

ORIGINAL RESEARCH ARTICLE

AMTransformer: A Koopman theory-based
transformer for learning additive manufacturing
dynamics in laser processesSuk Ki Lee and Hyunwoong Ko*

School of Manufacturing Systems and Networks, Ira A. Fulton Schools of Engineering, Arizona State University, Mesa, Arizona, United States of America

Abstract

Recent advancements in machine learning (ML) have shown unprecedented promise in understanding and predicting additive manufacturing (AM) dynamics. However, existing ML studies on AM often lack a comprehensive approach to address the multi-scale complexities inherent in AM processes and tend to employ context-specific methods. To address these limitations, we present a foundational method for formulating AM dynamics suitable for ML modeling. We then introduce a novel approach, the AMTransformer, designed to comprehend complex spatiotemporal dynamical dependencies among physical entities and their properties within the AM process. To enhance the understanding of AM dynamics, our method adapts Koopman's theory to generate latent embeddings of AM states and their transitions, effectively extracting hidden features related to physical properties and dynamical dependencies. In addition, by utilizing the transformer's attention mechanism, the proposed approach enhances the learning of non-local, non-linear dynamical dependencies across multiple scales. Our experiments, conducted using melt pool data from a laser powder bed fusion process, demonstrate that the AMTransformer outperforms traditional transformer and convolutional long short-term memory models. Specifically, the AMTransformer achieved structural similarity, mean absolute error, and accuracy metric values of 0.9206, 0.0009 mm², and 92.73%, respectively. These results indicate the AMTransformer's superior ability to predict future AM states, attributed to its improved learning of complex AM dynamics. By combining linear Koopman-based methods with non-linear transformer-based approaches, the AMTransformer significantly improves data-driven modeling for AM, providing a more comprehensive understanding of AM dynamics. Furthermore, the generalizability of the proposed method facilitates the expansion of the model's scope and enhances its applicability across various fields.

***Corresponding author:**Hyunwoong Ko
(hyunwoong.ko@asu.edu)

Citation: Lee SK, Ko H. AMTransformer: A Koopman theory-based transformer for learning additive manufacturing dynamics in laser processes. *Int J AI Mater Design*. 2024;1(2):76-91. doi: 10.36922/ijamd.3919

Received: June 12, 2024**Accepted:** August 9, 2024**Published Online:** September 2, 2024**Copyright:** © 2024 Author(s).

This is an Open-Access article distributed under the terms of the Creative Commons Attribution License, permitting distribution, and reproduction in any medium, provided the original work is properly cited.

Publisher's Note: AccScience Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Keywords: Additive manufacturing; Koopman theory; Laser AM dynamics; Machine learning; Predictive modeling; Transformer

1. Introduction

Additive manufacturing (AM) is a revolutionary technique in modern manufacturing that enables the production of products with enhanced performance characteristics, often unattainable with conventional methods. Through its layer-by-layer manufacturing

approach, AM offers unparalleled capabilities for creating designs with advanced geometrical, hierarchical, material, and functional complexities. These capabilities have significantly advanced innovation in both the design and functionality of final products.¹⁻⁴

Among the various AM processes, laser-based methods, such as laser powder bed fusion (LPBF) and direct energy deposition (DED), stand out. These methods use a laser beam as an energy source to fuse powder-like raw materials and build final products.^{5,6} In these processes, melt pools are generated when the laser interacts with the raw materials. Melt pools are the areas where the laser melts the powder materials, dynamically changing their states as the laser beam moves along the designated path.^{7,8} The melt pools undergo continuous dynamic changes due to spatial and temporal dependencies between the physical entities involved in the AM processes. For example, the high heat transfer within the melt pool results in spatial variations in temperature distribution. Furthermore, the thermal behavior of the melt pools changes over time due to the accumulation of layers and the presence of adjacent, previously formed melt pools along the path.⁹ The physical dynamics of the melt pools during the process significantly impact the quality and properties of the manufactured products.^{10,11}

Machine learning (ML) can significantly enhance our understanding of the complex dynamics involved in laser-based AM processes. By leveraging novel *in situ* data generated during LPBF and DED operations (e.g., thermography,^{13,14} visual imaging,^{9,14-16} acoustic signals,^{17,18} and spectral data^{19,20}), ML algorithms can identify intricate patterns and dependencies that traditional analytical or simulation methods might overlook.^{21,22} For example, ML models can predict the spatial and temporal dependencies within and between melt pools, accounting for factors like high heat transfer and the cumulative effects of successive layers.^{9,10} Enhancing the understanding of AM dynamics through ML-driven real-time monitoring and predictive analytics allows for adaptive control of the AM process, thereby improving consistency and reducing defects. This data-driven approach not only enhances process efficiency but also contributes to the development of more robust and reliable AM systems capable of producing high-quality components with complex geometries and material processing. Advances in process monitoring technologies have further enabled the collection of novel *in situ* data from AM processes, underscoring the importance of ML in comprehending and anticipating AM dynamics.²²

However, existing methods using ML for analyzing complex AM dynamics often lack generalizability. They rely on specific AM cases or data types, rendering them

inflexible and unable to adapt to different datasets or AM methods. In addition, AM processes involve a complex hierarchy of interconnected points, lines, and layers. Therefore, it is crucial to develop an ML method that provides a comprehensive understanding of this entire hierarchy, something that existing methods are insufficient to achieve. Consequently, there is an urgent need for an alternative approach capable of encompassing the multi-scale spatial and temporal modeling required for a complete understanding of AM dynamics.

To address this challenge, we first formulated the key dependencies in AM dynamics in this study. Building on this formulation, we proposed a novel method called the AMTransformer, designed to improve the understanding of melt pool dynamics and predict its behaviors. The AMTransformer leverages Koopman theory and a transformer architecture to capture spatiotemporal dependencies within and between melt pools at multiple scales, thereby providing a deeper understanding of the underlying dynamics that govern AM processes.

The remainder of this paper is organized as follows: In Section 2, we review relevant literature on melt pool modeling, a representative area of AM dynamics modeling. Section 3 establishes the problem statement, while Section 4 introduces the proposed method. In Section 5, we present a case study. Finally, Section 6 includes a discussion, and Section 7 offers concluding remarks and outlines future work.

2. Literature review

Deep learning methods have proven effective in identifying and extracting significant spatial features from data, thereby enabling precise predictions or classifications of melt pool characteristics. Yang *et al.*,²³ used a convolutional neural network (CNN) architecture, coupled with fully connected layers, to classify melt pool types using real-time melt pool monitoring (MPM) images. Similarly, Zhang *et al.*²⁴ proposed a method that utilizes a hybrid of two CNNs to monitor and predict melt pools. Fathizadan *et al.*¹⁴ adopted a convolutional autoencoder model to derive a comprehensive representation of melt pool areas, which they used to distinguish anomalies through a clustering algorithm.

In addition, studies have explored the application of time-series neural networks to learn temporal dependencies involved in melt pool dynamics. Larsen and Hooper²⁵ developed a method for predicting melt pool behavior by integrating a variational autoencoder with a recurrent neural network (RNN). In their approach, MPM images are first embedded into a latent space using a variational autoencoder. This latent representation then serves as the input for a dynamics model, which employs an RNN to predict the distribution of the next state based on previous

input data. Ko *et al.*⁹ utilized convolutional long short-term memory (LSTM) to classify melt-pool anomalies, taking into account the spatiotemporal dependencies between melt pools. Zhang *et al.*,¹⁶ also leveraged LSTM networks to grasp the dependencies among input data characteristics, including process parameters and *in situ* MPM images. Their model uses the LSTM networks to predict melt pool size and subsequently utilizes a conditional generative adversarial network to generate predicted melt pool images.

Recently, transformer methods have begun to emerge in the field of AM. Fernandez-Zelaia *et al.*²⁶ proposed a video transformer framework that efficiently captures both spatial and temporal patterns within a compact latent space representation. This latent representation holds important physical information and is used to create a data-driven process-structure model. The study utilized thermal simulation data from a reduced-order model to predict changes in spatial and temporal responses during manufacturing processes. Guirguis *et al.*,²⁷ used *in situ* monitoring images that contain the dynamic changes in the melt pools. They then leveraged spatiotemporal data to classify the process into different defect types and conditions using video vision transformers. These studies primarily focus on classifying and extracting features of melt pools.

