Ontology-guided Attribute Learning to Accelerate Certification for Developing New Printing Processes

Tsegai O. Yhdego [1], Hui Wang [1], Zhibin Yu [1], and Hongmei Chi [2]
[1] Department of Industrial and Manufacturing Engineering
Florida A&M University- Florida State University College of Engineering
[2] Department of Computer and Information Sciences, Florida A&M University
Tallahassee, Florida, USA

March 23, 2024

Abstract

Identifying printing defects is vital for process certification, especially with evolving printing technologies. However, this task proves challenging, especially for micro-level defects necessitating microscopy, which presents a scalability barrier for manufacturing. To address this challenge, we propose an attribute learning methodology inspired by human learning, which identifies shared attributes among seen and unseen objects. First, it extracts defect class embeddings from an engineering-guided defect ontology. Then, attribute learning identifies the combination of attributes for defect estimation. This approach enables it to recognize previously unseen defects by identifying shared attributes, even those not included in the training dataset. The research formulates a joint optimization problem for learning and fine-tuning class embedding and ontology and solves it by integrating natural language processing, metaheuristics for exploration and exploitation, and stochastic gradient descent. In a case study involving a direct-ink-writing process for creating nanocomposites, this methodology was used to learn new defects not found in the training data using the optimized ontology. Compared to traditional zero-shot learning, this ontology-based approach significantly improves class embedding, outperforming transfer learning in one-shot and two-shot learning scenarios. This research represents an early effort to learn new defect concepts, potentially reducing the need for extensive measurements in defect identification.

Keywords: Additive Manufacturing; Attribute learning; Ontology; Defect identification; Process certification

1 Introduction

In recent decades, the development pace of new processes for additive manufacturing has been accelerating; however, many research efforts stopped at proof-of-concept. One grand challenge is identifying defects and anomalies in printed structures that are helpful for

process certification. Automated defect identification is critical for qualifying and certifying printing processes and parts, as it enables the evaluation of whether printed layers or parts are defect-free or have specific types of defects. This capability is especially useful when developing new printing processes. Typically, manufacturers rely on their experience for offline inspection of printed parts or online inspection of printed layers for process certification. However, if they lack experience and measurement capability or budget, defect identification becomes challenging, hindering the development of repeatable processes for scale-up production.

The defect identification requires significant efforts in conducting measurements to collect labeled data that identifies the defect types. Afterward, machine learning algorithms can be implemented to train classifiers that can categorize defects to build a process window that reveals how processes impact defects. Nevertheless, the measurements usually involve microstructure characterization by microscopy, which could be expensive and time-consuming. For inexperienced researchers and manufacturers, there exist entry barriers to developing repeatable processes for the scale-up manufacturing and commercialization of new printing technologies within a short period of time.

Recent research presents potential solutions to the classification problem given limited measurements or experiments. Transfer learning, including multi-task learning (Pandiyan et al. (2022); Cheng et al. (2017)), stands out as one method that has been attempted by recent research, aiming to improve the learning accuracy through the transfer of knowledge from a related learning task. The existing framework of transfer learning still expected some labeled data to be collected for a target learning task. However, a zero-shot learning (ZSL) scenario can be more realistic for new process development, i.e., some anomalies or defects are completely unobserved in the training data.

This paper proposes a method to deal with ZSL by imitating human recognition behavior to recognize new objects by recognizing the combinations of attributes. The method explores the development of a defect ontology as a library of managing defect knowledge to obtain attributes, a significant challenge in implementing the ZSL approach. An ontology can be established for process anomalies by developing a unified way of representing the defect information, called attributes, from published data and engineering knowledge that can be shared across a range of processes. The learning of defects in a new process may

leverage the established ontology to characterize the printing anomalies or defects. This way may enable the identification of unseen defects, i.e., a ZSL scenario. The ontology-based approach has recently become popular in the manufacturing industry, especially in developing maintenance strategies for system failures. Its potential in learning printing defects for new processes has not been understood.

1.1 *State-of-the-art*

This section reviews the related research on the certification and small-sample quality control for additive manufacturing (AM) and state-of-the-art research on ontology, primarily in manufacturing applications.

Defect Detection for Process Certification. Several studies have been proposed to identify printing defects for process certification that evaluates whether the printed layers are defect-free. Depending on the defect types, different inspections were implemented for post-printing processes or in real time. For real-time detection, researchers employed spatial data cloud by 3D scanners (Ye et al. (2021)), thermal images by infrared or co-axial pyrometer cameras (Khanzadeh et al. (2019); Seifi et al. (2019)), and/or stream videos by high-speed cameras Vora and Sanyal (2020), acoustic emissions (Wu et al. (2016)), while post-printing inspections include microscopy, scanning electron microscopy (SEM), X-ray computed tomography, magnetic resonance imaging, and/or ultrasonic testing as reviewed in Vora and Sanyal (2020).

