Development of Novel Attention-Aware Deep Learning Models and Their Applications in Computer Vision and Dynamical System Calibration

Maede Maftouni

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Industrial and Systems Engineering

Zhenyu "James" Kong, Chair

Navid Ghaffarzadegan

Xiaowei Yue

Xi Chen

May 3, 2023

Blacksburg, Virginia

Keywords: Deep Learning, Attention Mechanism, Transformer, Computer Vision, Video Object Segmentation, Manufacturing, Healthcare, System Dynamics

Development of Novel Attention-Aware Deep Learning Models and Their Applications in Computer Vision and Dynamical System Calibration

Maede Maftouni

(ABSTRACT)

In recent years, deep learning has revolutionized computer vision and natural language processing tasks, but the black-box nature of these models poses significant challenges for their interpretability and reliability, especially in critical applications such as healthcare. To address this, attention-based methods have been proposed to enhance the focus and interpretability of deep learning models. In this dissertation, we investigate the effectiveness of attention mechanisms in improving prediction and modeling tasks across different domains. We propose three essays that utilize task-specific designed trainable attention modules in manufacturing, healthcare, and system identification applications. In essay 1, we introduce a novel computer vision tool that tracks the melt pool in X-ray images of laser powder bed fusion using attention modules. In essay 2, we present a mask-guided attention (MGA) classifier for COVID-19 classification on lung CT scan images. The MGA classifier incorporates lesion masks to improve both the accuracy and interpretability of the model, outperforming state-of-the-art models with limited training data. Finally, in essay 3, we propose a Transformer-based model, utilizing self-attention mechanisms, for parameter estimation in system dynamics models that outpaces the conventional system calibration methods. Overall, our results demonstrate the effectiveness of attention-based methods in improving deep learning model performance and reliability in diverse applications.

Development of Novel Attention-Aware Deep Learning Models and Their Applications in Computer Vision and Dynamical System Calibration

Maede Maftouni

(GENERAL AUDIENCE ABSTRACT)

Deep learning, a type of artificial intelligence, has brought significant advancements to tasks like recognizing images or understanding texts. However, the inner workings of these models are often not transparent, which can make it difficult to comprehend and have confidence in their decision-making processes. Transparency is particularly important in areas like healthcare, where understanding why a decision was made can be as crucial as the decision itself. To help with this, we've been exploring an interpretable tool that helps the computer focus on the most important parts of the data, which we call the "attention module". Inspired by the human perception system, these modules focus more on certain important details, similar to how our eyes might be drawn to a familiar face in a crowded room. We propose three essays that utilize task-specific attention modules in manufacturing, healthcare, and system identification applications. In essay one, we introduce a computer vision tool that tracks a moving object in a manufacturing X-ray image sequence using attention modules. In the second essay, we discuss a new deep learning model that uses focused attention on lung lesions for more accurate COVID-19 detection on CT scan images, outperforming other top models even with less training data. In essay three, we propose an attention-based deep learning model for faster parameter estimation in system dynamics models. Overall, our research shows that attention-based methods can enhance the performance, transparency, and usability of deep learning models across diverse applications.

Acknowledgments

I am immensely grateful for the support and assistance I received during my Ph.D. My humble appreciation goes to the wonderful individuals who made this endeavor possible. First and foremost, I am wholeheartedly grateful to my advisor, Dr. Zhenyu ”James” Kong, for his unwavering support and his exceptional mentorship throughout my doctoral journey. His wisdom and patience have been instrumental in my growth and the fulfillment of this doctoral degree. I would like to express my sincere gratitude to my committee members, Dr. Navid Ghaffarzadegan, Dr. Xiaowei Yue, and Dr. Xi Chen, for their invaluable insights and constructive guidance at different stages of my Ph.D. program. It has been an honor to work with them all. I want to thank all my wonderful colleagues and lab members, especially Bo Shen, Rongxuan Wang, Andrew Chung Chee Law, Jihoon Chung, Benjamin N. Standfield, and Raghav Gnannasambandam, for their assistance in shaping and conducting the research works and contributing to writing the technical papers. I also appreciate the funding provided by the National Science Foundation (NSF) and the Office of Naval Research (ONR), and the ISE department at Virginia Tech which made my research possible. Above all, I appreciate my beloved husband, Seied Ali Safiabadi Tali, for being my constant source of strength and support amidst the challenges I faced during my Ph.D. years. Had it not been for his uninterrupted cheering and boundless love, it would have been impossible for me to complete this chapter of my life. I am also truly blessed to have my parents, sister, and brother who have endowed me with unconditional love and ever-present belief in me which have been the bedrock of my journey of realizing my dreams. Last but not least, I want to express my heartfelt appreciation to my friends whose presence has transformed my Ph.D. into a beautiful voyage enriched with countless enjoyable moments and everlasting memories.

Table of Contents

Chapter 1: Introduction

Deep learning is a subfield of artificial intelligence that has made remarkable strides in recent years, revolutionizing computer vision and natural language processing. Its groundbreaking performance has revolutionized tasks such as image classification, video segmentation, and captioning in computer vision, while in NLP, deep learning has enabled machines to process human language with remarkable accuracy and efficiency, achieving significant advancements in tasks such as machine translation, sentiment analysis, question-answering, and text summarization. These achievements have opened up new possibilities for developing cutting-edge technologies that have the potential to impact various industries and fields.

Despite the remarkable results that deep learning methods have achieved in various domains, their black-box nature presents a significant limitation. This characteristic makes it challenging to interpret how the models arrive at their decisions, raising concerns about the reliability of the models, especially in critical healthcare applications where transparency and interpretability are crucial.

In an attempt to address the interpretability limitation of deep learning, novel complementary techniques such as saliency mapping and feature visualization, have been proposed. By visualizing the pixel locations that majorly contribute to the decision, these techniques aid in understanding deep learning's prediction logic and resolving failure modes. Nonetheless, it is important to note that these techniques primarily serve as visualization tools for the prediction results of deep learning and do not create a meaningful, human-like focusing behavior in the learning process. Improving the focus can be achieved by attention-based methods inspired by the human perception system, assigning higher weights to specific surrounding attributes important for decision-making.

Attention-based methods first emerged in the field of natural language processing, where they were used to improve the performance of recurrent neural networks by selectively attending to different parts of input sequences. Since then, attention-based methods have become a powerful technique to improve the performance of deep learning models in a variety of tasks, particularly those involving long sequences or variable-length inputs. These methods generate attention-aware features that focus on the most informative parts of the input data, leading to improved accuracy and robustness of the models. Moreover, attention-based methods create attention maps that resemble human attention behavior, enhancing the interpretability of deep learning models by providing valuable insights into the regions of input that contribute most to the model's decision-making process. Overall, attention-based methods offer a promising approach to improving both the performance and interpretability of deep learning models in a range of applications.

Our three-essay dissertation investigates the effectiveness of attention mechanisms for improving prediction and modeling tasks across diverse domains. Specifically, we aim to determine how task-specific attention mechanisms can improve deep learning model performance and reliability in diverse applications.

In Essay 1, we introduce a novel computer vision tool that leverages attention modules to automatically track the melt pool in X-ray images of laser powder bed fusion. The melt pool tracking task is challenging due to factors like high noise level, illumination fluctuation over time, and lack of sharp image boundary between the molten pool and solid hot metal. We propose a video object segmentation system to track the changing size melt pool in a highly noisy background and improve the tracking accuracy through spatiotemporal attention modules. The tracked melting pool can then be used for understanding the thermomechanical behaviors of the final manufactured product. We validate the effectiveness of our attention mechanism through ablation studies on annotated X-ray image sequences from the additive manufacturing inspection dataset. The impact of the added attention mechanisms on the general video object segmentation performance can be further investigated on the standard benchmark datasets, such as DAVIS and YouTube-VOS.

The use of efficient computer-aided medical diagnosis has never been more critical, given the global extent of Covid-19 and the consequent depletion of hospital resources. Artificial intelligence (AI) powered COVID19 detection can facilitate an early diagnosis of this highly contagious disease and further reduce the infectivity and mortality rates. The preferred imaging option for COVID19 screening and diagnosis is computed tomography (CT). In Essay 2, we present a robust COVID19 classifier on lung CT scan images. Our deep learning model is a mask-guided attention (MGA) classifier that simultaneously learns to focus on lesions to improve both the accuracy and interpretability of the model, outperforming state-of-the-art models with limited training data. Additionally, we built a large and diverse Covid-19 CT scan dataset by combining open-source datasets to improve the trained network's generalizability. Extensive experimental evaluations on the dataset illustrate the improved performance and data efficiency of our proposed model. Future research could enhance the proposed methodology by incorporating clinical and paraclinical examination results and extending it to analyze CT scan volumes rather than slices.

System dynamics provides transparent models that facilitate human decision-making by enabling the modeling of the behavior of a system. Calibrating the low-level parameter values in dynamic systems is challenging due to complex nonlinear interactions. Despite the extensive application and significant time demand of parameter estimation, there are limited efficient and accurate tools for parameter estimation in system dynamics. In this regard, in Essay 3, we propose a Transformer-based model, utilizing self-attention mechanisms, for parameter estimation in system dynamics models. The approach can compete with the conventional system calibration methods of Powell and Markov Chain Monte Carlo in terms of accuracy while having a lower time requirement when used frequently. We provide a proof-of-concept example using the Epidemic modeling of Covid-19 and show the potential of DL techniques in solving the inverse problem (parameter estimation) in SD modeling. The proposed approach is especially promising when applied to large-scale complex models, noisy data, and frequent calibration needs. Future work for the proposed DL-based calibrator includes its transformation into a model-generic software package integrated into existing SD software like Vensim, leveraging active learning to enhance implementation efficiency, and studying the interpretability of the parameter estimation by exploring the model's attention using saliency techniques.

As summarized in Figure 1, the shared use of attention-aware deep learning models among these essays explores the impact of attention-based learning in solving complex problems across various fields, from additive manufacturing to medical diagnosis and system identification. As attention-aware deep learning models continue to evolve, they offer a promising solution to a diverse range of complex problems and hold significant potential for future research and applications.

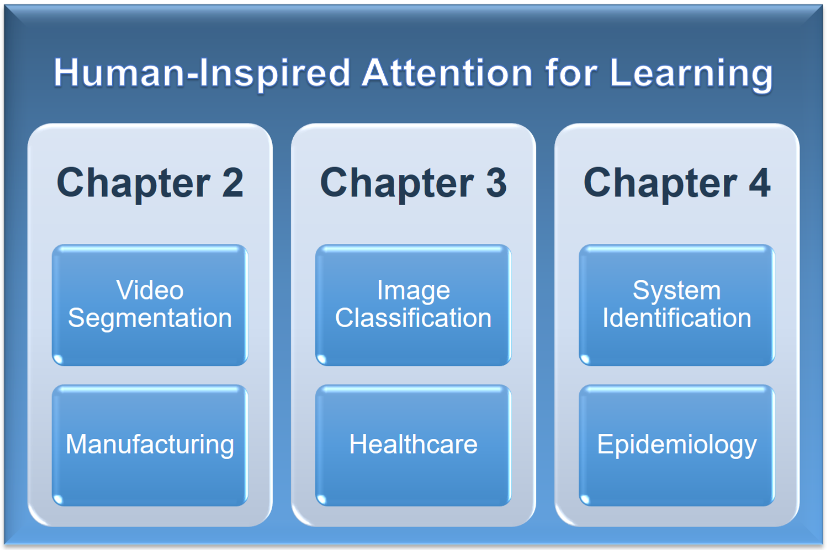


Figure 1: A high-level overview of dissertation chapters.

Chapter 2: Attention-Aware Melt Pool Video Segmentation

Abstract

Laser powder bed fusion (LPBF) is the most extensively used technique for metal additive manufacturing (AM). In LPBF, a laser is used to heat the metal powder sufficiently to form a molten pool, known as the melt pool. The size and shape of the melt pool play a critical role in determining the microstructure in additively manufactured metals. High-speed X-ray imaging has been used to study the subsurface melt pool phenomena in real time. The melt pool segmentation in X-ray images is challenging due to the high noise level, change in illumination over time, and lack of sharp image boundary. The typical methods to segment the melt pool boundary from X-ray data include manual annotating, which is time-consuming, and basic image processing techniques, which lack sufficient accuracy and robustness to noise. This paper implements a video object segmentation (VOS) deep learning model to automatically segment and track the melt pool in the X-ray image sequence for the first time in the literature. The proposed model is semi-supervised, only requiring the manual annotation of the melt pool boundary at the first frame to predict it for the rest of the video. The proposed model incorporates spatiotemporal attention modules to learn the correlations in X-ray image sequences effectively. The experimental results indicate that adopting attention modules improves the melt pool segmentation accuracy. Furthermore, by only training the model on single spot melt printing, our proposed method shows excellent extrapolation results when testing on data from other scan cases, such as linear scan printing.

1. Introduction

Metal additive manufacturing (AM) technologies have become an emerging technique to produce complex, functional components in small runs. Laser powder bed fusion (LPBF) is a common AM technique that has many applications in biomedical, aerospace, automobile, and defense [1]. Specifically, a part is printed layer-wise during the LPBF process, and each layer is filled with either hatch patterns by line scanning laser or random patterns by spot melting laser. The former is the current general practice, and the latter is a newly emerging technique that can create a preferred microstructure [2]. LPBF is complex by extreme thermal conditions from repeated layer-by-layer melting, cooling, and solidification.

The complex nature of the LPBF process makes mathematical modeling of the relationship between controllable parameters and printing quality infeasible. The practical solution is to monitor the laser keyhole's interactions with the surrounding material and previous layers in realtime. To that end, different types of vision and thermal cameras such as Charge-coupled device (CCD), complementary metal-oxide-semiconductor (CMOS), 3D scanner, High speed or high-resolution IR camera, DIC, X-ray, and fiber optics are used for additive manufacturing in-situ monitoring. Figure 2 shows one of our lab's setups for LPBF monitoring.

The molten metal region generated by laser irradiation on the powder bed surface is referred to as a melt pool. In the literature, there are different types of in-situ monitoring systems to study melt pool dynamics. The formation, behavior, and solidification of the melt pool impact the material microstructure and, thus, greatly influence the resultant properties and performance of the AM parts [3]. For example, the defects such as porosity, rough surface, residual stress, and phase grain structures in the finished parts may relate to melt pool geometries [4]. Therefore, it is vital to characterize the size and shape of melt pools during laser melting.

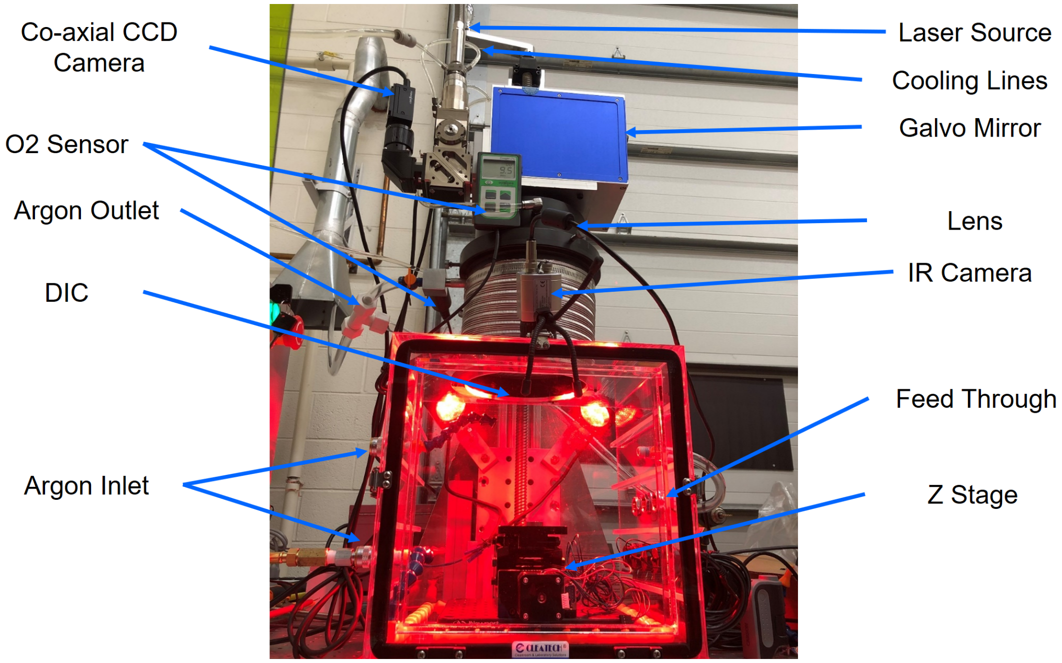


Figure 2: The LPBF layout with Multi sensing capabilities in our lab.

The imaging equipment can capture the melt pool's size, shape, consistency, and temperature distribution. This information helps to identify the uneven melt pool, unfocused laser power, and material powder contamination. High-speed visible cameras [5, 6], and infrared (IR) cameras [7] are conventional in-situ characterization tools. However, as depicted in Figure 3, they can only capture the surface of the melt pool since they cannot measure the interior of parts. LPBF is a rapid melting and cooling process (around 1000 mm/s in line scan mode and 0.5-2 ms dwell time in spot melting mode [2]), and the melt pool is relatively small (typically less than 200 µm in width or diameters). This brought tremendous challenges to in-situ melt pool monitoring due to the limited speed and spatial resolution, as well as the inability to measure the melt pool interior of the sensors. Synchrotron X-ray imaging is a unique in-situ technique to observe the solid-liquid interface beneath the surface of the melt pool during laser melting. Figure 4(a) shows one unprocessed X-ray image using the high-speed X-ray imaging system of the Advanced Photon Source (APS) at Argonne national laboratory. Due to the low-density difference between solid and liquid phases, it is also challenging to identify the solid-liquid interface in the X-ray image, as shown in Figure 4(a).

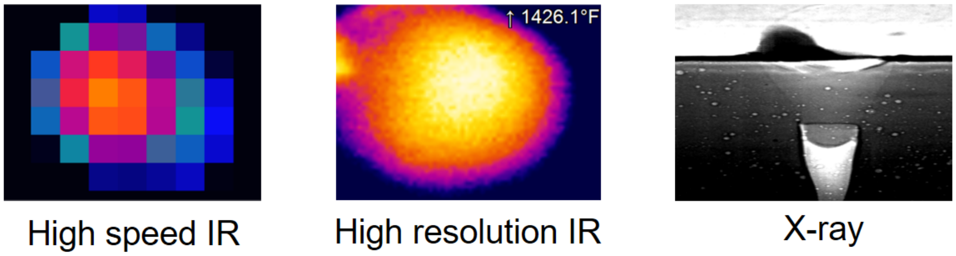


Figure 3: Imaging the surface temperature by IR camera vs the part's interior by X-ray [2].

Some preprocessing techniques [2, 8, 9] were applied to enhance the melt pool boundary before its annotation. For example, the processed X-ray image in Figure 4(b) is obtained by subtracting the X-ray image captured four frames earlier from the current one, followed by contrast enhancement. This operation can enhance the melt pool boundary, as marked in red in Figure 4(c), which can be more easily detected in the preprocessed image. However, the improved boundary contrast comes at the expense of more noise being introduced, which challenges boundary detection and segmentation accuracy. Therefore, there is a need for fast, accurate, and automatic segmentation methods for melt pool boundary detection. To this end, this paper develops a deep learning model capable of melt pool segmentation from highly noisy and low-contrast X-ray videos.

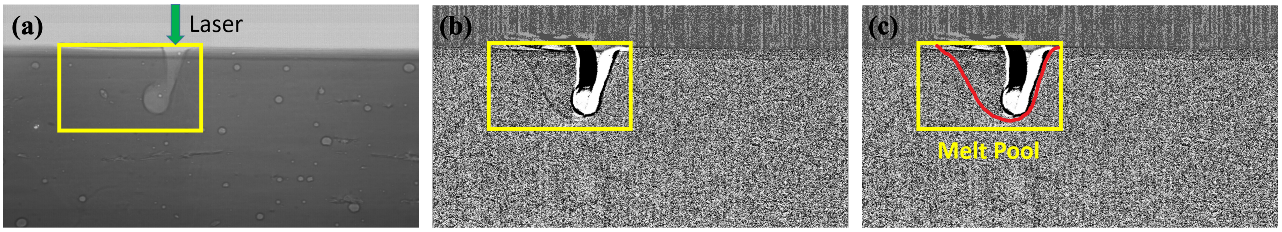


Figure 4: (a) Unprocessed X-ray image monitoring the melt pool; (b) Processed X-ray image; (c) Processed X-ray image with the manually annotated solid-liquid interface (Yellow rectangle boxes in (a-c) are the regions of the melt pool, the red boundary in (c) is the solid-liquid interface) [10].

Although image analysis automation for monitoring similar tasks such as laser welding has been studied in the past [5, 11, 12, 13], manual annotations and image interpretation are the typical choices in current industry practice, which is time-consuming and highly subject to human errors. This essay aims to improve the accuracy and speed of melt pool segmentation on X-ray images. For that purpose, we propose a semi-supervised deep learning structure capable of object segmentation from highly noisy and low contrast videos. Video-based segmentation algorithms have several theoretical advantages over image-based algorithms, incorporating both the video's spatial and temporal information.

The use of deep learning segmentation to make melt pool extraction automatic has not been investigated on LPBF X-ray images to the best of our knowledge. This is the first work on attention-based deep semantic segmentation in additive manufacturing, while the popular implementations are in the medical field, augmented reality, and self-driving cars. Concretely, the main contributions of our paper are as follows:

1. A novel automatic melt pool segmentation model is developed to facilitate the melt pool boundary detection in an X-ray sequence. The proposed method is a semi-supervised video object segmentation (VOS) that utilizes the spatiotemporal attention modules to effectively track the melt pool in both spatial and temporal domains.
2. The case study indicates that our model is robust to noise and can extract the melt pool accurately. Furthermore, by only training on single spot melt printing X-ray videos, our proposed method shows excellent extrapolation performance when testing the melt pool data with different scan cases, such as linear scan printing.

The next section will cover the related work to additive manufacturing segmentation and state of the art in video object segmentation. Next, I will present the dataset used in our study, our proposed segmentation method, and the results.

2. Related Works

In the following subsections, the related literature on melt pool monitoring and video object segmentation, aligned with the topic of melt pool segmentation, is reviewed.

2.1 Melt Pool Monitoring

Studies show a strong correlation between the solid/liquid boundary velocity and material microstructure [14]. The microstructure is directly correlated with the material's mechanical properties such as hardness and yield strength [15]. Therefore, segmenting out the melt pool streamlines the subsequent analysis of structural evolution during the laser-metal interaction.

In most previous related papers, the melt pool is monitored from CCD/CMOS axial melt pool images through binary thresholding or machine learning methods [5]. CCD and CMOS are less costly than other imaging equipment and can capture light in the visible range and only above the powder bed surface. On the other hand, to observe the melt pool solid/liquid boundary velocity, high-speed (over 10 kHz) and high-spatial-resolution (2 µm) X-ray is needed because the melt pool cooling time is short (less than 0.5 ms) and the size is small (less than 200 µm in diameter). As depicted in figure 5, segmentation of melt pool from side view X-ray images provides information on the melt pool morphology, melt flow velocity, keyhole dynamics, powder ejection velocity, and solidification rate. Therefore, there is a high demand for rapid, reliable, and automatic segmentation methods that can in turn improve the workflow efficiency in material science discovery of additive manufacturing processing.

The common practice for melt pool characterization and monitoring in X-ray data applies image processing techniques [1, 4, 8, 9]. Zhao et al. [4] studied LPBF X-ray images to find the characteristics of conduction and keyhole melting modes and how they are related to laser power. They used simple image processing techniques such as dividing the images by the frame at time zero (when no melting occurs) for contrast enhancement. The authors

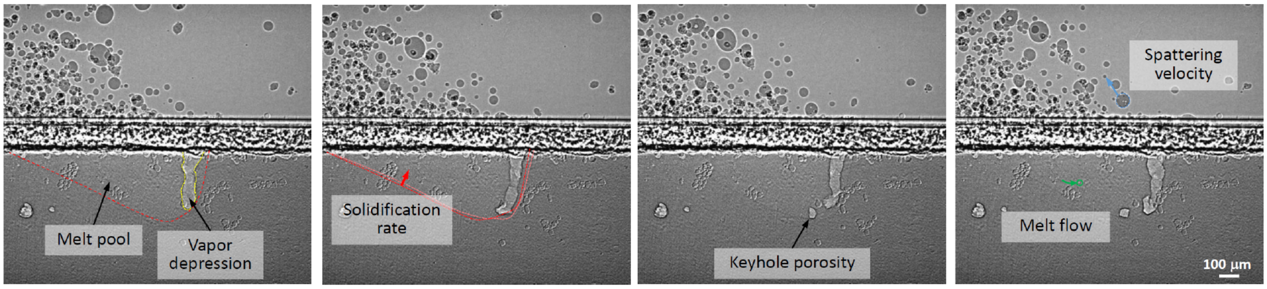


Figure 5: Visualization of melt pool morphology, solidification rate, generation of keyhole pore, melt flow velocity, and powder ejection velocity on X-ray images [1].

then used the image intensity profile for melt pool segmentation approximation and the fast compressive tracking technique for finding the powder ejection trajectory. [8] used X-ray images from the LPBF process to investigate the property variation of parts produced under constant input energy density. The authors showed the separate role of scan speed and laser power in shaping the melt pool, basing their analysis on the average melt pool length, width, and depth under different settings. ImageJ software was used to enhance the contrast of the solid-liquid interface by dividing frames. They characterized four melt pool modes: no melt pool, conduction, transition (or shallow depression), and keyhole (or deep depression), showing that depression zone depth is connected to laser power and its width depends on scan speed.

Image processing techniques for edge detection employ user-defined algorithms, which are tailored methods devised or selected by users to adeptly tackle challenges such as noise, varying contrast, and object complexity. These algorithms demonstrate considerable sensitivity to parameter choices and are highly dependent on the particular application context. Additionally, they are not robust to the low and variable edge contrast, low signal-to-noise ratio, and discontinuities of image edge. On the other hand, the recent deep learning segmentation techniques make it possible to directly learn a robust set of visual features for the task and require less fine-tuning of hyper-parameters. However, melt pool edge detection is challenged by the low-density difference between the liquid and the solid, resulting in low edge contrast. Therefore, there is a need to streamline the melt pool segmentation and tracking by an automatic computer vision tool.

So far, the deep learning-based melt pool segmentation has not been widely investigated on LPBF X-ray images. However, some work has applied the machine learning approach in their melt pool analysis of other image types. For example, Ref. [5] presented a machine learning-based in-situ melt pool classifier on visible light high-speed camera images. The authors used Scale Invariant Feature Transform (SIFT) for feature extraction and Histogram of Oriented Gradients (HOG) to build visual fingerprints for both spatters and melt pools, and SVM for classification of melt pool appearances. They note that improved representation learning is possible using Deep Learning techniques. pools. Ref. [16] developed a deep learning-based melt pool classification method to predict melt pool size. Based on a convolutional neural network, a classifier was trained with melt pool images captured from a high-speed camera. In [7], the melt pool boundary was extracted on the coaxial infrared images using the temperature gradient distribution of Ti-6Al-4V. Then, the average melt pool width was predicted based on the extracted melt pool. They denoised their highly noisy images using Fast non-local means. [17] presented a convolutional neural network to predict laser power on high-speed camera melt pool image. Khanzadeh et al. [18] proposed in-situ monitoring of melt pool thermal profile images for porosity prediction. They clustered melt pools based on the similarity of their thermal distributions, and mark the cluster with a small number of melt pools as the anomalies, assumed equivalent to the microstructure anomalies.

2.2 Video Object segmentation

Convolutional Neural Network (CNN) has been revolutionary in the computer vision field, first proved successful in the image classification task. CNN is composed of two main parts: representation learning (feature extraction) and fully connected layers for classification. CNN is powerful because it can automatically learn the most relevant features at different levels of representation. These features can be fine-tuned for tasks other than classification, including object detection and segmentation.

Semantic segmentation can be thought of as a dense pixel-wise classification. The challenge in semantic segmentation is that it requires classification, localization, and exact boundary detection all at the same time. Classification should be transformation-invariant while localization should be transformation-sensitive [19]. This contradictory network requirement makes semantic segmentation more challenging than either of the tasks.

Long et al. paper [20] on Fully Convolutional Network (FCN) sparked a breakthrough in image semantic segmentation exceeding the state-of-the-art accuracy and computational efficiency. They formulated the dense feature extractor idea in [21] to substitute the fully connected portion of CNN with convolutional layers to make large feature maps. The fully convolutional network will keep spatial information and learns the representation that suits the pixel-wise dense prediction task. The FCN network structures often consist of two parts: the encoder that successively reduces the feature map's spatial dimension to learn the contextual features and the decoder that gradually reconstructs the segmentation mask image through the upsampling and deconvolution layers. To retain fine features and location information lost in the feature extraction process, skip connections connects the corresponding convolutional layers between the encoder and the decoder.