Transformer models have been proposed in these studies to overcome the limitations of existing time-series neural networks. The pivotal concept of these models is the attention mechanism, providing a new paradigm for understanding and processing sequences in ML models.²⁸ The attention mechanism allows an ML model to simultaneously attend to all positions in the input data globally. This interaction enables the model to assess the importance of all elements in a sequence when processing a particular element. As a result, even elements that are far apart in the sequence can effectively influence each other. In practical terms, the attention mechanism allows the model to effectively handle dependencies between elements, regardless of their distance from each other. This approach contrasts with RNNs and LSTMs, which process data sequentially, requiring each time step to be analyzed only after all preceding time steps have been processed. In the context of the AM dynamic systems, the ability of transformer models to examine interdependencies among sequential melt pools is fundamental for a comprehensive understanding of the dynamics governing melt pool phenomena.

3. Problem statement

Although existing ML studies in AM dynamics have shown promising performance, learning the fundamental dynamics of AM processes remains a significant challenge. The AM

processes are characterized by their complex nature, with interrelated points, lines, and layers of material evolution during the processes at multiple scales. To accurately capture AM dynamics, a comprehensive approach is required – one that enhances the understanding of this multi-scale complexity. A suitable modeling approach should not only extract the hidden features of various AM states but also account for the underlying dependencies at multiple spatiotemporal scales. Understanding these multi-scale dependencies is crucial, as they significantly impact AM processes such as melt pool behaviors and, consequently, the overall quality of the final product.

Furthermore, it is important to note that previous studies employing ML for AM dynamics have often delineated their models within specific contexts, rendering them ad hoc and lacking adaptability to alterations in data or the varieties of physical phenomena. Given the diversity of AM types, each with distinctive and different kinds of data depending on its data acquisition systems, it is crucial to consider a formal modeling approach when analyzing the dynamics of AM.

4. Methods

This section introduces a novel ML method for learning AM dynamics: the AMTransformer. We begin by presenting the dependencies between the physical entities and their properties involved in AM dynamics. To achieve this, we formulate the dynamical dependencies involved in AM phenomena using the concepts of state and rate properties. Following this, we detail the AMTransformer, which enhances the ability to learn the dynamics of AM processes and accurately predict their behaviors.

4.1. Dynamical dependency formulations

In modeling with the AMTransformer, we adapt the concept of dynamical dependency to represent the spatiotemporal mechanisms of physical entities and properties involved in AM dynamics.¹⁰ This adaptation provides a foundational representation of AM dynamics for ML modeling. Dynamical dependencies illustrate the mechanisms of AM and their impacts on changes in AM processes and parts. The properties of physical entities and the dependencies among them encapsulated within these dynamical dependencies, represent the AM dynamics. The dynamical dependencies in AM dynamics are described as follows:

- A physical entity, such as powdered metals, engages in an AM process, like the scanning of a laser. [Figure 1](#) illustrates various physical entities in an AM process.
- Each physical entity has a physical property, which is divided into two sub-classes: state property and rate property. A state property quantitatively characterizes the amount or momentum of a physical entity involved

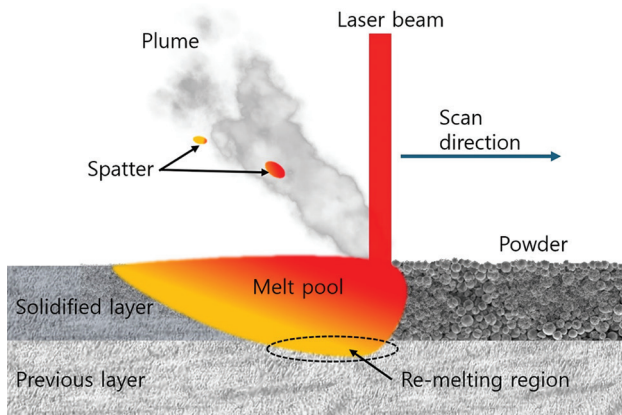


Figure 1. A schematic illustration of physical entities in melt pool generation

in an AM process, while a rate property quantitatively characterizes a flow rate or a force applied to a physical entity during an AM process.

- A dynamical dependency refers to the relationship between the state and rate physical properties that interact with each other. For example, when a laser is applied and moves, the size of a melt pool changes. In this AM process, the melt pool size is a state property influenced by rate properties like the speed of the melt pool or the laser energy per unit length, in addition to the state properties of other physical entities involved in the process, such as the areas or temperatures of adjacent metal powders melting and solidifying.
- At any given moment, an AM process exists in a specific physical state, which we define as an AM state. The magnitude of state and rate properties, such as the shape, size, and location of the melt pool, as well as laser energy per unit and speed, determines the AM states. These state and rate properties, along with their dynamical dependencies, influence the magnitude of AM state changes, state transitions, and the dynamic behaviors of AM processes over time.

Laser AM involves various processes characterized by consecutive physical phenomena that occur in a layer-by-layer manner. Particularly, when the laser interacts with the material, dynamic changes in AM states occur as the laser moves along its designated scanning path, and the material undergoes continuous melting and solidifying. To analyze these phenomena, we discretize the changes in AM states into T -time steps. We define AM states as a function of a set of state properties and rate properties. An AM state can be expressed as Equation I:

$$\varphi_t = f(x_p, c_t), \varphi_t \in S \subset \mathbb{R}^m, t \in T \subset \mathbb{R}^+ \quad (\text{I})$$

where $x_t \in X = \{x_1, x_2, \dots, x_{n-1}, x_n\}$ represents state properties, $c_t \in C = \{c_1, c_2, \dots, c_{n-1}, c_n\}$ represents rate properties, and

$\varphi_t \in \mathbb{R}^d$ indicates an AM state variable in the time domain T and spatial domain S . Figure 2 provides a schematic representation of AM states.

Dynamical dependencies in AM can be categorized into three types: (i) dependencies between state properties, (ii) dependencies between rate properties, and (iii) dependencies between state and rate properties. For instance, the volume of a melt pool depends on the volume of an adjacent melt pool track (a dependency between state properties); the laser's energy per unit property depends on the laser speed (a dependency between rate properties); and the melt pool's volume is influenced by the melt pool's flow rate (a dependency between a state property and a rate property). Together, these three types of dynamical dependencies illustrate how physical entities transfer or manage energy flow and evolve during AM processes. These dependencies trigger transitions between AM states, represented by the transition function in Equation II:

$$g: \varphi_t \rightarrow \varphi_{t+1} \quad (\text{II})$$

Based on Equation II, Equation III captures the concatenation of AM states, representing all the successive AM state changes from the initial state of a build in the first layer (φ_0) to the final state in the last layer (φ_M):

$$\Phi = \prod_{t=0}^M \varphi_t, 0 \leq M < \infty \quad (\text{III})$$

where $\Phi = \varphi_0 \cdot \varphi_1 \cdots \varphi_M$ such that $\varphi_t = g(\varphi_{t-1}) = g(g(\varphi_{t-2}))$.

4.2. AMTransformer

The AMTransformer is designed to observe the physical characteristics of AM states and their transitions over time, capturing the dynamic dependencies between them. In addition, based on this understanding, the AMTransformer is capable of predicting future AM states. This capability is achieved through two fundamental components: (i) an AM state embedder and (ii) a transformer. Figure 3 presents the overall architecture of the AMTransformer.

4.2.1. AM state embedder

The primary objectives of the AM state embedder are to (i) learn the dynamical dependencies within an AM state and during its state transition to the next state and (ii) encapsulate this understanding into data representations by projecting these dependencies and state transitions into latent-space embedding vectors.

