [13] | 0.837        |
| 2       | $n = 8, m = 0$ [13] | 0.925        |
| 3       | $n = 15, m = 0$     | 0.92         |
| 4       | $n = 30, m = 0$     | 0.867        |
| 5       | $n = 60, m = 0$     | 0.847        |
| 6       | $n = 8, m = 8$      | 0.932        |
| 7       | $n = 8, m = 30$     | 0.934        |
| 8       | $n = 60, m = 30$    | <b>0.938</b> |

As can be seen from Table 1, we first increase the dynamic dependence time ( $n$ ) from  $n = 1$  (model #1 in Table 1, no dynamics introduced) to  $n = 8$  (model #2 in Table 1). This introduction of the dynamic time was effective in improving the sufficiency of the raw information (the sufficiency of the input of the underlying process) as can also be seen from Fig. 7 for their visual comparison. However, the improvement does not continue when  $n$  further increases (from model #2 through to #5) as shown in Table 1 and in Figs. 8 and 9. Instead, we see decreases in accuracy.

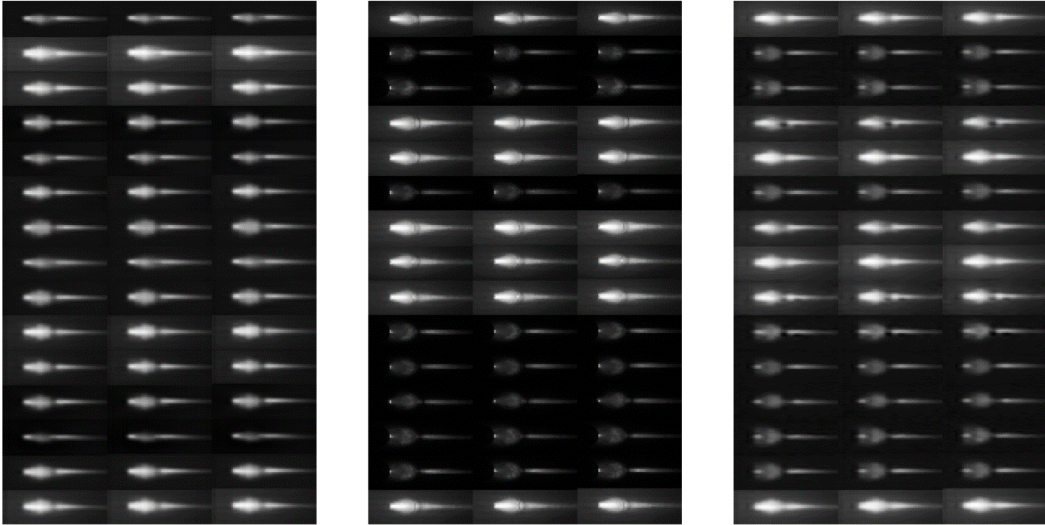


Figure 7: Result from model #1 ( $n = 1, m = 0$ ) and #2 ( $n = 8, m = 0$ ) showing the effect of introducing dynamics. Left: generated using ( $n = 1, m = 0$ ); middle: real top images; right: generated using ( $n = 8, m = 0$ ) [13]. The comparison of the left and right with the middle shows the effect of dynamics in improving the sufficiency of the model input of the underlying process.

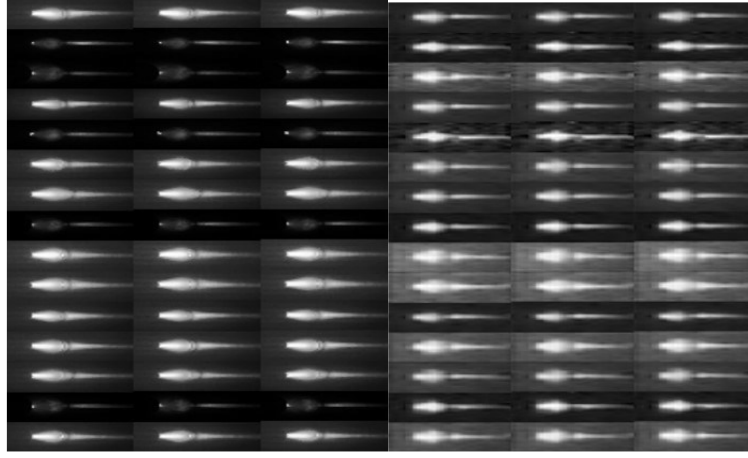


Figure 8 Result from Model #4 in Table 1 ( $n = 30, m = 0$ ). Left:  $I_t$ ; right:  $\hat{I}_t$ .