Defect Feature Extraction. Recent research extract features to characterize and detect defects and their underlying causes by using non-traditional defect detection methods. *Infrared thermal imaging technology* proposed by Bartlett et al. (2018) uses a long-wave infrared camera to detect the shape and contour of defects as thermal radiation differences. Thermal analysis was also proposed in Bappy et al. (2022) in order to detect abnormalities in the morphological dynamics of melt pools and heat-affected zones. Similarly, Esfahani et al. (2022) carried out thermal image series analyses to characterize defects by studying the dynamics in the layer-wise thermal history. *Penetration defect detection* in Chen et al. (2021) uses capillary action and fluorescent or colored dyes to identify surface-opening defects. After applying the penetrant fluorescent dye, a developer is applied to characterize the defect. *Eddy current testing* proposed by Chen et al. (2021) uses electromagnetic in-

duction to identify defects in conductive materials by measuring changes in induced eddy currents. Such changes represent the test piece's material, defect, shape, and size. *Ultrasonic testing* (Du et al. (2018)) uses ultrasonic waves to inspect internal defects and is more sensitive to cracks, incomplete penetration, and infusion defects. In ultrasonic testing, reflected high-frequency sound waves characterize defects. *Laser ultrasonic detection* methods have been proposed to improve the performance of ultrasonic testing for crack defects and have better performance in defect detection (Millon et al. (2018)). *Laser ultrasonic detection*, on the hand, detects sound waves representing AM defects using a pulsed laser and optical detection via interferometers.

Small-sample quality control for AM. Recent research has been developed to reduce the measurements and samples needed in data modeling or machine learning for geometric deviation and shape error compensation Luan and Huang (2016); Wang et al. (2016); Jin et al. (2016); Cheng et al. (2017). This line of research decomposes the product shapes in the shape error model. It progressively develops the shape error prediction model from simple shapes, such as cylinders and polygons, to free-form shapes. Sabbaghi et al. (2018); Jin et al. (2016); Sabbaghi and Huang (2016) developed a statistical framework of effect equivalence for transferring information across multiple shapes and printing processes. Overall, the related research focuses on geometric errors of the printed shapes and starts modeling from specific common shapes to establish knowledge transfer. However, the internal defects in each layer, as more concerned with emerging printing processes for functional structures using novel materials, are not considered. Furthermore, a transfer learning technique was also utilized by Scime and Beuth (2018) to efficiently train a multi-scale convolution neural network using a pre-trained AlexNet (MsCNN), which detected powder irregularity for the classification of defects.

Zero-shot learning in computer vision. In the past two decades, a significant breakthrough has been made for small-sample learning in computer vision. One accomplishment is zero-shot learning (ZSL), aiming to recognize objects that are not included in training data. In other words, testing and training class sets are disjointed under a ZSL environment. This line of approach is to identify new objects by transferring attributes from heterogeneous objects (Lampert et al. (2013)). One common strategy in ZSL is to learn a linear compatibility function between the features extracted from input images and attributes associated

with classes (class embedding) as annotated by experts or learned from elsewhere. The related approaches include ALE (Akata et al. (2015)) and Latent Embeddings (LatEm) Xian et al. (2016). These methods only focus on the visual recognition of new objects without any engineering background. Additionally, there were limited discussions on how to identify defect attributes to improve the ZSL accuracy.

1.2 *Research gaps and proposed work*

This paper identifies the following research gaps in small-sample learning for AM.

- Little work was found on identifying AM defect and morphology anomalies when the labeled data about certain defects are all missing in the training data. Such a ZSL scenario presents a major challenge to traditional classification algorithms and even recent transfer learning or semi-supervised learning, which still requires limited data from the target process.

- There is a lack of methods for effectively extracting class attributes about printing defects to ensure the ZSL accuracy. Existing ZSL methods in computer vision primarily rely on the expert annotation of data to generate different attributes and create class embedding for ZSL algorithms. Also, attribute extraction and ZSL were studied separately and implemented in sequence. However, the selection of proper class attributes significantly affects ZSL accuracy. It is essential to avail of a method that can integrate engineering knowledge with machine learning for selecting appropriate attributes shared between seen and unseen defect classes.

This survey demonstrate that knowledge on defect formation and characteristics is available with literature or online dataset to supplement data for supporting ZSL, thereby proving the feasibility of the proposed attribute learning. Hence, this research proposes an ontology-guided attribute learning framework that addresses the challenges in (1) ZSL classification of printing defects that are unobserved in training data by recognizing attributes shared between seen and unseen classes and (2) the learning of defect attributes and associated class embedding to facilitate ZSL. This paper proposes to develop a defect ontology initialized by published/shared data and information from the literature to extract class embedding. A random walk algorithm is developed to explore the ontology structure

to generate a semantic description of defects, allowing natural language processing to parse the ontology to defect embedding as attributes. The attribute learning framework formulates a joint optimization problem to tune the defect ontology and determine an embedding method. The problem is solved by integrating stochastic gradient descent with metaheuristics by exploration/exploitation. The established defect ontology can help identify defects in more printing processes under a ZSL scenario. A case study establishes/optimizes a defect ontology for a direct-ink-writing of new composite materials and utilizes it to identify unseen defects. The study also discusses the impact of learning parameters on defect classification accuracy and the robustness of the algorithm to noises in the input data.

In a broader aspect, this study is expected to become a building block for a pipeline that scrapes data from literature and online databases to build a unified representation of defects knowledge through ontology. This utilization and re-structuring of open-access engineering information are expected to reduce the need for extensive testing and measurement (especially microscopy). Eventually, the outcome of this study can expedite the new process certification and lower the entry barriers for researchers/practitioners to commercialize their technologies for market demand promptly.

The remainder of this paper is organized as follows. Section 2 proposes the methodology of ontology-guided attribute learning, including an ontology-guided method to extract attributes to facilitate ZSL in section 2.1, learning problem formulation, and a two-stage solution algorithm in section 2.2, followed by a flowchart in supplemental materials. Section 3 presents a case study on a direct-ink-writing process for nanocomposites with results and discussions. Finally, Section 4 summarizes the paper and outlines future research directions.