The AM state embedder is an autoencoder designed for non-linear AM processes. It is an embedding network consisting of three main elements: an encoder, a Koopman operator, and a decoder. The encoder component of the AM state embedder, which consists of multi-layer neural networks, plays a crucial role in projecting the observation

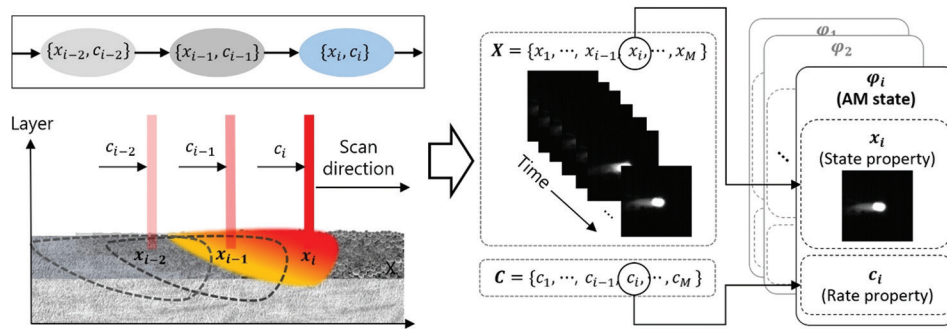


Figure 2. A schematic representation of additive manufacturing states and their transitions with melt pool area images as examples of melt pool's state properties

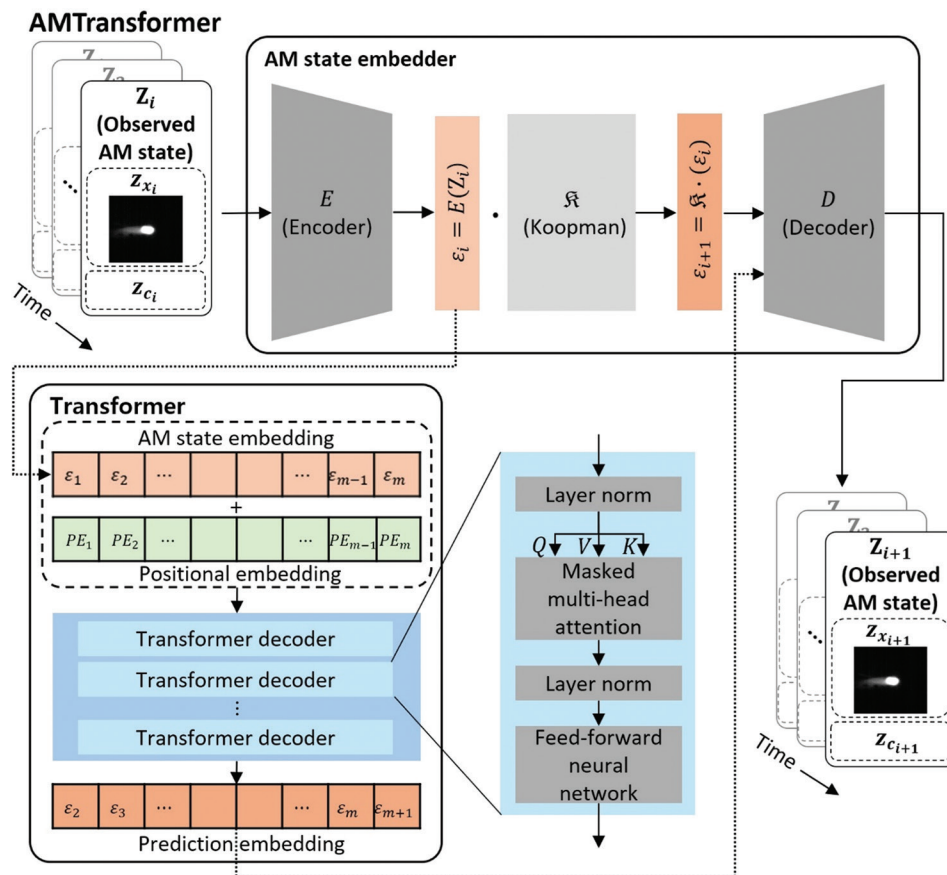


Figure 3. An overall architecture of the AMTransformer in an additive manufacturing process

data Z_i of current state φ_i into the embedding vector ε_i within the observable space, denoted as $E: \mathbb{R}^d \rightarrow \mathbb{R}^e$, as shown in Equation IV. Our model uses observed data to model rate and state properties along with their dynamical dependencies. As an input of the encoder, Z_i includes the observed state and rate property data of AM states. The output of the encoder is an AM embedding, which captures the data features representing the dynamical dependencies

(as per function f), and represents the AM state at time i (φ_i) as described in Equation I.

$$\varepsilon_i = E(Z_i) \quad (\text{IV})$$

By applying the embedding process to all paired sets of state and rate property data, the AM state encoder equips the Koopman operator (\mathcal{K}) with a set of embedding vectors. These vectors represent the observed data and capture the

dynamical dependencies between state and rate properties within a state across all state transitions in AM. The AM state embedder then applies the Koopman operator repeatedly, capturing the dynamical dependencies in AM state transitions within the observable space through the evolution of the AM embedding vectors, as shown in Equation V:

$$\varepsilon_{i+1} = \mathfrak{K}\varepsilon_i, \varepsilon_{i+2} = \mathfrak{K}^2\varepsilon_i, \varepsilon_{i+3} = \mathfrak{K}^3\varepsilon_i, \dots \quad (\text{V})$$

Using the AM state embedding as inputs, the Koopman operator projects the AM state transitions onto the embedding state transition in an infinite-dimensional observable space as a linear operator, as shown in Figure 4. The Koopman operator offers a method to examine non-linear dynamical systems by transforming them into a linear framework with infinite dimensions.²⁹ The relationship can be defined in Equation VI:

$$g \circ f \equiv \mathfrak{K} \circ E \quad (\text{VI})$$

The Koopman operator is based on the Koopman theory and employs a dynamic mode decomposition (DMD) approach. DMD is a method for representing the Koopman operator using finite-dimensional approximations based on available data.³⁰⁻³⁵ In the AM state embedder, DMD identifies essential measurement functions that control the dynamics of an AM system, along with a corresponding finite approximation of the Koopman operator. Direct application of the Koopman operator in its infinite-dimensional form is impractical, so DMD provides a feasible alternative. DMD can identify coherent structures in high-dimensional data by analyzing multiple snapshots over time, aiding in the prediction of future states. From the collected two snapshot matrices, which are sets of sequential embedding vectors, denoted as $S_a = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_t]$ and $S_b = [\varepsilon_2, \varepsilon_3, \dots, \varepsilon_{t+1}]$ with one state-transition step difference, DMD determines the dominant spectral decomposition, that is, eigenvectors and eigenvalues, of the best-fit linear operator linking

the two matrices. This linear operator is adopted as \mathfrak{K} , as shown in Equation VII:

$$S_b \approx \mathfrak{K}S_a, \mathfrak{K} = \underset{\mathfrak{K}}{\operatorname{argmin}} \|S_b - \mathfrak{K}S_a\| \quad (\text{VII})$$

The decoder part, consisting of a multi-layer neural network symmetric to the encoder part, reconstitutes the embedding vectors from the Koopman operator back into the original physical space, $D: \mathbb{R}^e \rightarrow \mathbb{R}^d$, during training. The data representing the AM state obtained from the decoder can be expressed as Equation VIII:

$$Z_{i+1} = D(\varepsilon_{i+1}) \quad (\text{VIII})$$

The AM state embedder is trained using the loss function shown in Equation IX. This loss function includes two crucial components. The first component guarantees a uniform mapping to and from the embedded representation, capturing the dependencies between the state and rate properties at an AM state while minimizing reconstruction loss. The second component, the Koopman dynamics loss, encourages embeddings to adhere to linear dynamics, minimizing errors in capturing the dynamical dependencies during each AM state transition.

$$L_{AMSE} = \sum_{i=1}^N \sum_{j=1}^T \lambda_0 \underbrace{l_{\text{recon}}(Z_j^i, D \circ E(Z_j^i))}_{\text{Dynamical dependencies at an AM state}} + \lambda_1 \underbrace{l_{KD}(Z_j^i, D \circ \mathfrak{K}^{j-1}E(Z_j^i))}_{\text{Dynamical dependencies at AM state transitions}} \quad (\text{IX})$$

In this equation, L_{AMSE} is a total loss function of the AM state embedder, Z_j^i represents the j^{th} observed data point corresponding to an AM state in the i^{th} observation data sequence, N is the number of observation data sequences, T is the length of an observation sequence (i.e., the number of AM state embeddings constituting a data sequence), l_{recon} and l_{KD} represent the reconstruction loss and the Koopman dynamics loss, respectively, and λ_0 and λ_1 are coefficients.

Figure 5 illustrates the architecture of the AM state embedder. During training, the embedder constantly updates each part by identifying the evolutionary characteristics of the AM states. The architecture of the AM state embedder, grounded in the Koopman theory, is designed to comprehend the intrinsic dynamics of AM processes. This approach guarantees that the resulting trained embeddings accurately encapsulate the AM states, consisting of their state and rate properties and transitions, thereby facilitating the capture of their evolutionary trajectory and inherent dynamical dependencies.