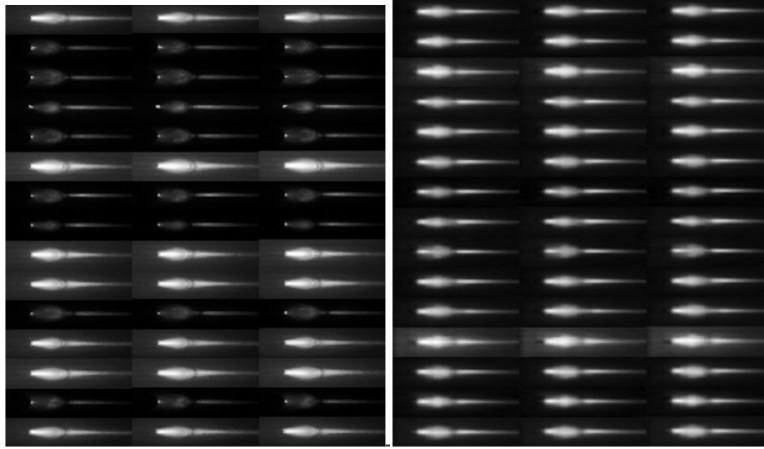
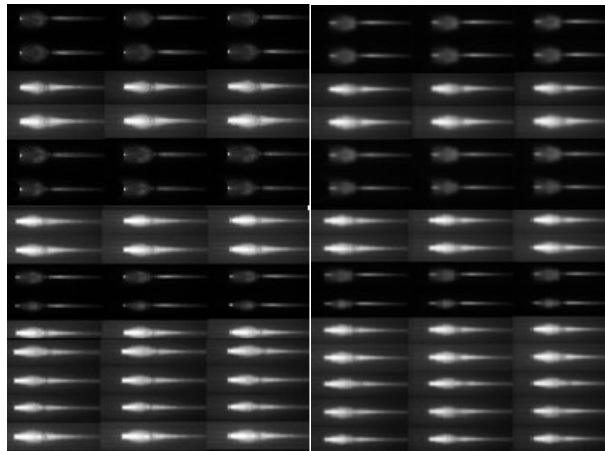


Figure 9 Result from Model #5 in Table 1 ( $n = 60, m = 0$ ). Left:  $I_t$ ; right:  $\hat{I}_t$ .

Theoretically, the SSIM score should not decrease as the model complexity ( $n$ ) increases. Practically, to train a more complex model, more data is needed. The training also tends to be more challenging. The decrease in the SSIM score reflects those possible benefits from increasing the model complexity (model #4, #5 from #2) are less significant in comparison with the adverse effect in increases in data needed and in training difficulty. As such, the modeling accuracy of the human-observed behaviors is first drastically increased by introducing dynamic dependence (from model #1 to #2), but such improvement does not continue indefinitely.



(a) Real image  $I_t$  (b) Generated image  $\hat{I}_t$

Figure 10: Result from this study showing the effect of introducing welding current waveform. Model #6 with  $(n = 8, m = 8)$ .

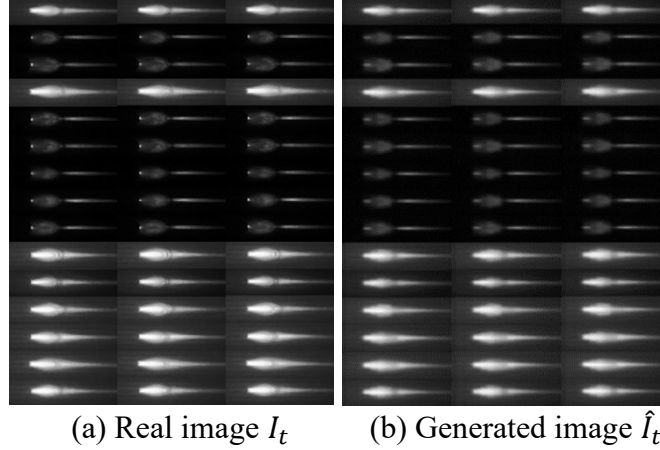


Figure 11: Result from this study showing the coupling effect from the dynamic dependence on the penetration with the welding current waveform. Model #8 with  $(n = 60, m = 30)$ .