2 Ontology-guided Attribute Learning

The idea of attribute learning is to imitate the way how a human being would learn a new concept, as discussed in Lampert et al. (2013). Figure 1 illustrates an analogous example in which a human learns to recognize an unseen class (ostrich) through shared attributes with the panda, chicken, and other small birds. Although the person has never seen an ostrich before, he/she can recognize the unique combination of feather, bill, and leg. Meanwhile, literature may report an ostrich is a large bird that does not fly and has dark/light colors.

All the combination of the attribute information relevant to an ostrich can help identify it with a highly successful chance.

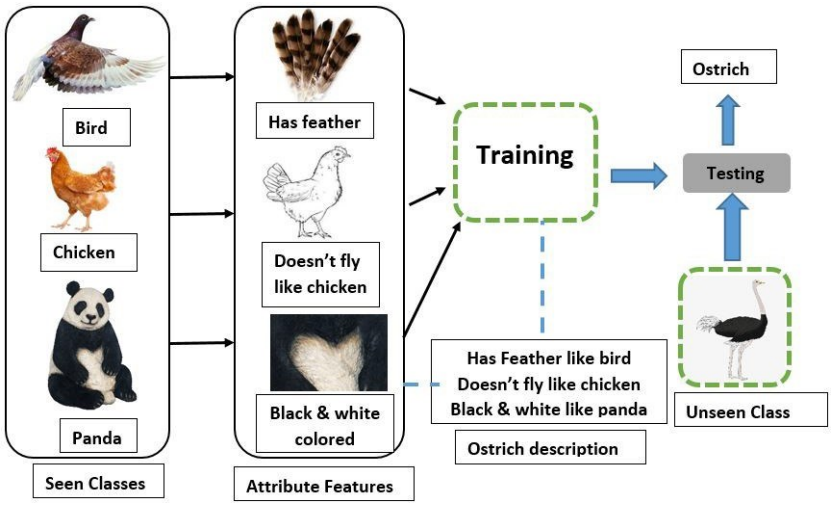

Figure 1: Illustration of attribute learning leveraging information from seen classes, along with text, to identify unseen class ostrich.

Figure 2 presents a high-level illustration of learning by attributes. X is a dataset, including training and testing data that cover the information for K discrete types (classes) of defects $Y^{tr} = \{y_1, y_2 \ldots y_K\}$. Generally, training and testing splits can be represented as $Y^{ts} - Y^{tr} = \phi$, indicating that seen and unseen classes can all be present in the testing data. However, ZSL represents a special testing case where testing data contains defect classes $Y^{ts} = \{z_1, z_2, \ldots z_N\}$ that are not fully observed in or disjoint with training data, i.e., $Y^{ts} \cap Y^{tr} = \emptyset$. In the case study, both testing scenarios will be explored. Each defect class can be associated with a set of attributes $A = \{a_1, a_2 \ldots a_L\}$, which describe the physical property, material, and printing condition, representing multi-level properties of classes shared across different processes or objects. The arrow within the attribute layer indicates the relationship between the attributes, e.g., the dependency between properties. The diagram shows that defects unobserved in training data can leverage the attributes shared with seen defects in the training data to identify the defects.

The formulation of attribute learning for ZSL scenarios can be described as follows. Given dataset of N defects $S = \{(\mathbf{x}_n, y_n), n = 1....N\}$, where $y_n \in Y^{tr}$ belongs to the training set of defects or seen classes and $\mathbf{x}_n$ is the corresponding data, such as images

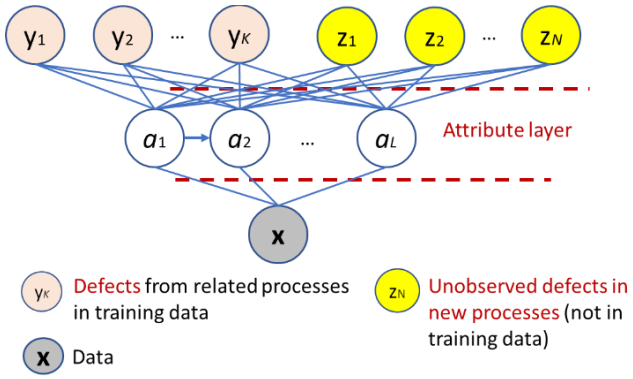

Figure 2: Attribute learning of unseen classes by leveraging the information from attribute layers shared between seen and unseen classes.

represented by vectors. The objective is to map input data, such as microscopic images, to its corresponding class even if images representing the class were not included in the training dataset. Similarly to Xian et al. (2017), for a given dataset with input images X and class label sets Y, a mapping function f needs to be trained such that $f : X \rightarrow Y$, where the mapping function $f(\mathbf{x}; \mathbf{W})$ can be defined as:

$$f(\mathbf{x}; \mathbf{W}) = argmax_{y \in Y} F(\mathbf{x}, y; \mathbf{W}) \quad (1)$$

For ZSL testing, the objective is to assign a class label to the data/image with the highest compatibility. In the testing phase, the performance of the trained classifier was evaluated to identify unseen classes that were not present in the training dataset. The compatibility function aims to find the most compatible class with the testing data/images. The compatibility function is defined on the features of input data $\boldsymbol{\theta}(\mathbf{x})$ and attributes of class $\boldsymbol{\phi}(y)$ and can be represented as:

$$F(\mathbf{x}, y; \mathbf{W}) = max(\boldsymbol{\theta}'(\mathbf{x})\mathbf{W}\boldsymbol{\phi}(y)) \quad (2)$$