4.2.2. Transformer

The transformer part enhances the elucidation of spatial and temporal dynamical dependencies, primarily based

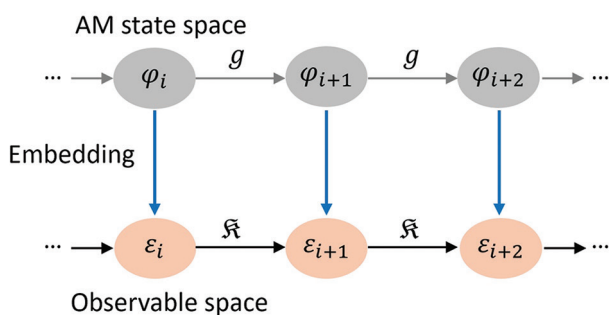


Figure 4. A projection of additive manufacturing (AM) states and their transitions into AM state embeddings and their transitions within an observable space

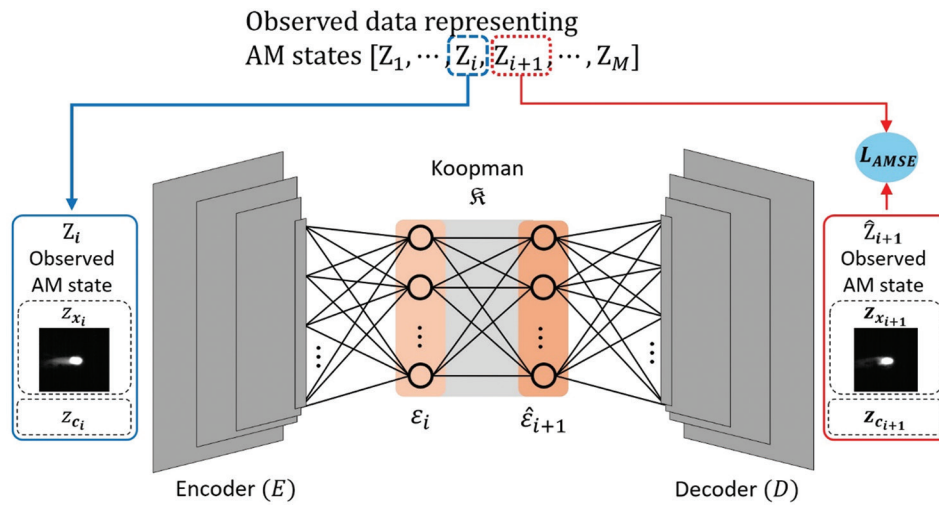


Figure 5. An illustration of the additive manufacturing state embedder architecture

on the attention mechanisms. In addition to capturing local dynamical dependencies at a single point in time or during a single state transition, the proposed transformer also identifies non-local dynamical dependencies across concatenations of AM state embedding vectors. These concatenated vectors, generated by the trained AM state embedder through its iterative embedding process, represent multiple successive state transitions at various spatial and temporal scales.

To model non-local dynamics and enhance their representation as dynamical dependencies across multiple AM state transitions at various scales, the transformer is equipped with positional embeddings. These embeddings incorporate the relative positional information of the concatenations of the AM state embeddings into the AMTransformer. The positional embeddings are defined by Equation X:^{28,32}

$$PE_{p,2j} = \sin\left(\frac{p}{10000^{2j/e}}\right), PE_{p,2j+1} = \cos\left(\frac{p}{10000^{2j/e}}\right) \quad (X)$$

where p represents the relative position of the AM embedding vector in the input concatenations, and $2j$ and $2j + 1$ indicate the positions in the embedded vector for the even and odd elements among the e elements of a vector. Each positional embedding vector contains information about the corresponding AM embedding, enhancing the AMTransformer's ability to identify and analyze the relationships between AM states.

By modeling non-local dynamical dependencies in a time series, the transformer enables the prediction of future AM states. For prediction purposes, the AMTransformer employs the transformer decoder architecture of the generative pre-trained transformer model,³⁶⁻³⁸ specifically

designed for sequential prediction. The transformer decoder takes the sum of the positional embedding and the AM state embedding as its input. This input passes through multiple layers of the transformer decoder, each containing an attention layer and a feed-forward neural network. This multi-layer architecture enables the model to learn intricate dynamical dependencies at various levels of abstraction. Each layer operates on the concatenations with its own set of learned weights, progressively improving the learning of dependencies as it goes deeper into the network. This depth is critical for effectively handling complex dependencies in AM.

The transformer uses self-attention based on scaled-dot product attention as its main mechanism for learning the dependencies. The core of this mechanism involves the concepts of keys, queries, and values, which enable the proposed model to selectively concentrate on specific positional and AM state embeddings within the input concatenations. A query corresponds to the current embedding that requires attention and is employed to identify which parts of the input are relevant. The keys are associated with all embeddings that the model should focus on, aiding in determining the extent to which each input component should contribute to the output at every step. Each input embedding is linked to specific values, representing the actual content used to construct the output. The model identifies relevant input embeddings by matching the query with the keys and then utilizes the corresponding values to generate the output. Calculations for each key, query, and value vector are performed using neural networks N_k , N_q , and N_v on the AM state embeddings, which are combined with positional embeddings. Then, the AMTransformer employs the softmax function to calculate attention using the sets of queries, keys, values,

and a transpose operator, denoted by Q , K , V , and T , respectively, as shown in Equation XI:^{28,31}

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (XI)$$

where $q_i = N_q(\varepsilon_i) \in Q$, $k_i = N_k(\varepsilon_i) \in K$, and $v_i = N_v(\varepsilon_i) \in V$. Each neural network N_q , N_k , and N_v contains weight matrices that perform linear transformations on the input AM state embedding ε_i . These transformations are applied as shown in Equation XII:²⁸

$$q_i = \varepsilon_i W_Q, k_i = \varepsilon_i W_K, v_i = \varepsilon_i W_V \quad (XII)$$

where ε_i is the input embedding, and W_Q , W_K , and W_V are the weight matrices for the queries, keys, and values, respectively. In the multi-head attention mechanism, the outputs of all attention heads are concatenated and linearly transformed to produce the final attention output, as shown in Equation XIII:²⁸

$$MultiHead(Q, K, V) = Concat(h_1, \dots, h_n)W^O \quad (XIII)$$

where W^O is the output weight matrix that projects the concatenated outputs back to the dimensions of input embeddings, h_j represents the output of the j^{th} attention head, and n is the number of attention heads.

Multi-head attention allows the AMTransformer to explore various complexities of AM processes captured by the AM state embeddings at different positions. Each attention head focuses on different parts of the embedding concatenations, capturing diverse dynamical dependencies. AM processes have a hierarchical structure, with each point, line, or layer contributing to the overall shape and properties of complex manufactured objects. The AMTransformer captures this complexity through multi-head self-attention mechanisms, allowing the model to simultaneously attend to different levels of abstraction. As the model employs multi-head masked attention layers, the dimension of the query and key for each layer is obtained by dividing the dimension of the AM embedding vector by the number of parallel attention layers, which is equivalent to the number of heads. The masking in the attention mechanism enables the decoder to selectively focus on past and present AM states during training. This masking mechanism ensures that the AMTransformer learns to predict the future AM states in the concatenations based solely on the relevant spatial and temporal dynamical dependencies from the AM states that have already occurred at that time, without any information from the future, as illustrated in Figure 6. In addition, masking enables the AMTransformer to handle multiple sequence positions in parallel, up to the number of heads.

The last part of the decoder is the feed-forward neural network. In each decoder layer of the architecture, a feed-

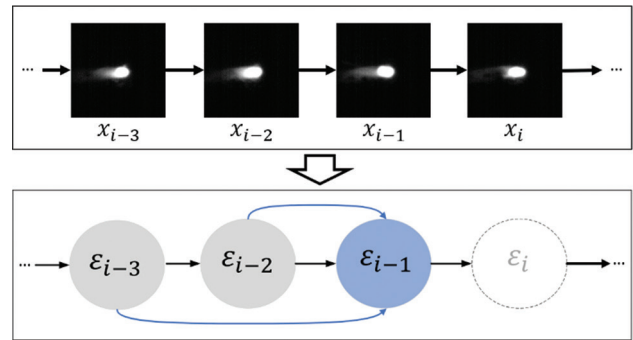


Figure 6. An example of a masked attention mechanism applied to melt pool states, showing how dynamical dependencies are captured. When the query is ε_{i-1} , the decoder masks future information, focusing only on the previous sequence. The black arrows signify transition, while the blue arrows represent dependencies captured through attention.

forward neural network is used to apply a non-linear transformation to every individual AM state in the input sequence. This part enhances the model's representational ability by introducing non-linear processing and facilitating interaction between different AM states in the input sequence. This non-linearity is crucial because it allows the AMTransformer to capture intricate patterns and relationships inherent in the data that cannot be represented by linear transformations alone. The output of the transformer consists of predicted embedding vectors, which the decoder of the AM state embedder reconstructs back into the original physical space.