One-Shot Video Object Segmentation introduced by S. Caelles et al. [22], is the first FCN architecture used for video object segmentation and has inspired many following works in recent years. One-shot VOS is a semi-supervised segmentation approach that proposes to train an FCN on the general task of separating the foreground object from the background and then fine-tune the trained model for a specific object we are interested in at the test time. The trained model will be further trained (fine-tuned) on the first frame of its ground truth mask (hence the name one-shot) and then tested on the rest of the sequence. The user is free to choose the trade-off between accuracy and speed by selecting the level of fine-tuning. They simplify video segmentation into processing frames independently without the explicit use of time information. They argue that this approach avoids propagating errors temporally, especially in occlusions and abrupt motion.

Recent advancements in computer vision, potential applications in self-driving cars, augmented reality, and the appearance of publicly available annotated benchmark datasets (such as SegTrack (2013), DAVIS (2016, 2017), and Youtube-VOS (2018)) have led to a surge of interest in Video Object Segmentation (VOS). These segmentation challenges aim at separating an object of interest from the background in an entire video as accurately as possible. Video Object Segmentation can be categorized from different aspects into semi-supervised (given the first frame ground truth mask)[22, 23, 24, 25, 26, 27] vs. unsupervised, Considering the temporal dimension vs. processing frames independently [22, 26, 28], fully end-to-end trainable vs. requiring post-processing, multiple vs. single object segmentation, and relying on object proposals vs. attention-based.

The simplified approach to VOS is to segment the object in each frame separately. U-Net architecture [30] is a widely used light-weight image segmentation network, mainly applied in the medical field. U-Net has skip connections between the encoder and decoder to combine low-level location and higher-level semantic information and give a more precise pixel-level localization. U-Nets offer good performance even on low contrast noisy images. They also have lower training data requirements and use GPU memory efficiently [31]. Using the convolutional part of a pretrained on ImageNet classification network such as VGG-11 (depicted in Figure 6), VGG-16, and ResNet [29, 32] as the encoder has proven successful to improve performance and reduce the required data.

While early VOS work did not consider the video's temporal aspect, different approaches

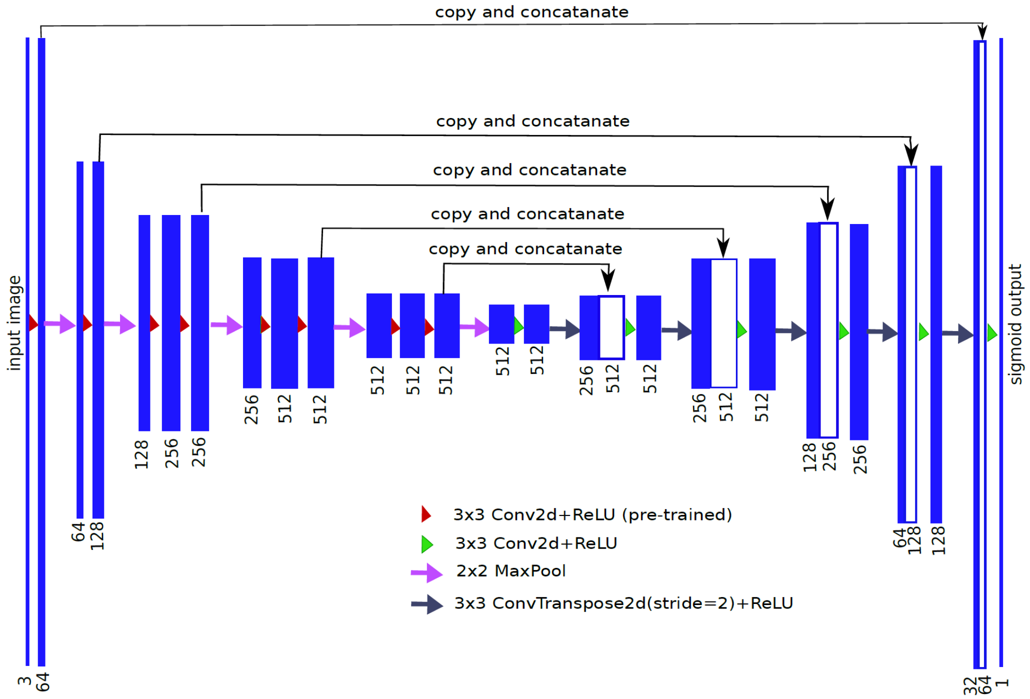


Figure 6: U-net with VGG-11 encoder (pre-trained on ImageNet)[29].

have been proposed to leverage temporal information. In [33, 34, 35], optical flow models temporal dependencies, which slows down the computation. Alternatively, 3D convolutional [36] and Recurrent Neural Networks (RNNs) [37, 38], can learn spatio-temporal features from the video sequence in an end-to-end manner.

An alternative approach to incorporate temporal information is to propagate some of the previous frames and their predicted masks as an additional resource for current frame mask prediction. Figure 7 illustrates possible approaches to VOS from the perspective of learning from previous frames. One approach (Figure 7(a))) is only to propagate the last predicted mask as a guide to gain temporal coherence [24, 39]. The problem in such approaches is error propagation. Next approach (Figure 7(b))) is to use the first frame to guide object detection [22, 40]. In contrast to the previous one, this method cannot work when there is a drastic change in object appearance over time. Using both the previous and first frame as a guide for object mask in the current frame has been proven successful in [27, 39] (Figure 7(c)), improving accuracy and reducing the run time. They reached temporal consistency while avoiding error propagation in time by consolidating mask propagation with representation learning by using both the last predicted mask and the first frame embedding for the current frame mask prediction. The most recent trend is to use the memory from more frames (possibly all the frames) (Figure 7(d)). This approach uses spatio-temporal attention to decide which frames, and where inside those frames, are the most relevant for the current frame segmentation.

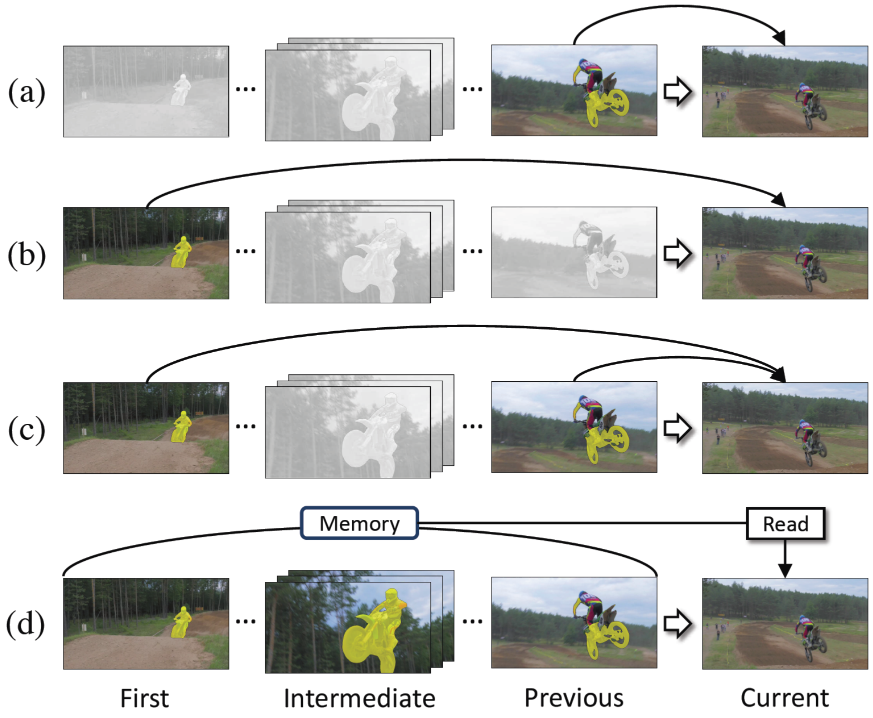


Figure 7: Comparing possible approaches to VOS based on the use of previous frames [41].

There exists an emerging trend of incorporating the attention mechanism in the area of image semantic segmentation [42, 43] as well as video object segmentation [44, 45]. The attention mechanism's successful implementation first emerged as an additional weight module over the encoder decoder-based neural machine translation system in natural language processing (NLP) [46, 47]. Focusing on specific regions is a desirable feature in many applications and is consistent with the human selectively attending perception generated in the prefrontal cortex. After showing promising results in NLP, various attention mechanisms were quickly adopted in image captioning, action recognition, and generative adversarial network. Attention mechanisms were used inside convolutional networks in [48, 49]. In the video object segmentation, the commonly used attention mechanisms can be categorized into four approaches: co-attention (capture the correlation across different frames) [45], spatial attention (prioritization of an area within the visual field and overlooking the background clutter) [44, 50], Channel Attention (for selecting semantic attributes such as motion information) [51], human visual attention (using human visual gaze with eye-tracking)[52], and temporal attention (directing attention to specific point in time) [41].

Given the excellent performance of VOS, we propose a semi-supervised deep learning structure capable of melt pool segmentation from highly noisy and low-contrast X-ray videos. The locality of features, high noise level, and lack of sharp melt pool boundary, especially at the end of the video, call for well-focused and spatiotemporal attention-aware features. In the rest of this essay, our new semi-supervised mask propagating video object segmentation framework, incorporating attention modules, is presented and applied for melt pool tracking.

3. Our Dataset

The X-ray data used in this work were collected at the 32-ID-B beamline of the Advanced Photon Source (APS), Argonne National Lab [53]. APS is a synchrotron facility that can generate powerful X-ray irradiation for high-speed in-situ monitoring. Beamline 32-ID-B is equipped with a laser powder bed fusion simulator that can perform single-layer melting. During the experiment, a YAG laser beam with 1070 nm wavelength was shot on the top of a 3 mm (H) × 50 mm (L) × 0.5 mm (T) sample made of Ti-6Al-4V (Ti 64) to simulate the selective laser melting (SLM) process. At the same time, the X-ray beam was projected through the thickness side of the sample to monitor the melt pool generated by the laser. The X-ray can distinguish the melt pool and substrate because the density between the liquid and solid material is different. After penetrating the substrate, the X-ray then illuminates a scintillator, and the resulting pattern is collected by a high-speed camera with 2µm spatial resolution at 70 kHz [2].

Table 1 specifies the dataset split between training, validation, and testing used in Section 5. The method uses single spot melt videos for training. We included the line scan videos in the test split to assess the generalizability of our model. Additionally, data augmentation methods, including random flipping, random cropping, and reversal of the sequence, are applied to the training set to avoid overfitting.

Table 1: Number of images from each of the classes in each of the splits.

	Train		**Validation**		**Test**	
Type	Video	Frame	Video	Frame	Video	Frame
Single Spot Melt	9	1,001	1	129	2	147
Line Scan	-	-	-	-	2	408

4. Method

The overall proposed methodology consists of two steps, as outlined in Figure 8. Step 1 (Section 4.1) applies a set of preprocessing steps on the image sequence to improve the melt pool edge contrast. Step 2 (Section 4.2) develops a segmentation model to track the melt pool in the processed X-ray image sequence.

4.1 X-ray Frames Preprocessing

The preprocessing step aims to enhance the melt pool boundary contrast as it is barely visible in the original x-ray image. As illustrated in the upper part of Figure 8, we are applying the following five transformations on the raw X-ray frames:

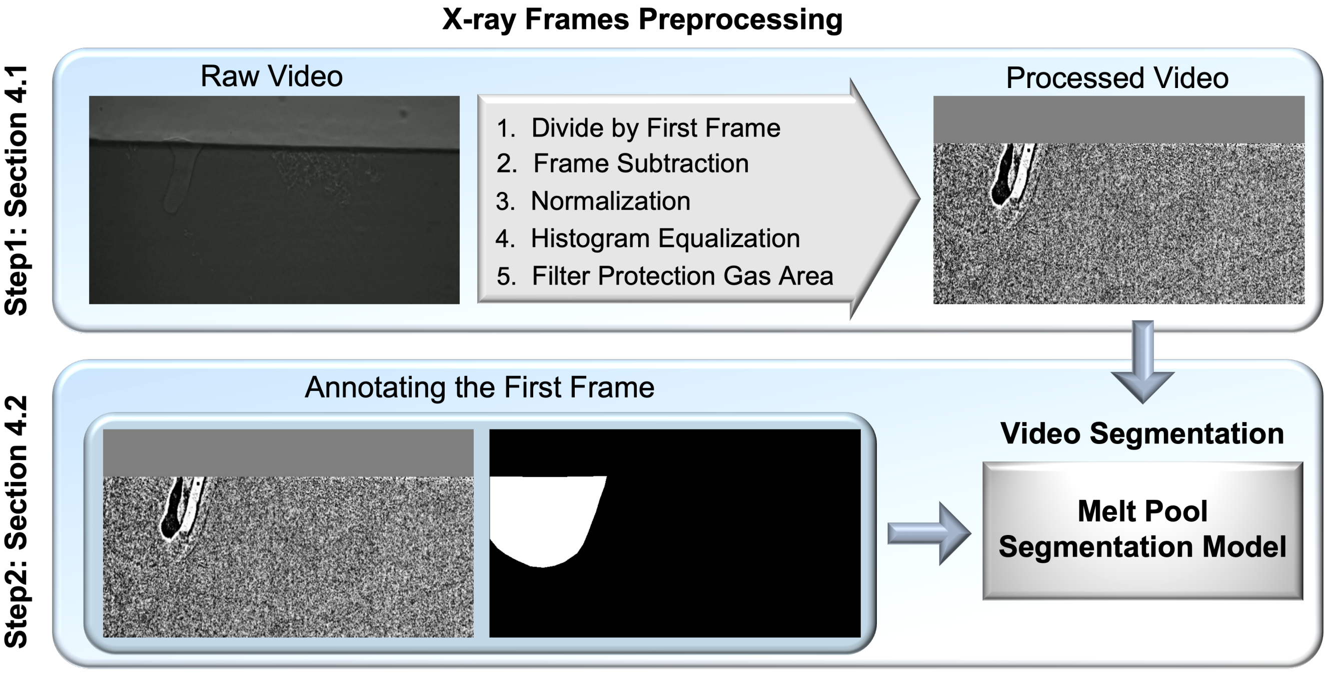


Figure 8: The two main steps of the proposed method.

1. Divide by First frame: all images in the sequence are divided by the first frame (background frame) to enhance the melt pool features.
2. Frame Subtraction: For each image X_t at timestep t from Step 1, the new image is obtained by the subtraction operation, namely, $Y_t = X_t - X_{t-4}$, with the 4-frame lag determined through ablation studies for optimal performance.
3. Normalization: All the Y_t frames have their pixel values normalized between 0 and 1.
4. Histogram Equalization: Histogram equalization is applied on under-exposed and low-contrast images to gain a higher contrast by stretching out the image intensity range.
5. Filter Protection Gas Area: In the absence of physical filtering, a thick metal piece that blocks the X-ray, the protection gas area filtering, should be applied to the image.

The procedure involves several common image processing techniques that are aimed at improving image quality and extracting relevant information from the images. These techniques include dividing the image sequence by the first frame to enhance contrast, frame subtraction to highlight changes over time, normalization to standardize pixel values, histogram equalization to enhance contrast, and filtering to remove artifacts. These techniques are widely used and proven effective in various image processing applications.

4.2 Melt pool Segmentation in the X-ray Video

Segmentation may be thought of as a pixel-wise classification that performs object localization and boundary detection tasks simultaneously. As shown in Figure 8, the melt pool segmentation model takes the preprocessed X-ray image sequence from the first step and the melt pool segmentation mask of the first frame, annotated by the user. Therefore, our proposed segmentation model is a semi-supervised model that allows high prediction performance from a small amount of training data. Inspired by [39], our melt pool segmentation model, shown in Figure 9, employs a Siamese encoder-decoder network structure consisting of the following components.

4.2.1 Siamese Encoder

The Siamese encoder consists of two identical ResNet50 networks. The upper stream is the target stream, taking the current frame and previously predicted mask as the input. The previous masks are from the earlier predictions of the model, except the first frame, which is the ground truth. The lower stream is the reference stream, taking the first frame and its corresponding mask as the input. These two streams help the model simultaneously learn from both the previously predicted and the first frame masks. The encoder weights are initialized from the ImageNet pre-trained model. They are modified to take four channels (three of them from the frame image and one from the mask) at the first convolution. Each stream consists of 5 residual blocks that spatially downsample its input and learn more filters, according to the scales and number of channels denoted in Figure 9. The feature embedding at a scaled-down image size can localize the melt pool, while fine edges are better captured at a scaled-up image size.

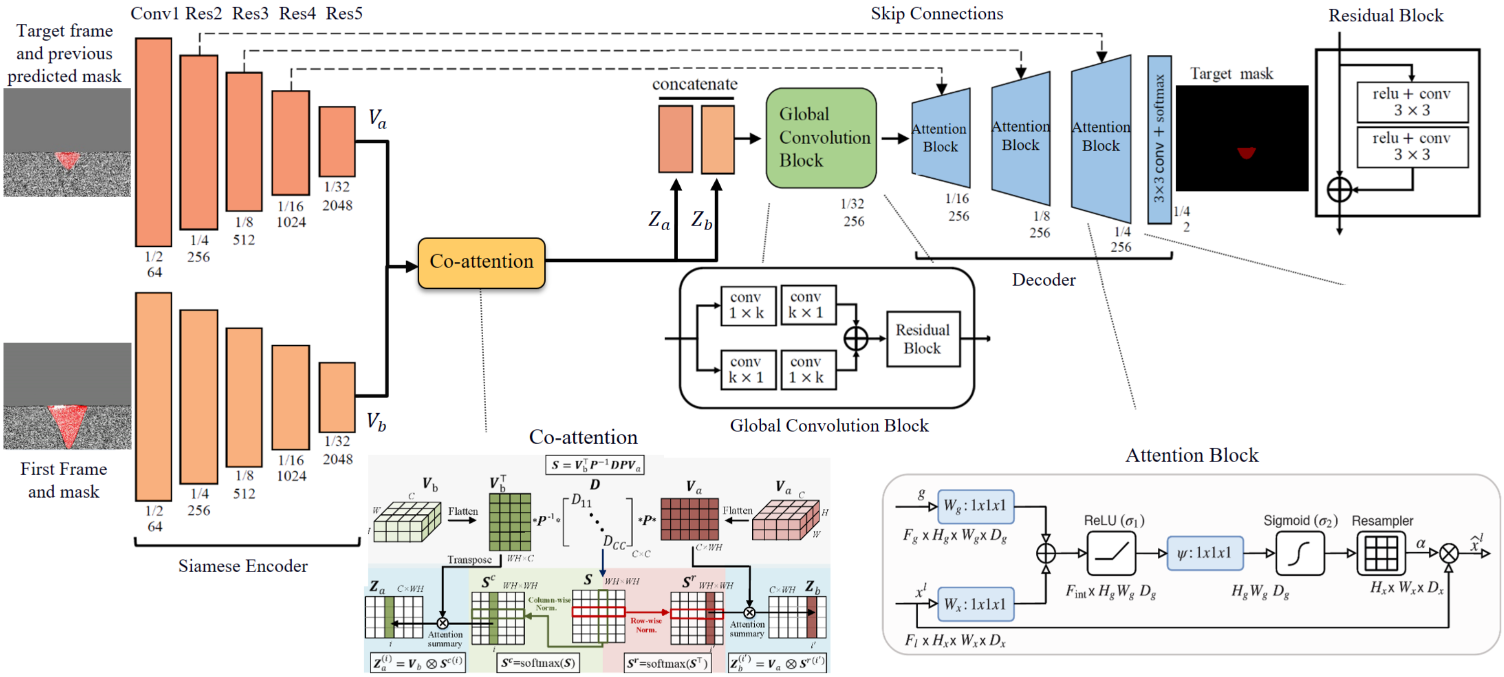


Figure 9: Attention-based reference-guided mask propagating video object segmentation

4.2.2 Co-Attention Module

The Co-Attention module [45] learns to encode the correlations between the resultant embedding tensors from the flattened target (V_a) and reference (V_b) representations and leverages their similarity matrix (S) to encode their correlations. Specifically, the co-attention module encodes the correlations between V_a and V_b and outputs the co-attention enhanced feature as follows:

$$S = V_b^T W V_a = V_b^T P^{-1} D P V_a \tag{1}$$

$$S^c = \text{Softmax}(S) \quad , \quad S^r = \text{Softmax}(S^T) \tag{2}$$

$$Z_a = V_b S^c \quad , \quad Z_b = V_a S^r \tag{3}$$

where W is a square weight matrix decomposed into an invertible matrix (P) and a diagonal matrix (D). Co-attention takes the two encoder branches' feature maps, V$_a$ and V$_b$ and learns their correlations (S, Equation 1). The similarity matrix S is then normalized row-wise (S^r) and column-wise (S^c) with a Softmax function to generate attention weights (Equation 2), and multiplied by the flattened embedding tensor (V_a, V_b) to generate the co-attention enhanced features (Z_b, Z_a) (Equation 3).

4.2.3 Global Convolution Block

The global convolution block (GCB) [19] performs the global feature matching to localize the object in the current frame. GCB utilizes a combination of 1×k + k×1 and k×1 + 1×k convolutional layers to enlarge the receptive field for better localization.

4.2.4 Decoder

Finally, the decoder, consisting of attention blocks [31], produces the target mask. The attention block amplifies and upsamples the object mask's relevant features and details by combining the GCB output with skip-connected target encoding at three scales. Finally, a convolutional layer produces the current frame mask prediction from the output of the last attention block. The final predicted mask is 1/4 of the image size. Improving the output resolution can improve the melt pool edge and is a future direction to be considered.

Instead of the commonly used cross-entropy loss function (Equation 4), we have used Lovasz-Softmax loss function ([54], Equation 8). Cross entropy loss evaluates the class predictions pixel-wise and outputs their average:

$$\text{CE Loss}(f) = -\frac{1}{p}\sum_{i=1}^{p} \log f_i(y_i^*) \tag{4}$$

$$f_i(c) = \text{Softmax}(F_i(c)) = \frac{e^{F_i(c)}}{\sum_{c' \in C} e^{F_i(c)}} \qquad \forall i \in [1, p], \forall c \in C. \tag{5}$$

Where f represents $f_i(c)$ a vector of Softmax normalized class probabilities outputted by the network(Equation 5), p denotes the number of pixels, y_i^* is the i_{th} pixel's ground truth class in a given image, and $f_i(y_i^*)$, is the probability assigned to the correct pixel's class.

Jaccard index or intersection-over-union (IOU) is a segmentation performance measure. Given the vector of ground truth y^* and predicted $\tilde{y}$ labels, the Jaccard index of class c and its corresponding risk function are defined as in Equations 6 and 7. Equation 8 is Lovasz-Softmax loss function, which is the continuous and convex surrogate function of the Jaccard index. $m_i(c)$ is the per-pixel errors and $\overline{\Delta}$ is the tight convex closure of Δ. Jaccard is chosen here for its superior performance, especially for small objects.

$$J_c(y^*, \tilde{y}) = -\frac{|\{y^* = c\} \cap \{\tilde{y} = c\}|}{|\{y^* = c\} \cup \{\tilde{y} = c\}|} \tag{6}$$

$$\Delta_{J_c}(y^*, \tilde{y}) = 1 - J_c(^*, \tilde{y}) \tag{7}$$

$$\text{Lovasz-Softmax loss}(f) = \frac{1}{C} \sum_{c \in C} \overline{\Delta_{J_c}}(m(c)) \tag{8}$$

$$m_i(c) = \begin{cases} 1 - f_i(c), & \text{if } c = y_i^*, \\ f_i(c), & \text{otherwise} \end{cases} \tag{9}$$

5. Results

Figure 10 provides the qualitative comparison of the predicted masks on a test video sequence. The ground truth masks are obtained by manually marking the visible boundary of the melt pool in the X-ray images using an image annotation tool. To evaluate the effectiveness of our proposed attention-based model, we compared its performance with the Reference-Guided Mask Propagation (RGMP) model proposed in [39], as the base model without the attention and co-attention modules. To further examine the impact of the attention module, we performed an ablation study by modifying the RGMP architecture with two changes: replacing the refinement modules with attention blocks and incorporating the co-attention module before concatenating the embeddings of the two branches.We observe

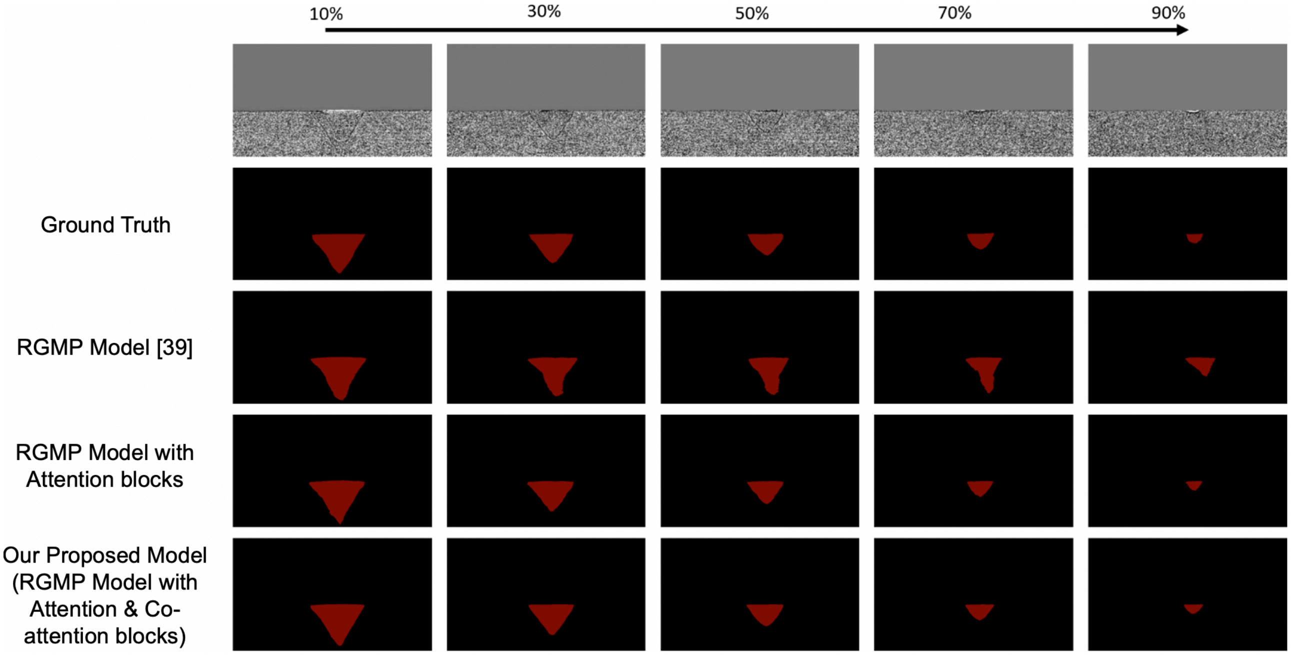

10%
30%
50%
70%
90%
Ground Truth
RGMP Model [39]
RGMP Model with Attention blocks
Our Proposed Model (RGMP Model with Attention & Co-attention blocks)

Figure 10: The melt pool segmentation results of different networks over the video sequence

that, our proposed model with co-attention and attention blocks generate masks that more closely resemble the ground truth masks, as shown in Figure 10. Additionally, quantitative metrics, such as the IOU (Equation 10), Dice coefficient (Equation 11), and frame per second (FPS), are used to benchmark the models' performance, which are averaged over the test set and summarized in Table 2. Our experimental results confirm that incorporating the attention module in our proposed model leads to improved melt pool boundary prediction capabilities compared to the baseline models.

$$\text{Jaccard Index} = \frac{TP}{TP + FP + FN} \tag{10}$$

$$\text{Dice score} = \frac{2 \times TP}{(TP + FP) + (TP + FN)} \tag{11}$$

All the model were trained using an Adam optimizer with a learning rate of 1e-5 and truncated back-propagation through time (BPTT) with step 5 and length 25. We used a batch

Table 2: Melt Pool segmentation performance Comparison (the best performance in each metric is in red).

Model	**IOU (%)**	**Dice (%)**	**FPS**
RGMP Model [39]	71.39	82.67	**60**
RGMP Model with Attention Block	77.44	86.93	56
RGMP Model with Co-Attention and Attention Block (Proposed)	**78.29**	**87.60**	55

size of 1 and trained the network for 100 epochs, saving the weights with the best validation IOU performance. These hyperparameters are tuned using Bayesian Optimization. We used 1 NVIDIA GeForce RTX 2080 Ti GPU and Pytorch library to train our models.