When applying the classifier to the features extracted from testing images, different candidate classes are evaluated to determine which class embedding can maximize the compatibility with features from images. The objective of classification during the testing stage is to identify the candidate class that has the best compatibility with the input features of the testing images and report it as the output class. This compatibility-based

classification during the testing stage is a common practice in computer vision (Xian et al. (2017)). Therefore, unseen classes in the testing data will be assigned a label identified by this compatibility mapping during the testing stage and later compared to actual labels for performance evaluation. Assume that the measured data from a new process are $\mathbf{x\_test}_n$. The classification is achieved by finding a class k so that the compatibility between input features $\boldsymbol{\theta}(\mathbf{x\_test}_n)$ and the kth class embedding $\boldsymbol{\phi}(y_k)$ is maximized, i.e.,

$$\hat{k} = argmax\{\boldsymbol{\theta}'(\mathbf{x\_test}_n)\hat{\mathbf{W}}\,\boldsymbol{\phi}(y_k)\} \tag{3}$$

There exists abundant research on extracting features $\boldsymbol{\theta}(\mathbf{x})$ from images, such as Img2Vec python library for image embedding, which uses ResNet50 (He et al. (2016)) model and its weights pre-trained on the ImageNet dataset (Deng et al. (2009)). By contrast, the research on the extraction of class attributes $\boldsymbol{\phi}(y)$ for printing defects is limited. As such, the key challenge in successfully implementing attribute learning is to obtain appropriate combinations of attributes $\boldsymbol{\phi}(y)$ from data, as will be discussed in the next section.

2.1 *Defect ontology and class embedding for attribute extraction*

This paper proposes to extract attributes from the defect ontology learned from the information with literature, online databases, and other process data. The defect ontology represents a way of managing the knowledge of defect classes with a relationship structure that can be shared across a range of processes. The attributes of defects can be extracted from the class embedding (vectorial representation) from the ontology. This section first formulates a joint learning problem for tuning ontology structure and method of creating embedding vectors. Based on the embedding, this section develops a learning algorithm to solve the problem by integrating stochastic gradient descent with metaheuristics based on exploration and exploitation.

2.1.1 *Defect knowledge organization by ontology*

This section develops a semantic representation of engineering knowledge to extract class attributes $\boldsymbol{\phi}(y)$. A structural knowledge representation of a manufacturing defect, known as a defect ontology, is used to represent information about defects such as morphology,

causes, and materials. In this work, ontology is used to capture the knowledge about different aspects of defects that occur during printing, such as morphology, underlying causes, and associated materials.

The steps involved in building an ontology for defects in additive manufacturing are as follows. First, it is necessary to identify the relevant concepts, relationships, and constraints associated with the defects. These concepts and relationships are then represented as class/subclass properties within the ontology. Next, they are organized under a hierarchical structure using subclass and superclass relationships. Supplementary S1 and S2 outline the details of the ontology-building process and updating with new information respectively.

Assume that the ontology has N classes and $y \in Y = \{y_1, \ldots\ldots, y_N\}$, and each class in the structure has a set of M attributes $\{a_i,\ i = 1, 2 \ldots . M\}$ to describe the class. Each class can be defined in an M-dimension attribute vector as $\boldsymbol{\varphi}(\mathbf{y}) = [\rho_{y,1} \ldots\ldots \rho_{y,M}]$, where $\rho_{y,i}$ represents the measure of the association between each attribute a_i and class y. If the attribute and class are not associated, the corresponding value is zero. Stacking up attribute vectors for N classes leads to an $N \times M$ matrix $\boldsymbol{\varphi}$ as shown in Figure 6, where grey cells indicate that the attributes in the row are selected to be associated with the defect class (column) for class embedding in the next step.

2.1.2 *Parsing ontology for defect embedding*

This section proposes an ontology exploration strategy integrated with natural language processing (NLP) to convert or parse the ontology into class embedding in vector representation to facilitate ZSL. To parse the ontology, a ”walk” algorithm was developed. This algorithm generates sentences that describe various aspects of the defect class using domain-specific keywords such as morphology, color, surface texture, causes, and printing parameters. These sentences are generated by traversing the ontology from the root to the end nodes. The exploration starts with different defect nodes followed by different branches in the ontology. For representation purposes, each combination of ontology branches and segments can be represented as a gene. Take the ontology in Figure 3 for example. Gene 13 can represent that defect 1 is associated with branch 3, gene 16 with branch 6, and gene 17 with branch 7. The types of the association are labeled by different styles of arrows. Table 1 shows an example of exploring ontology to generate genes and their explanations.

The exploration of the ontology along different branches leads to different keyword sentences. The table also summarizes the keyword sentences extracted corresponding to each gene.

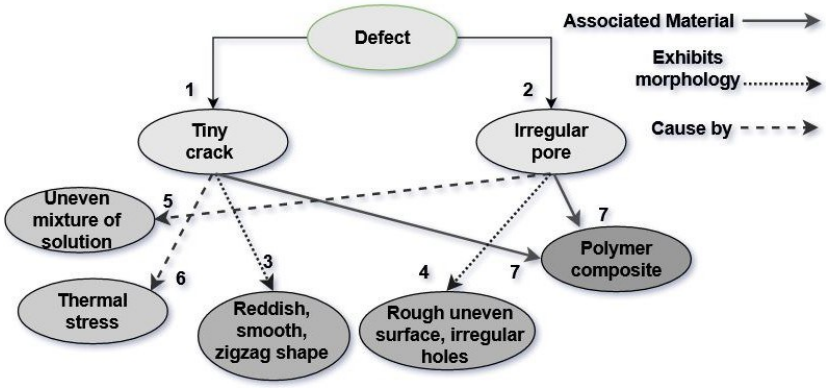

Figure 3: An example of a defect ontology consisting of different relationships to identify sentences containing specific attributes that describe the defect.