The transformer undergoes training utilizing a loss function that enhances its accuracy and performance, as shown in Equation XIV:

$$L_T = \sum_{i=1}^N \sum_{j=1}^T l_T(\varepsilon_j^i, \hat{\varepsilon}_j^i) \quad (XIV)$$

where L_T stands for the total loss function of the transformer, ε_j^i represents the actual embedded vector for the j^{th} AM state in the i^{th} AM state sequence, $\hat{\varepsilon}_j^i$ represents the transformer's predicted embedding for $\hat{\varepsilon}_j^i$, N represents the number of the AM state sequences, T represents the length of an AM state sequence, which is the number of AM state embeddings constituting an AM state sequence, and l_T represents the loss incurred for each future AM state embedding prediction.

5. Case study

In this section, we present a case study that demonstrates the proposed AMTransformer using LPBF melt pool experimental data. The objective of this case study is to assess the effectiveness of the AMTransformer, with a specific focus on predicting future melt pools.

5.1. Data selection

To assess the learning capabilities of the AMTransformer, we utilized the AM metrology testbed (AMMT): overhang part X4 dataset, which was created to facilitate the development of data-driven predictive models, as referenced in Lane and Yeung.³⁹ This dataset was compiled through research conducted at the National Institute of Standards and Technology. The AMMT building process involves constructing four overhang structures measuring 5 mm × 9 mm × 5 mm. The process involves a laser with a specified power of 100 W and a scanning speed of 900 mm/s for the pre-contour phase. During the infill hatching phase, the laser power is increased to 195 W, and the scanning speed is adjusted to 800 mm/s. The component is fabricated by accumulating 250 layers, each with a thickness of 20 μm, with a 90° rotation applied between each layer.

The AMMT dataset encompasses a collection of *in situ* MPM images captured during the execution of an LPBF process, as well as a set of process control commands and measured data. The MPM images are captured using a co-axial MPM camera integrated into the AMMT. To obtain stationary monitoring images of the melt pool during the 3D build process, the laser beam and melt pool emission are optically aligned in the system, ensuring that the high-speed camera's field of view is fixed on the melt pool. The camera captures 120 × 120-pixel grayscale images of melt pools at a sampling rate of 100 μs per frame, with pixel values ranging from 0 to 255. The command data for the process contains information regarding the location and power required for controlling the laser.³⁹

In this study, we employed a total of 13,233 MPM images, specifically from the fourth part of the dataset. We selected 4812 images from Layer 1, 3609 images from Layer 150, and another 4812 images from Layer 210, along with the corresponding process control information. These layers were chosen because they exhibited a high incidence of anomalies, thereby providing a comprehensive dataset for our model analysis. To ensure the reliability of our results, we set aside 18% (2406 images) of the total dataset as the test dataset. This selection was done randomly across the different layers (Layer 1, Layer 150, and Layer 210) to ensure a representative sample. The test dataset was used to evaluate the model's performance and validate its accuracy. The remaining 82% (10,827 images) of the data was used for training.

5.2. Data pre-processing and structure

The AMTransformer used the pre-processed MPM images representing the melt pool, spatter, and plume areas, along with the melt pool's x and y locations as state properties of the material, laser power as state property of the laser,

and laser velocity and energy per unit length (mm) as rate properties of the laser to understand their dynamical dependencies and predict future melt pools. To extract features from the raw data, we performed three types of pre-processing on the raw MPM images: denoising, centering, and cropping, in addition to pre-processing the control data. These datasets encompass a variety of dynamic AM phenomena, including but not limited to melt pools, spatter, and plumes, as shown in Figure 7. The extracted features of these phenomena serve as critical indicators for understanding and categorizing the state and rate properties within the LPBF process.

The denoising process removed any noise from the raw MPM images, such as the noise highlighted in Figure 7, leaving only the features of interest: the main melt pool, spatter, and plume. After denoising, we centered and cropped the main melt pools. Centering reduced bias related to the locations of the melt pools in the images, while cropping decreased the size of the original MPM images from 120 × 120 pixels to 64 × 64 pixels for enhanced resource utilization. Figure 8 illustrates the pre-processing of the MPM images.

We also processed the control data to extract additional features such as laser velocity and energy per unit length. For laser velocity, we measured the times and the distances traveled by the laser. Distances were measured based on the x and y coordinates using the Pythagorean theorem. We then extracted the laser velocity feature by dividing the distance by the time, according to the formula for velocity in physics. Using the laser's velocity and power, we also extracted the feature for the energy per unit length – a measure of how much energy is applied per unit length – by dividing the power by the velocity.

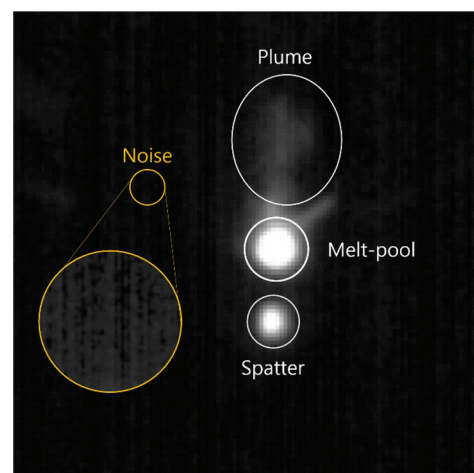


Figure 7. An example of a melt pool monitoring (MPM) image: the orange circle highlights a noise instance, displaying random greyscale variations in the MPM images outside the melt pool, spatter, and plume

Following the proposed method, the case study systemically matched the state and rate property features at each time step to generate concatenations of consecutive AM states. Figure 9 details this process. At time step i , an MPM image mainly focuses on the melt pool's top surface area (e.g., size and morphology), which can be expressed as a state property of the melt pool ($x_{i,mp}$). Melt pool location (x_{i,mp_loc}), and laser power ($x_{i,laser_p}$) also represent state

properties in the laser AM process. These state properties of the LPBF process are expressed as x_i^L , and the observation data ($z_{c_i}^L$) represents these properties. Simultaneously, rate property observation data ($z_{c_i}^L$) represents velocity ($c_{i,laser_v}$) and energy density ($c_{i,laser_d}$) obtained from process control data. The observed data on state and rate properties (Z_i) is then provided as input to the AMTransformer in sequential order.

5.3. Melt pool prediction of the AMTransformer

To handle the input data, the AM state embedder incorporated CNNs into its architecture. In this case study, the AM state embedder comprised four 2D convolutional layers in an encoder and four 2D transposed convolutional layers in a decoder. The AM state embedder is designed to combine multiple observed inputs, each representing an AM state (ϕ_i), to produce a single embedding vector (ε_i). This embedding vector serves as a latent representation of the AM state, encapsulating the state transition characteristics of the LPBF process. The Koopman operator within the AM state embedder then captures the local dynamic dependencies, as depicted in Figure 9B. In this case study, the latent vector embedding had 128 dimensions. We used rectified linear units as the activation function for the AM state embedder, leveraging its simplicity and effectiveness in mitigating the vanishing

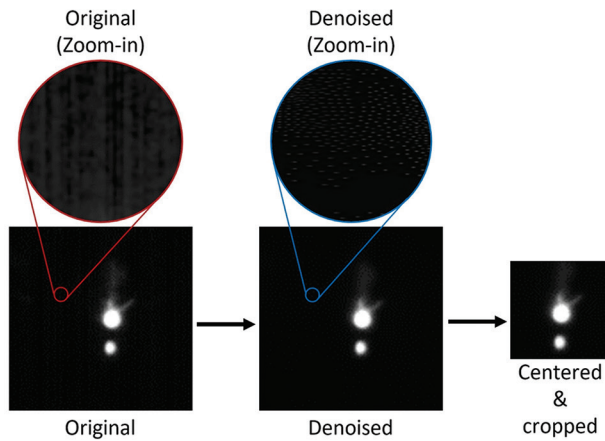


Figure 8. An illustration of melt pool monitoring image pre-processing, with zoomed-in views of the original and denoised images to demonstrate the improvement in image quality