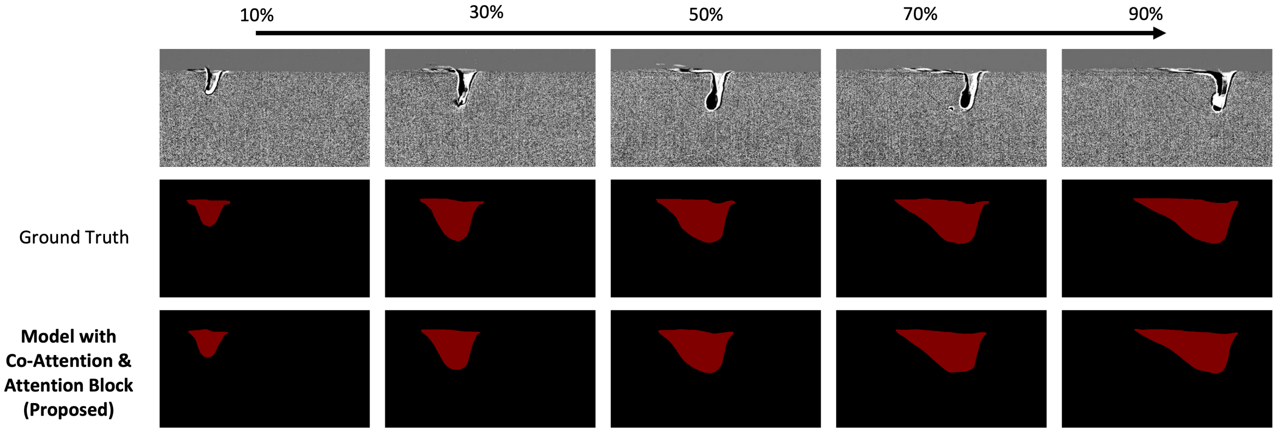


Figure 11: The melt pool segmentation results over a line scan video sequence

Our results show the high accuracy and speed of our proposed deep learning network for melt pool segmentation from highly noisy and low contrast X-ray videos for the first time. Our video-based segmentation model's high performance results from incorporating both the spatial and temporal information in the video. Additionally, our model shows a good generalization performance when tested on a video sequence from a different experimental setup. Specifically, Figure 11 shows the segmentation results of our method on the video of 3D printing using a line scan. Even though our model has been trained on single spot melt videos, it could successfully extrapolate to predict the melt pool in the line scan videos.

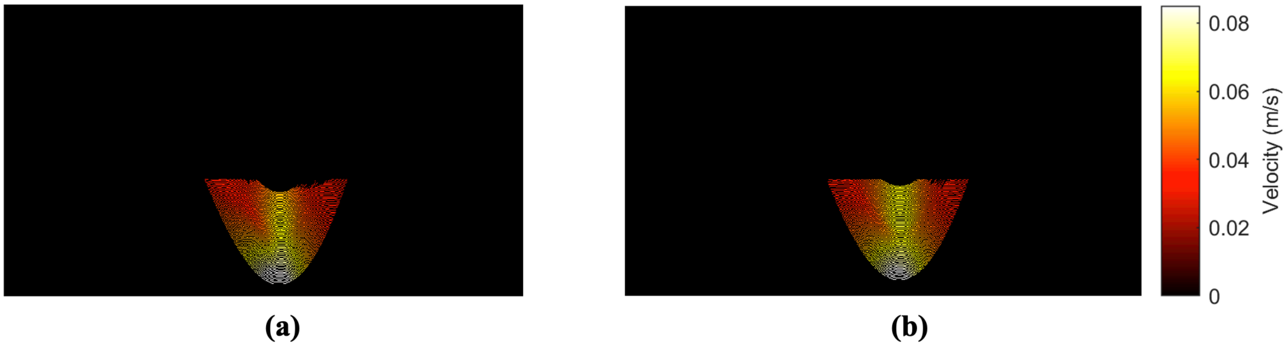


Figure 12: The velocity maps calculated from the (a) ground truth and (b) prediction masks.

The proposed method can be applied to extract the solidification (Solid/liquid boundary) velocity from the melt pool boundary and provide a reliable approach for microstructure prediction. The melt pool boundary can be characterized by the lower part of the predicted mask boundary. Since X-ray images are time-series data, we first use the ”locally weighted scatter plot smooth” method from the Curve Fitting Toolbox in MATLAB to fit a 3D surface. Next, the solidification velocity can be calculated following the same approach as [4]. The velocity maps calculated from the ground truth mask and prediction mask are shown in Figures 12(a) and 12(b), respectively. The two velocity maps look almost the same visually. Cooling down velocity is pointed to the keyhole center in both cases.To compare them quantitatively, relative root error (RRE), namely, $\frac{\|X-\hat{X}\|_F}{\|X\|_F}$, is used. The RRE calculated based on the locations where both velocity maps have non-zero values is 0.0966. This result validates that using our predicted melt pool boundary can provide a very close melt pool characterization as the work-intensive manual annotation.

6. Conclusion and Future Direction

Monitoring the melt pool’s size and shape is significant for the fabrication quality control and microstructure prediction of complex parts through LPBF process. This essay presents an automatic segmentation method to track the melt pool boundary in an SLM process sequence of X-ray images. The speed, accuracy, and robustness of our proposed automatic segmentation method can significantly improve workflow efficiency in additive manufacturing analysis. Our method utilizes a semi-supervised VOS approach that only requires the melt pool annotation in the first frame to segment the melt pool boundary throughout the X-ray sequence. Our proposed network incorporates the first frame and the previous frame embeddings in output mask prediction in tandem. The promising performance and generalization of our proposed method demonstrated the significant potential of using the VOS recent advancements for additive manufacturing.

The future work includes investigating the impact of the encoder backbone choice and incorporation of plug-in modules (such as Atrous Spatial Pyramid Pooling (ASPP) and feature pyramid network (FPN)) on the video object segmentation performance, resolution, and speed. The work can also be expanded to simultaneously segment the melt pool on IR and X-ray images, to correlate the melt pool shape with the thermal characteristics in real time.

References

[1] N. Parab, C. Zhao, R. Cunningham, L. I. Escano, K. Fezzaa, A. Rollett, L. Chen, and T. Sun, "In situ characterization of laser powder bed fusion using high-speed synchrotron x-ray imaging technique," *Microscopy and Microanalysis*, vol. 25, no. S2, pp. 2566–2567, 2019.

[2] R. Wang, D. Garcia, R. R. Kamath, C. Dou, X. Ma, B. Shen, H. Choo, K. Fezzaa, H. Z. Yu, and Z. J. Kong, "In situ melt pool measurements for laser powder bed fusion using multi sensing and correlation analysis," *Scientific Reports*, vol. 12, no. 1, pp. 1–17, 2022.

[3] T. DebRoy, H. Wei, J. Zuback, T. Mukherjee, J. Elmer, J. Milewski, A. M. Beese, A. d. Wilson-Heid, A. De, and W. Zhang, "Additive manufacturing of metallic components–process, structure and properties," *Progress in Materials Science*, vol. 92, pp. 112–224, 2018.

[4] C. Zhao, K. Fezzaa, R. W. Cunningham, H. Wen, F. De Carlo, L. Chen, A. D. Rollett, and T. Sun, "Real-time monitoring of laser powder bed fusion process using high-speed x-ray imaging and diffraction," *Scientific reports*, vol. 7, no. 1, pp. 1–11, 2017.

[5] C. Knaak, G. Kolter, F. Schulze, M. Kröger, and P. Abels, "Deep learning-based semantic segmentation for in-process monitoring in laser welding applications," in *Applications of Machine Learning*, vol. 11139. International Society for Optics and Photonics, 2019, p. 1113905.

[6] L. Scime and J. Beuth, "Using machine learning to identify in-situ melt pool signatures indicative of flaw formation in a laser powder bed fusion additive manufacturing process," *Additive Manufacturing*, vol. 25, pp. 151–165, 2019.

[7] L. Zheng, Q. Zhang, H. Cao, W. Wu, H. Ma, X. Ding, J. Yang, X. Duan, and S. Fan, "Melt pool boundary extraction and its width prediction from infrared images in selective laser melting," *Materials & Design*, vol. 183, p. 108110, 2019.

[8] Q. Guo, C. Zhao, M. Qu, L. Xiong, L. I. Escano, S. M. H. Hojjatzadeh, N. D. Parab, K. Fezzaa, W. Everhart, T. Sun *et al.*, "In-situ characterization and quantification of melt pool variation under constant input energy density in laser powder bed fusion additive manufacturing process," *Additive Manufacturing*, vol. 28, pp. 600–609, 2019.

[9] Q. Guo, C. Zhao, M. Qu, L. Xiong, S. M. H. Hojjatzadeh, L. I. Escano, N. D. Parab, K. Fezzaa, T. Sun, and L. Chen, "In-situ full-field mapping of melt flow dynamics in laser metal additive manufacturing," *Additive manufacturing*, vol. 31, p. 100939, 2020.

[10] B. Shen, R. R. Kamath, H. Choo, and Z. Kong, "Robust tensor decomposition based background/foreground separation in noisy videos and its applications in additive manufacturing," *IEEE Transactions on Automation Science and Engineering*, 2022.

[11] M. Grasso, V. Laguzza, Q. Semeraro, and B. M. Colosimo, "In-process monitoring of selective laser melting: spatial detection of defects via image data analysis," *Journal of*

Manufacturing Science and Engineering, vol. 139, no. 5, 2017.

[12] A. Olabi, G. Casalino, K. Benyounis, and M. Hashmi, "An ann and taguchi algorithms integrated approach to the optimization of co2 laser welding," *Advances in Engineering Software*, vol. 37, no. 10, pp. 643–648, 2006.

[13] J.-Y. Jeng, T.-F. Mau, and S.-M. Leu, "Prediction of laser butt joint welding parameters using back propagation and learning vector quantization networks," *Journal of Materials Processing Technology*, vol. 99, no. 1-3, pp. 207–218, 2000.

[14] R. Acharya, J. A. Sharon, and A. Staroselsky, "Prediction of microstructure in laser powder bed fusion process," *Acta Materialia*, vol. 124, pp. 360–371, 2017.

[15] R. M. Mahamood, E. T. Akinlabi, M. Shukla, and S. Pityana, "Scanning velocity influence on microstructure, microhardness and wear resistance performance of laser deposited ti6al4v/tic composite," *Materials & design*, vol. 50, pp. 656–666, 2013.

[16] Z. Yang, Y. Lu, H. Yeung, and S. Krishnamurty, "Investigation of deep learning for real-time melt pool classification in additive manufacturing," in *2019 IEEE 15th international conference on automation science and engineering (CASE).* IEEE, 2019, pp. 640–647.

[17] O. Kwon, H. G. Kim, W. Kim, G.-H. Kim, and K. Kim, "A convolutional neural network for prediction of laser power using melt-pool images in laser powder bed fusion," *IEEE Access*, vol. 8, pp. 23 255–23 263, 2020.

[18] M. Khanzadeh, S. Chowdhury, M. A. Tschopp, H. R. Doude, M. Marufuzzaman, and L. Bian, "In-situ monitoring of melt pool images for porosity prediction in directed energy deposition processes," *IISE Transactions*, vol. 51, no. 5, pp. 437–455, 2019.

[19] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4353–4361.

[20] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[21] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2012.

[22] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, “One-shot video object segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 221–230.

[23] V. Jampani, R. Gadde, and P. V. Gehler, “Video propagation networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 451–461.

[24] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, “Learning video object segmentation from static images,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2663–2672.

[25] P. Tokmakov, K. Alahari, and C. Schmid, “Learning video object segmentation with visual memory,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4481–4490.

[26] P. Voigtlaender and B. Leibe, “Online adaptation of convolutional neural networks for video object segmentation,” *arXiv preprint arXiv:1706.09364*, 2017.

[27] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos, “Efficient video object segmentation via network modulation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6499–6507.

[28] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, “Video object segmentation without temporal information,” *IEEE trans-*

actions on pattern analysis and machine intelligence, vol. 41, no. 6, pp. 1515–1530, 2018.

[29] V. Iglovikov and A. Shvets, “Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation,” *arXiv preprint arXiv:1801.05746*, 2018.

[30] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[31] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, “Attention u-net: Learning where to look for the pancreas,” *arXiv preprint arXiv:1804.03999*, 2018.

[32] A. A. Shvets, A. Rakhlin, A. A. Kalinin, and V. I. Iglovikov, “Automatic instrument segmentation in robot-assisted surgery using deep learning,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 624–628.

[33] L. Bao, B. Wu, and W. Liu, “Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5977–5986.

[34] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, “Segflow: Joint learning for video object segmentation and optical flow,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 686–695.

[35] S. D. Jain, B. Xiong, and K. Grauman, “Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos,” in *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 2017, pp. 2117–2126.

[36] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal

features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[37] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang, "Youtube-vos: Sequence-to-sequence video object segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 585–601.

[38] C. Ventura, M. Bellver, A. Girbau, A. Salvador, F. Marques, and X. Giro-i Nieto, "Rvos: End-to-end recurrent network for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5277–5286.

[39] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim, "Fast video object segmentation by reference-guided mask propagation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7376–7385.

[40] Y.-T. Hu, J.-B. Huang, and A. G. Schwing, "Videomatch: Matching based video object segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 54–70.

[41] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9226–9235.

[42] T. Zhang, G. Lin, J. Cai, T. Shen, C. Shen, and A. C. Kot, "Decoupled spatial neural attention for weakly supervised semantic segmentation," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2930–2941, 2019.

[43] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 325–341.

[44] X. Li and C. Change Loy, "Video object segmentation with joint re-identification and attention-aware mask propagation," in *Proceedings of the European Conference on Com-*

puter Vision (ECCV), 2018, pp. 90–105.

[45] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, “See more, know more: Unsupervised video object segmentation with co-attention siamese networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 3623–3632.

[46] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[47] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” *arXiv preprint arXiv:1508.04025*, 2015.

[48] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” *arXiv preprint arXiv:1705.03122*, 2017.

[49] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[50] M. Guo, D. Zhang, J. Sun, and Y. Wu, “Symmetry encoder-decoder network with attention mechanism for fast video object segmentation,” *Symmetry*, vol. 11, no. 8, p. 1006, 2019.

[51] M. Choi, H. Kim, B. Han, N. Xu, and K. M. Lee, “Channel attention is all you need for video frame interpolation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 10 663–10 671.

[52] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S. C. Hoi, and H. Ling, “Learning unsupervised video object segmentation through visual attention,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 3064–3074.

[53] T. Sun and K. Fezzaa, “Hispod: a program for high-speed polychromatic x-ray diffraction experiments and data analysis on polycrystalline samples,” *Journal of synchrotron*

radiation, vol. 23, no. 4, pp. 1046–1053, 2016.

[54] M. Berman, A. Rannen Triki, and M. B. Blaschko, “The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4413–4421.

Chapter 3: A Mask-guided Attention Deep Learning Model for COVID-19 Diagnosis based on an Integrated CT Scan Images Database

(This chapter draws from the paper “A Mask-Guided Attention Deep Learning Model for COVID-19 Diagnosis Based on an Integrated CT Scan Images Database” by Maftouni, M., Shen, B., Law, A. C. C., Yazdi, N. A., Hadavand, F., Ghiasvand, F., and Kong, Z., published in IISE Transactions on Healthcare Systems Engineering (2022): 1-18.)

Abstract

The global extent of COVID-19 mutations and the consequent depletion of hospital resources highlighted the necessity of effective computer-assisted medical diagnosis. COVID-19 detection mediated by deep learning models can help diagnose this highly contagious disease and lower infectivity and mortality rates. Computed tomography (CT) is the preferred imaging modality for building automatic COVID-19 screening and diagnosis models. It is well-known that the training set size significantly impacts the performance and generalization of deep learning models. However, accessing a large dataset of CT scan images from an emerging disease like COVID-19 is challenging. Therefore, data efficiency becomes a significant factor in choosing a learning model. To this end, we present a multi-task learning approach, namely, a mask-guided attention (MGA) classifier, to improve the generalization and data efficiency of COVID-19 classification on lung CT scan images. The novelty of this method is compensating for the scarcity of data by employing more supervision with lesion masks, increasing the sensitivity of the model to COVID-19 manifestations, and helping both generalization and classification performance. Our proposed model achieves better overall performance than the single-task (without MGA module) baseline and state-of-the-art models, as measured by various popular metrics. In our experiment with different percentages of data from our curated dataset, the classification performance gain from this multi-task learning approach is more significant for the smaller training sizes. Furthermore, experimental results demonstrate that our method enhances the focus on the lesions, as witnessed by both attention and attribution maps, resulting in a more interpretable model.

1. Introduction

The coronavirus pandemic has struck the world since late 2019, causing a global crisis and countless deaths. As of January 11th, 2021, the World Health Organization has reported more than 308.46 million confirmed cases and 5.49 million deaths due to this virus. Furthermore, COVID-19 mutations have significantly constrained hospitals' capacity resulting in delayed care and increased risks for patients suffering from other critical conditions. COVID-19's global reach has brought together experts from a wide range of fields to combat the disease. One of the ongoing research topics has been to improve the COVID-19 diagnosis. Early diagnosis has two main benefits: (1) lowering the infectivity rate by isolating patients; and (2) reducing the fatality rate through early intervention.

While the reverse transcription-polymerase chain reaction (RT-PCR) test, which falls under the category of nucleic acid amplification tests (NAATs), has become the gold standard for detecting COVID-19, it has drawbacks such as limited sensitivity to the new variants, short supply of testing kits, and lengthy wait time for results [1, 2, 3, 4]. Alternatively, lung computed tomography (CT) has proven to be a rapid and relatively accurate method of detecting COVID-19 and severity assessment [2, 3, 4, 5]. Infected patients' lung CT scans may exhibit distinctive characteristics such as ground-glass opacification, bilateral involvement, and diffuse distributions [3, 4, 6]. However, interpreting CT scans is a complex task requiring extensive radiology expertise. The number of radiologist experts is limited, and they face a heavy workload during an outbreak, increasing the risk of human errors. Therefore, transferring expert knowledge into intelligent models is valuable in order to improve healthcare accessibility, reduce the medical specialists' workload, and unnecessary exposure.

Deep learning has become one of the most extensively used approaches for building intelligent models, which can learn the underlying representation of images and classify them in a time-efficient manner. Notably, deep learning approaches have been successful for COVID-19 diagnosis in lung CT scans. [7] proposed an AI system that can identify COVID-19 markers and lesion properties using an extensive CT database of 3,777 patients. [8] used a mix of CT scans, lung, and lesion masks to train a COVID-19 diagnosis model leveraging multi-task and self-supervised learning. [9] presented a fast, accurate, and fully automated method for COVID-19 diagnosis from the patient's chest CT scan images. There have been several other studies on deep learning-based COVID-19 diagnosis [10, 11, 12, 13]. Most of the work uses a single-task approach and devotes the learning model to only one task. On the other hand, jointly learning multiple related tasks, namely, multi-task learning (MTL), has been shown to overcome over-fitting and improve generalization by implicit data augmentation, attention focusing, and regularization [14].

Despite the promising learning ability of deep models, the generalization power of the trained network depends on the size, distribution, and quality of the training dataset. Inadequate training datasets can easily lead to over-fitted deep learning models that cannot generalize well on a new dataset. Some COVID-19 datasets have been made publicly available [8, 9, 15, 16, 17, 18, 19, 20]. [8] introduced the COVID-CT dataset, which includes 349 COVID-19 CT images from 216 patients and 463 non-COVID-19 (a mix of normal cases and patients with other diseases). [6] reported improving classification performance by categorizing negative COVID-19 cases into specific groups and creating the COVID-19 CT Radiograph Image Data Stock dataset with careful data split. [15] built an open-sourced dataset named COVID-CT-MD, comprising COVID-19, Normal, and community-acquired pneumonia (CAP) cases. The

COVID-CT-MD is accompanied by lobe-level, slice-level, and patient-level labels to aid in developing deep learning methods. Notwithstanding, researchers continue to require more data for deep learning models' training in order to provide better insights and generalization performance. To this end, our COVID-19 lung CT-scan dataset is curated from seven open-source datasets.

Our proposed method applies a deep learning model with an attention module, which is a state-of-the-art technique in machine learning, to improve the performance of COVID-19 detection. For an image input, the attention module infers the attention map, which is a collection of pixel-level weights, to prioritize the image features by the level of importance for the task [21]. It attempts to mimic human visual perception that focuses on specific locations, objects, and attributes in the scene by filtering out irrelevant information. For example, an expert radiologist knows precisely where to focus in a CT scan to find a particular pathology. So, intuitively, the attention map learns which areas on the image are more relevant to the performed task, such as medical diagnosis. The use of attention modules in deep learning networks originated and proved successful in neural machine translation [22, 23]. Motivated by this success and its consistency with human perception, visual attention modules were adopted in different computer vision applications such as image captioning [24], visual question answering [25], and image classification [26]. The Residual Attention Network in [26] achieved state-of-the-art object recognition performance on several benchmark datasets and showed improved robustness against noisy labels. Later, Woo et al. [21] proposed a lightweight convolutional block attention module (CBAM) that could be integrated into any convolutional neural network (CNN) architecture to infer and refine attention. They showed that integrating CBAM inside various state-of-the-art CNN models improves the classification and detection performance. Accordingly, CBAM is incorporated into our model for enhanced performance through attention map learning and feature refinement.

To summarize, the objective of this paper is to improve the generalization and performance of COVID-19 detection deep learning models. Specifically, the main contributions of our paper are as follows:

- A large and broadly representative lung CT scan dataset for COVID-19 detection is built by curating seven open-source datasets. To the best of our knowledge, this is the largest publicly available COVID-19 CT dataset, accompanied by patient metadata. The dataset includes cases from 13 countries and has three classes: COVID-19, Normal, and CAP. The dataset also consists of COVID-19 frames with corresponding lesion masks merged from three of the datasets.
- A novel mask-guided attention (MGA) classifier for COVID-19 diagnosis is developed that improves classification performance, data efficiency, and interpretability. Our experimental results demonstrate the proposed method's superior performance over the baseline and improved focus on the COVID-19 lesions.

The remainder of this paper is organized as follows. In Section 2, a brief review of related research work on COVID-19 diagnosis, lesion segmentation, MGA methods, and multi-task learning is provided. Next, the proposed research methodology is summarized in Section 3. Section 4 introduces our curated CT scan dataset. Our proposed MGA deep learning model for COVID-19 diagnosis is detailed in Section 5, followed by the experimental results and ablation studies in Section 6. Finally, the conclusions and future directions are discussed in Section 7.

2. Related Work

The related works in deep learning-based COVID-19 diagnosis and lesion segmentation on CT scans is reviewed first in Section 2.1. Next, the multi-task learning related to COVID-19 are introduced in Section 2.2. The research gap is identified in Section 2.3.

2.1 COVID-19 Diagnosis and Lesion Segmentation on CT Scans

Deep learning has been the method of choice in most existing works on diagnosing COVID-19 infection from CT scans [9, 10, 11, 12, 20]; owing to the success of deep learning methods in image classification. [20] tested seven state-of-the-art deep classification models including VGG-16 [27], ResNet18, ResNet-50 [28], DenseNet-121, DenseNet-169 [29], EfficientNet-b0, and EfficientNet-b1 [30]. They integrated contrastive self-supervision [31] into the transfer learning process to further improve the performance of deep classification algorithms. In [9], a two-stage system was proposed for detecting COVID-19. The first stage filtered out those CT frames in which the inside of the lung is not properly observable. At the next stage, they applied a new feature pyramid network designed for classification problems using a ResNet-50V2 baseline [28], allowing the model to investigate different resolutions of the image and maintain the data from small objects. [10] proposed a light Convolutional Neural Network design, based on the SqueezeNet model [32], for the efficient differential diagnosis of COVID-19 CT scans from other community-acquired pneumonia infections and healthy CT scans. [11] proposed a novel transfer learning-based and uncertainty-aware framework for reliable detection of COVID-19 cases from X-ray and CT images. In [12], the attentional convolution network [26] is proposed to focus on the infected areas of the chest so that the network can provide a more accurate prediction. [33] proposed an optimized CNN model, named OptCoNet, for the automatic screening of COVID-19 patients based on X-ray images and used Grey Wolf optimizer for CNN hyperparameter optimization. [34] compared the performance of five pre-trained convolutional neural network models (ResNet50, ResNet101, ResNet152, InceptionV3, and Inception-ResNetV2) for COVID-19 classification on X-ray images and reports ResNet50 to have the best overall performance. In [35], the human-machine collaborative strategy is applied to design a deep convolutional neural network tailored to detect COVID-19 on chest X-ray images with improved sensitivity. They also introduce COVIDx, their large and public benchmark dataset of COVID-19 X-ray images.

Lesion segmentation is another task on CT scan images that is well suited for deep learning [36, 37, 38, 39, 40]. Generally, this task entails automatically predicting binary lesion masks, assigning the same label to all types of lesions. The problem can be expanded to the semantic segmentation of different types of lesions and within and outside lung regions if a sufficient number of lesion-specific ground truth masks are available. Nonetheless, the binary lesion masks are adequate for assessing the extent of involvement and manifestations of the disease in the lung of a confirmed or suspected COVID-19 patient [41]. [36] proposed to automatically segment ground-glass opacities (GGO) and areas of consolidation together using a DenseUNet [42]. [37] proposed CovidENet: an ensemble of 2D and 3D CNNs based on AtlasNet [43] for binary lesion segmentation and achieved human-level segmentation performance in terms of Dice Score and Hausdorff distance. [38] proposed the NormNet, a voxel-level anomaly modeling network to recognize normal voxels from possible anomalies. A decision boundary for normal contexts of the NormNet was learned by separating healthy tissues from the diverse synthetic "lesions," which can segment COVID-19 lesions without training on any labeled data. To focus more on the lesion areas, a novel lesion attention module was developed to integrate the intermediate segmentation results.

2.2 Multi-task Learning (MLT)

In general, MTL is known as a machine learning approach that assimilates information from correlated tasks to improve the generalization capability of the overall learning model [44]. There are two approaches in multi-task learning: hard parameter sharing and soft parameter sharing of hidden layers [14]. The hard parameter sharing is commonly found in the literature, in which multiple tasks (networks) share some hidden layers while keeping their separated output layers. Soft parameter sharing is when each task has its separate model and respective parameters, but the parameters from the tasks are jointly regularized.

MTL has been adopted for COVID-19 diagnosis improvement. [45] proposed end-to-end multi-task learning to detect and assess the severity of COVID-19 cases with improved performance using only a relatively small dataset of 1329 CT scans. [46] developed a multi-task deep learning model with three tasks of classification, segmentation, and reconstruction from chest CT images. [47] deployed a two-task deep learning model to identify COVID-19 cases and quantify the disease severity. [39] developed a novel Joint Classification and Segmentation (JCS) system to perform real-time and explainable COVID-19 chest CT diagnosis. [40] developed a dual-branch combination network (DCN) for COVID-19 diagnosis to simultaneously achieve individual-level classification and lesion segmentation. These papers reported an improvement over the single-task benchmark models. Furthermore, multi-task learning has improved the performance of smaller datasets more significantly [48, 49].

Another form of MTL is MGA models that extend the attention convolutional neural networks. The attention weights that the model assigns to each input element are generally learned without dedicated supervision; therefore, they might also converge to irreverent parts of the image for the task. For example, in classifying lung CT scans, the main focus should be on the inside lung manifestations, and assigning high attention weights to outside lung pixels is useless. Accordingly, recent research adopts extra supervision on attention map training. For instance, [50] designed a contrastive attention model guided by binary masks. It can generate a pair of body-aware and background-aware attention maps, which can produce features of body and background for Person Re-Identification. [51] introduced a novel MGA network that fits into popular pedestrian detection pipelines. The attention network emphasizes visible pedestrian regions while suppressing the occluded parts by modulating full body features. [52] proposed an MGA model that provides auxiliary supervision from predicted masks from a pre-trained segmentation model for discriminative and patchy representation learning.

2.3 Research Gap

The research gaps in the COVID-19 diagnosis approaches listed in Sec. 2.1 and Sec. 2.2 are identified as: (1) most proposed COVID-19 diagnosis methods are single-task, which may be more susceptible to over-fitting; (2) Training an accurate COVID-19 diagnosis model requires a large amount of broadly representative sample data, which many of the existing research efforts lack; and (3) Some COVID-19 diagnosis applications using a multi-task approach demonstrated improved performance over the single-task model; however, their models lack explainable choices of diagnosis results. In summary, there is a need for a novel approach to improve the generalization, interpretability, and data efficiency of the deep learning model for COVID-19 diagnosis applications. Therefore, in this work, a multi-task COVID-19 detection model, jointly supervising the attention maps and class labels, is developed to fill the aforementioned research gaps. The multi-task learning approach is implemented through an MGA module integrated inside the COVID-19 classifier to supervise its attention map with segmented lesions. Additionally, one of the strengths of our model is that it is trained and tested on a more broadly representative dataset, which promotes its generalizability.