Table 1: presents an example of key-phrase sentences generated by the *RandomWalk* algorithm, as depicted in Figure 3. Each sentence is represented by a ”gene” that corresponds to the branch from which it was generated. For example, gene 13 represents that defect 1 is associated with branch 3.

Gene	**Extracted key-word sentences**
13	Tiny crack exhibits zigzag-shaped morphology on a reddish smooths surface.
16	Tiny crack is caused by thermal stress.
17	Tiny crack is associated with polymer composite material.
24	Irregular pore has a morphology of irregular holes on rough uneven surfaces.
25	Irregular pores are caused by the uneven mixture of solutions.
27	Irregular pore is associated with polymer composite material.

Each sentence is then tokenized, breaking down the sentence into individual words or phrases using an open-source python library called ”TransformerWordEmbeddings.” These tokens are then passed through the BERT model for embedding. BERT (Devlin et al. (2018)), which stands for ”Bidirectional Encoder Representations from Transformers,” is a pre-trained transformer-based neural network that has been trained on a large corpus

of text data. The embeddings generated by BERT capture the meaning and context of the defect class in the sentence. By extracting the embedding for the token representing the class, this information can be used for attribute-based learning. The ontology walk algorithm is repeated to optimize the embedding process and generate different sentences for each class. By performing BERT on the newly generated sentences after each walk, a representation of the class can be obtained with shared attributes between seen and unseen classes.

Next, the extracted keyword sentences are converted into contextualized embedding vectors ($\boldsymbol{\phi}$) by using BERT, a self-supervised transformers model pre-trained on a massive corpus of multilingual data. The self-supervision implies that the model was pre-trained entirely on raw texts, with no human labeling. A popular BERT model is provided by Huggingface's transformers package (Wolf et al. (2019)). BERT will convert an extracted sentence to a column vector of dimension M for class embedding, i.e., $\boldsymbol{\phi}(y)$. The embedding vectors for all N classes can be collected in a $\boldsymbol{\phi}$ matrix with dimension $M \times N$ (e.g., Figure 4). Once the extracted sentences are converted to a column vector of dimension M through BERT, the embedding of each class (column) can be fine-tuned by swapping between rows in the $\boldsymbol{\phi}$ matrix (Figure 4). Such a manipulation essentially applies linear mapping to the initial embedding matrix converted from the ontology to obtain the finalized embedding vector. The proposed procedure of converting/parsing defect ontology $(\boldsymbol{\varphi})$ to the vector representation of class (i.e., defect class embedding $\boldsymbol{\phi}$) is summarized in Figure 5.

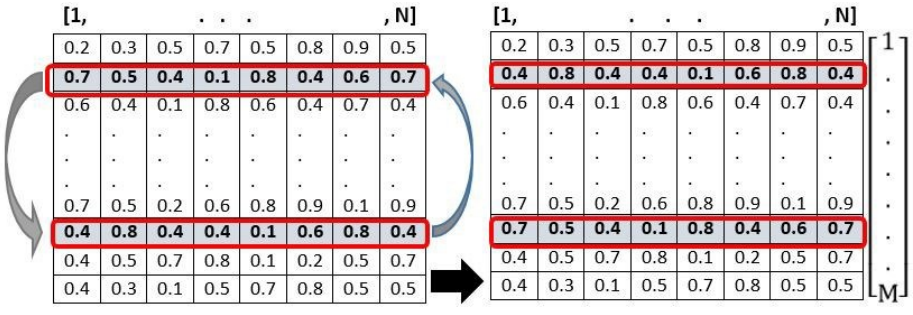

Figure 4: Linear mapping, such as row swapping, to fine-tune the embedding vector of defect classes

The proposed methodology essentially relies on experts' domain knowledge to create attribute vectors that are suitable for defects in additive manufacturing. However, it has

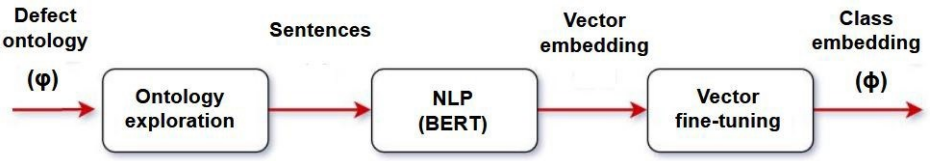

Figure 5: The logic flow of using ontology and natural language processing to obtain defect class embedding.

the advantage of providing a generalized, uniform representation of engineering knowledge about defects in printing applications (Please refer to Supplementary S3 for details).

2.2 *Joint ontology tuning and class embedding guided by ZSL*

This section discusses the optimization problem formulation for tuning the ontology and class embedding guided by ZSL. It focuses on the challenge in the formulated problem and proposes a solution algorithm integrating stochastic gradient descent with metaheuristics based on exploration/exploitation.