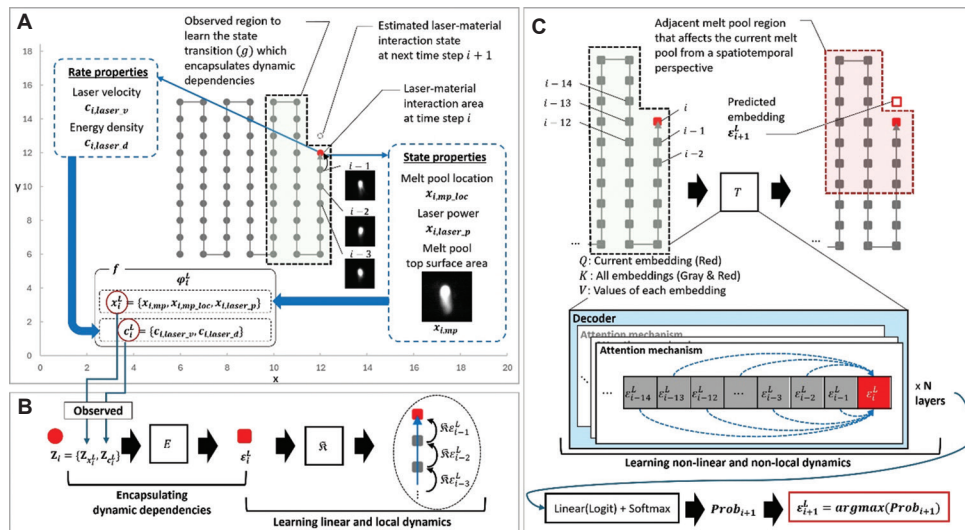


Figure 9. Diagram illustrating the implementation and operation of the AMTransformer within the case study. (A) An example of the laser powder bed fusion (LPBF) layer from the case study, showing data and target dynamical dependencies. Each dot represents a melt pool along the laser scanning path, with the shaded area indicating the observed melt pool region used to learn the dependencies captured in functions f and g . (B) The additive manufacturing (AM) state embedder operation and LPBF data flow: the circle represents observed data (Z_i), including observation of state and rate properties. The rounded square denotes AM embedding, encapsulating dynamic dependencies within AM states. The Koopman operator captures linear local state transitions. (C) The transformer operation and its LPBF data flow: the transformer processes all embeddings to reveal the adjacent melt pool region influencing the current melt pool (red dot) from a spatiotemporal perspective. Multi-head attention and multiple decoder layers consider the relationships among embeddings of the LPBF process, enabling the learning of non-linear and non-local dynamics. The decoder's output, which is a contextualized embedding, is passed through a linear layer followed by a softmax function, converting it into a probability. The highest probability event is selected as the prediction of the future LPBF states.

gradient problem.^{40,41} In this case study, we trained the AM state embedder for 300 epochs, after which the trained embeddings were passed to the transformer. As a sequence of AM state embeddings from the LPBF process is processed through the transformer, the proposed model identifies non-local dynamic dependencies in LPBF using the attention mechanism and infers future states based on its understanding, as shown in Figure 9C. The transformer consists of six decoder layers, with each decoder having four heads of attention. The feed-forward neural networks in the transformer adopted the GELU activation function, which offers smoother and more probabilistic activation, potentially enhancing the model's performance.⁴² The length of each AM state embedding sequence input for the transformer was set to 16, and the transformer was trained for 200 epochs. Figure 10 shows the learning curve of the AMTransformer, demonstrating how the model's loss decreases over time, indicating convergence. The decoder reconstructed the predicted melt pool images from the outcome embeddings of the transformer.

Experiments were conducted in a Linux environment with an Intel Xeon CPU (2 cores @2.00GHz), 12.7GB RAM, and an NVIDIA Tesla T4 GPU. The software used in this study included Python 3.10, PyTorch 2.3, and CUDA 12.2.

5.4. Results

In this study, we conducted a comprehensive evaluation of the predictive accuracy of the proposed AMTransformer. This assessment was grounded in two primary criteria: (i) the extent of congruence between the predicted future and actual MPM images and (ii) the congruence between the predicted and actual sizes of the melt pools. We compared the AMTransformer model against a transformer with a basic autoencoder model and a convolutional LSTM (ConvLSTM) model based on these criteria.

To evaluate the generated MPM images, we used the structural similarity (SSIM) metric. SSIM assesses this similarity between two images – namely, the ground truth

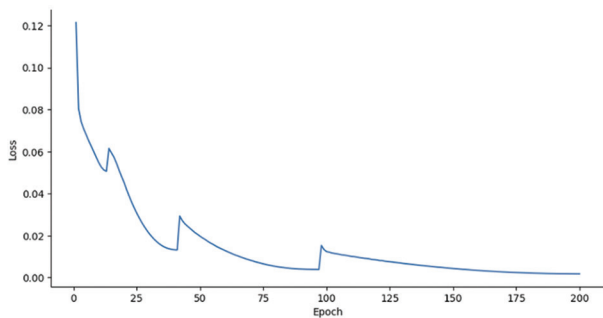


Figure 10. The learning curve of the AMTransformer

image x and the predicted image y – based on luminance, contrast, and structure.⁴³ We calculated each comparison factor using Equation XV. Utilizing the results from this equation, we derived the SSIM using Equation XVI, where μ and σ represent the average and variance, respectively, and c denotes a constant.

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}, c(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2},$$

$$s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \quad (XV)$$

$$SSIM(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y)$$

$$= \frac{(2\mu_x\mu_y + c_1)(2\sigma_x\sigma_y + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (XVI)$$

To assess the agreement between the predicted and actual size of the melt pools, we extracted the melt pool areas from the predicted and actual MPM images. Calculating the size required setting a threshold value to define the boundary of the melt pool area. We set the threshold value to 150 based on previous research that verified this value.^{15,23} Since the melt pool areas are larger than the spatter areas, we examined all the contours in the images and considered only the largest contour as the melt pool area. The size of the melt pools was determined by counting the number of pixels in the maximum contour area and multiplying it by the actual measured size value corresponding to the pixel.

We used mean absolute error (MAE) and accuracy to evaluate the prediction of melt pool size. Equation XVII presents the formula used to compute the MAE:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\alpha_i - \hat{\alpha}_i| \quad (XVII)$$

where α_i denotes the size of the melt pool in the target image and $\hat{\alpha}_i$ represents the predicted melt pool size. Accuracy is a metric that calculates the absolute error ratio between the predicted and target melt pool size using Equation XVIII:

$$Accuracy_i = 1 - \frac{|\alpha_i - \hat{\alpha}_i|}{\alpha_i} \quad (XVIII)$$

Before conducting model comparisons, the case study configured the AMTransformer by experimenting with different numbers of decoder layers and attention heads. Each configuration was evaluated using SSIM, MAE, and accuracy metrics. In the initial experiments varying the number of layers, we used a configuration of four attention heads. As shown in Table 1, increasing the number of

decoder layers generally improved model performance, with six layers achieving the best SSIM of 0.9206, an MAE of 0.0009 mm², and an accuracy of 92.73%. While adding more layers slightly enhanced MAE and accuracy, it had a negligible effect on SSIM. Thus, a six-layer configuration was deemed the most suitable for achieving high image congruence.

Table 2 presents the results of varying the number of attention heads while keeping the six decoder layers constant. The configuration with four attention heads was the most effective, delivering the highest SSIM and an optimal MAE while maintaining good accuracy. Increasing the number of attention heads slightly decreased performance, likely due to the complexity of the LPBF tool path. While each head captures distinct local patterns, there can be interdependencies between embeddings processed by different heads. For instance, the initial embeddings from the first head may have strong relationships with the last embeddings of the sixth head when they are spatially close but temporally distant. This observation indicates a need for future research on AM-specific multi-head attention mechanisms, which will be discussed further in Section 7.

Based on these findings, we selected the configuration with six decoder layers and four attention heads, as it offered the best overall performance across all metrics. With this configuration set, we compared the AMTransformer's performance against other models, including a transformer with a basic autoencoder and a ConvLSTM model.

Table 1. Performance comparisons across a varying number of decoder layer

Layer	SSIM	MAE (mm ²)	Accuracy (%)
2	0.4163	0.0057	21.87
4	0.7815	0.0016	77.52
6	0.9206	0.0009	92.73
8	0.9169	0.0008	93.04

Notes: MAE: Mean absolute error; SSIM: Structural similarity index measure.

Table 2. Variations in the number of attention heads with six decoder layers

Head	SSIM	MAE (mm ²)	Accuracy (%)
2	0.7584	0.0041	44.6
4	0.9206	0.0009	92.73
8	0.9047	0.0011	90.13
16	0.8518	0.0013	83.32

Notes: MAE: Mean absolute error; SSIM: Structural similarity index measure.