3. Proposed Research Methodology

This work aims to develop a data-efficient deep learning model for COVID-19 diagnosis based on chest CT scan slices, with good generalization and interpretability. The performance of the deep learning model is highly dependent on the training data. However, a comprehensive CT scans dataset for COVID-19 is not publicly available to the researchers in the current literature. To fill this gap, a new CT scans dataset for COVID-19 is created and introduced in Section 4.

A CT scan cross-section or slice is reconstructed from the measurements of attenuation coefficients (intensity reduction) of x-ray beams as it passes through the tissues. Tissues with higher attenuation (such as bones) are bright, whereas tissues with little attenuation (such as air and water) appear dark. Since a normal lung looks dark in the CT scan, the abnormal increase in the attenuation in an inside lung area points at lesions related to different diseases (e.g. COVID-19 or CAP). Radiologists have characterized the key lung lesion patterns, or lesion types, for COVID-19 diagnosis. Our dataset contains the marking of these patterns as follows,

1. ground-glass opacities (hazy gray opacities that do not obscure the underlying vessels),
2. consolidation (areas of increased attenuation that obscure the underlying vessels), and
3. pleural effusion (excess fluid build-up between the lung and chest cavity).

These patterns are revealed through the lesion annotations manually marked by radiologist experts. As depicted in Figure 13, the lesion annotations are employed to derive the binary lesion masks of each image. Namely, black pixels are non-lesion while white ones are lesions. Therefore, in this paper, all different COVID-19 lesion types are combined as one type used in the classification analysis for COVID-19 diagnosis.

Our idea is to fully utilize the available domain knowledge through the COVID-19 lesion patterns to improve the deep learning model performance while lowering its data requirement. The overall proposed model architecture, depicted in Figure 14, is a two-step approach:

- Step 1 (Section 5.1): A lesion segmentation model based on Hierarchical Multi-scale Attention Network [53] (HMSANet) is implemented to automatically create lesion masks for the images that radiologists did not mark with lesion masks since manual marking is costly and time-consuming. The lesion masks are then used in the MGA module to supervise the spatial attention map (created by CBAM in Step 2) for the purpose of assigning higher attention weights to the pixels resembling lesions.

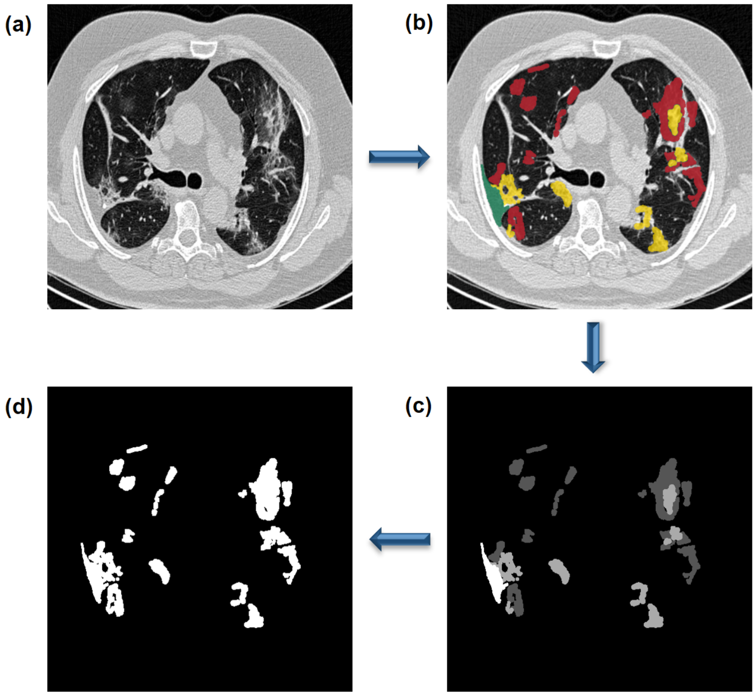


Figure 13: Example of Deriving binary lesion mask from radiologist annotations on a chest CT scan slice. (a) The chest CT scan slice (b) The radiologist annotation from [19]. Red, yellow, and green colors indicate ground-glass opacities, consolidation, and pleural effusion lesion types, respectively. (c) Semantic segmentation mask that maps non-lesion pixels to black and assigns each lesion type to a different class (level of gray) (d) Binary (black and white) lesion mask, after mapping all lesion types to the general category of lesions. Black represents non-lesion, and white represents lesion.

- Step 2 (Section 5.2): The deep learning classification model is applied to classify the input CT image, guided with the lesion mask generated in Step 1, and provides the diagnosis result, namely, Normal, COVID-19, or CAP case. Our classification model uses CBAM and MGA modules to enhance the model's focus on lesion locations. Particularly, the spatial attention map created by CBAM is guided towards the lesions through the MGA module during training.

These two steps introduced above are integrated through the hard parameter sharing multi-task learning model (namely, the share of some hidden layers). The first task, accomplished through the MGA module, directly supervises the network's attention map using the lesion

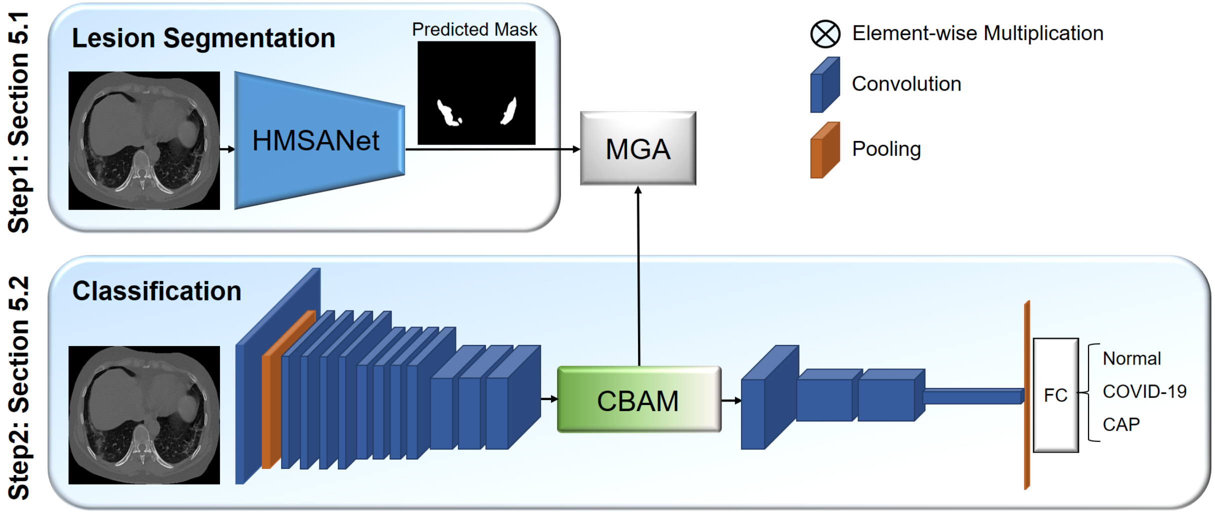


Figure 14: The proposed method architecture with two main parts: Lesion segmentation and classification

masks predicted in Step 1. The second task, implemented in the second step, applies supervision on the class predictions with the ground-truth class labels. This multi-task learning model has the following advantages.

1. First, the increased focus on the lesion regions, which are the COVID-19 manifestations, improves the accuracy of COVID-19 diagnosis and alleviates over-fitting by lowering the effective dimensionality of the data.
2. Second, it lowers the training data requirement. Our experiments show that the proposed model offers fewer training data sample requirements by utilizing additional supervision through the lesion data.
3. Third, the model prediction is more interpretable and reliable when focusing on the lesions instead of the entire image with many irrelevant parts to the illness.

4. Dataset Creation

CT scans show promise in providing COVID-19 screening and testing accurately and quickly [8]. We created a large lung CT scan dataset for COVID-19 to aid in developing the diagnosis models. The dataset includes curated data from [8, 9, 15, 16, 17, 18, 19]. Each of the seven datasets is illustrated by an example image in Figure 15. These datasets have been utilized publicly in COVID-19 diagnosis literature and have proven effective in deep learning applications. As a result, the combined dataset is expected to increase the generalization capacity of deep learning models by learning from all of these resources together.

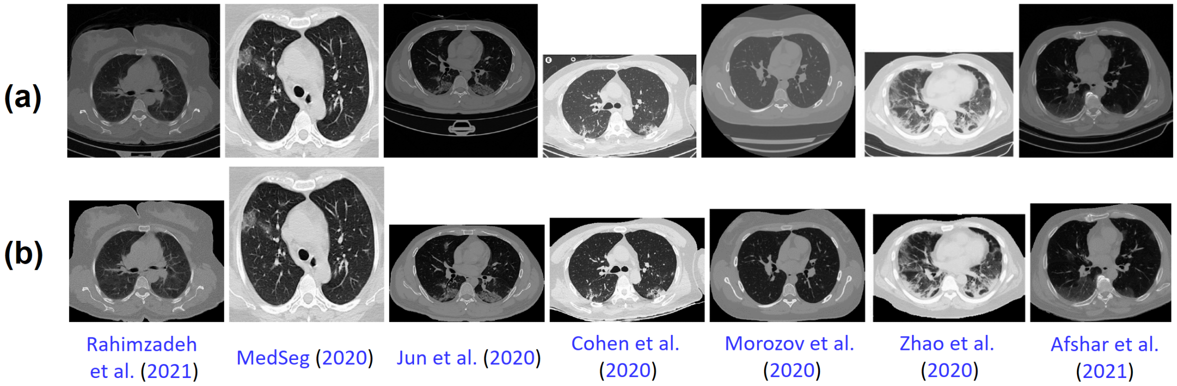


Figure 15: (a) An example lung CT frame from the seven open-source datasets included in our dataset; (b) Same images after initial preprocessing, including background removal, cropping, and normalization on the foreground segment.

Our objective is to provide a large dataset of axial chest CT scan slices with three labels, namely, (1) COVID-19, (2) Normal, and (3) CAP, together with their corresponding metadata and lesion masks if available.

Our study integrates seven public datasets of CT images from different sources across multiple countries. In this regard, the datasets are quite heterogeneous in terms of the operational parameters of the generation, resolution, and formatting (e.g., NIfTI, DICOM, TIFF, PNG, and JPG). Some datasets consist of class labeled CT slices (CT scan cross-sections, also referred to as frames or images). In contrast, Other datasets include 3D CT scan volumes (slices stacked on top of each other) with slice-level annotations. Section **??** details the steps to preprocess these heterogeneous datasets into our unified dataset of consistent format.

It should be noted that not all of the 3D CT volumes in the dataset were annotated with class labels at the slice level, and we worked with our radiologist to annotate the remaining CT images. To ensure the dataset quality, we excluded the chest slices that do not carry information about inside lung manifestations, as well as the adjacent slices with almost identical appearances. Additionally, we removed images lacking clear class labels or patient information. We have collected 7,593 COVID-19 images from 466 patients, 6,893 normal images from 604 patients, and 2,618 CAP images from 60 patients in total. Our CAP images are all from the dataset [15], in which 25 cases are already annotated. Our radiologist has annotated the remaining 35 CT scan volumes.

Table 3 summarizes the number of frames from COVID-19 and normal classes, the availability of specific metadata and masks, and the initial data format of each of the seven datasets. As previously stated, all of the cases have patient ID, necessary for data splitting. As listed in the table, three of the datasets have lesion masks [17, 18, 19], providing us with 2,729 COVID-19 lesion masks (36% of the COVID cases) to be used to train the mask segmentation model, explained in Section 5.1. The distinct categories of lesions in [19] are mapped to a binary lesion mask for consistency across datasets.

Figure 16 depicts multiple statistics from the dataset. The country and gender distributions on the entire dataset are shown in the subfigures (a-b). Figure 16(a) indicates that the cases come from 13 countries, with Iran, Russia, and China ranking first through third. According to Figure 16(b) most of the cases are male, and this male dominance holds for all Normal, COVID-19, and Cap classes. Figure 16(c) compares the age distribution of the three classes and shows that all the age groups are represented in the dataset. The median age of Normal, COVID-19, and CAP classes are 50, 49, and 59, respectively. Figure 16(d) compares the prevalence of distinctive CT characteristics in the 796 COVID-19 cases with CT scan reports, highlighting that ground-glass opacities, bilateral involvements, and consolidation have frequently been reported. And patterns attributed to higher severity, such as diffuse

Table 3: Seven Datasets summary.

Dataset	Country	COVID-19 Slices	COVID-19 Cases	Normal Slices	Normal Cases	Masks	Gender & Age	Data Format
[8]	China, Japan	349	213	NA	NA	NA	Missing	PNG, JPG
[15]	Iran	3,815	55	760	76	lobe level annotation	Available	DICOM
[16]	Multiple	34	17	NA	NA	NA	Missing	PNG, JPG
[17]	Russia	785	50	5,080	254	lesions	NA	NIfTI
[9]	Iran	666	68	1,053	274	NA	Available	TIFF
[18]	Multiple	1,844	20	NA	NA	lung and lesions	Available	NIfTI
[19]	Italy	100	43	NA	NA	lung and lesions	Available	NIfTI
Ours	**Multiple**	**7,593**	**466**	**6,893**	**604**	**64% Missing**	**9% Missing**	**PNG**

Note: NA stands for not available, and Missing is available but with missing values.

distribution [54], are also present. These statistics indicate that the dataset population is broad and representative, having cases from various ages, gender, nationality, and severity groups.

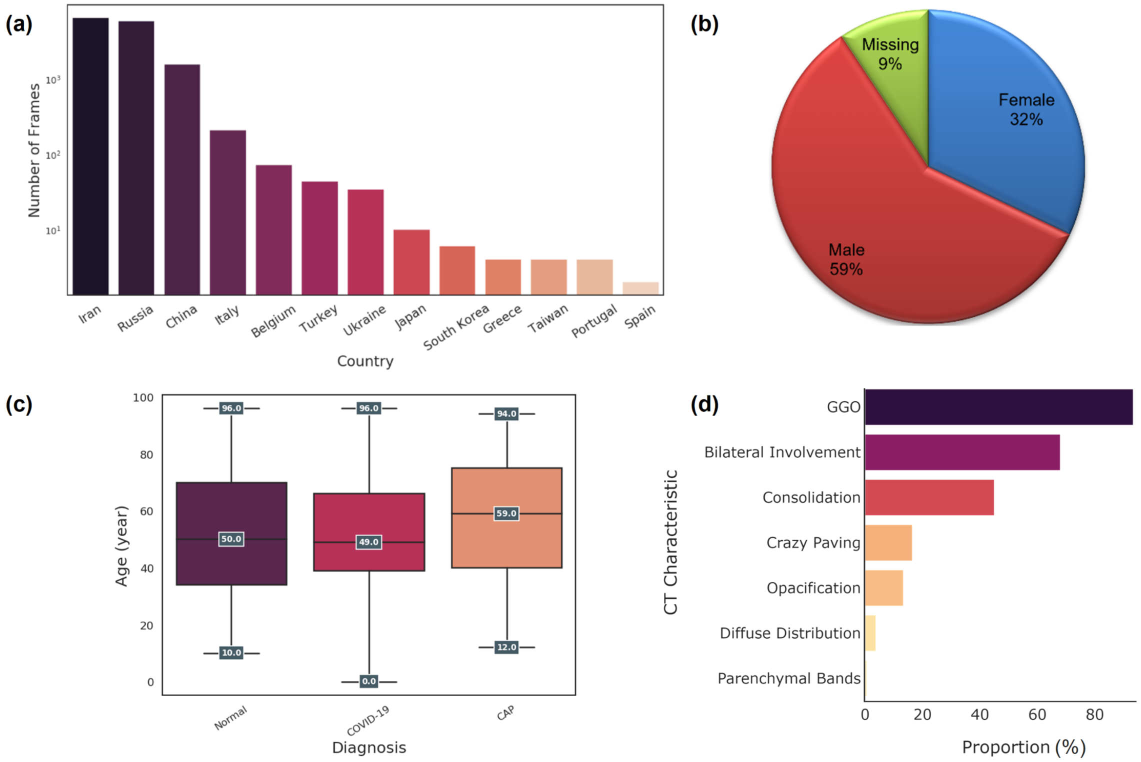


Figure 16: (a) Country and (b) Gender distribution of images in our dataset (c) Comparison of the three classes' age ranges (d) The proportion of critical COVID-19 manifestations in the available CT scan reports.

The combination of datasets from diverse sources may introduce heterogeneity in the resulting dataset, thereby necessitating data preprocessing. To address this issue in our CT COVID-19 dataset, we have implemented a data preprocessing approach, comprising two steps.

- Convert all of the CT volumes data into labeled frames. For the CT volumes, we used their slice-level annotations to extract the label of each frame. All the extracted frames are then converted to 8-bit PNG file format for better uniformity and accessibility for the deep learning analysis. A few examples of extracted frames are shown in Figure 15.

- A series of image transformations have been applied, including background removal, cropping lung area, image normalization, and resizing to 224 pixels by 224 pixels. The background is removed to prevent artifacts and appearance differences between datasets from impacting the COVID-19 diagnosis. The lung area is cropped out since it is our region of interest. Figures 15 (a) and (b) are CT scans before and after transformation, respectively. Image normalization and resizing are applied to create a uniform style of different images.

This paper uses a random resized crop of scale $(0.5, 1)$, random image rotation with a maximum of 10 degrees, and horizontal flipping with a probability of 0.5 for training set augmentations.

5. COVID-19 Diagnosis Using Deep Learning with MGA Model

This section presents the multi-task learning model using MGA in detail, consisting of two steps (See Figure 14). Step 1: in Section 5.1, a lesion mask prediction model is implemented based on the 2729 COVID-19 available lesion masks and then applied to generate the lesion mask in all the images that were not annotated with lesions. Step 2: in Section 5.2, a classification model is developed to classify if the input image is Normal, COVID-19 or CAP. Additionally, the significance and method of interpreting the model predictions is introduced in Section 5.3.

5.1 Segmentation Model for Lesion Mask Prediction

Semantic segmentation is to classify every pixel in the image into one of the classes of interest. The problem in this paper is simplified to binary segmentation when the aim is to separate out a single class, namely, lesions. Segmentation may be thought of as a pixel-wise classification that requires object localization and boundary detection at the same time.

Localization and boundary detection require different image resolutions and network receptive fields (the extent of an image exposed to a single neuron within the model). Predicting object location is better handled at a scale-down image size because the network's receptive field can observe more of the image context. In contrast, detecting fine edges and thin structures is better handled at a scaled-up image size, leading to a smaller receptive field. Therefore, multi-scale inference is an effective means to address both of these underpinning segmentation requirements. The challenge is how to combine the multiple-scale predictions effectively. The simplest way is to combine the results with averaging or max pooling. A more effective approach is to find the weighted average of the multiple scale-level predictions based on pixel-level weight maps learned within the model. HMSANet [53] uses the second approach and hierarchically combines the multiple scale predictions using the learned weight map, also called attention map. This model can learn the relative weighting between adjacent scales during training and enables the inclusion of other scales during inference on the test images.

HMSANet is adopted in this study for the lesion segmentation because its multi-scale and high resolution learning facilitates lesion localization and accurate boundary detection, especially as lesions appear in different sizes and shapes. Additionally, our results presented in 6.1 shows that HMSANet outperforms other segmentation methods on the lesion segmentation task.

The way that the HMSANet model is adopted for lesion segmentation is shown in Figure 14 (upper part), which is the first step of the proposed methodology. HMSANet model structure is depicted in Figure 17 in which the lesion mask is inferred using three frame scales. These image scales pass through a network trunk for both scale-level lesion mask and

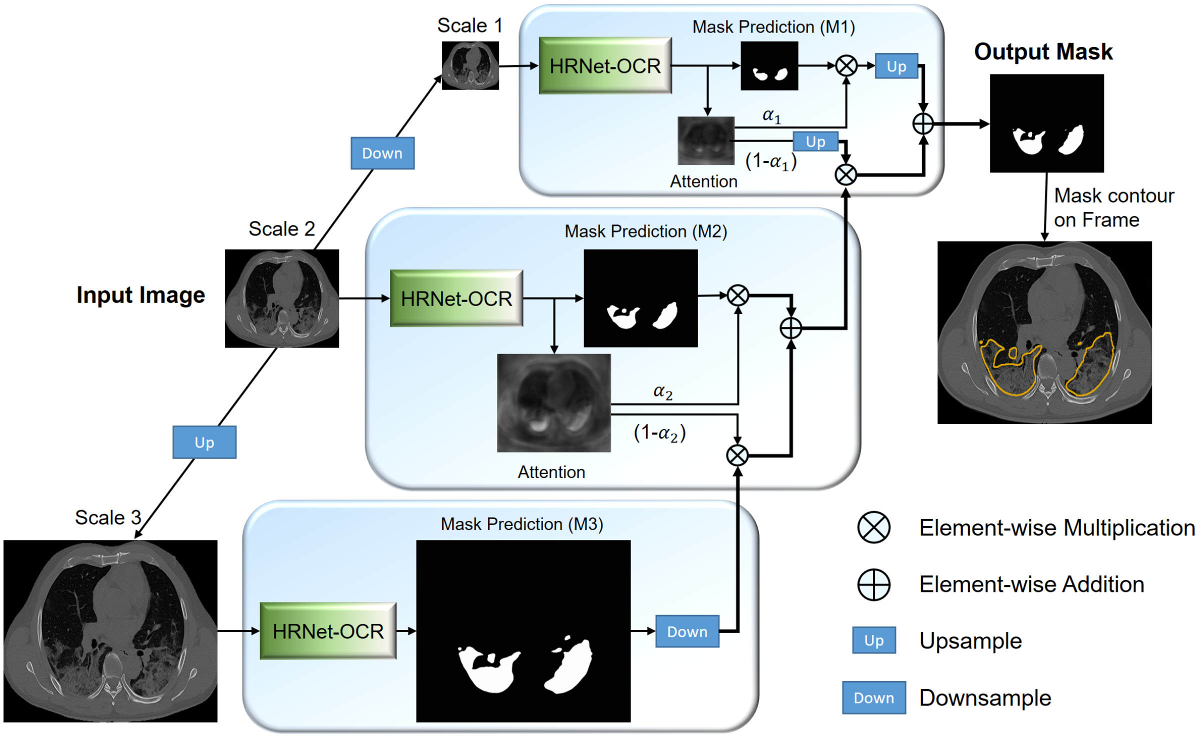


Figure 17: The lesion segmentation model (the adopted HMSANet module [53] structured in Figure 14). HMSANet infers the lesion mask of the same size as the input image by hierarchically combining predictions at multiple scales, weighted by the hierarchically learned attention weights. Lower scales determine the general lesion location, while higher scales refine its details and edges.

attention map inference. The High-Resolution network Object-Contextual Representations (HRNet-OCR) model with ResNet-101 baseline [55] is the best-performing scale-level trunk for the HMSANet model showing competitive performance on several semantic segmentation benchmarks. As shown in Figure 17, these scale-level mask predictions are combined to generate the final lesion mask by applying a chain of element-wise multiplication between the attention maps (α_n) and the mask predictions (M_n), followed by element-wise addition among the multiple scales. The chain starts at the lowest scale of the image, namely scale 1 in Figure 17, which captures the most global features, and is further refined for details at the following higher scales in order (Scale 2 and 3). Since lower scales take precedence, they take out their contribution share ($0 < \alpha_n(i, j) < 1$), higher (whiter) at the pixels of increased confidence, and pass the remaining attention $(1 - \alpha_n(i, j))$ to the following higher scales. Specifically, the final predicted mask ($\mathbf{M}$) is calculated by Equation (12), in which $\mathbf{U}$ is bilinear upsampling and $\mathbf{D}$ is downsampling.

$$\mathbf{M} = \mathbf{U}(\alpha_1 \otimes \mathbf{M}_1) + \mathbf{U}(1 - \alpha_1) \otimes [(\alpha_2 \otimes \mathbf{M}_2) + ((1 - \alpha_2) \otimes \mathbf{D}(\mathbf{M}_3))] \tag{12}$$

The HMSANet model is trained using the cross-entropy loss function, batch size of 1 per GPU, image scales of 0.5 (scaled down to half the size) and 1.0, stochastic gradient descent optimizer with the learning rate of 0.01, the momentum of 0.9, and the weight decay of $5e^{-4}$. These segmentation model hyperparameters are determined based on their values in the base paper [53], achieving a new state-of-the-art performance, and showed the best performance on our validation set. We used four NVIDIA GeForce RTX 2080 Ti GPUs and Pytorch library to train the model. The 2729 COVID-19 frames and their ground truth lesion masks are split into the training, validation, and test sets in sizes of 2329, 200, and 200, respectively.

After evaluating the segmentation performance, the trained segmentation model is employed to predict COVID-19 lesions masks on all the images without lesion masks, regardless of their class. Then, all the images are paired with their corresponding masks to be used as the ground-truth of the network's attention map in the MGA module, as laid out in the next section.

5.2 Classification Model for COVID-19 Diagnosis

The lightweight Residual Network [28] with 18 layers (ResNet18) is selected in this work to serve as the backbone of our COVID-19 classification architecture. The residual networks resolve the vanishing gradient and performance degradation problems of deep networks through skip connections, also known as residual connections. Specifically, Resnet18 is chosen for its lightweight architecture, computational efficiency, and competitive performance in COVID-19 diagnosis [56, 57]. The ResNet18 architecture is our baseline model but without attention.

We have embedded CBAM [21] as the attention module in the ResNet18 architecture to enhance the activation of discriminate parts of the input image. For the second step of the proposed methodology, namely, the lower part of Figure 14, the more detailed structure is shown in Figure 18.

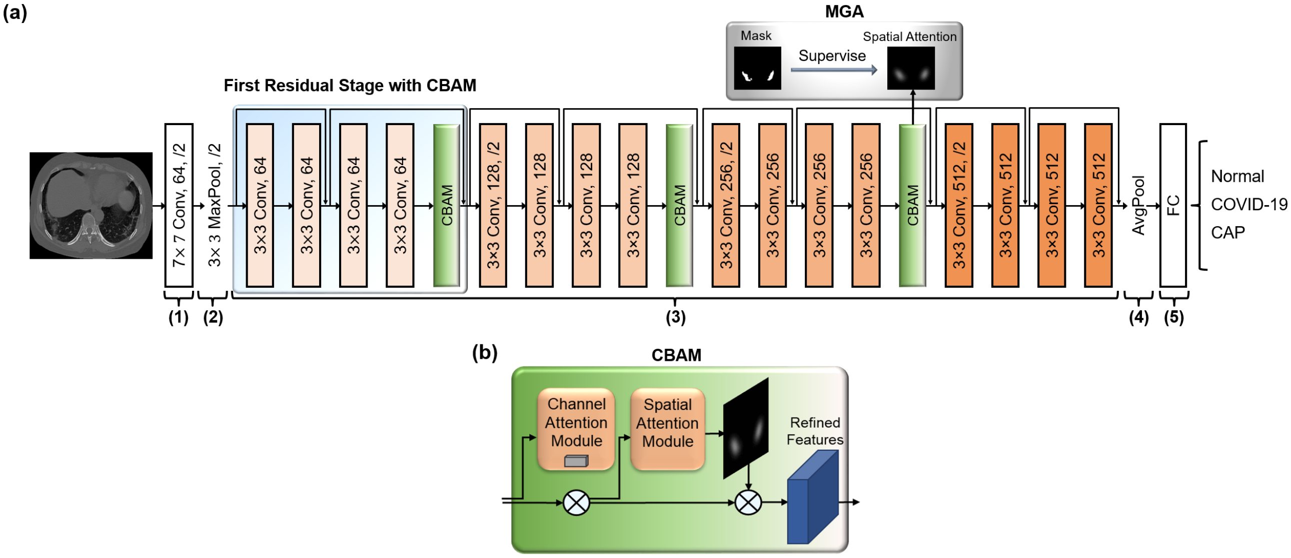


Figure 18: (a) The classification model's network structure (b) CBAM structure

our classification model's network structure consists of the following components:

1. a convolutional layer (with 7×7 filter size, 64 filters, and stride of 2) to learn 64 filters,
2. a max pooling layer (with 3×3 filter size, and stride of 2) to reduce the input spatial size,
3. four residual stages (four successive convolutional layers with two residual connections and the same number of filters, distinguished by the color in Figure 18), to allow the information flow between layers while gradually reducing the spatial size and learning more filters. CBAM is embedded only in the first three residual stages to save the computation,
4. an average pooling layer, to spatially down-sample the feature map into a vector, and
5. a fully connected layer at the end for classification.