2.2.1 *Formulation of joint optimization*

Once features $\boldsymbol{\theta}(\mathbf{x})$ and attributes $\boldsymbol{\phi}(y)$ are extracted, the objective Eq. 2 can be detailed with an optimization formulation. The proposed framework learns the feature-to-attribute mapping matrix $\boldsymbol{W}$, attribute stacking matrix $\boldsymbol{\varphi}$ and class embedding matrix $\boldsymbol{\phi}$ by minimizing the objective function:

$$min_{\mathbf{W},\boldsymbol{\varphi},\boldsymbol{\phi}} \frac{1}{N} \sum_{n=1}^{N} L(y_n, f(\mathbf{x}_n; \mathbf{W}, \boldsymbol{\varphi}, \boldsymbol{\phi})) \tag{4}$$

where L is the loss function, defined as a ranking-based loss function (Xian et al. (2016)):

$$L = \sum_{y \in Y^{tr}} max\{0, \Delta(y_n, y) + F(\mathbf{x}_n, y; \mathbf{W}) - F(\mathbf{x}_n, y_n; \mathbf{W})\}. \tag{5}$$

where $\Delta(y_n, y) = 1$ if $y \neq y_n$ and 0 otherwise. This loss function can force the model to produce higher compatibility between the input data (e.g., image) embedding and the class embedding of the true label than between the image embedding and the class embedding of incorrect labels. In the training phase, the mapping $\hat{\mathbf{W}}$, ontology $\hat{\boldsymbol{\varphi}}$, and class embedding

$\hat{\boldsymbol{\phi}}$ are jointly estimated through (4) and (5) based on labeled samples from published data sources and literature.

Challenge in the joint optimization: The formulation (4) and (5) have the challenge in the combinatorial search for ontology and defect class embedding and minimization of a non-smooth ranking-based loss function, which is not jointly convex in all the $\mathbf{W}'s$. The combinatorial search in this optimization can be briefly explained in an example in Figure 6, where an ontology with two-level structures is simplified as a matrix. On the left panel, light grey cells indicate all candidate attributes (row) associated with each class (column) guided by engineering knowledge. The right panel is the selection of attributes as indicated by dark grey cells after tuning. The ontology can be restructured by exploring different combinations of attributes and connections between levels and sub-levels in ontology $\boldsymbol{\varphi}$. Furthermore, the search for the embedding of the sentences from ontology exploration, as shown in Figure 4, adds to the computational cost of obtaining defect class embedding. Such combinatorial search can be computation-expensive if the ontology has a large structure with multiple levels. The ontology exploration and search for class embedding vectors are guided by performance feedback from ZSL based on the minimization of the loss function. This performance feedback represents the ability of the current embedding to represent the class and promote better attribute sharing between seen and unseen classes.

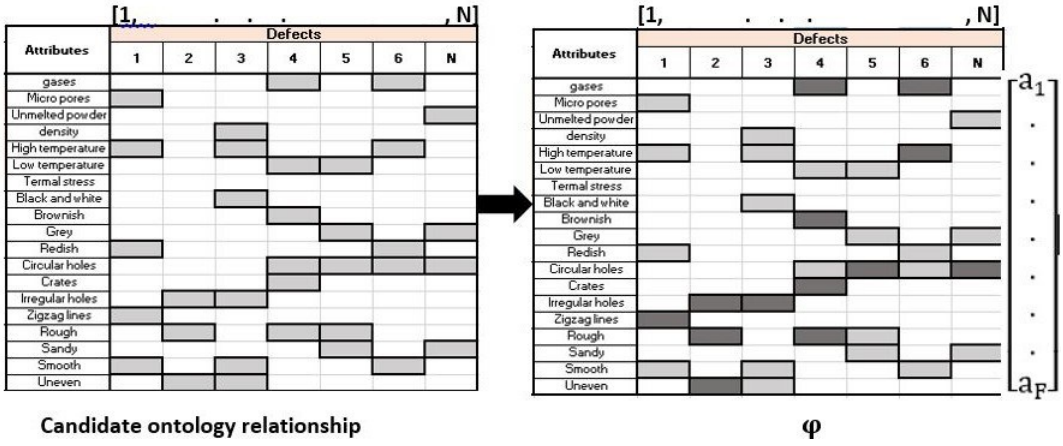

Figure 6: An illustration of ontology optimization. The left panel shows all candidate attributes based on domain knowledge, represented by light grey cells. The right panel shows the selected attributes after tuning, represented by dark grey.

Remark: If engineering knowledge about the ontology structure $\boldsymbol{\varphi}^A$ is available, the objective (4) can be regularized to ensure the closeness with $\boldsymbol{\varphi}$ such as:

$$min_{\mathbf{W},\boldsymbol{\varphi},\boldsymbol{\phi}} \frac{1}{N} \sum_{n=1}^{N} L(y_n, f(\mathbf{x}_n; \mathbf{W}, \boldsymbol{\varphi}, \boldsymbol{\phi})) + \lambda \|\boldsymbol{\varphi} - \boldsymbol{\varphi}^A\|^2,$$

where λ is the regularization parameter that can be later determined by cross-validation. This formulation is to fine-tune the ontology according to the data collected.

2.2.2 *Integrating metaheuristics with stochastic gradient descent*

The optimization in the training phase can adopt a two-stage strategy: (1) search for mapping $\mathbf{W}$ in the compatibility function given class embedding $\boldsymbol{\phi}$ and ontology structure $\boldsymbol{\varphi}$ by stochastic gradient descent to minimize the ZSL loss function in Eq. (4) and (2) find class embedding $\boldsymbol{\phi}$ and ontology structure $\boldsymbol{\varphi}$ through exploration and exploitation. The ZSL loss in stage (1) will guide the tuning of ontology $\boldsymbol{\varphi}$ and embedding $\boldsymbol{\phi}$. Supplementary S8 provides the flow chart of the optimization process.