Model #2 with  $(n = 8, m = 0)$  can be considered a milestone improvement but, from Figure 7, its accuracy in generating  $\hat{I}_t$  is still far from being sufficient. Per our analysis of the underlying process, adding the current waveform may introduce a novel way to improve the sufficiency of the input of the underlying process. As can be seen in Table 1, we improved the SSIM score from 0.925 (model #2 with  $(n = 8, m = 0)$ ) to 0.932 (model #6 with  $(n = n, m = 8)$ ) simply by adding a current waveform and continued the improvement by increasing the model complexity in both  $n$  and  $m$  directions. The improvements are clearly visualized by comparing Fig. 7 with Fig. 10 and with Fig. 11.

One may note that the improvement in the SSIM score appears to be small. Model #8 with  $(n = 60, m = 30)$  demonstrates a perfect set of generated images with SSIM=0.938. However, the improvement over model #2 with  $(n = 8, m = 0)$  in the SSIM is only 0.013. This provides us a sense of the sensitivity of modeling accuracy with the SSIM score. With this in mind, we below discuss how the accuracy changes with  $n$  and  $m$  after introducing the current waveform whose length is measured by  $m$ .

First, introducing the current waveform, from model #2  $(n = 8, m = 0)$  to model #6  $(n = 8, m = 8)$  increases the accuracy/SSIM score from 0.925 to 0.932. The increase in the SSIM score is 0.007. Although 0.007 increase appears to be very small, from the sensitivity of the modeling accuracy with the SSIM score, we can appreciate that this improvement in the accuracy is quite significant. As such, introducing the current waveform is fundamental in modeling the underlying process in understanding human-observed dynamic behaviors.

Next when we further increase the length of the current waveform from  $m = 8$  to  $m = 30$  (from model #6  $(n = 8, m = 8)$  to model #7  $(n = 8, m = 30)$ ), there demonstrated further improvement. Then we increase the number of the bottom images to 60 (corresponding to 1 second) (model #8), we see further improvements in the modeling accuracy rather than decrease.

We have discussed the relationship between the significance in increasing the model sufficiency and the resultant increase in the model complexity when the training data is given. If the model complexity increase does not improve the model sufficiency, the trained model may not improve the modeling accuracy or may even reduce the accuracy. After introducing the welding current waveform, the modeling accuracy keeps improving even when increasing the number of the bottom images.

It is interesting to note that increasing the bottom images reduces accuracy but improves the accuracy after introducing the current waveform. This probably implies that it is the coupling between the current (welding process input) and penetration (welding process output) that has longer dynamic effect on the observed dynamic behaviors rather than the input and output of the welding process alone separately. This makes sense as there is a dynamic process from the welding parameters to change the penetration and the changed penetration (changed weld pool) to come back to influence the observed behaviors. Of course, such an

explanation is highly hypothetical and is subject to systematic study to test which exceeds the scope of this present study.

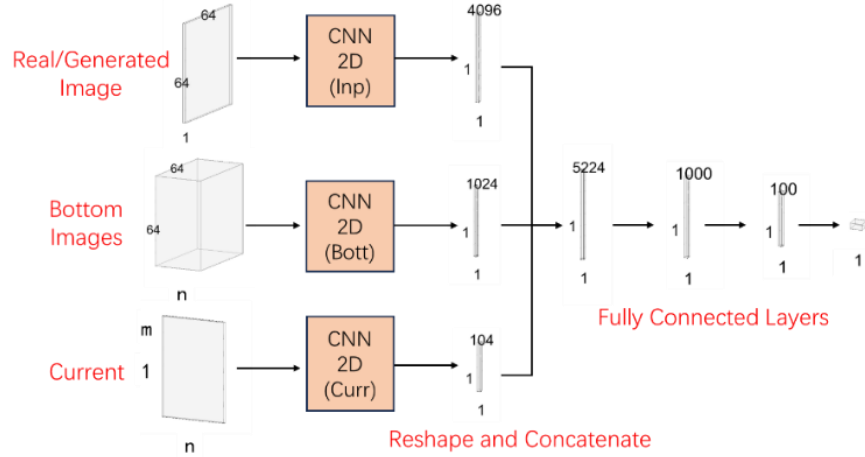


Figure 12: An experimental discriminator.