Each convolutional layer outputs a 3D tensor called a feature map with (height, width) as the spatial axes and multiple-output channels (C) based on the number of filters. The feature maps of convolutional layers in each residual stage have the same dimension. From one residual stage to the next, the feature maps' height and width are halved (noted by /2 in Figure 18) by convolution stride, and the output channels are doubled (64, 128, 256, and 512, respectively). The attention module's role is to reweight the feature map. Since the feature maps are 3D tensors, the feature map re-weighting can be performed spatially (by spatial attention module) or on the channels (by channel attention module). The spatial attention module assigns higher weights to more informative parts of the input, while the channel attention module weights the channels based on their relevance and importance by multiplying the channel weights with the feature map. CBAM has a consecutive channel and spatial attention (sub)modules, which is shown to be the best performing combination.

The ResNet18 model with embedded CBAM is our baseline with attention but without direct supervision of attention map learning. In addition to applying attention reweighting, our proposed model uses an MGA module to directly supervise the spatial attention map of one of the three CBAMs by the predicted masks. This extra supervision makes our method multi-task learning because we are jointly optimizing the two tasks of classification and attention to lesions through two distinct loss functions specified in the following paragraphs. Figure 18 shows our classification model's network structure when the MGA module is placed at the third residual stage. The optimal placement of the MGA module has been studied in Section 6.3.

In order to create the spatial attention map, the spatial attention module average-pools and maximum-pools the channel-attended feature map of dimension (H,W,C) to aggregate and squeeze its channel information into two (H,W,1)-dimensional tensors. Then, these two poolings are concatenated in the channel dimension (H,W,2), and transformed into the spatial attention map via a convolutional layer with one channel output, padding of 3, filter size of 7×7 ($f^{7\times7}$), and a sigmoid activation function (σ), as formulated in Equation (13). Therefore, the spatial attention map is a one-channel tensor with the same height and width as its corresponding feature map (H,W,1) in which all the values are between zero and one.

$$SA(f_i) = \sigma(f^{7\times7}([AvgPool(f_i); MaxPool(f_i)])) \tag{13}$$

As indicated by Equation (14), the image features extracted at the j^{th} residual stage denoted by $f_j \subseteq \mathbb{R}^{H\times W\times C}$ are spatially multiplied by the spatial attention map $SA_1 \subseteq \mathbb{R}^{H\times W}$ to construct the attended features ${f_j}^{att}$. H, W, and C denote height, weight, and the number of channels, respectively. In the element-wise multiplication of the broadcasted (copied) one-channel spatial attention map with multi-channel features, i signifies the channel index.

$${f_j}^{att}(i) = f_j(i) \otimes SA(f_{j(i)}) \tag{14}$$

We directly supervise one of the spatial attention maps (SA) with the same sized predicted lesion mask (M) from Step 1 (section 5.1) by minimizing the pixel-wise mean squared error loss function L_{att}:

$$L_{att} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \|M_{i,j} - SA_{i,j}\| \tag{15}$$

The MGA module is intended to direct the spatial attention map emphasis to the inside lung manifestations and give extra attention to lesions and lung parts that resemble lesions. Since the predicted masks might not completely match the ground truth lesion masks, it is critical that our model performance is not overly sensitive to them. The residual connection right after the CBAM module (see Figure 18) facilitates the flow of unattended features via the skip connections and helps prevent the error propagation from inaccurate masks. The sensitivity of classification performance to the predicted masks has been further studied in Section 6.3.

The classification task is supervised with the cross-entropy loss between the predicted class probabilities ($\hat{y}$) and one-hot encoded ground truth class labels (y) of the three classes as stated in Equation (16).

$$L_{ce}(\hat{y}, y) = -\sum_{k}^{3} y^{(k)} \log\, \hat{y}^{(k)} \tag{16}$$

Our proposed classifier is therefore applying supervision over two tasks, namely, the attention map using the attention mean squared error loss (L_{att}) and supervision over the class label predictions using classification cross-entropy loss (L_{ce}). As represented in Equation (17), we adopted learning with uncertainty loss weighting [58] between the L_{ce} and L_{att} because it has shown superior performance over using fixed weights [49]. This weighting scheme lets the model adjust the weight of each loss by learning the observation noise parameters σ_1 and σ_2 alongside the model weights (W). Smaller values of the observation noise parameter will increase the contribution of its associated loss function. These noise parameters are regularized to avoid very large values, which diminishes the contribution of each of the tasks.

$$L(W, \sigma_1, \sigma_2) = \frac{1}{2\sigma_1^2} L_{ce}(W) + \frac{1}{2\sigma_2^2} L_{att}(W) + \log\sigma_1 + \log\sigma_2 \tag{17}$$

The model is trained using an Adam optimizer with a learning rate of 0.0001, a cosine annealing scheduler, a batch size of 32, and 100 epochs with early stopping with the patience of 10. These hyperparameters are tuned using Bayesian Optimization [59]. We used four NVIDIA GeForce RTX 2080 Ti GPUs and Pytorch library to train our models.

Table 4 specifies one example dataset split between training, validation, and testing. Since the images from a single patient are naturally dependent, all the data splits are made in a patient-aware manner to avoid performance overestimation from the data leak [60, 61]. Patient-aware splitting keeps images from each unique patient together in one of the train, validation, or test splits. On the other hand, having multiple slices from each patient in the training set is not problematic because it can have a similar effect as data augmentation. We also applied stratification in our splitting which means that the splits have relatively the same proportion for each of the classes. Patient-aware splitting must be strictly adhered to. Limited by the patient-aware splitting, stratification is performed as much as feasible.

Table 4: Train, Validation, and Test splits distribution.

Data Split	COVID-19	Normal	CAP	Total
Train	5,563	4,643	1,773	11,979
Validation	1,508	1,736	643	3,887
Test	522	514	202	1,238

5.3 Interpreting the Model's Prediction

So far, we have introduced our proposed classification model that provides the COVID-19 diagnosis prediction but without interpretability. Achieving highly accurate but uninterpretable decisions makes deep learning models less trustable and has an adverse impact on their clinical applications. Although deep learning has a black box nature, much recent work has investigated the flow of information and input-output connections in deep neural networks to shed light on how it predicts. Such explanation methods help increase trust in the model when it predicts correctly and identifies the failure modes (such as data corruption and learning wrong patterns) when wrong. The gradient-based attribution methods [62, 63, 64, 65] provide input-specific explanations of the deep learning predictions by assigning an attribution value to each input feature. Each gradient-based attribution method has a slightly different formulation for identifying the contribution of each feature to the model's output through backpropagating the output prediction and decomposing it on the input image. The result is an attribution map, an image with the same size as the input containing the pixel level contribution scores.

Attribution maps are often shown as heatmaps, representing the attribution map with colors. For instance, red indicates features that contribute positively to the activation of the target output; the blue color distinguishes features that have a suppressing effect on it; and the white color indicates the insignificance for the derived output. In this work, we use two prominent attribution methods called Integrated Gradient [64] and DeepLIFT [65] methods to highlight disease features in the CT images. The Integrated Gradient method calculates the integral of gradients of each feature along the path from a baseline (such as a black image) to input, while DeepLIFT is its faster approximation.

6. Results and Discussion

This section presents the performance of the lesion mask segmentation method (Section 6.1) and the proposed classification model with attention (Section 6.2). Additionally, Section 6.3 covers the ablation studies to determine the placement of the MGA module, the effectiveness of MGA classification with different training set sizes, and sensitivity of the classification performance to the predicted masks. Next, Section 6.4 presents the interpretability of the decisions of our deep learning model. Finally, Section 6.5 discusses our work from the physician's perspective.

6.1 Segmentation Performance

The HMSANet architecture, presented in Section 5.1, is employed as the mask prediction method because of its state-of-the-art segmentation performance. We compared the HMSANet's performance with UNet [42], SegNet [66], and DeepLabV3 [67] architectures, which are among the most widely used segmentation methods in the literature. As reported in Table 5, HMSANet achieves the highest intersection over union (IOU), Dice coefficient, precision, and recall on the test set (consisting of 200 COVID-19 frames). The segmentation models' complexity is assessed by their number of trainable parameters and floating-point operations (FLOPs). These columns point to the trade-off between the model's computational cost and accuracy. Figure 19 provides the qualitative comparison of the predicted masks on five sample test images. According to this figure, HMSANet predicted masks most closely resemble the ground truth masks and are our best choice for the lesion prediction.

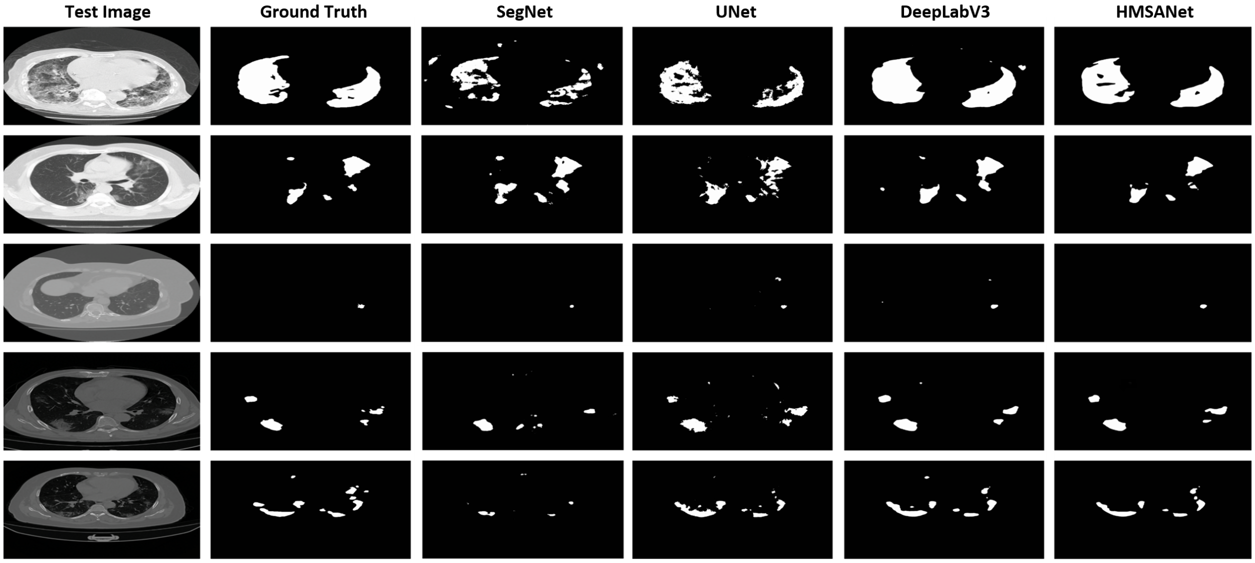


Figure 19: Lesion mask prediction comparison on five test images. HMSANet most closely resembles the ground truth.

Table 5: Lesion segmentation performance Comparison (the best performance in red).

Method	IOU (%)	Dice (%)	Precision (%)	Recall (%)	Parameters (M)	FLOPs (G)
SegNet	43.48	60.60	50.00	76.92	39.87	79.89
UNet	54.05	70.17	68.50	71.94	31.06	16.59
DeepLabV3	70.42	82.64	81.96	83.33	60.99	121.06
HMSANet	**74.63**	**85.47**	**84.03**	**86.96**	72.12	151.32

6.2 Classification Performance

Table 6 compares our proposed classification model (Section 5.2) with two baseline models without and with CBAM attention modules (ResNet18 and ResNet18 + CBAM), and four state-of-the-art benchmark models (ResNet50, MobileNetV2, VGG16, and DenseNet121).

For the sake of consistency, all the models are trained from scratch. The performance metrics are the average over three random patient-aware stratified test splits (one of which is reported in Table 4). The four benchmark architectures are among the most used deep learning architectures, showing the best performance on our dataset. ResNet50, VGG16, and DenseNet121 are also reported to achieve high COVID-19 diagnosis accuracy from CT scan by [20]. MobileNetV2 by [68] is included in the comparison for its competitive speed-accuracy trade-off, which is useful for mobile applications. For our medical application, achieving the highest diagnosis accuracy and F1 score takes precedence over training speed. According to table 6, our proposed approach achieves the highest accuracy, F1 score and recall at a reasonable speed. Particularly, the recall, also known as sensitivity, has shown significant improvement since the better focus on the lesions has boosted the detection of COVID-19 cases. Regarding the average recall, our model is the best and outperforms the second and third best-performing methods by 2.06% and 3.34%, respectively. The average COVID-19 prediction accuracy of our model for each country in the test split is 92.35 (Iran), 90.35 (Russia), 91.30 (China), and 87.5 (Italy). Please note that these per country accuracy scores are reported based on the COVID-19 cases since our dataset does not have all three classes (CAP, COVID-19, and Normal) per country.

Table 6: Classification averaged performance results (the best performance in red).

Method	Accuracy (%)	F1 score (%)	Recall (%)	Precision (%)	ROC AUC (%)	time (s)	Parameters (M)	FLOPs (G)
ResNet18	93.01	91.16	85.80	97.23	97.82	0.31	11.69	1.81
ResNet50	93.62	92.02	87.41	97.14	97.62	0.59	25.56	3.86
MobileNetV2	93.11	91.25	85.61	**97.68**	97.62	**0.17**	**3.3**	**0.42**
VGG16	93.25	91.70	87.04	96.89	97.92	0.76	138.36	15.5
DenseNet121	93.54	91.81	87.09	97.06	97.95	1.05	7.89	2.88
ResNet18 + CBAM	94.07	92.45	88.37	96.92	98.11	0.62	11.76	1.82
Proposed	**94.98**	**93.64**	**90.43**	97.09	**98.33**	0.68	11.76	1.82

The ROC curve measures the true-positive rate (sensitivity) and false-positive rate (1 – specificity) trade-off, and its area under the curve (ROC AUC, also referred to as AUC) has meaningful interpretation for disease classification and is extensively used in medical diagnosis. F1 score, which is the harmonic mean of recall and precision, is another reported metric. Our model's enhanced AUC and F1 score metrics indicate that the increased sensitivity did not come at the expense of more false positives. The proposed multi-task learning improves generalization by leveraging the domain-specific knowledge contained in the training data and makes it capable of learning a more meaningful representation.

Table 6 also reports the measured minibatch training time using four NVIDIA GeForce RTX 2080 Ti GPUs, the number of parameters, and FLOPs of a single forward pass. While the time column compares the speed of the models during training, FLOPs measures the computational overhead at the inference time. According to this Table, the proposed model notably improves the classification performance while keeping the number of parameters and FLOPs at the same level as the simple ResNet18 model. Moreover, despite training two tasks, our approach is 35% quicker than the DenseNet121 model, which is less memory efficient due to the dense concatenation operations.

6.3 Ablation Studies

Since the attention supervision can be applied inside any residual stage (Figure 18), the first ablation study determines the best placement of the MGA module.

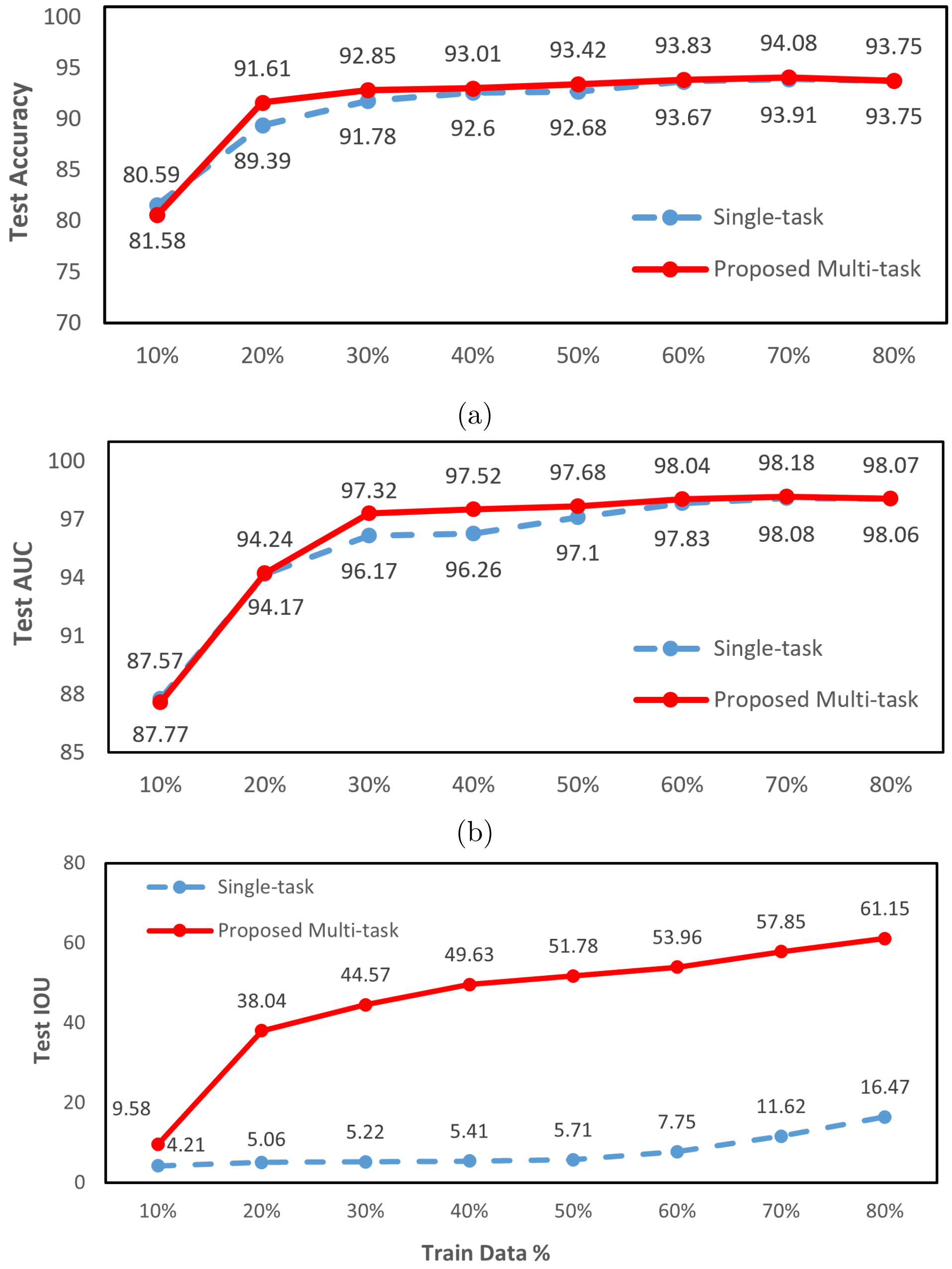
The results in Table 7 indicate that the best performance is achieved by placing the MGA module at the third residual stage. One possible explanation from the perspective of multi-task learning is that increasing the number of shared hidden layers between the highly related tasks helps the performance [69] and representation learning. Since the number of parameters and forward pass FLOPs is not impacted by the MGA module placements, all the reported models in Table 7 have the same values for these columns as the ones in the proposed row of Table 6. However, the backward pass FLOP slightly increases as the MGA module moves to deeper residual stages. Consequently, the training time of the third residual stage model has slightly increased. Additionally, comparing the results in Table 7 with the other models' performance in Table 6 shows that using the MGA module in any location improves the overall classification performance.

Table 7: Comparison of different MGA module placements (the best performance in red).

Method	Accuracy (%)	F1 score (%)	Recall (%)	Precision (%)	ROC AUC (%)	time (s)
First Residual Stage	94.65	93.49	90.86	96.29	98.08	**0.66**
Second Residual Stage	94.82	93.71	**91.25**	96.3	98.56	0.67
Third Residual Stage	**95.15**	**94.07**	91.05	**97.3**	**98.59**	0.68

The second experiment investigates the effectiveness of MGA classification for different training set sizes. We simplified the model and ran this experiment on a ResNet18 base model with only one embedded CBAM at the first residual stage to save the computation. Specifically, the single-task and multi-task models are identical, except that the spatial attention map of the CBAM is supervised with the predicted lesion masks for the multi-task learning.

Figures 20 (a) and (b) show the test classification performance comparison, and (c) is the IOU between the lesion masks and their binarized attention maps of the single and multi-task classification for different train set sizes. While the test set is separated and fixed, the remaining data is split between the train and validation according to the train data size. It is worth noting that the percentages are not exact since the data should be divided in a patient-aware and stratified manner. Consistent with the literature [48, 49], the results in subfigures (a-b) show that multi-task learning improves performance, especially when the training data is small and sufficient for the learning to happen. We can see that from 20% to 60% there is the most improvement. 10% is too small for learning, and for the large train sets, there is less difference in performance, yet the generalization and interpretability advantages remain. In other words, the proposed multi-task learning stands out in the model performance when the train set size is sufficient but relatively small. Moreover, a 70-30 data split between the train and validation has given the best performance; therefore, it is the ratio we used for comparing all the models.



(a)

(b)

(c)

Figure 20: The Single-task vs. Multi-task classification performance of the simplified model measured by (a) Accuracy (b) AUC and (c) attention map and mask IOU on the test set for different train set sizes

Figure 20 (c) employs intersection over union as a measure to quantify and compare the focus on the lesions. Each point is calculated by averaging the IOU results of test images with non-zero masks. We can see that the IoU of the proposed method is significantly better than the baseline. Additionally, increasing the train data has improved the focus of both learners. The same patterns can be observed from the attention map visualizations in Figure 21. This figure corroborates that, as the train data increases, the attention map of both the single-task learner and the multi-task learner converges to the lesions. However, the latter one starts to converge using only 30% of the data, while the improved focus emerges in the former after using 80% of the data. Therefore, attention supervision can help the fast convergence of the attention map with a smaller required train data size.

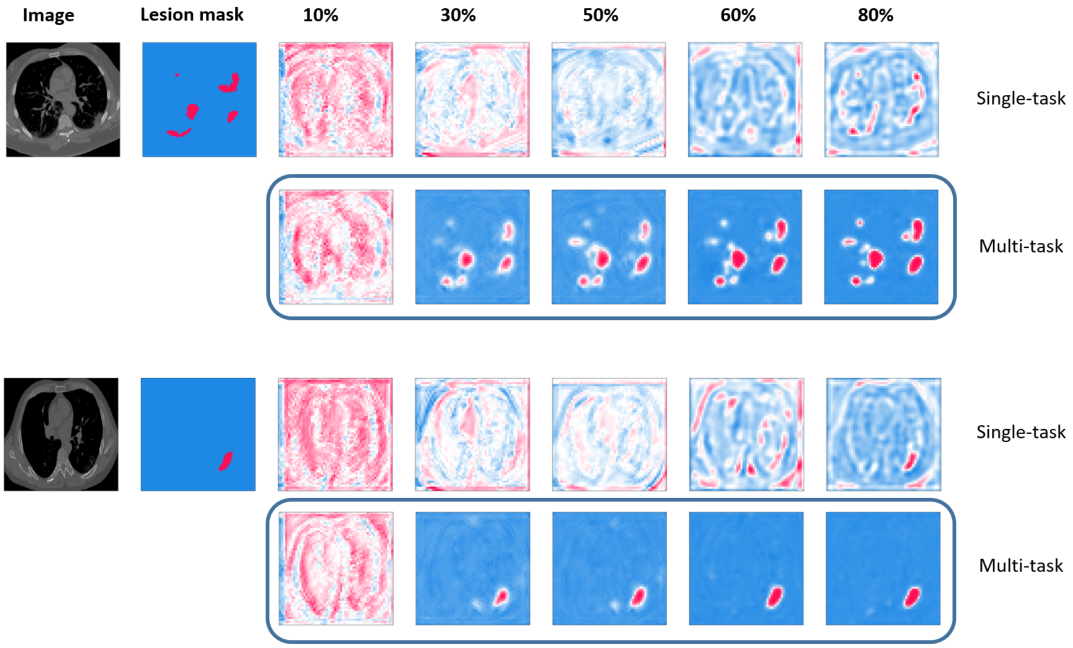


Figure 21: The Single-task vs. Multi-task attention maps of two example frames when different percentages of training data is used. The color changes from blue to red as the pixel's attention weight increases. While attention maps of both methods converge to highly score the lesions, the supervised attention maps in the multi-task classification converge using considerably less training data (20%). The unsupervised attention map takes a lot more data, 80% in our case, to focus on the lesions.

Our last experiment interrogates the impact of the error propagation from the segmentation step into the downstream classification task. We test this impact by applying the following transformations to the predicted lesion masks:

- Erosion: shrinking the lesion mask by removing pixels on the boundaries (of size 9)
- Dilation: expanding the lesion mask by adding pixels to lesion boundaries (of size 9)
- Shifting: displacing the lesion mask (by 9 pixels downwards)

Our overall results reported in Table 9 show that the model is not sensitive to the exact boundaries of the lesions, and the mask guidance results in performance gain using masks that point to the overall location of the mask. According to Table 9, the performance gain is higher for the dilated predicted masks. On the other hand, a comparison between the results in Table 6 and Table 9 indicates that high levels of mask erosion result in only slightly better performance and mask shifting in the same level of performance as the single-task model.

Table 8: The impact of the predicted mask transformations on the classification performance **of our proposed method**.

Mask Transformation	Accuracy (%)	F1 score (%)	Recall (%)	Precision (%)	ROC AUC (%)
Shifted Mask	94.00	92.62	89.28	96.22	98.5
Eroded Mask	94.98	93.87	91.57	96.29	98.52
Dilated Mask	95.15	94.07	90.52	97.91	98.62

This robustness to mask changes is attributed to the residual connections right after the CBAM module that facilitates the flow of unattended features (features before attention weights are applied). These skip connections may prevent the error propagation from inaccurate masks. Also, using uncertainty loss weighting between the classification cross-entropy and attention mean squared error losses equips the network with enough flexibility to adjust the weights of each loss function and gloss over the attention loss if it is not consolidating the classification task. Therefore, using an imperfect mask doesn't reduce performance compared to single-task learning, and its only downside is delayed convergence.

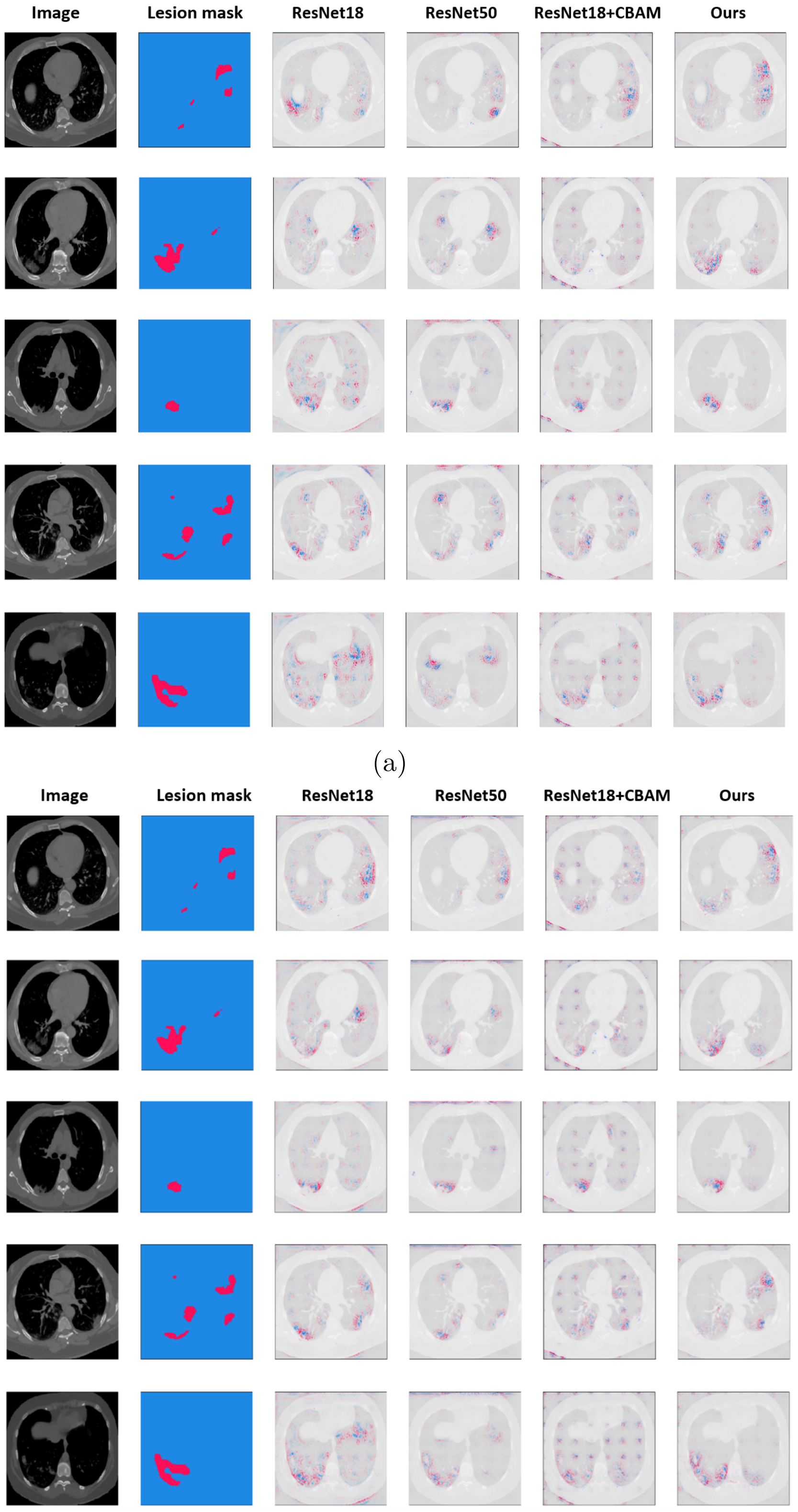
6.4 Interpretability using Attribution Maps

Figure 22 compares the attribution maps of our model with the other models for five COVID-19 frames, using 22a DeepLIFT (Rescale) [65] and 22b Integrated Gradient attribution [64] methods. We can see that the red and blue regions (pointing to influential features) of our model's attribution map highly overlap with the lesion regions (represented with red color in the lesion mask). In other words, the lesion regions highly contribute to our model decision while other models are less focused on the lesions. This visualization further emphasizes our multi-task learning approach's effectiveness in improving the model's attention to the relevant regions. This is because, compared to the single-task (classification), the two integrated tasks (namely, attention supervision and classification) can provide evidence for the relevance or irrelevance of specific features.