The procedure is summarized in Algorithm 1. The input data and parameters include *In dt* - sample features; *owl*- defect ontology; *lr* - learning rate; *ep* - total number of epochs; *erly stp* - early stop epochs; *rd sd* - random seed; Th_0 - ontology exploration rate; and m - the iteration number. The features $\boldsymbol{\theta}(x)$ are extracted in line 4. The ontology ($\boldsymbol{\varphi}^A$) for class embedding is initialized in line 5, and the mapping matrix W is randomly initialized in line 6. An iterative search process begins at line 10. Each iteration implements the two-stage search, where lines 8-16 implement the stochastic gradient descent in stage (1) to update the mapping matrix $\mathbf{W}$ from training data, and lines 17-29 search for the tuning of ontology and class embedding in stage (2). In stage (1), the stochastic gradient descent draws sample ($\mathbf{x}_n$, y_n) at each step t and searches for the highest compatible class $y = y_n$ as shown in line 11, where $\eta(t)$ is the learning step size. The trained $\mathbf{W}$ is applied to the data for tuning ontology to estimate classification accuracy in lines 14.

In stage (2), this accuracy is compared with an ontology exploration rate Th_0 to determine if the algorithm continues ontology structure exploration (lines 25-26) based on a random walk and NLP or refine class embedding (lines 23). To extract information from ontology, a random walk is performed iteratively through different branches of the ontology (line 25). At iteration m, a function *randomWalk* is executed to perform this task based on section 2.1.2 and Table:1 gene formulation. The iterative step includes three functions:

- *OntologyWalk*($\boldsymbol{\varphi}^{(m)}$) → $Gene^{(m)}$: Depending on the performance of the previous ontology structure, an ontology walk is performed across the branches to extract new information.

- *Mutation*($Gene^{(m)}$) → $Gene^{(m+1)}$: A new set of attributes are generated by mutating local gene segment, e.g., 16 mutated to 17. Each gene represents a unique set of paths on the walk.

- *Decoding*($Gene^{(m+1)}$) → $\boldsymbol{\varphi}^{(m+1)}$: The newly generated genes (representing a new path) are then converted (decoded) to sentences with attributes.

In each iteration, the best-discovered ontology and embedding will be updated (lines 19-21). The iteration will continue when certain convergence criteria are not satisfied (line 10), such as ZSL accuracy no longer improves or the cut-off iterations have been reached. The algorithm outputs the mapping matrix $\mathbf{W}^*$, tuned ontology $\boldsymbol{\varphi}^*$, and class embedding $\boldsymbol{\phi}^*$ for future ZSL implementation.

3 *Case study*

This section presents a case study to demonstrate the proposed ontology-guided attribute learning in helping identify defects that are unobserved in training data.

3.1 *A direct-ink-writing process for creating nanocomposites*

The case study focuses on a printing process for creating a new nanowire-polymer composite structure (Shan et al. (2019)) as photoactive coatings that detect visible light. Supplementary S7 presents the details of the printing process. For the simplicity of illustration, this research only focuses on ten defect types of cracks and pores in different morphologies as captured by microscopic images from different data sources. An example of these defects is given in Figure 7. The samples for classes 1,3, 7, and 9 were obtained from a printing process of functional composites (Shan et al. (2019)), while the remaining classes 2, 4, 5, 6, 8, and 10 were obtained from other AM processes, such as selective laser melting (SLM). It should be noted that although samples obtained from SLM are very different, certain defect morphologies still share some common attributes with the direct-ink-writing process.

Algorithm 1 Zero-shot Learning algorithm with adaptive class embedding optimization

procedure ZSL(In_dt, owl, lr, erly_stp, ep, rd_sd, Th_0, m)
  In: In_dt, owl, lr, erly_stp, ep, rd_sd, Th_0, m
  out: Zero-shot class labels; optimum class embedding
  $\boldsymbol{\theta}(\mathbf{x}) \leftarrow loadFeatures(In\_dt)$; $\boldsymbol{\varphi}^{\Lambda} \leftarrow loadClassEmbed(owl)$
  $\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi}^{\Lambda}$; $\boldsymbol{\phi}^{(m)} \leftarrow BERT(\boldsymbol{\varphi})$ # Initialization
  $\mathbf{W} \leftarrow RandInitialize(rd\_sd)$; τ - training data (images vs. class label)
  while $Termination\_criteria$ is not met **do**
    while $t \leq ep$ **do**
      Draw $(\mathbf{x}_n, y_n) \in \tau$ and $y \neq y_n$
      if $F(\mathbf{x}_n, y) + 1 > F(\mathbf{x}_n, y_n)$ **then**
        $\mathbf{W}^{(t+1)} \leftarrow \mathbf{W}^{(t)} - \eta_{(t)} \mathbf{X}'_n (\boldsymbol{\phi}(y) - \boldsymbol{\phi}^{(m)}(y_n))$
      end if
      Return W
      $Acc^m \leftarrow$zsl_acc($\mathbf{x}\_test_n, \mathbf{W}, test\_labels$)
      $t \leftarrow t + 1$
    end while
    if $Acc^{(m)} > Th_0$ **then**
      if $Acc^{(m)} > Acc^{(m-1)}$ **then**
        $\boldsymbol{\varphi}^{*} \leftarrow \boldsymbol{\varphi}^{(m)}$ # update the best ontology discovered $\boldsymbol{\varphi}^{*}$
        $\boldsymbol{\phi}^{*} \leftarrow \boldsymbol{\phi}^{(m)}$ # update the best class embedding discovered $\boldsymbol{\phi}^{*}$
        $\mathbf{W}^{*} \leftarrow \mathbf{W}$ # Update the best discovered $\mathbf{W}^{*}$
      end if
      $\boldsymbol{\phi}^{(m+1)} \leftarrow \boldsymbol{I}^{(m)} \boldsymbol{\phi}^{(m)}$
    else
      $\boldsymbol{\varphi}^{(m+1)} \leftarrow RandomWalk(\boldsymbol{\varphi})$
      $\boldsymbol{\phi}^{(m+1)} \leftarrow BERT(\boldsymbol{\varphi}^{(m+1)})$
    end if
    $m \leftarrow m + 1$
  end while
  Return $\mathbf{W}^{*}$, $\boldsymbol{\phi}^{*}$ and $\boldsymbol{\varphi}^{*}$