The AMTransformer achieved an SSIM of 0.9206, which is higher than the SSIMs of the transformer with a basic autoencoder (0.8699) and the ConvLSTM model (0.9069). This result indicates that the AMTransformer can predict the MPM images more accurately than the other models. For melt pool size prediction, the AMTransformer achieved an MAE of 0.0009 mm², whereas the transformer with a basic autoencoder and ConvLSTM had MAEs of 0.0017 mm² and 0.0014 mm², respectively. In addition, the proposed method for predicting melt pool size achieved an accuracy of 92.73%. Table 3 displays the comparison of model performance. According to the results, the AM – Transformer demonstrated the best overall performance. Figure 11 presents a comparison between the predicted and target images, illustrating the correspondence between the model's outputs and the ground truth.

6. Discussion

In the framework of the AMTransformer, the proposed AM dynamics formulation links the Koopman and transformer approaches to AM data. These approaches complement each other by leveraging their respective strengths for AM. The AM state embedder improves the learning of significant features of physical properties and their dynamical dependencies for each AM state within the embeddings, which are latent vector representations of these physical property features and dependencies at each time step. In addition, in our study, the adaptation of the Koopman operator with AM state embeddings in latent space representations focuses on transforming non-linear dynamical AM systems into a linear framework. This transformation enables improved analysis and prediction of key dynamical dependencies in AM using a linear method. The Koopman operator can linearize aspects of the dynamical dependencies that are amenable to linear analysis, providing a robust foundation for understanding underlying AM dynamics. In the case study, by comparing the AMTransformer with the transformer using a basic autoencoder, we explored how the AM state embedder with the Koopman operator improves the understanding of AM dynamics.

Meanwhile, the adaptation of the transformer, which employs a multi-head attention mechanism, is adept at

Table 3. Comparison of performance across models

Model	SSIM	MAE (mm ²)	Accuracy (%)
AMTransformer	0.9206	0.0009	92.73
Basic AE+Transformer	0.8699	0.0017	70.55
ConvLSTM	0.9069	0.0014	89.50

Notes: AE: Autoencoder; ConvLSTM: Convolutional long short-term memory; MAE: Mean absolute error; SSIM: Structural similarity index measure.

handling non-linear relationships within the dynamical dependencies, capturing complex patterns across various levels of abstraction. The transformer can manage and interpret the non-linear dependencies that the linear perspective of the AM state embedder's Koopman operator might miss. In AM processes, the characteristics of each point, line, or layer are often interrelated in a non-linear and non-local manner. The transformer part of the proposed method effectively captures long-range, non-linear, and non-local dynamical dependencies through self-attention mechanisms. This self-attention mechanism allows each AM state embedding to attend to all others in the concatenated inputs of AM state embedding vectors in latent representations, as shown in Figure 12. This capability enables an understanding of how changes in one physical state in AM can affect future states at

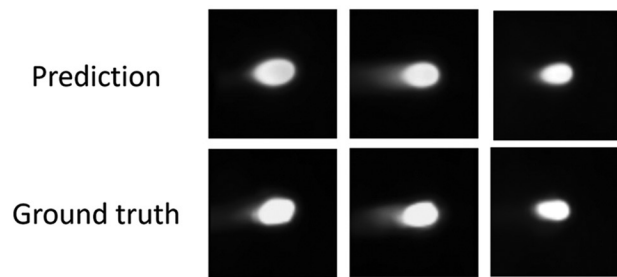


Figure 11. Comparison of predicted melt pool monitoring (MPM) images (top row) with original MPM images (bottom row)

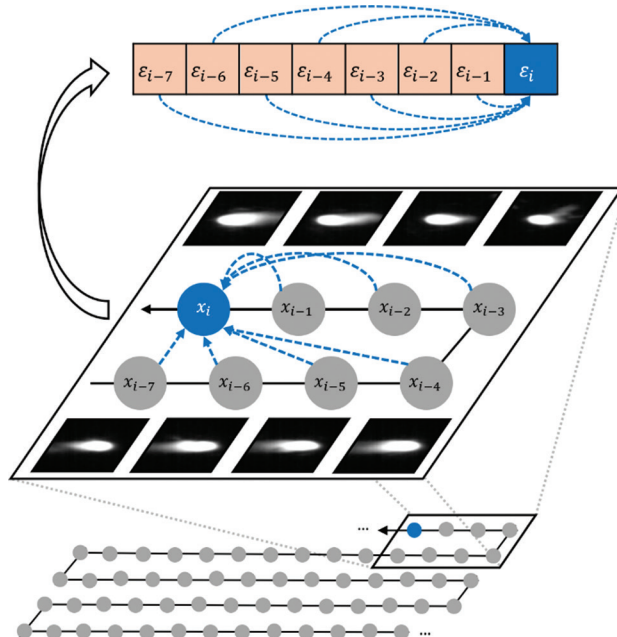


Figure 12. The AMTransformer's attention mechanism learning dynamical dependencies at multi scales: The blue dotted lines show how the AMTransformer learns dependencies among melt pools

multiple spatial and temporal scales, even if they are not adjacent in space and time. This advantage of the proposed AMTransformer contributed to its superior performance over the ConvLSTM model, which can only refer to the dependency among states sequentially.

The new modeling approach for AM proposed in this paper is impactful because AM part geometries, hierarchies, materials, and functionalities can be extremely complex due to its unique design freedom. This capability is crucial for understanding how control of a state may influence future states across multiple scales in complex patterns. The integration of linear and non-linear methods in the AMTransformer enhances data-driven modeling in AM by providing a more comprehensive understanding of the dynamics involved in AM processes. In addition, the linear method can stabilize and expedite the learning process, while the non-linear method can increase the model's adaptability and accuracy in AM scenarios where complex interactions between physical entities occur.

Our approach involves formulating the dynamics of AM into generalizable representations, which allows us to model AM without being restricted to specific types of AM processes or data. This generalizability of the proposed method facilitates the expansion of the model's scope and enhances its applicability in various fields.

7. Conclusion

This paper presents a novel method, the AMTransformer, and proposes a formal representation of AM dynamics to provide a foundation for ML models to capture these dynamics. In addition, we introduce a Koopman theory-based transformer that enables improved prediction of future AM states. The proposed method offers a fundamental modeling approach to comprehend complex spatiotemporal dependencies among physical entities and their properties in AM. Our study adapts Koopman theory and the transformer's attention mechanism to enhance the generation of latent embeddings that capture key information about AM states, their spatial-temporal dependencies, and their evolution in AM processes.

In the future, we will focus on gaining a deeper understanding and refining the design of AM-specific attention mechanisms within the model. While case study results indicate that structural changes affect attention dynamics, further exploration is needed to comprehend how the attention mechanism captures relevant dynamical dependencies both spatially and temporally. This future work will involve analyzing the alignment of attention patterns with melt pool locations and tool paths to interpret the dynamical dependencies the model captures.

By interpreting these dynamics, we aim to uncover how the attention mechanism enhances the model's explanation of foundational dynamical dependencies in AM. In addition, we will explore incorporating a priori physical knowledge into the AMTransformer to improve the predictions and interpretation. This understanding will help develop a new method to improve alignment with physical phenomena.

Acknowledgments

We wish to acknowledge the data provided by the National Institute of Standards and Technology (NIST), which was instrumental in facilitating this research.

Funding

This research was supported by Arizona State University startup funds (Award number: CC1379 PG14421), as well as by PADT and the Arizona State University Science and Technology Centers (Award number: AWD00037762).

Conflict of interest

Hyunwoong Ko is an Editorial Board Member of this journal, but was not in any way involved in the editorial and peer-review process conducted for this paper, directly or indirectly. Separately, other authors declared that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Author contributions

Conceptualization: All authors

Formal analysis: Suk Ki Lee

Investigation: All authors

Methodology: All authors

Writing – original draft: All authors

Writing – review & editing: All authors

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data

This study used publicly available data obtained through <https://doi.org/10.6028/jres.125.027>.