Moreover, DeepLIFT (Rescale) and Integrated Gradients have generated highly correlated attribution maps, consistent with past works, while DeepLIFT is considerably faster to execute. Current attribution methods do not explain how the network combines the features to produce the answer and scores them independently, but DeepLIFT (RevealCancel) method takes dependencies into account. For future exploration, it would be interesting to derive and compare the DeepLIFT (RevealCancel) attribution maps, which claimed to outperform the two other techniques when Pytorch support is available.

6.5 Physician's Perspective

Chest CT scans can help COVID-19 diagnosis in patients with a high clinical suspicion of the infection and pulmonary involvement [70, 71]. They are also helpful for assessing the disease severity and guiding its management [72]. The understanding of COVID-19-related abnor-



(a)

(b)

Figure 22: (a) DeepLIFT (Rescale) and (b) Integrated Gradient attribution comparison between different models.

malities in CT images has evolved since the onset of the pandemic. Employing intelligent systems that can accumulate and share knowledge about emerging diseases like COVID-19 across the globe may expedite understanding of the disease and facilitate its diagnosis and management. The current work showcases the possibility of accumulating knowledge about CT scan findings into intelligent machines and using them to make interpretable diagnoses by focusing on the abnormalities.

Even though CT scans can differentiate between most cases of CAP, COVID-19, and Normal, differential diagnosis of a broader range of disease classes necessitates the inclusion of clinical and paraclinical examination results [73]. Deep learning models can distinguish between many class labels and learn from various data formats (e.g., images, text, and tabular data) if an adequate dataset is available. Therefore, building a comprehensive and integrated database of patients' information (CT scans, clinical and paraclinical results, etc.) is a requirement for creating more practical systems that can address the following challenges:

- **Cases with non-typical CT findings:** when the signs in the CT scan are non-typical or non-specific, accompanying clinical and paraclinical symptoms are required.
- **Cases with multiple medical conditions:** Usually, the high-risk patients simultaneously present various medical conditions (e.g., diabetes, cardiovascular disorders, immunosuppressive therapy, etc.). Therefore, such cases require identifying more than one complication.

7. Conclusion and Future Direction

This paper presented the MGA-based classification model, a novel multi-task learner for COVID-19 diagnosis based on CT scan images. Specifically, the proposed model leveraged the predicted lesion masks to impose extra supervision on the classifier's attention module. Since attention supervision and classification are consolidatory tasks, their multi-task learning yielded a significant performance improvement over the single-task baseline (i.e., the baseline model without MGA module) and the state-of-the-art deep learning methods in image classification.

Our experiments also showed that the proposed method benefits from improved data efficiency and interpretability, which are especially valuable in the medical domain in which data may be often limited, and reliability is paramount. Additionally, in this work, a large, nationally diverse, and broadly representative COVID-19 CT slice classification dataset has been curated for conducting experiments and serving as a benchmark dataset for the research community. The quality of our dataset is ensured using slices with patient identification and precise labels.

This research could be extended to include an MGA module that segments both the lungs and the lesions to improve the overall inside lung learning, especially for normal cases. Additionally, as only two groups of COVID-19 and non-COVID-19 are examined in most of the literature, the effect of having more precisely categorized disease classes on COVID-19 detection could be further investigated.

Data Availability Statement

The data that support the findings of this study are available in Kaggle. These data were curated from the following resources available in the public domain: [8, 9, 15, 16, 17, 18, 19].

References

[1] S. Tahan, B. A. Parikh, L. Droit, M. A. Wallace, C.-A. D. Burnham, and D. Wang, "Sars-cov-2 e gene variant alters analytical sensitivity characteristics of viral detection using a commercial rt-pcr assay," *Journal of Clinical Microbiology*, pp. JCM–00 075, 2021.

[2] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen, W. Lv, Q. Tao, Z. Sun, and L. Xia, "Correlation of chest ct and rt-pcr testing in coronavirus disease 2019 (covid-19) in china: a report of 1014 cases," *Radiology*, p. 200642, 2020.

[3] X. Xie, Z. Zhong, W. Zhao, C. Zheng, F. Wang, and J. Liu, "Chest ct for typical 2019-ncov pneumonia: relationship to negative rt-pcr testing," *Radiology*, p. 200343, 2020.

[4] E. Trivizakis, N. Tsiknakis, E. E. Vassalou, G. Z. Papadakis, D. A. Spandidos, D. Sarigiannis, A. Tsatsakis, N. Papanikolaou, A. H. Karantanas, and K. Marias, "Advancing covid-19 differentiation with a robust preprocessing and integration of multi-institutional open-repository computer tomography datasets for deep learning analysis," *Experimental and therapeutic medicine*, vol. 20, no. 5, pp. 1–1, 2020.

[5] Y. Fang, H. Zhang, J. Xie, M. Lin, L. Ying, P. Pang, and W. Ji, "Sensitivity of chest ct for covid-19: comparison to rt-pcr," *Radiology*, vol. 296, no. 2, pp. E115–E117, 2020.

[6] K. Misztal, A. Pocha, M. Durak-Kozica, M. Wator, A. Kubica-Misztal, and M. Hartel, "The importance of standardisation–covid-19 ct & radiograph image data stock for deep learning purpose," *Computers in Biology and Medicine*, vol. 127, p. 104092, 2020.

[7] K. Zhang, X. Liu, J. Shen, Z. Li, Y. Sang, X. Wu, Y. Zha, W. Liang, C. Wang, K. Wang *et al.*, "Clinically applicable ai system for accurate diagnosis, quantitative measurements, and prognosis of covid-19 pneumonia using computed tomography," *Cell*, 2020.

[8] J. Zhao, Y. Zhang, X. He, and P. Xie, "Covid-ct-dataset: a ct scan dataset about covid-19," *arXiv preprint arXiv:2003.13865*, 2020.

[9] M. Rahimzadeh, A. Attar, and S. M. Sakhaei, "A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct scan dataset," *Biomedical Signal Processing and Control*, p. 102588, 2021.

[10] M. Polsinelli, L. Cinque, and G. Placidi, "A light cnn for detecting covid-19 from ct scans of the chest," *Pattern Recognition Letters*, vol. 140, pp. 95–100, 2020.

[11] A. Shamsi, H. Asgharnezhad, S. S. Jokandan, A. Khosravi, P. M. Kebria, D. Nahavandi, S. Nahavandi, and D. Srinivasan, "An uncertainty-aware transfer learning-based framework for covid-19 diagnosis," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[12] S. Yazdani, S. Minaee, R. Kafieh, N. Saeedizadeh, and M. Sonka, "Covid ct-net: Predicting covid-19 from chest ct images using attentional convolutional network," *arXiv preprint arXiv:2009.05096*, 2020.

[13] M. Maftouni, A. C. C. Law, B. Shen, Z. J. K. Grado, Y. Zhou, and N. A. Yazdi, "A robust ensemble-deep learning model for covid-19 diagnosis based on an integrated ct scan images database," in *IIE Annual Conference. Proceedings.* Institute of Industrial and Systems Engineers (IISE), 2021, pp. 632–637.

[14] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[15] P. Afshar, S. Heidarian, N. Enshaei, F. Naderkhani, M. J. Rafiee, A. Oikonomou, F. B. Fard, K. Samimi, K. N. Plataniotis, and A. Mohammadi, "Covid-ct-md, covid-19 computed tomography scan dataset applicable in machine learning and deep learning," *Scientific Data*, vol. 8, no. 1, pp. 1–8, 2021.

[16] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, and M. Ghassemi, "Covid-19 image data collection: Prospective predictions are the future," *arXiv preprint arXiv:2006.11988*, 2020.

[17] S. Morozov, A. Andreychenko, N. Pavlov, A. Vladzymyrskyy, N. Ledikhova, V. Gombolevskiy, I. A. Blokhin, P. Gelezhe, A. Gonchar, and V. Y. Chernina, "Mosmeddata: Chest ct scans with covid-19 related findings dataset," *arXiv preprint arXiv:2005.06465*,

2020.

[18] M. Jun, G. Cheng, W. Yixin, A. Xingle, G. Jiantao, Y. Ziqi, Z. Minqing, L. Xin, D. Xueyuan, C. Shucheng, W. Hao, M. Sen, Y. Xiaoyu, N. Ziwei, L. Chen, T. Lu, Z. Yuntao, Z. Qiongjie, D. Guoqiang, and H. Jian, “COVID-19 CT Lung and Infection Segmentation Dataset,” https://doi.org/10.5281/zenodo.3757476, 2020, [Online].

[19] MedSeg, “COVID-19 CT segmentation dataset,” http://medicalsegmentation.com/covid19/, 2020, [Online].

[20] X. He, X. Yang, S. Zhang, J. Zhao, Y. Zhang, E. Xing, and P. Xie, “Sample-efficient deep learning for covid-19 diagnosis based on ct scans,” *medRxiv*, 2020.

[21] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “Cbam: Convolutional block attention module,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[22] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[24] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International conference on machine learning*. PMLR, 2015, pp. 2048–2057.

[25] J. Lu, J. Yang, D. Batra, and D. Parikh, “Hierarchical question-image co-attention for visual question answering,” *arXiv preprint arXiv:1606.00061*, 2016.

[26] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in *Proceedings of the IEEE conference on*

computer vision and pattern recognition, 2017, pp. 3156–3164.

[27] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[28] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[30] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[31] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[32] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size,” *arXiv preprint arXiv:1602.07360*, 2016.

[33] T. Goel, R. Murugan, S. Mirjalili, and D. K. Chakrabartty, “Optconet: an optimized convolutional neural network for an automatic diagnosis of covid-19,” *Applied Intelligence*, vol. 51, no. 3, pp. 1351–1366, 2021.

[34] A. Narin, C. Kaya, and Z. Pamuk, “Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks,” *Pattern Analysis and Applications*, vol. 24, no. 3, pp. 1207–1220, 2021.

[35] L. Wang, Z. Q. Lin, and A. Wong, “Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images,” *Scientific Reports*, vol. 10, no. 1, pp. 1–12, 2020.

[36] S. Chaganti, P. Grenier, A. Balachandran, G. Chabin, S. Cohen, T. Flohr, B. Georgescu, S. Grbic, S. Liu, F. Mellot *et al.*, “Automated quantification of ct patterns associated with covid-19 from chest ct,” *Radiology: Artificial Intelligence*, vol. 2, no. 4, p. e200048, 2020.

[37] G. Chassagnon, M. Vakalopoulou, E. Battistella, S. Christodoulidis, T.-N. Hoang-Thi, S. Dangeard, E. Deutsch, F. Andre, E. Guillo, N. Halm *et al.*, “Ai-driven ct-based quantification, staging and short-term outcome prediction of covid-19 pneumonia,” *arXiv preprint arXiv:2004.12852*, 2020.

[38] Q. Yao, L. Xiao, P. Liu, and S. K. Zhou, “Label-free segmentation of covid-19 lesions in lung ct,” *IEEE Transactions on Medical Imaging*, 2021.

[39] Y.-H. Wu, S.-H. Gao, J. Mei, J. Xu, D.-P. Fan, R.-G. Zhang, and M.-M. Cheng, “Jcs: An explainable covid-19 diagnosis system by joint classification and segmentation,” *IEEE Transactions on Image Processing*, vol. 30, pp. 3113–3126, 2021.

[40] K. Gao, J. Su, Z. Jiang, L.-L. Zeng, Z. Feng, H. Shen, P. Rong, X. Xu, J. Qin, Y. Yang *et al.*, “Dual-branch combination network (dcn): Towards accurate diagnosis and lesion segmentation of covid-19 using ct images,” *Medical image analysis*, vol. 67, p. 101836, 2021.

[41] S. Tilborghs, I. Dirks, L. Fidon, S. Willems, T. Eelbode, J. Bertels, B. Ilsen, A. Brys, A. Dubbeldam, N. Buls *et al.*, “Comparative study of deep learning methods for the automatic segmentation of lung, lesion and lesion type in ct scans of covid-19 patients,” *arXiv preprint arXiv:2007.15546*, 2020.

[42] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical

image segmentation," in *International Conference on Medical image computing and computer-assisted intervention.* Springer, 2015, pp. 234–241.

[43] M. Vakalopoulou, G. Chassagnon, N. Bus, R. Marini, E. I. Zacharaki, M.-P. Revel, and N. Paragios, "Atlasnet: multi-atlas non-linear deep networks for medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2018, pp. 658–666.

[44] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2017.

[45] G. Bao, H. Chen, T. Liu, G. Gong, Y. Yin, L. Wang, and X. Wang, "Covid-mtl: Multi-task learning with shift3d and random-weighted loss for automated diagnosis and severity assessment of covid-19," *arXiv preprint arXiv:2012.05509*, 2020.

[46] A. Amyar, R. Modzelewski, H. Li, and S. Ruan, "Multi-task deep learning based ct imaging analysis for covid-19 pneumonia: Classification and segmentation," *Computers in Biology and Medicine*, vol. 126, p. 104037, 2020.

[47] M. Goncharov, M. Pisov, A. Shevtsov, B. Shirokikh, A. Kurmukov, I. Blokhin, V. Chernina, A. Solovev, V. Gombolevskiy, S. Morozov *et al.*, "Ct-based covid-19 triage: deep multi-task learning improves joint identification and severity quantification," *Medical image analysis*, vol. 71, p. 102054, 2021.

[48] G. Crichton, S. Pyysalo, B. Chiu, and A. Korhonen, "A neural network multi-task learning approach to biomedical named entity recognition," *BMC bioinformatics*, vol. 18, no. 1, pp. 1–14, 2017.

[49] T. Gong, T. Lee, C. Stephenson, V. Renduchintala, S. Padhy, A. Ndirango, G. Keskin, and O. H. Elibol, "A comparison of loss weighting strategies for multi task learning in deep neural networks," *IEEE Access*, vol. 7, pp. 141 627–141 632, 2019.

[50] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Mask-guided contrastive attention

model for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1179–1188.

[51] Y. Pang, J. Xie, M. H. Khan, R. M. Anwer, F. S. Khan, and L. Shao, "Mask-guided attention network for occluded pedestrian detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4967–4975.

[52] J. Wang, X. Yu, and Y. Gao, "Mask guided attention for fine-grained patchy image classification," *arXiv preprint arXiv:2102.02771*, 2021.

[53] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," *arXiv preprint arXiv:2005.10821*, 2020.

[54] Q. Lei, G. Li, X. Ma, J. Tian, Y. fan Wu, H. Chen, W. Xu, C. Li, and G. Jiang, "Correlation between ct findings and outcomes in 46 patients with coronavirus disease 2019," *Scientific Reports*, vol. 11, no. 1, pp. 1–6, 2021.

[55] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer, 2020, pp. 173–190.

[56] T. D. Pham, "A comprehensive study on classification of covid-19 on computed tomography with pretrained convolutional neural networks," *Scientific reports*, vol. 10, no. 1, pp. 1–8, 2020.

[57] A. Helwan, M. K. S. Ma'aitah, H. Hamdan, D. U. Ozsahin, and O. Tuncyurek, "Radiologists versus deep convolutional neural networks: A comparative study for diagnosing covid-19," *Computational and Mathematical Methods in Medicine*, vol. 2021, 2021.

[58] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[59] F. Nogueira, “Bayesian Optimization: Open source constrained global optimization tool for Python,” 2014. [Online]. Available: https://github.com/fmfn/BayesianOptimization

[60] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, “Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning,” *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[61] Y. Yang, L.-F. Yan, X. Zhang, Y. Han, H.-Y. Nan, Y.-C. Hu, B. Hu, S.-L. Yan, J. Zhang, D.-L. Cheng *et al.*, “Glioma grading on conventional mr images: a deep learning study with transfer learning,” *Frontiers in neuroscience*, vol. 12, p. 804, 2018.

[62] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” *arXiv preprint arXiv:1312.6034*, 2013.

[63] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[64] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic attribution for deep networks,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 3319–3328.

[65] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 3145–3153.

[66] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[67] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” *arXiv preprint arXiv:1706.05587*, 2017.

[68] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[69] R. Caruana, “Multi-task learning,” *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[70] Y. Wu, Y.-l. Xie, and X. Wang, “Longitudinal ct findings in covid-19 pneumonia: case presenting organizing pneumonia pattern,” *Radiology: Cardiothoracic Imaging*, vol. 2, no. 1, p. e200031, 2020.

[71] S. Machnicki, D. Patel, A. Singh, A. Talwar, B. Mina, M. Oks, P. Makkar, D. Naidich, A. Mehta, N. S. Hill *et al.*, “The usefulness of chest ct imaging in patients with suspected or diagnosed covid-19: a review of literature,” *Chest*, vol. 160, no. 2, pp. 652–670, 2021.

[72] F. Pan, T. Ye, P. Sun, S. Gui, B. Liang, L. Li, D. Zheng, J. Wang, R. L. Hesketh, L. Yang *et al.*, “Time course of lung changes at chest ct during recovery from coronavirus disease 2019 (covid-19),” *Radiology*, vol. 295, no. 3, pp. 715–721, 2020.

[73] M. Parekh, A. Donuru, R. Balasubramanya, and S. Kapur, “Review of the chest ct differential diagnosis of ground-glass opacities in the covid era,” *Radiology*, 2020.

Chapter 4: A Deep Learning Technique for Parameter Estimation in System Dynamics Models

Abstract

We propose a deep learning method for parameter estimation in system dynamics models. Our method relies on training a deep learning model with synthetic data, that ultimately receive a real-world trend (e.g., daily cases of an infectious disease) and produces parameter values of a system dynamics model that represents the trend (e.g., normal contact rate of an epidemic model). We demonstrate the performance of the method using a formerly validated behavioral epidemic model, and compare our estimation accuracy and speed against two conventional estimation methods of Powell and Markov Chain Monte Carlo. The analysis includes various scenarios of stochasticity and data generating models. The results show that our method can outperform the others in speed and accuracy when the level of stochasticity increases. This is especially promising when large-scale complex models are used, data are noisy, and model calibration is needed to be performed frequently and quickly.

Keywords: parameter estimation, machine learning, deep learning, transformer, system dynamics, epidemic modeling

1. Introduction

Accurate parameter estimation is essential in complex dynamic models to ensure that simulation results effectively capture and represent the real-world system being modeled. Advances in computing power and access to large volumes of data have made it more feasible to collect and process large data volumes and inform such models. As a result, an increasing emphasis is made on model estimation techniques and data-driven modeling. Without these approaches, simulation models can fall substantially short of making accurate predictions, and suggest misleading policy recommendations.

Several formal approaches for parameter estimation of complex dynamic models exist. Many of these approaches rely on statistical techniques to minimize the error between the model's predictions and the observed data. Statistical estimation methods arrive at the point estimates of the unknown parameters by minimizing a measure of fit between the simulation results and the available data or maximizing the likelihood of observing the data given the model. Minimizing the measure of fit is typically done through iterative optimization algorithms (e.g., Powell's method [1]) at multiple random starting points. On the other hand, Bayesian statistical approaches, such as Markov Chain Monte Carlo (MCMC) (Metropolis et al. [2]), treat the parameters as random variables with prior probability distributions and assess the posterior distribution by incorporating the new information brought by the observed data.

The conventional approaches, especially the Powell method and MCMC are powerful techniques for parameter estimation, but they have certain limitations that can inhibit their performance. The Powell method is a local optimization technique, sensitive to the starting point, and thus can get trapped in local optima. In addition, it does not provide estimates of confidence intervals. On the other hand, MCMC is computationally intensive and can be challenging in practice for users. It also requires several tuning parameters that can affect the accuracy of estimates. Both methods can also suffer from data noises, and can become slow when dealing with large volumes of data and high-dimensional models.

The most recent experience of the COVID-19 pandemic and the need for offering timely model-based projections for policymakers perfectly depicts the challenge of timely parameter estimation in complex models. For example, one of the well-developed, feedback-rich models of the pandemic that offered timely estimations of COVID-19 spread in 92 nations was by Rahmandad, Lim and Sterman[3]. When asked about the process of modeling and the main challenges, Rahmandad indicates: "*The COVID-19 model went through 79 versions; driven by comparing the match between the model and different aspects of data, from waves of the pandemic to fatality rates and findings from statistical analyses, across over 90 countries. Iterations were thus driven by calibrating a model with over 2000 parameters against data. Even with parallel computation over 40+ cores, each calibration cycle could take multiple days to complete. Overall, figuring, implementing, and fine-tuning the calibration process was easily the most time-consuming component of this project.*" The statement clearly describes data-driven modeling challenges. Particularly, in situations like this, depending solely on traditional methods of modeling and parameter estimation can hinder the primary objective of the model.

Less attention has been paid to the potential application of machine learning techniques for the calibration of dynamic models (exception, [4, 5, 6]). Generally, machine learning algorithms can be trained to match patterns in data and make predictions or decisions based on these patterns [7]. For parameter estimation, these algorithms can be used to estimate the relationship between parameter values and model outcome, and once it is trained, the algorithm can be used to offer a quick estimation of parameter values [8]. Among such methods, deep learning is the state-of-the-art technique that can quickly learn the representation of the data with more flexibility. Deep learning is a subset of machine learning that uses a deep neural network architecture to learn and recognize patterns directly from the data without requiring explicit feature engineering [9]. Deep learning algorithms have achieved state-of-the-art results in pattern recognition from a wide range of data formats, including images, text, speech, and time series data [10, 11, 12]. Transformer, a type of deep learning architecture [13], revolutionized tasks involving sequential data, such as natural language processing (NLP) and speech recognition by learning longer-term dependencies without recurrent connections.

In this paper, we investigate the potential of using deep learning (DL) models for the parameter estimation of system dynamics (SD) models. There have been considerable advancements in DL given the increase in the available computational resources and labeled data [9, 10]. We propose the utilization of DL architectures as surrogate models for SD parameter calibration. We argue and provide evidence from a pilot test that, once the surrogate model is built, it can be used to quickly and accurately calibrate the model parameters by learning its dynamic patterns, and the estimate values, if not better, are often as good as conventional approaches. The remainder of this paper is organized as follows. Next, we provide a review of parameter estimation techniques and advancements in DL with applications in dynamical system modeling. Then, our proposed methodology is detailed, followed by our experimental design of testing the proposed method. Lastly, the results of our analysis are reported and discussed.

2. Parameter Estimation Techniques

Parameter estimation, also referred as model calibration, involves finding the best values for certain model parameters of a model by comparing the model's behavior to that of the actual system it represents. It is important to note that model calibration is only one of many required approaches for building trust in models, and the parameter calibration process will only produce reliable results if the model structure is adequate and passes the common validation and verification tests [14, 15, 16, 17].

Choosing the calibration method heavily depends on the available data type (time series vs. statistical moments), the complexity of the model, and statistical inference approaches (Frequentist vs. Bayesian). Minimizing a measure of the fit error (e.g., mean squared error, mean absolute error, mean absolute percentage error) between the observed data and the values predicted by the model is the common way for finding the point estimates of the parameters. A common approach to finding the optimal parameter values is the Powell (1964) optimization algorithm. A built-in feature in many software, including Vensim, the Powell method starts by choosing a set of initial parameter values, calculating the objective function for those values, and then searching for the best values through hill-climbing algorithms. The method is commonly used in system dynamics for parameter estimation (e.g., [18, 19, 20, 21]).

Another commonly used, but computationally more intensive, approach, is MCMC. MCMC is often employed in Bayesian inference to generate samples from the posterior distribution of parameters by performing a random walk on the likelihood surface. In Bayesian inference, probability distributions represent degrees of belief and incorporate prior knowledge about parameter values by using prior probability distributions. The method updates the prior distribution based on observed data, obtaining the most credible interval and plausible parameter value from the posterior distribution. Not as commonly as Powell, but still there are system dynamics models that employ MCMC for parameter estimation (e.g., [3, 22, 23, 24]). In their recent work, Andrade et al. [25] suggest Hamiltonian Monte Carlo as a more efficient algorithm for Bayesian parameter inference in system dynamics, and provide a detailed workflow for effectively communicating the outcomes of model calibration using an SEIR model calibration as an example.

The Method of Simulated Moments [26] is also a well-established estimation method in econometrics and is applied to SD models [27, 28]. The central idea behind this method is to define a set of appropriate statistical moments (e.g., mean (first moment), variance (second moment), etc.) of empirical data and then find the parameters by minimizing the weighted difference between these moments calculated on data and their simulated counterparts. This method can use not only time-series data but also cross-sectional population statistics. The greater availability of cross-sectional data opens many opportunities for dynamic modelers to work on domains traditionally dominated by regression models. Moreover, the indirect inference method proposed in [29] is very similar to the Method of Simulated Moments in essence. However, it provides more flexibility by relaxing the constraint of choosing the statistics from statistical moments and states that a more comprehensive set of functions (not just statistical moments) can be matched for estimating the parameters. Hosseinichimeh et al. [30] offers a detailed guide for applying indirect inference calibration to SD models.

More recent advancements include filtering techniques to better calibrate models when data are noisy. Kalman filtering [31] for example is a mathematical technique that enables the estimation of a system's state by leveraging noisy measurements. It is widely used in a variety of applications, including control systems, signal processing, and navigation. The fundamental concept behind Kalman filtering involves a recursive algorithm that considers the system's noisy measurements to adjust and improve its estimation of the state. The Kalman filtering algorithm is composed of two main steps. The first step, referred to as the prediction step, involves forecasting the upcoming state of the system based on the prior state estimate and the SD model. The second step, the update step, utilizes the current measurements to refine and update the state estimation. Kalman filtering is used in SD to improve parameter calibration by reducing the effect of noise in the data [32, 33].

Recently, to a limited extent, machine learning techniques have been applied to support and automate SD validation testing and calibration tasks. Mert Edali [4] proposes a random forest meta-model for automated and interpretable analysis of input-output relationships in system dynamics models, illustrated through case studies. In their previous work [5], Edali and Yücel implemented the proposed explainable meta-model procedure with active learning sampling design on an influenza epidemic case study, demonstrating its effectiveness in assessing various intervention strategies and identifying transitions between different model behavior modes. In his work [8, 34], Duggan employs the R programming language to provide a clear and practical illustration of how machine learning techniques can be effectively utilized to analyze time series outputs from a system dynamics model. The paper [6] proposes a novel policy design method for system dynamics models based on recurrent neural networks, which can search in both the system structure and parameter space simultaneously.

Supervised machine learning is the task of learning to map a set of inputs (x_i) to their corresponding outputs(y_i) given an adequate number of training examples $\{(x_1, y_1), .., (x_N, y_N)\}$. Based on the universal approximation theorem [35], a single-layered artificial neural network (ANN) with a sufficient number of neurons and certain activation functions can approximate any continuous function to a reasonable accuracy. Given that, a properly trained ANN network can be used as a surrogate for any SD model. There is a limited number of works in the literature that use ANNs for approximating the nonlinear mapping of the SD outputs and input parameters. In [36], neural networks are employed as parameter estimators of a parameterized model structure, where the model structure is utilized to generate training examples. More specifically, the system response (output of the system) is used as the ANN model input and its associated SD model parameters, selected at random from a range, as the ANN output. Consequently, the ANN model is trained on the task of estimating the model parameters from the observed data. The authors intentionally introduce some noise to their simulation outputs in order to improve the robustness of their learning method and test their model for identifying the processing delay in an online controller.

In a study, three machine learning approaches, including multi-layer perceptron, convolutional neural network (CNN), and long short-term memory (LSTM), were tested by [37] to estimate the reproduction number, latent period, and recovery rate parameters of a respiratory virus, formulated as a supervised regression problem. First, they use their SEIR individual-based model to generate simulations for random parameters. Next, they train their machine learning models to learn the mapping between simulated daily infected and the corresponding parameters. An ANN with time encoding is reported to outperform Recurrent and convolutional neural networks in terms of both accuracy and computational speed by [37].