## 5. Additional Efforts

We have made additional efforts to see if the results in Table 1 reasonably reflect the best possible modeling accuracy based on which we analyzed how the sufficiency changes with the information sources. One effort is to use another discriminator shown in Figure 12 where different kinds of information are separated processed first to form the features to see if it may improve the generative capability.

This another discriminator has three CNNs: (1): CNN 2D (Inp) with the real/generated image as the input. It has two convolution layers with a batch norm applied in between. The kernel sizes are 4 and 3, respectively with stride 2 and zero padding set to 1 for both convolution layers. The number of channels besides the input image is set to 8 and 16 for the two convolution layers. (2) CNN 2D (Bott) with  $n$  bottom images as the input. It also has two convolution layers, with numbers of channels 16 and 4, and kernel size 4, stride 2 and zero padding 1 for both layers and batch norm in between layers; (3) CNN 1D with the welding current  $m \times n$  matrix as the input. It has two 1D convolution layers with kernel size 3, numbers of channels 16 and 4. Then the outputs from all three CNNs are reshaped as [batch size  $\times$  dim] (where based on the given hyperparameters dim for CNN 2D (Inp) is 4096, the dim for CNN 2D (Bott) is 1024 and 104 for CNN 1D (Curr). We incorporate two Fully - Connected layers (size 1000 and 100) and then obtain the output (one number that shows if the image is from the real data or the generated images). Leaky ReLU with coefficient 0.2 is the activation function applied in between layers besides the last layer, where sigmoid activation function is employed.

The ( $n = 60, m = 30$ ) conditional GAN using this discriminator generates SSIM=0.936. As such, the GAN with the concatenated input for the discriminator (Fig. 4) which gives SSIM=0.938 reasonably effectively utilized the input information. The scores given in Table 1 thus reasonably measure the sufficiency of the various raw information as the input of the underlying process governing the human-observed dynamic behaviors.

Table 2: SSIM scores from experiments using NC-GRU

| GAN condition    | SSIM score   |
|------------------|--------------|
| $n = 8, m = 8$   | 0.932        |
| $n = 8, m = 30$  | 0.933        |
| $n = 60, m = 30$ | <b>0.935</b> |

Another effort we have taken is to experiment using NC-GRU in place of GRU. The NC-GRU uses a Neumann series-based Scaled Cayley transformation to train orthogonal matrices in GRU and a previous study has shown that NC-GRU significantly outperforms GRU as well as several other RNNs [22]. Table 2 documents the results trying different models  $(n, m)$ . The results are very close to those in Table 1 which is believed to reasonably measure the sufficiency of the various raw information as the input of the underlying process governing the human-observed dynamic behaviors.

The last additional effort we wish to report is that we have also tried the RMSE loss in Table 3, which is the average of all images (test data), pixel to pixel differences between all  $\hat{I}_t$ 's and their respective  $I_t$ 's. In this case, the model accuracy increases as the RMSE reduces. As can be seen, the results related to how the accuracy is related to model structure  $(n, m)$  are similar as in Table 1. Table 4 is the results using NC-GRU and REMS loss. Again, the results are similar to those in Table 1.

Table 3: REMS Losses

| GAN condition       | REMS          |
|---------------------|---------------|
| $n = 1, m = 0$ [13] | 0.108         |
| $n = 8, m = 0$ [13] | 0.079         |
| $n = 15, m = 0$     | 0.079         |
| $n = 30, m = 0$     | 0.100         |
| $n = 60, m = 0$     | 0.104         |
| $n = 8, m = 8$      | 0.073         |
| $n = 8, m = 30$     | 0.0707        |
| $n = 60, m = 30$    | <b>0.0700</b> |

Table 4: REME loss from experiments using NC-GRU

| GAN condition    | SSIM score    |
|------------------|---------------|
| $n = 8, m = 8$   | 0.072         |
| $n = 8, m = 30$  | 0.071         |
| $n = 60, m = 30$ | <b>0.0703</b> |

As such, we have taken reasonable efforts to study the sufficiency of the various raw information as the input of the underlying process governing the human-observed dynamic behaviors and are confident that we are able to conclude on the correlation between the modeling capability, for the underlying process governing the human-observed dynamic behaviors, and model inputs.

## 6. Conclusions

We have developed a conditional GAN to model the dynamics during gas tungsten arc welding. With the right input information as the condition of the GAN, the dynamic process can be accurately modeled with no prior knowledge of the welding process dynamics.