References

- Gibson I, Stucker B, Khorasani M. *Additive Manufacturing Technologies*. Vol. 17. Berlin: Springer; 2010.
doi: 10.1007/978-1-4419-1120-9
- Ko H, Moon SK, Hwang J. Design for additive manufacturing in customized products. *Int J Precis Eng Manuf*. 2015;16:2369-2375.
doi: 10.1007/s12541-015-0305-9
- Chua CK, Leong KF. *3D Printing and Additive Manufacturing: Principles and Applications (with Companion Media Pack)-of Rapid Prototyping*. Singapore: World Scientific Publishing Company; 2014.
doi: 10.1142/9008
- Gao W, Zhang Y, Ramanujan D, et al. The status, challenges, and future of additive manufacturing in engineering. *Comput Aided Des*. 2015;69:65-89.
doi: 10.1016/j.cad.2015.04.001
- King WE, Anderson AT, Ferencz RM, et al. Laser powder bed fusion additive manufacturing of metals; physics, computational, and materials challenges. *Appl Phys Rev*. 2015;2(4):041304.
doi: 10.1063/1.4937809
- Bikas H, Stavropoulos P, Chryssolouris G. Additive manufacturing methods and modelling approaches: A critical review. *Int J Adv Manuf Technol*. 2016;83:389-405.
doi: 10.1007/s00170-015-7576-2
- Bourell DL, Frazier WE, Kuhn HA, Seifi M. *Additive Manufacturing Processes*. Vol. 24. OH, USA: ASM International Novelty; 2020.
- Kruth JP, Levy G, Klocke F, Childs T. Consolidation phenomena in laser and powder-bed based layered manufacturing. *CIRP Ann*. 2007;56(2):730-759.
doi: 10.1016/j.cirp.2007.10.004
- Ko H, Kim J, Lu Y, Shin D, Yang Z, Oh Y. Spatial-Temporal Modeling Using Deep Learning for Real-Time Monitoring of Additive Manufacturing. In: *Proceedings of the ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 2: 42nd Computers and Information in Engineering Conference (CIE)*. St. Louis, Missouri, USA; 2022.
doi: 10.1115/DETC2022-91021
- Ko H, Lu Y, Yang Z, Ndiaye NY, Witherell P. A framework driven by physics-guided machine learning for process-structure-property causal analytics in additive manufacturing. *J Manuf Syst*. 2023;67:213-228.
doi: 10.1016/j.jmsy.2022.09.010
- Yan W, Lin S, Kafka OL, et al. Data-driven multi-scale multi-physics models to derive process-structure-property relationships for additive manufacturing. *Comput Mech*. 2018;61:521-541.
doi: 10.1007/s00466-018-1539-z
- Marshall GJ, Thompson SM, Shamsaei N. Data indicating

- temperature response of Ti-6Al-4V thin-walled structure during its additive manufacture via Laser Engineered Net Shaping. *Data Brief*. 2016;7:697-703.
doi: 10.1016/j.dib.2016.02.084
13. Mitchell JA, Ivanoff TA, Dagle D, Madison JD, Jared B. Linking pyrometry to porosity in additively manufactured metals. *Addit Manuf*. 2020;31:100946.
doi: 10.1016/j.addma.2019.100946
14. Fathizadan S, Ju F, Lu Y. Deep representation learning for process variation management in laser powder bed fusion. *Addit Manuf*. 2021;42:101961.
doi: 10.1016/j.addma.2021.101961
15. Yang Z, Lu Y, Yeung H, Krishnamurthy S. From scan strategy to melt pool prediction: A neighboring-effect modeling method. *J Comput Inf Sci Eng*. 2020;20(5):051001.
doi: 10.1115/1.4046335
16. Zhang Z, Sahu CK, Singh SK, Rai R, Yang Z, Lu Y. Machine learning based prediction of melt pool morphology in a laser-based powder bed fusion additive manufacturing process. *Int J Prod Res*. 2024;62(5):1803-1817.
doi: 10.1080/00207543.2023.2201860
17. Tempelman JR, Wachtor AJ, Flynn EB, *et al*. Detection of keyhole pore formations in laser powder-bed fusion using acoustic process monitoring measurements. *Addit Manuf*. 2022;55:102735.
doi: 10.1016/j.addma.2022.102735
18. Kononenko DY, Nikonova V, Seleznev M, van den Brink J, Chernyavsky D. An *in situ* crack detection approach in additive manufacturing based on acoustic emission and machine learning. *Addit Manuf Lett*. 2023;5:100130.
doi: 10.1016/j.addlet.2023.100130
19. Lough CS, Escano LI, Qu M, *et al*. In-situ optical emission spectroscopy of selective laser melting. *J Manuf Process*. 2020;53:336-341.
doi: 10.1016/j.jmapro.2020.02.016
20. Montazeri M, Nassar AR, Dunbar AJ, Rao P. In-process monitoring of porosity in additive manufacturing using optical emission spectroscopy. *IISE Trans*. 2020;52(5):500-515.
doi: 10.1080/24725854.2019.1659525
21. Liu J, Ye J, Silva Izquierdo D, Vinel A, Shamsaei N, Shao S. A review of machine learning techniques for process and performance optimization in laser beam powder bed fusion additive manufacturing. *J Intell Manuf*. 2023;34(8):3249-3275.
doi: 10.1007/s10845-022-02012-0
22. Cai Y, Xiong J, Chen H, Zhang G. A review of *in-situ* monitoring and process control system in metal-based laser additive manufacturing. *J Manuf Syst*. 2023;70:309-326.
doi: 10.1016/j.jmsy.2023.07.018
23. Yang Z, Lu Y, Yeung H, Krishnamurthy S. *Investigation of Deep Learning for Real-Time Melt Pool Classification in Additive Manufacturing*. United States: IEEE; 2019:640-647.
doi: 10.1109/COASE.2019.8843291
24. Zhang Y, Soon HG, Ye D, Fuh JYH, Zhu K. Powder-bed fusion process monitoring by machine vision with hybrid convolutional neural networks. *IEEE Trans Ind Inf*. 2019;16(9):5769-5779.
doi: 10.1109/TII.2019.2956078
25. Larsen S, Hooper PA. Deep semi-supervised learning of dynamics for anomaly detection in laser powder bed fusion. *J Intell Manuf*. 2022;33(2):457-471.
doi: 10.1007/s10845-021-01842-8
26. Fernandez-Zelaia P, Dryepondt SN, Ziabari AK, Kirka MM. Self-supervised learning of spatiotemporal thermal signatures in additive manufacturing using reduced order physics models and transformers. *Comput Mater Sci*. 2024;232:112603.
doi: 10.1016/j.commatsci.2023.112603
27. Guirguis D, Tucker C, Beuth J. Accelerating process development for 3D printing of new metal alloys. *Nat Commun*. 2024;15(1):1-12.
doi: 10.1038/s41467-024-44783-5
28. Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. *arXiv*. Preprint posted online 2017.
doi: 10.48550/arXiv.1706.03762
29. Koopman BO. Hamiltonian systems and transformation in Hilbert space. *Proc Natl Acad Sci*. 1931;17(5):315-318.
30. Rowley CW, Mezić I, Bagheri S, Schlatter P, Henningson DS. Spectral analysis of nonlinear flows. *J Fluid Mech*. 2009;641:115-127.
doi: 10.1017/S0022112009992059
31. Brunton SL, Kutz JN. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge, UK: Cambridge University Press; 2019.
doi: 10.1017/9781108380690
32. Geneva N, Zabarar N. Transformers for modeling physical systems. *Neural Netw*. 2022;146:272-289.
doi: 10.1016/j.neunet.2021.11.022
33. Mezić I. Analysis of fluid flows via spectral properties of the Koopman operator. *Annu Rev Fluid Mech*. 2013;45:357-378.
doi: 10.1146/annurev-fluid-011212-140652
34. Schmid PJ. Dynamic mode decomposition and its variants. *Annu Rev Fluid Mech*. 2022;54:225-254.

- doi: 10.1146/annurev-fluid-030121-015835
35. Schmid PJ. Dynamic mode decomposition of numerical and experimental data. *J Fluid Mech.* 2010;656:5-28.
doi: 10.1017/S0022112010001217
36. Radford A, Narasimhan K, Salimans T, Sutskever I. Improving language understanding by generative pre-training. 2018, OpenAI Blog. Available from: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf [Last accessed on 2024 Aug 29].
37. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. *Open AI Blog.* 2019;1(8):9.
38. Wolf T, Debut L, Sanh V, *et al.* HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv.* Preprint posted online 2019.
doi: 10.48550/arXiv.1910.03771
39. Lane B, Yeung H. Process monitoring dataset from the additive manufacturing metrology testbed (ammt): Overhang part x4. *J Res Natl Inst Stand Technol.* 2020;125:1-18. doi: 10.6028/jres.125.027
40. Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics.* JMLR Workshop and Conference Proceedings; 2011:315-323.
41. Glorot X, Bordes A, Bengio Y. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In: *Proceedings of the 28th International Conference on Machine Learning;* 2011:513-520.
42. Hendrycks D, Gimpel K. Gaussian error linear units (GELUs). *arXiv.* Preprint posted online 2016.
doi: 10.48550/arXiv.1606.08415
43. Wang Z, Simoncelli EP, Bovik AC. Multiscale structural similarity for image quality assessment. In: *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers.* IEEE; 2003:1398-1402.
doi: 10.1109/acssc.2003.1292216