As stated, the traditional methods of model calibration have several limitations which influence their effectiveness. Particularly, most of the methods are computationally intensive and time-consuming, sensitive to noise in data, and inaccurate when dealing with large volumes of data and high-dimensional models. Our purpose is to contribute to calibration methods by applying a DL technique and examining its performance against two conventional methods. Our proposed method is described in the next section.

3. Proposed DL-based Calibration Method

The SD simulation model generates the output or dynamic behavior of a system, given the model parameters. Thus, it can be viewed as a forward problem. On the other hand, the process of parameter calibration for an SD model (and for any other models) aims to determine the input parameters of the model that yield the best match between the simulated behavior of the model and the observed behavior of the system. This process can be seen as an instance of the inverse problem, where the input parameter is inferred from the observed output (Figure 23). A common approach to the inverse problem is optimization, that is, minimizing a function that depicts error. Our proposed idea includes offering a learning model (instead of an optimization model) that takes the dynamic trend of interest (input) and produces model parameters for the SD model (output). Such a learning model should be, first, trained to map the observed outputs to underlying model parameters, making them a promising approach for calibrating SD parameters. Once a model that maps dynamic trends to parameter values is trained, then it can be used for parameter estimation purposes.

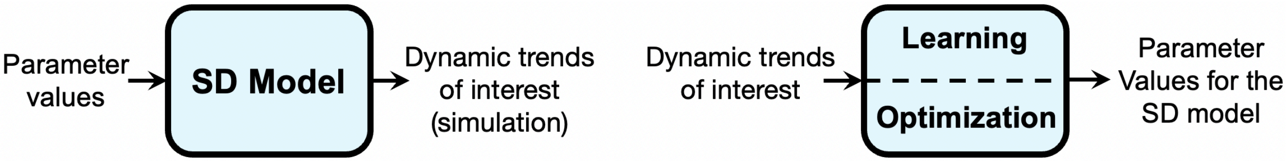

Parameter values
SD Model
Dynamic trends of interest (simulation)
Dynamic trends of interest
Learning
Optimization
Parameter Values for the SD model

Figure 23: SD VS DL model

Our proposed DL calibrator requires the following steps:

- Employing the SD model to generate a large amount of data with various model parameters and noise streams to be used for DL model training.
- Training the DL model on the generated data to learn the mapping of the model's output to their corresponding model parameters.
- Applying the trained DL model to predict the model parameters given new input data.

We particularly propose a transformer-based deep network that is well-suited for capturing complex and nonlinear relationships between input and output sequences. The parallel processing capabilities of the transformer model further improve its performance, making it a promising choice for our application.

- **Transformer Network**

Transformers, a type of neural network architecture initially introduced for machine translation tasks in [13], has quickly revolutionized many DL tasks, especially natural language processing and sequence learning. DL networks have long been dominated by convolutional or recurrent layers, with an appended attention mechanism in recent years. The Transformer, however, is built solely on self-attention mechanisms to weigh the importance of different elements of an input sequence. This allows for better long-range dependencies modeling and the avoidance of vanishing gradient problems. Additionally, transformers are highly parallelizable, making them well-suited for modern hardware architectures. These advantages have made transformer-based models the go-to solution for learning from sequential data.

In contrast to a recurrent neural network, which decodes the output sub-sequences through a general learned representation from the whole input sequence, the Transformers' attention function creates a learning mechanism for the decoder to pick the input part to pay closer attention to. As a result, an attention function can model the input and output dependencies irrespective of their distance in the sequence because the information's path length is shorter. The transformer has the conventional encoder-decoder structure in which the encoder takes the input and learns its representation or context (left branch of the Transformer structure in Figure 24), and the decoder generates the output sequence one step at a time based on the encoded representation (right branch of the Transformer structure in Figure 24). N is the number of repeating and stacking of encoder and decoder layers. The Input and Output Embedding part is specific to natural language processing (NLP) and transforms words into numerical vectors. Since the model has no recurrence or convolution, the positional information of the sequence is lost. Positional Encoding is to retain the sequence's order by encoding and adding it back into the vectors. This way, the network can still tell the relative positions of sub-sequences when weighting them.

More specifically, the encoder of the input sequence builds key(K), value(V) pairs, and the previous decoder layer makes the queries(Q). The attention function maps the query to the output at the next step, computing the next output as a weighted sum of the values, with the weight assigned based on the similarity of the query with the corresponding keys:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q.K^T}{\sqrt{d_k}}\right).V \tag{18}$$

This function is depicted in the Scaled Dot-Product Attention part of Figure 24. d_k denotes the keys and queries dimension. The scaling factor of $\frac{1}{\sqrt{d_k}}$ is shown to improve the dot product attention performance for larger d_ks.

In addition to the encoder-decoder attention function explained above (the top right multi-head attention in Figure 24), there are also self-attention modules in the encoder and decoder to assign importance weights to their sub-sequences. In a self-attention layer, the keys, values pairs are all the current encodings, and the queries are the output of the previous layer. The only difference in the decoder's self-attention is that only the so far produced outputs are used, and there is no leak of information from the future.

Instead of attending to just one part of the input, the multi-head attention, depicted on the left side of Figure 24, performs h attention functions in parallel on the linearly projected lower-dimensional key, value, and queries. The head outputs are then concatenated and projected to produce the output values.

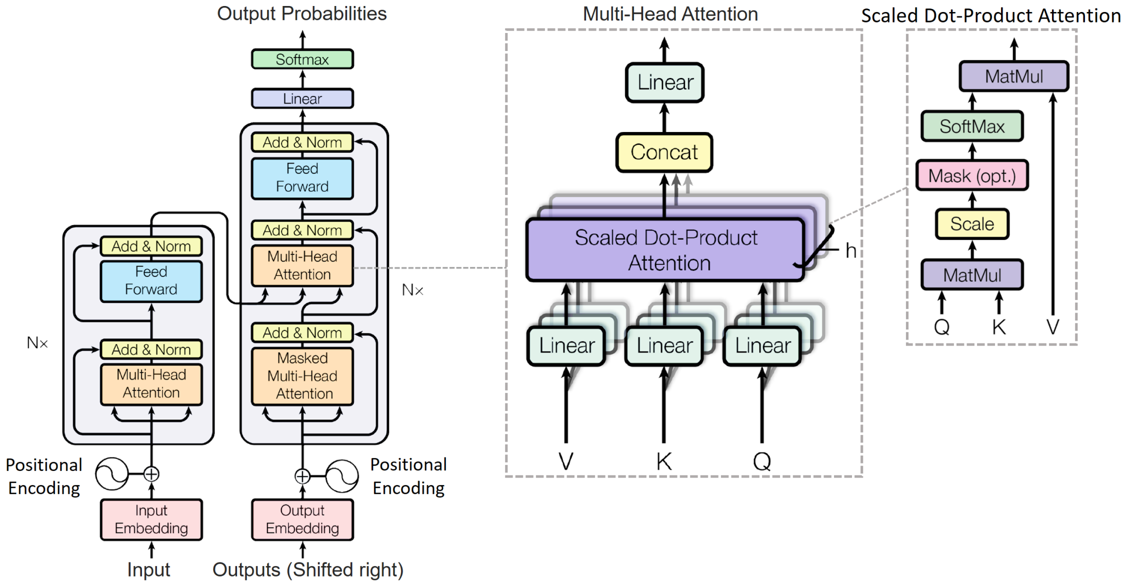


Figure 24: The Transformer - model architecture [13]

The Transformer network has proven to be a powerful tool for modeling and forecasting time-series data. In particular, [38] demonstrated that their modified Transformer network could effectively capture intricate patterns and dynamics in time-series data, as well as accurately forecast system state variables using time delay embeddings.

While the Transformer network has shown promise in various time-series modeling tasks, it has not yet been widely used for parameter estimation of SD models. However, there are several reasons why the Transformer network could be advantageous for this task. For example, its flexible nature allows for updating parameters over time and adapting to new observations. Additionally, it can handle variable input and output time series lengths, and exhibit a more interpretable and human-like focusing behavior in the sequence structure, compared to other neural network architectures, such as recurrent neural networks. Finally, the Transformer network allows for significantly more parallelization than recurrent neural networks, which can greatly reduce training time and improve scalability.

4. Method

The common approach to evaluate an SD parameter calibration framework is to test it on synthetic data generated by the SD model, known parameters, and added autocorrelated noise [3, 39]. Following this approach, we compare the performance of our DL calibrator against the Powell and MCMC methods on the SD-simulated data.

- **Step 1: Synthetic data generation**

We generate synthetic data employing a stochastic SEIRb model. The SEIRb model, proposed by Rahmandad et al. [18], builds upon the traditional SEIR framework, which divides the population into Susceptible (S), Exposed (E), Infected (I), and Removed (R) compartments, flowing from left to right over time through the following differential equations:

$$N = S(t) + E(t) + I(t) + R(t) \tag{19}$$

$$\frac{dS}{dt} = -\beta S \frac{I}{N} = -(c\beta_0) S \frac{I}{N} \tag{20}$$

$$\frac{dE}{dt} = \beta S \frac{I}{N} - \frac{E}{\gamma_e} \tag{21}$$

$$\frac{dI}{dt} = \frac{E}{\gamma_e} - \frac{I}{\gamma_i} \tag{22}$$

$$\frac{dR}{dt} = \frac{I}{\gamma_i} \tag{23}$$

where N is the total population, β is the infectious contacts depending on the disease infectivity (β_0) and the average number of contacts (c), γ_e indicates the average duration between exposure and symptom onset, and γ_i denotes the average infection period from symptom onset to death or recovery.

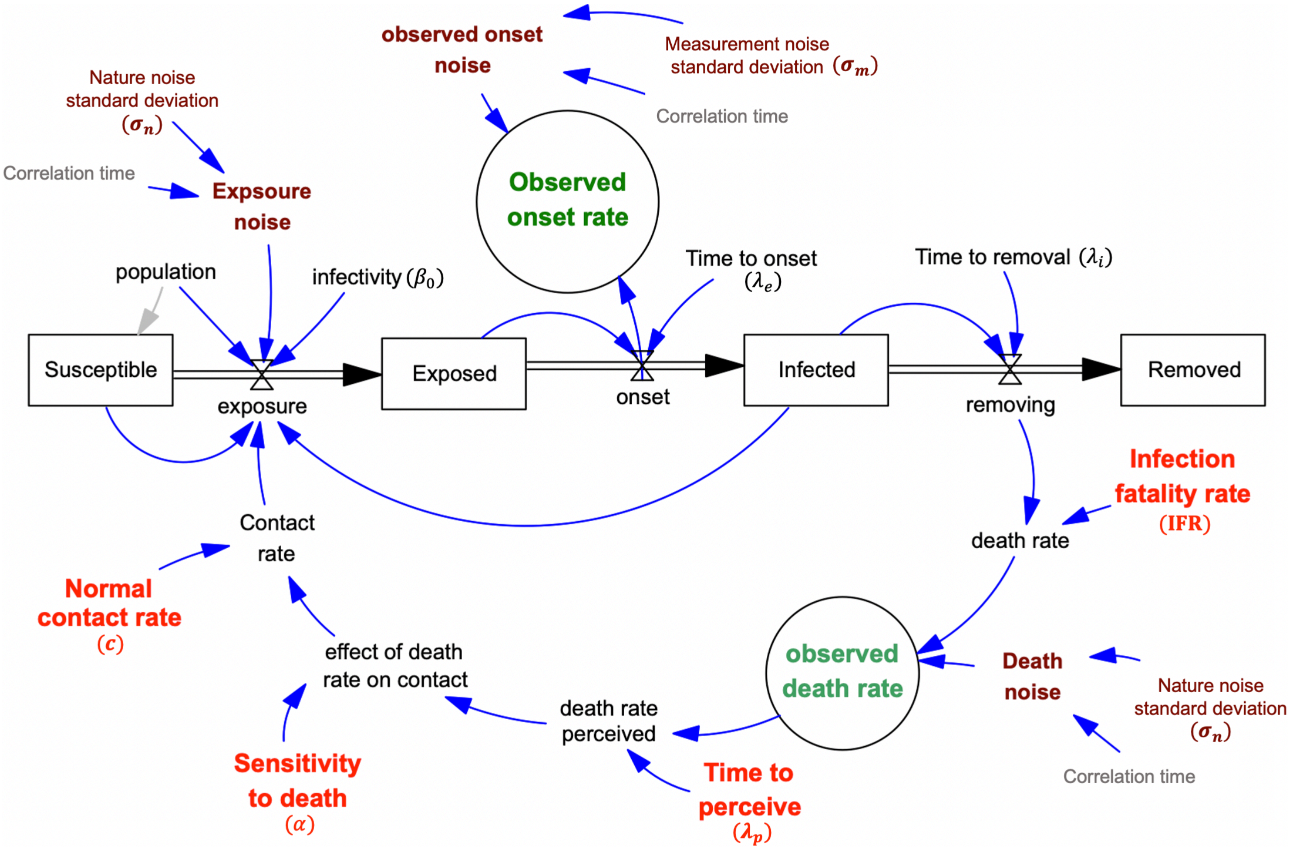

observed onset noise
Measurement noise standard deviation (σ_m)
Correlation time
Nature noise standard deviation (σ_n)
Correlation time
Expsoure noise
Observed onset rate
population
infectivity (β_0)
Time to onset (λ_e)
Time to removal (λ_i)
Susceptible
exposure
Exposed
onset
Infected
removing
Removed
Infection fatality rate (IFR)
death rate
Contact rate
Normal contact rate (c)
effect of death rate on contact
observed death rate
Death noise
Nature noise standard deviation (σ_n)
Correlation time
death rate perceived
Sensitivity to death (α)
Time to perceive (λ_p)

Figure 25: SEIRb model [18] with three noise generators. The circled variables denoted in green are dynamic trends of interest and the four parameters, denoted in red, are calibrated.

The SEIRb model, depicted in Figure 25, captures human risk response to an evolving pandemic through a behavioral feedback loop to improve the predictive power of the basic SEIR model. In this model, as the risk of death increases, people decrease their contacts which decreases the spread of the disease and risk of death, forming a balancing feedback loop. This feedback loop is argued to be essential in modeling the spread of an infectious disease [40, 41, 42]. In the model, specifically, the population's perception of the death rate is formulated as a lagged variable of observed daily deaths (f) (as shown in Equation 25). As the mortality rate increases, risk perception rises, leading to decreased social interactions (as shown in Equation 26) and, consequently, a decline in the number of cases. The new behavioral loop includes three parameters denoted as IFR (infection fatality ratio), α (sensitivity to death), and γ_p (time to perceive risk). The SEIRb model has been previously validated as a competitive predictive model for COVID-19 trajectories when compared to the best models in the Centers for Disease Control and Prevention (CDC) forecasting model set [42].

$$f = \text{IFR} \times \frac{dR}{dt} \tag{24}$$

$$\frac{df'}{dt} = \frac{f' - f}{\gamma_p} \tag{25}$$

$$\beta = c\beta_0 = e^{-\alpha f'}\beta_0 \tag{26}$$

The model parameters used for calibration in this work are as follows:

- **Infection Fatality Rate** (IFR): proportion of infected who die from the infection.
- **Time to Perceive** (γ_p): average lag time to perceive to the confirmed death cases.
- **Sensitivity to Death** (α): average public sensitivity to perceived death.
- **Normal Contact Rate** (c): pre-epidemic average number of contacts per person.

The two time-series inputs to our calibration models are daily reported infected cases (Observed Onset Rate) and daily reported death cases (Observed Death Rate). The error in the reports is modeled by auto-correlated noise (pink noise). Through SD simulations on the SEIRb model and randomly choosing the four parameters using the Latin Hypercube sampling [43] method, we generated 100,000 synthetic onset and death rates time series datasets that were employed to train and validate the DL calibrator. The model's remaining fixed parameters and initial values are set as follows:

Table 9: Model parameters.

Parameter	Unit	Value
Population (N)	Person	1,000,000
Initial Susceptible (S_0)	Person	999,999
Initial Exposed (E_0)	Person	1
Initial Infected (I_0)	Person	0
Initial Removed (R_0)	Person	0
Correlation time	Day	15
Noise standard deviation (σ)	Dimensionless	(0 - 0.45)
Infectivity (β_0)	Dimensionless	0.05
Time to onset (γ_e)	Day	6
Time to removal (γ_i)	Day	10

- **Step 2: Training DL**

We translated our Vensim model to Python using the PySD library [44] to streamline the simulations, data manipulation, and storage. Data generation took about 30 minutes. First, we split the 40,000 simulated data into training and validation using the split ratio of 90:10. Then, we separately generated 100 test data points with unique auto-correlated noise parameters to be used for comparing the DL and iterative optimization calibrators. We kept the test set relatively small as the benchmark iterative calibrator is time-intensive.

The DL calibrator network architecture, illustrated in Figure 26 (a), is designed to accurately estimate the four specified parameters from the two time series inputs. The Time2Vec block introduced in [45] and depicted in Figure 26 (c) learns the time embedding of the two inputs, which is essential for capturing the temporal dynamics in the data. Next, the concatenated input and time embedding is fed to the Transformer block (Figure 26 (b)) to capture the complex dynamics in the inputs and model their dependencies over time through its multi-

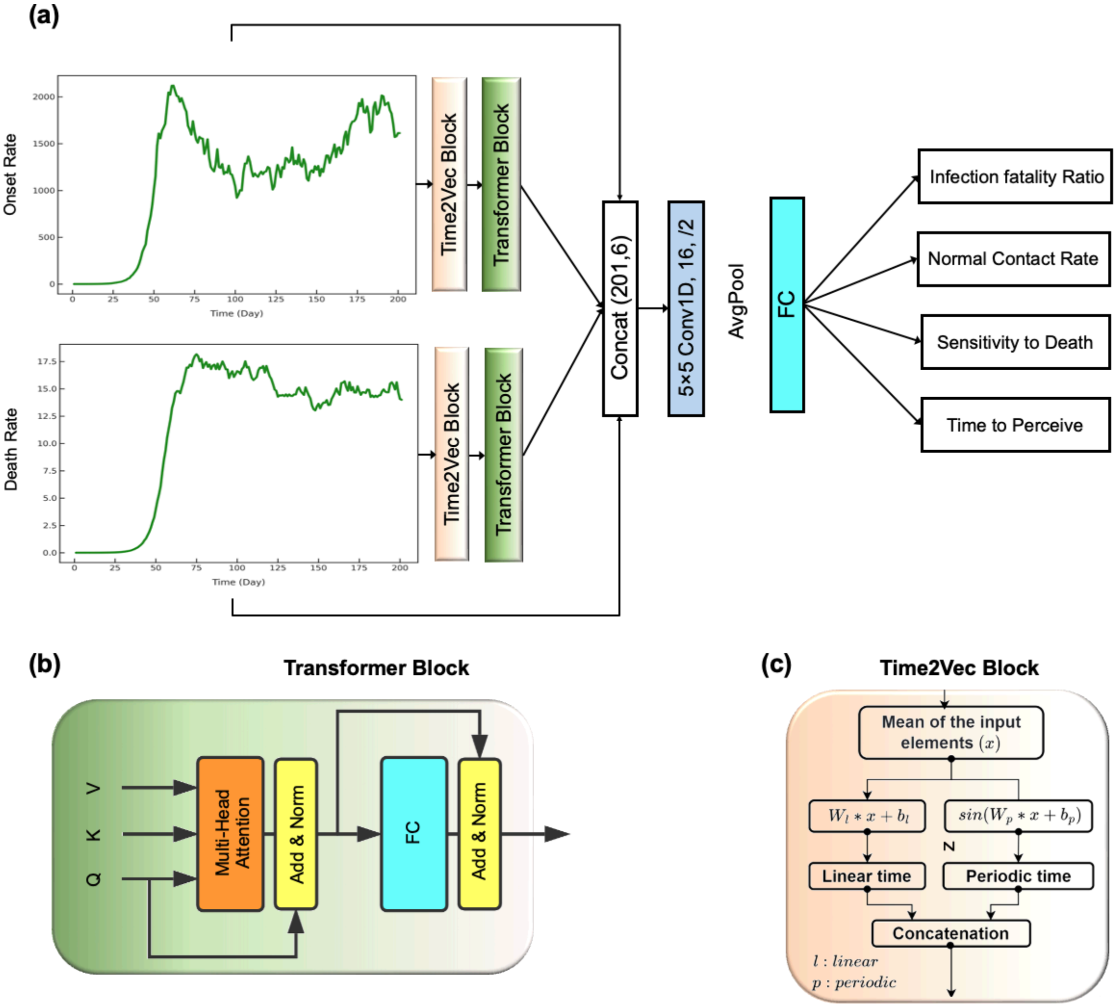


Figure 26: (a) DL calibrator network architecture, taking the dynamic trends as input and predicting the model parameters (b) Transformer block (c) Time2Vec Block for capturing time embedding

head attention mechanism. The transformer block outputs of the two branches are then concatenated and processed by a series of convolutional and dense layers. The last dense layer is branched out to output each of the four parameters. Incorporating the Time2Vec and Transformer blocks into our DL calibrator network architecture yielded improved calibration performance. This improvement can be attributed to these blocks' capacity to capture the temporal dynamics and intricate dependencies in the data while effectively handling the noise in the input.

The model was trained using an Adam optimizer with a learning-rate warm-up for greater stability and a mean square error loss function. We selected a batch size of 16 and trained the network for 50 epochs, saving the epoch's weights with the best validation performance. We used 1 NVIDIA GeForce RTX 2080 Ti GPU and Keras library to train our model.

- **Step 3: Test DL**

The trained DL model can be then employed as the surrogate model of the inverse SD model to estimate the unknown parameters of unseen simulated data at a much higher speed compared to the optimization-based methods. As shown in Figure 25, auto-correlated noise generators are embedded in the SEIRb model to generate the process and measurement noises to make the synthetic data more realistic. The auto-correlated noise parameters (standard deviation and correlation time) are also selected randomly to account for the unknown noise processes. All other model parameters were fixed.

5. Experimental Design

We conduct three general assessments to evaluate the performance of our proposed method against the conventional Powell and MCMC calibrations. For Powell and MCMC, we employed Vensim's calibration tool to minimize the fit error between the observed and simulated data. We assigned weights to the two data streams, onset and death rates, based on the inverse standard deviation of each stream to account for their varying scales and spreads.

The first experiment evaluates the ability of our method to replicate the dynamic trends in the onset and death rate over ten randomly generated synthetic datasets. The experiment is conducted by first generating the synthetic data with low-level noise (σ <0.15), then asking each method to predict the parameters. The parameters are then used to produce the onset and death rates given the model structure with the noise generator turned off. We turn off the noise generator at this stage because the actual noise generator is unknown, and we used it initially to make the synthetic data more similar to real-world data. We use the averaged mean absolute error (MAE) of each of the rates as our performance metric.

The second assessment examines the accuracy of the predicted parameters in matching their ground truth values under different levels of stochasticity. This experiment includes 12 conditions (3 estimation methods X 4 noise levels). We are specifically interested in comparing our method against two other methods of Powell optimization and MCMC, under four different levels of stochasticity. We defined four levels of noise based on the standard deviation (σ) of the noise, categorized as low (σ <0.15), medium (0.15 < σ <0.25), high (0.25 < σ <0.35), and very high (0.35 < σ <0.45). Figure 25 depicts the SEIRb model with embedded auto-correlated noise generators used to generate both process and measurement noises.

Lastly, we set up an experiment to examine the impact of the model's accuracy on the calibration performance. For this, we utilized the SEIR model as the model structure to train the DL calibrator. Then, we evaluated its performance by testing it on data generated from the SEIRb model. This approach enabled us to determine the impact of the differences between the model structure and the actual system on the calibration performance.

6. Results

6.1 Relative Performance of DL

We compare the effectiveness of three calibration methods (the proposed DL, Powell, and MCMC) in capturing the onset and death rate trends on randomly generated synthetic datasets. Figure 27 shows an example of calibration results from three different methods for a specific set of parameter values with a low level of noise in synthetic data. As shown all methods are fairly replicating the data. In order to systematically compare the methods under various noise levels, we repeated the test 10 times. Table 10 presents the average mean absolute error between the synthetic data and the dynamics generated using the calibrated

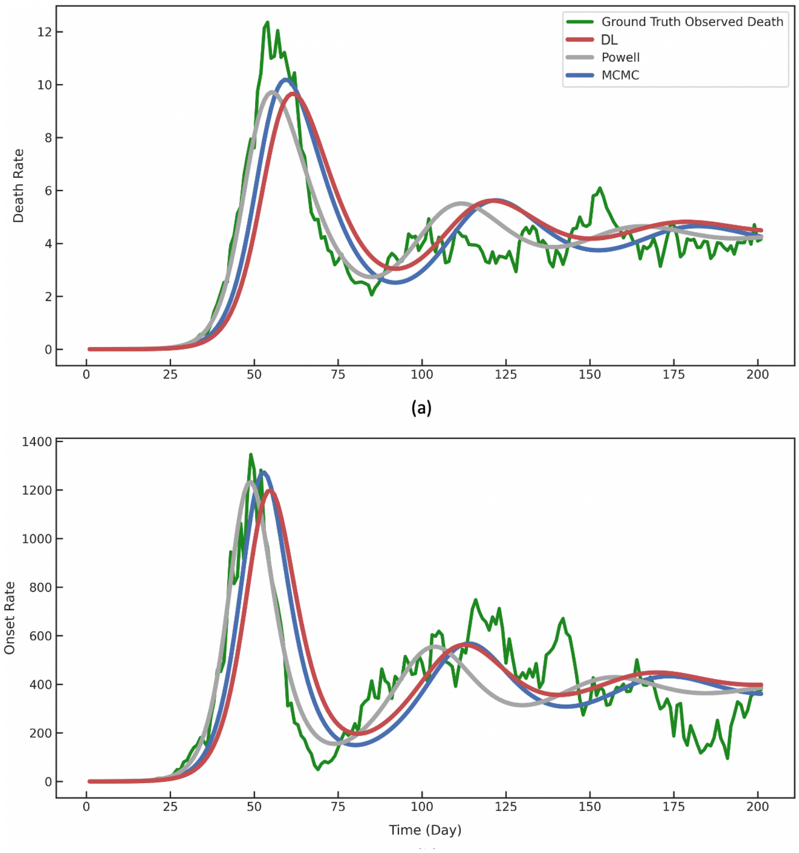


Figure 27: (a) Simulated death rate (b) Simulated onset rate, with ground truth parameters and low-level noise (green), the proposed DL method (red), Powell (grey), and MCMC (blue) calibrated parameters and no noise

parameters over the 10 datasets. Both qualitative and quantitative evaluations indicate that our proposed DL calibrator can capture the dynamic trends at a level that is comparable to the Powell and MCMC methods.

Furthermore, the proposed DL calibrator requires significantly less calibration time than the other two methods, as reported in Table 10. While Powell and MCMC rely on iterative optimization at multiple random starts to avoid local optima, the DL calibrator only requires a pre-trained function evaluation. However, it is worth noting that, unlike the other methods, the DL approach necessitates two additional steps of data generation and model training. Nonetheless, these steps are only required once per model, unlike the calibration time that must be repeated for each new calibration.

Table 10: Evaluating the proposed DL calibrator against Powell and MCMC averaged over 10 randomly generated synthetic data.

Calibration approach	MAE (Onset Rate)	MAE (Death rate)	Calibration Time (s)	Training Time (s)
Powell	90.08	0.58	87.47	-
MCMC	125.32	0.93	107.12	-
DL (Proposed)	105.29	0.80	0.0024	497.53

6.2 Effect of Stochasticity

To investigate the calibration methods' robustness to different levels of noise in the data, we introduce three noise generators (observed onset noise, exposure noise, and death noise) into the SEIRb model (as demonstrated in Figure 25) to generate data at four levels of noise. Specifically, we generate onset rate and death rate data with fixed parameter values but with 100 different random noise streams at each level of noise. The impact of noise on the calibration methods is evaluated by analyzing the distribution of calibrated parameter values around the true parameter value.

Figures 28 presents box plots summarizing the results of Powell, MCMC, and DL calibrators for four parameters across four levels of noise, with the true parameter value shown as a dashed line. As expected, the increase in noise adversely affects all the methods, leading to a more widely dispersed distribution of calibrated values. By examining the range of values in the box plots, it is evident that the DL calibrator displays a lower susceptibility to noise as compared to the other two techniques.