The welding process behaviors human welders observe during gas tungsten arc welding process are dynamically determined by the weld penetration and welding current which are the output and input of the welding process. Without the welding current waveform, the human-observed process dynamic behaviors are unable to be fully modeled. This challenges the current popular practices that use the welding process observations to derive the weld penetration. To accurately monitor the weld penetration to robotize the welding process, the welding parameter waveform may be needed. This calls for a new foundation to monitor weld penetration.

## Acknowledgment

This work is partially supported by the National Science Foundation under Grants No. CMMI-2024614, DMS-2208314, and IIS-2327113, the Institute for Sustainable Manufacturing, Department of Electrical and Computer Engineering, and Department of Mathematics at the University of Kentucky.

## Declarations

Conflict of interest: The authors declare no competing interests.

## References

- [1] Robotic Welding System Design: A Need-to-Know Guide for Automated Welding, Assembly Magazine, April 16, 2020.
- [2] Kotecki DJ, Cheever D, Howden D. Mechanism of ripple formation during weld solidification. *Welding Journal*, 1972, 51:386s– 391s.
- [3] Renwick R, Richardson RW. Experimental investigation of GTA weld pool oscillations. *Welding Journal*, 1983, 62:29s–35s.
- [4] Nagarajan S, et al., 1992. Control of the welding process using infrared sensors. *IEEE Trans Robot Autom*, 8:86–93.
- [5] Zhang, Y.M., Kovacevic, R. and Li, L., 1996. Adaptive control of full penetration gas tungsten arc welding. *IEEE Transactions on Control Systems Technology*, 4(4), pp.394-403.
- [6] Chokkalingham S, Chandrasekhar N, Vasudevan M. Predicting the depth of penetration and weld bead width from the infrared thermal image of the weld pool using artificial neural network modeling. *J Intell Manuf* 2012.
- [7] Zhang YM, Kovacevic R. Neurofuzzy model-based predictive control of weld fusion zone geometry. *IEEE Trans Fuzzy Syst* 1998, 6:389–401.
- [8] Liu Y, Zhang W, Zhang Y. Dynamic neuro-fuzzy-based human intelligence modeling and control in GTAW. *IEEE Trans Autom Sci Eng* 2015.
- [9] Liu Y, Zhang Y. Iterative local ANFIS-based human welder intelligence modeling and control in pipe GTAW process: A data-driven approach. *IEEE/ASME Trans Mechatronics* 2015.
- [10] Liang, R., Yu, R., Luo, Y. and Zhang, Y., 2019. Machine learning of weld joint penetration from weld pool surface using support vector regression. *Journal of Manufacturing Processes*, 41: 23-28.
- [11] Yu, R., et al., 2023. Deep learning based real-time and in-situ monitoring of weld penetration: Where we are and what are needed revolutionary solutions?. *Journal of Manufacturing Processes*, 93, pp.15-46.
- [12] 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.
- [13] Muclari, E. et al., 2013. "Do We Need a New Foundation to Use Deep Learning to Monitor Weld Penetration?" *IEEE Robotics and Automation Letters* (2023), 8(6): 3669 – 3676.
- [14] Asif, K., Zhang, L., Derrible, S., Indacochea, J.E., Ozevin, D. and Ziebart, B., 2022. Machine learning model to predict welding quality using air-coupled acoustic emission and weld inputs. *Journal of Intelligent Manufacturing*, pp.1-15.
- [15] 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.
- [16] Goodfellow, I., et al., 2020. Generative adversarial networks. *Communications of the ACM*, 63(11), pp.139-144.
- [17] Mirza, M. and Osindero, S., 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
- [18] Radford, A., Metz, L. and Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.
- [19] Ioffe, S. and Szegedy, C., 2015, June. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). pmlr.

- [20] Cho, K., et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- [21] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, 9(8), pp.1735-1780.
- [22] Muclari, E., Zadorozhnyy, V., Pospisil, C., Nguyen, D. and Ye, Q., 2022. Orthogonal Gated Recurrent Unit with Neumann-Cayley Transformation. *arXiv preprint arXiv:2208.06496*.
- [23] Jing L, Gulcehre C, Peurifoy J, Shen Y, Tegmark M, Soljacic M, et al. Gated Orthogonal Recurrent Units: On Learning to Forget. *Neural Comput* 2019.
- [24] Arjovsky, M., Chintala, S. and Bottou, L., 2017, July. Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214-223). PMLR.
- [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.