We also systematically compare the estimations across the methods to examine if DL estimations' accuracy is statistically different from the other methods. Our analysis reported in Table 11 shows that DL estimated normal contact rate and time to perceive better than the other two methods consistently in medium, high, and very high levels of noise ($p<0.001$) while not doing worse when the noise level is low ($p>0.38$). In the case of the Infection

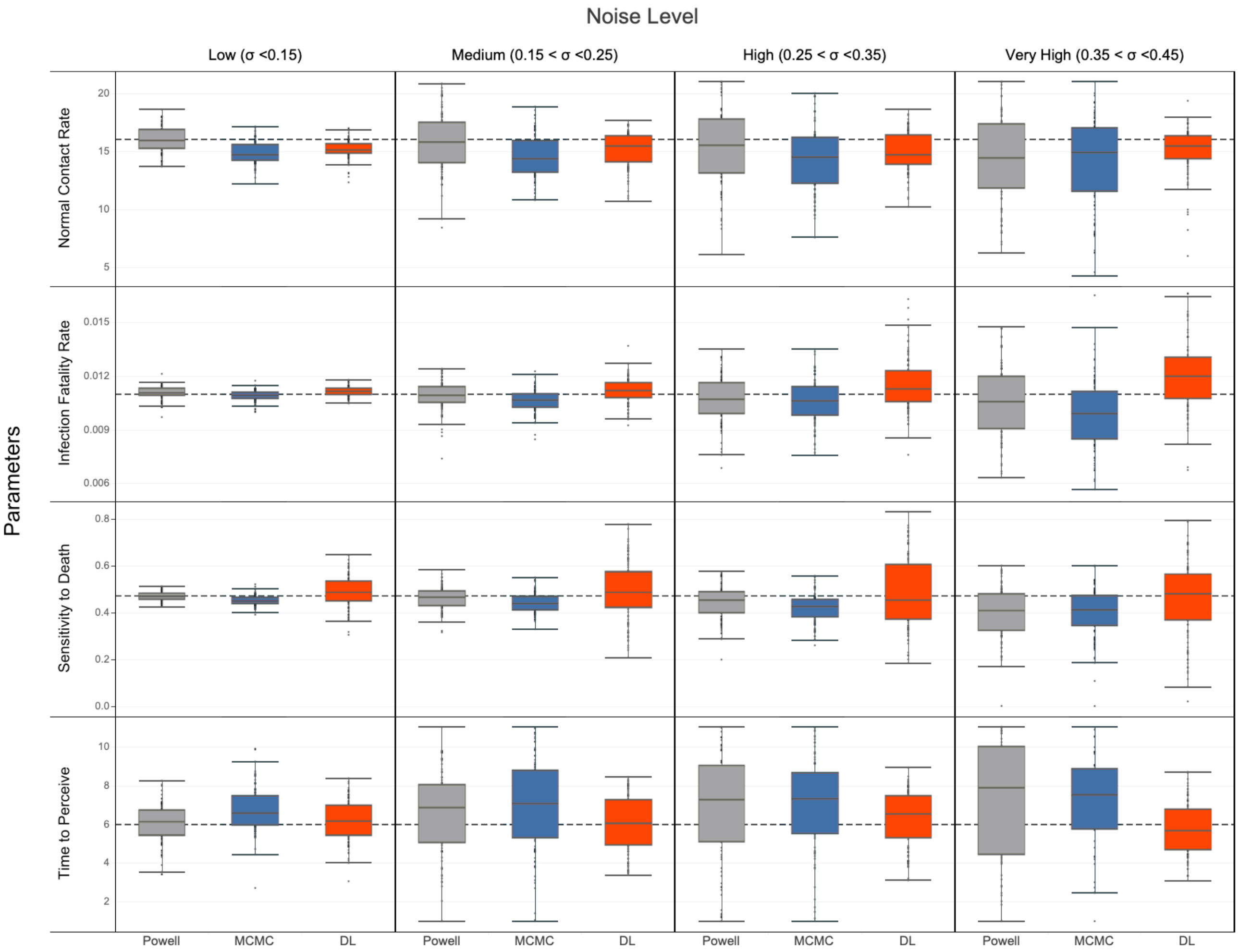


Figure 28: The effect of stochasticity on the Calibrators

Fatality Rate (IFR), there is no significant statistical difference between the Deep Learning (DL) estimations and those obtained using Powell and Markov Chain Monte Carlo (MCMC) methods. However, when it comes to sensitivity to death across low, medium, and high noise levels, the other methods demonstrate marginally better performance (evidenced by a p-value of less than 0.001 and an approximate 0.06 improvement in the average absolute estimation error).

Figure 29 summarizes these results and compares the methods in two metrics. Figure 29 (a) compares the time requirements of the three methods, with DL data generation and

Table 11: Paired t-tests comparing average absolute parameter estimation errors between DL and the other two methods (*** p<0.001, ** p<0.01, * p<0.05, with the number of asterisks indicating the level of statistical significance that the observed difference in performance occurred by chance).

Parameter	Noise Level	Pair	DL - Benchmark	p-value
Time to Perceive	Low	DL & Powell	-0.0772	0.3772
Time to Perceive	Low	DL & MCMC	0.1165	0.2869
Time to Perceive	Medium	DL & Powell	0.6574	<0.001***
Time to Perceive	Medium	DL & MCMC	1.0599	<0.001***
Time to Perceive	High	DL & Powell	1.4042	<0.001***
Time to Perceive	High	DL & MCMC	1.4863	<0.001***
Time to Perceive	Very High	DL & Powell	2.1022	<0.001***
Time to Perceive	Very High	DL & MCMC	2.4336	<0.001***
Normal Contact Rate	Low	DL & Powell	-0.0410	0.6660
Normal Contact Rate	Low	DL & MCMC	0.3097	0.0028**
Normal Contact Rate	Medium	DL & Powell	0.6485	0.0007**
Normal Contact Rate	Medium	DL & MCMC	0.6295	0.0003**
Normal Contact Rate	High	DL & Powell	1.1622	<0.001***
Normal Contact Rate	High	DL & MCMC	0.7592	0.0010**
Normal Contact Rate	Very High	DL & Powell	2.1530	<0.001***
Normal Contact Rate	Very High	DL & MCMC	1.8267	<0.001***
Infection Fatality Rate	Low	DL & Powell	0.0001	0.0982
Infection Fatality Rate	Low	DL & MCMC	0.0006	0.0168*
Infection Fatality Rate	Medium	DL & Powell	0.0000	0.5470
Infection Fatality Rate	Medium	DL & MCMC	0.0002	0.0139*
Infection Fatality Rate	High	DL & Powell	0.0000	0.9409
Infection Fatality Rate	High	DL & MCMC	-0.0001	0.6800
Infection Fatality Rate	Very High	DL & Powell	-0.0002	0.4990
Infection Fatality Rate	Very High	DL & MCMC	-0.0001	0.5726
Sensitivity to Death	Low	DL & Powell	-0.0393	<0.001***
Sensitivity to Death	Low	DL & MCMC	-0.0307	<0.001***
Sensitivity to Death	Medium	DL & Powell	-0.0612	<0.001***
Sensitivity to Death	Medium	DL & MCMC	-0.0573	<0.001***
Sensitivity to Death	High	DL & Powell	-0.0613	<0.001***
Sensitivity to Death	High	DL & MCMC	-0.0574	<0.001***
Sensitivity to Death	Very High	DL & Powell	-0.0298	0.0229*
Sensitivity to Death	Very High	DL & MCMC	-0.0358	0.0197*

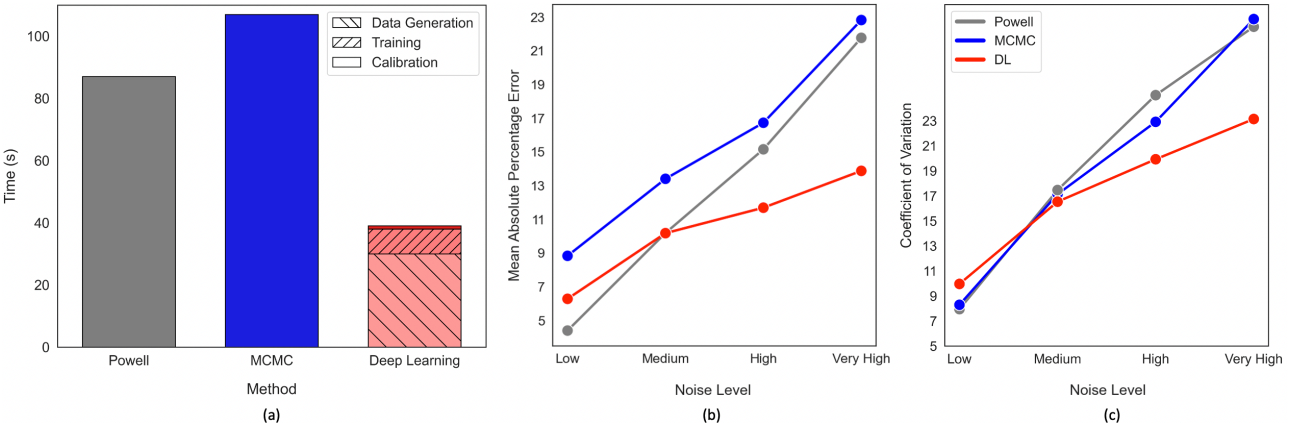


Figure 29: (a) Calibration time with DL data generation and training time prorated for 60 calibrations, (b) mean absolute percentage error (MAPE), and (c) coefficient of variation at different levels of noise (averaged over the four calibrated parameters).

training steps prorated for 60 calibrations, demonstrating that even after accounting for training time, our proposed approach has a lower time requirement when used frequently. To make a better sense of the efficiency of our model calibration consider a model like the one developed by Rahmandad et al [3] which is periodically calibrated, every time for 90 regions. In addition to the fact that one round of training can work for estimating 90 regions' parameters independently, as long as the model structure is not changed the same trained DL can be used for frequent refining parameter values in possibly a fraction of a second. It is possible to further reduce the time required for data generation and training steps of the DL calibrator through parallelization, active learning, and fine-tuning on pre-trained models.

Furthermore, the figure (panels b and c) reports Mean Absolute Percentage Error (MAPE) and Coefficient of Variation (CV). MAPE computes the average percentage difference between predicted and actual values, and is used to capture the accuracy, and CV measures the variability of a data set by dividing its standard deviation by the mean, which is used to capture the variability of the parameter estimation results given the noise level. Results in Figure 29 (b) and (c) demonstrate that the proposed DL calibrator outperforms the other two methods as the noise level increases. The DL calibrator exhibits lower mean absolute percentage error and coefficient of variation at High and Very High noise levels.

6.3 Effect of SD Training Model Accuracy

With the considerable accuracy of DL, how important is it to have a proper SD model? To examine the impact of SD model accuracy used for training DL, we train our DL model with an “imperfect” SD model of SEIR (imperfect as the test data are generated with SEIRb).

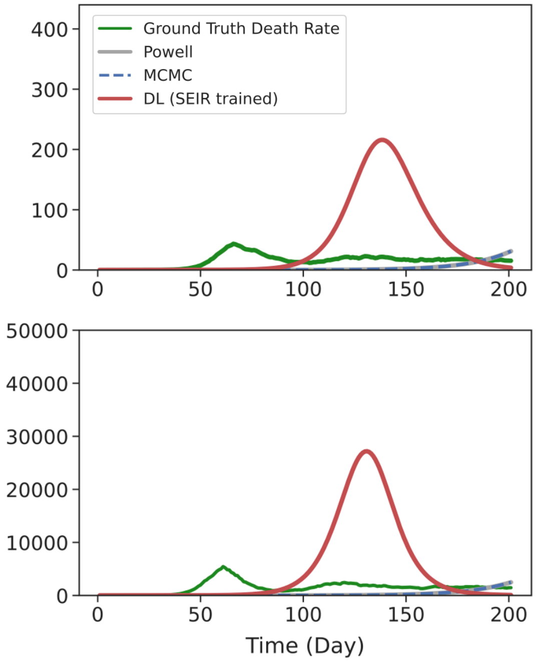


Figure 30: Examining the impact of model accuracy by calibrating the SEIR model on data generated from SEIRb at a low-level noise ($\sigma = 0.1$), Sensitivity to Death 0.1 and time to Perceive of 10, indicating all calibrators perform poorly when the model structure is wrong.

To that end, we switch the data generation model used to train the DL calibrator from SEIRb to SEIR, in order to investigate the importance of training the DL with a proper SD model. The test data is still being generated by SEIRb. Figure 30 compares the performance of the SEIR-trained DL calibrator with Powell and MCMC. Our results indicate that the performance of all three models depends on the accuracy of the model structure. An inaccurate model structure particularly hinders the DL calibrator’s ability to replicate system dynamics, emphasizing the importance of a well-defined model structure. Therefore, while DL can enhance calibration efficiency when a well-defined model structure is available, it cannot replace SD modeling, as it relies directly on the accuracy of the model structure.

7. Discussions and Conclusion

In this paper, we offered a novel DL-based approach for solving the inverse problem, also known as parameter calibration, in the context of dynamical systems. We tested the method for calibrating a behavioral epidemic model. Specifically, we demonstrated our proposed calibration approach through a proof-of-concept example on the SEIRb SD model.

Our results show that the proposed DL calibrator is capable of training a surrogate model for the inverse problem, allowing us to run parameter calibration with high accuracy while significantly reducing computation time compared to existing iterative optimization methods. Additionally, our proposed approach exhibited better performance in dealing with noisy or imperfect data. These findings suggest that DL techniques can be a powerful tool for calibrating SD models.

This work offers methodological contributions to the SD modeling literature. We build upon the previous works in SD validation testing and calibration tasks, including the approaches by Mert Edali [4, 5] and Duggan [8, 34] that utilized machine learning techniques. Our study extends their approaches by incorporating the state-of-the-art deep learning model with the aim of capturing the nonlinear and complex patterns present in noisy time-series data. Our study highlights the potential of using DL techniques for solving inverse problems in the context of dynamical systems, and we hope that our proposed approach can inspire further research and serve as a valuable tool for SD researchers and practitioners. Furthermore, by using the SEIRb model as a test case, this study adds to a growing body of health system dynamics modeling [46], and particularly studies that apply the SEIRb structure to epidemiology in different contexts [47, 48, 49], or investigate model validation challenges of behavioral epidemic models [50].

The DL-based approach proposed in this paper can be transformed into a software package integrated into existing SD software, such as Vensim. The package can be designed to be model generic, enabling modelers to calibrate any model structure with observable variables and parameters of interest efficiently. This would enable SD modelers to efficiently calibrate their model through a built-in function without having to rely on iterative optimization methods or manual parameter tuning.

This study has several limitations, which demand future work:

- Our study was focused on one single model from epidemiology. We invite the method to be applied to other SD models and its performance is evaluated in those cases.
- Leveraging active learning instead of the Latin hypercube sampling can enhance the implementation efficiency of the proposed method by reducing the data requirement and implementation time.
- The practical implementation of the proposed technique depends on efficient transfer learning from the pre-trained DL calibrator on similar SD model structures.
- Incorporating techniques presented by [51] to obtain confidence intervals from the estimated parameters using the proposed DL method would significantly enhance the applicability of the approach and enable decision-makers to quantify the uncertainty in the model's predictions.
- Using saliency techniques like attribution maps [52] enhances interpretability and accountability of the estimated parameters by providing valuable insights into the model's decision-making process.
- The proposed method can be used complementary to the conventional methods by providing closer initial values to optimal solutions, improving the efficiency and effectiveness of the optimization process.

Altogether, our study shows that there is a great potential to apply machine learning techniques to SD modeling in general and model calibration in particular. Especially given the need to use more sophisticated models with large volumes of data, ML techniques can help improve model accuracy and computational speed. We invite researchers to use the synergic opportunity and further enhance SD modeling practices.

References

[1] M. J. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *The computer journal*, vol. 7, no. 2, pp. 155–162, 1964.

[2] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[3] H. Rahmandad, T. Y. Lim, and J. Sterman, "Behavioral dynamics of covid-19: estimating underreporting, multiple waves, and adherence fatigue across 92 nations," *System Dynamics Review*, vol. 37, no. 1, pp. 5–31, 2021.

[4] M. Edali, "Pattern-oriented analysis of system dynamics models via random forests," *System Dynamics Review*, vol. 38, no. 2, pp. 135–166, 2022.

[5] M. Edali and G. Yücel, "Analysis of an individual-based influenza epidemic model using random forest metamodels and adaptive sequential sampling," *Systems Research and Behavioral Science*, vol. 37, no. 6, pp. 936–958, 2020.

[6] Y.-T. Chen, Y.-M. Tu, and B. Jeng, "A machine learning approach to policy optimization in system dynamics models," *Systems Research and Behavioral Science*, vol. 28, no. 4, pp. 369–390, 2011.

[7] I. El Naqa and M. J. Murphy, *What is machine learning?* Springer, 2015.

[8] J. Duggan, “Exploring the opportunity of using machine learning to support the system dynamics method: Comment on the paper by edali and yücel,” *Systems Research and Behavioral Science*, vol. 37, no. 6, pp. 959–963, 2020.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[10] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[12] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, “Deep learning for time series classification: a review,” *Data mining and knowledge discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *arXiv preprint arXiv:1706.03762*, 2017.

[14] P. M. Senge and J. W. Forrester, “Tests for building confidence in system dynamics models,” *System dynamics, TIMS studies in management sciences*, vol. 14, pp. 209–228, 1980.

[15] J. Sterman, “System dynamics: systems thinking and modeling for a complex world,” 2002.

[16] B. Richmond and S. Peterson, *An introduction to systems thinking.* High Performance Systems., Incorporated Lebanon, NH, 2001.

[17] Y. Barlas, “Formal aspects of model validity and validation in system dynamics,” *System Dynamics Review: The Journal of the System Dynamics Society*, vol. 12, no. 3, pp. 183–210, 1996.

[18] H. Rahmandad, R. Xu, and N. Ghaffarzadegan, “Enhancing long-term forecasting:

Learning from covid-19 models," *PLOS Computational Biology*, vol. 18, no. 5, p. e1010100, 2022.

[19] N. Ghaffarzadegan, A. Ebrahimvandi, and M. S. Jalali, "A dynamic model of post-traumatic stress disorder for military personnel and veterans," *PloS one*, vol. 11, no. 10, p. e0161405, 2016.

[20] A. Akhavan and P. Gonçalves, "Managing the trade-off between groundwater resources and large-scale agriculture: the case of pistachio production in iran," *System Dynamics Review*, vol. 37, no. 2-3, pp. 155–196, 2021.

[21] A. K. Wittenborn and N. Hosseinichimeh, "Exploring personalized psychotherapy for depression: A system dynamics approach," *Plos one*, vol. 17, no. 10, p. e0276441, 2022.

[22] N. Ghaffarzadegan and H. Rahmandad, "Simulation-based estimation of the early spread of covid-19 in iran: actual versus confirmed cases," *System Dynamics Review*, vol. 36, no. 1, pp. 101–129, 2020.

[23] T. Y. Lim, E. J. Stringfellow, C. A. Stafford, C. DiGennaro, J. B. Homer, W. Wakeland, S. L. Eggers, R. Kazemi, L. Glos, E. G. Ewing *et al.*, "Modeling the evolution of the us opioid crisis for national policy development," *Proceedings of the National Academy of Sciences*, vol. 119, no. 23, p. e2115714119, 2022.

[24] W. Wakeland and J. Homer, "Addressing parameter uncertainty in a health policy simulation model using monte carlo sensitivity methods," *Systems*, vol. 10, no. 6, p. 225, 2022.

[25] J. Andrade and J. Duggan, "A bayesian approach to calibrate system dynamics models using hamiltonian monte carlo," *System Dynamics Review*, vol. 37, no. 4, pp. 283–309, 2021.

[26] D. Duffie and K. J. Singleton, "Simulated moments estimation of markov models of asset prices," 1990.

[27] H. Rahmandad and N. S. Sabounchi, "Modeling and estimating individual and population obesity dynamics," in *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction.* Springer, 2012, pp. 306–313.

[28] M. Jalali, H. Rahmandad, and H. Ghoddusi, "Using the method of simulated moments for system identification," *Analytical methods for dynamic modelers*, 2015.

[29] C. Gourieroux, A. Monfort, and E. Renault, "Indirect inference," *Journal of applied econometrics*, vol. 8, no. S1, pp. S85–S118, 1993.

[30] N. Hosseinichimeh, H. Rahmandad, M. S. Jalali, and A. K. Wittenborn, "Estimating the parameters of system dynamics models using indirect inference," *System Dynamics Review*, vol. 32, no. 2, pp. 156–180, 2016.

[31] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[32] T. Li, H. Rahmandad, and J. Sterman, "Improving parameter estimation of epidemic models: Likelihood functions and kalman filtering," *Available at SSRN 4165188*, 2022.

[33] J.-W. Jeon, O. Duru, Z. H. Munim, and N. Saeed, "System dynamics in the predictive analytics of container freight rates," *Transportation Science*, vol. 55, no. 4, pp. 946–967, 2021.

[34] J. Duggan, "Using r libraries to facilitate sensitivity analysis and to calibrate system dynamics models," *System Dynamics Review*, vol. 35, no. 3, pp. 255–282, 2019.

[35] B. C. Csáji *et al.*, "Approximation with artificial neural networks," *Faculty of Sciences, Etvs Lornd University, Hungary*, vol. 24, no. 48, p. 7, 2001.

[36] T. Samad and A. Mathur, "Parameter estimation for process control with neural networks," *International Journal of Approximate Reasoning*, vol. 7, no. 3-4, pp. 149–164,

1992.

[37] H. L. Tessmer, K. Ito, and R. Omori, “Can machines learn respiratory virus epidemiology?: A comparative study of likelihood-free methods for the estimation of epidemiological dynamics,” *Frontiers in microbiology*, vol. 9, p. 343, 2018.

[38] N. Wu, B. Green, X. Ben, and S. O’Banion, “Deep transformer models for time series forecasting: The influenza prevalence case,” *arXiv preprint arXiv:2001.08317*, 2020.

[39] T. Li, H. Rahmandad, and J. Sterman, “Improving parameter estimation of epidemic models: Likelihood functions and kalman filtering,” 2020.

[40] H. Rahmandad, R. Xu, and N. Ghaffarzadegan, “A missing behavioural feedback in covid-19 models is the key to several puzzles,” *BMJ global health*, vol. 7, no. 10, p. e010463, 2022.

[41] C. T. Bauch and A. P. Galvani, “Social factors in epidemiology,” *Science*, vol. 342, no. 6154, pp. 47–49, 2013.

[42] S. Funk, M. Salathé, and V. A. Jansen, “Modelling the influence of human behaviour on the spread of infectious diseases: a review,” *Journal of the Royal Society Interface*, vol. 7, no. 50, pp. 1247–1256, 2010.

[43] M. McKay, R. Beckman, and W. Conover, “Comparison the three methods for selecting values of input variable in the analysis of output from a computer code,” *Technometrics;(United States)*, vol. 21, no. 2, 1979.

[44] J. Houghton and M. Siegel, “Advanced data analytics for system dynamics models using pysd,” *revolution*, vol. 3, no. 4, 2015.

[45] S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. Brubaker, “Time2vec: Learning a vector representation of time,” *arXiv preprint arXiv:1907.05321*, 2019.

[46] N. Darabi and N. Hosseinichimeh, "System dynamics modeling in health and medicine: a systematic literature review," *System Dynamics Review*, vol. 36, no. 1, pp. 29–73, 2020.

[47] H. Rahmandad and J. Sterman, "Quantifying the covid-19 endgame: Is a new normal within reach?" *System Dynamics Review*, vol. 38, no. 4, pp. 329–353, 2022.

[48] H. Rahmandad, "Behavioral responses to risk promote vaccinating high-contact individuals first," *System Dynamics Review*, vol. 38, no. 3, pp. 246–263, 2022.

[49] N. Ghaffarzadegan, "Simulation-based what-if analysis for controlling the spread of covid-19 in universities," *PloS one*, vol. 16, no. 2, p. e0246323, 2021.

[50] A. Osi and N. Ghaffarzadegan, "Data-informed parameter estimation in behavioral epidemic models," 2023, working paper. Department of Industrial and Systems Engineering at Virginia Tech.

[51] T. Pearce, A. Brintrup, M. Zaki, and A. Neely, "High-quality prediction intervals for deep learning: A distribution-free, ensembled approach," in *International conference on machine learning*. PMLR, 2018, pp. 4075–4084.

[52] D. Mercier, J. Bhatt, A. Dengel, and S. Ahmed, "Time to focus: A comprehensive benchmark using time series attribution methods," *arXiv preprint:2202.03759*, 2022.

Chapter 5: Conclusion

Deep learning has made remarkable progress in computer vision and natural language processing, but the interpretability limitations of black-box models have been a major bottleneck to their practical application. Attention-based models have revolutionized the field of deep learning by enabling the model to selectively focus on relevant parts of long input sequences, understand the context, and achieve human-like focusing behavior and performance in a wide range of applications of natural language processing, speech recognition, and computer vision. Additionally, they offer interpretability and transparency as attention weights reveal which parts of the input the model is focusing on, especially important in applications where the decisions made by the model have significant consequences, such as in healthcare.

This dissertation proposes the use of task-specific designed attention modules in three different applications: manufacturing, healthcare, and system identification, which involve video, medical image, and time series data, respectively. The attention modules are designed to focus on relevant data features specific to each application, leading to improved accuracy, transparency, and efficiency.

In Essay 1, we introduced a novel computer vision tool that tracks the melt pool in X-ray images of laser powder bed fusion (LPBF) additive manufacturing using attention modules. The proposed model used a semi-supervised video object segmentation (VOS) approach with spatiotemporal attention modules. This approach enabled automatic segmentation of the melt pool boundary, with manual annotation only necessary in the first frame, thus improving efficiency. Our proposed approach provided accurate and robust segmentation results and demonstrated excellent generalization performance. This work makes a significant contribution to the field of additive manufacturing by demonstrating the potential of using VOS for melt pool segmentation and tracking. We suggest future investigations on the deep learning model architecture including the encoder backbone choice and incorporation of plug-in modules (such as Atrous Spatial Pyramid Pooling (ASPP) and feature pyramid network (FPN)) for better performance, resolution, and speed. Furthermore, the model can be expanded to simultaneously segment the melt pool in infrared and X-ray images to correlate melt pool shape with thermal characteristics in real time.

In Essay 2, we addressed the urgent need for efficient computer-aided medical diagnosis and the consequent depletion of hospital resources amidst the global COVID-19 outbreak. We developed an AI-powered COVID-19 detection model to facilitate early diagnosis, aiming to reduce infectivity and mortality rates. Our multi-task learning approach, the mask-guided attention (MGA) classifier, uses lung CT scan images for COVID-19 diagnosis. The novelty of our proposed method is the use of lesion masks to compensate for the scarcity of data. This increases the model's ability to identify COVID-19 characteristics, and ultimately boosts both data efficiency and classification performance compared to single-task (ResNet18) and state-of-the-art models (ResNet50, MobileNetV2, VGG16, and DenseNet121). The MGA approach offers better interpretability and data efficiency, evident in attention and attribution maps, especially when training data is limited. We also contributed a large, diverse, and representative COVID-19 CT slice classification dataset, curated from seven open-source datasets, as a benchmark for future studies. Potential future research directions include incorporating lung segmentation masks in the MGA module for better lung analysis, investigating the impact of more precise disease classes on COVID-19 detection, extending the methodology to analyze CT scan volumes instead of slices, and integrating clinical and paraclinical examination results.

In Essay 3, we presented a novel deep learning (DL)-based approach to address the inverse problem, or parameter calibration, in dynamical systems. Our focus was on calibrating a behavioral epidemic model. We proposed a Transformer-based calibrator that employs self-attention mechanisms, demonstrating its ability to learn a surrogate model for the inverse problem. This approach resulted in competitive accuracy and significantly reduced computation time compared to conventional iterative optimization methods, such as Powell and Markov Chain Monte Carlo. Our experimental results demonstrated that our proposed calibrator is particularly promising when large-scale complex models are used, data are noisy, and model calibration is needed to be performed frequently and quickly. By building upon previous works in system dynamics (SD) modeling and extending their machine learning-based approaches with cutting-edge DL techniques, our study highlighted the potential for employing DL to tackle dynamic systems parameter calibration. We further suggest transforming our DL-based approach into a model-generic software package that can be integrated into existing SD software, allowing modelers to calibrate various model structures more efficiently. Despite our study's limitations, including its focus on a single epidemiological model and the need for further exploration of practical implementation aspects, it represents a promising direction for incorporating deep learning into SD modeling. By doing so, we can provide an efficient model structure validation tool that helps researchers and practitioners calibrate their models efficiently using a software built-in function.

In summary, this dissertation proposed the use of attention-based deep learning models to tackle significant challenges in various domains, such as manufacturing, healthcare, and system identification. By leveraging attention modules tailored to specific tasks, the proposed solutions showed improved accuracy, efficiency, and interpretability, contributing to the advancement of the respective fields. The results obtained demonstrated the potential of attention-based models in addressing real-world problems and highlighted future research directions to further improve their scalability and applicability. The attention-based models possess interpretability advantages and superior performance, which could drive progress in solving pressing problems across different fields.