$\boldsymbol{I}^{(m)} \leftarrow \boldsymbol{I}$
K = Acc - RandomUniform(0, 1)
If $K < 0$, #Exploration on $\boldsymbol{\phi}^{(m)}$
  $\boldsymbol{I}^{(m+1)} \leftarrow$ RandomSwap($\boldsymbol{I}^{(m)}$)
If $K > 0$, #Exploitation on $\boldsymbol{\phi}^{(m)}$
  $\boldsymbol{I}^{(m+1)} \leftarrow$ TwoRowSwap($\boldsymbol{I}^{(m)}$)

The idea can be analogous to the learning of a certain species by leveraging the attributes of different types of animals. As such, this study also explores the opportunity of attribute learning in leveraging information from more different printing processes as reported in published data and literature. The types of printing defects considered in this study are summarized in Supplementary S4.

Remark on between-process attribute sharing: Certain attributes are unique to each AM process or its family. When AM processes are intrinsically distinct, sharing attributes based on process parameters and their impacts on defects may not be feasible. In ontology, attributes related to causes and materials can be unique to each type of AM process and can only be shared within similar or closely related AM processes. Attributes related to morphology, on the other hand, are more transferable across different processes. For example, in SLM and other directed energy AM processes, cracks are caused by localized thermal stress, whereas in direct inc-writing processes, cracks are caused by tiny pores and rapid solidification during extrusion. The causes are mainly related to the physics of each process, and the process-defect relationship is not transferable. However, all cracks have common attributes of zig-zag shapes and long aspect ratios that can be shared across different processes. Similarly, pores caused by trapped gases during extrusion or material preparation that exhibit roundish morphologies on a surface can be shared across processes.

3.2 *Defect data preparation and embedding*

The dataset includes ten defect classes with five samples per class. All defect data, along with their labels, were collected and categorized into training and testing datasets. Some or all samples in the testing data are not present in the training data, making them unseen. The training data only includes seen classes that share common attributes and class embedding with unseen classes. For data with unseen classes, one part is reserved for fine-tuning the ontology. The rest is for testing the performance. The algorithm learns to map the features extracted from training samples to their respective attribute-based embedding. Shared attributes between seen and unseen classes exist, allowing the algorithm to extrapolate information on unseen classes based on their class embedding during training. The algorithm focuses on the attributes of the shared attributes of seen classes. The algorithm is then tested to determine the label of the unseen classes, and the prediction is compared against the actual label. Traditional classification metrics such as accuracy, F1 score, and confusion matrix are still applicable to evaluate the algorithm's performance in determining the unseen class label.

The case study uses different percentages of available classes to define classes unseen in the training phase, including 10%, 20%, 30%, 40%, and 50%. Figure 7 shows an example

of seen vs. unseen class split, by which the ZSL classifier will be trained on 70% (7 seen classes) and tested for 30% of available classes (3 unseen classes on the right column). Each class has its unique property in shape, texture, or color. Sharing attributes between seen and unseen classes is vital for ensuring the good performance of ZSL. Below lists some examples of shared features:

- Irregular pore-1 shares the attribute "grey colored surface" with Large pore-2 and the attribute "irregular shaped pores" with Irregular pore-2.
- Large pore-1 shares the attribute "reddish surface with small red dots" with Tiny crack-1 and the attribute "large circular pores" with Large pore-2.

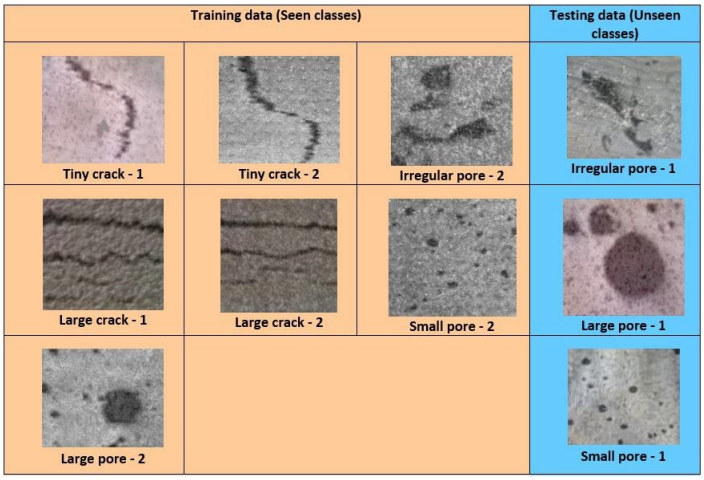

Figure 7: An example of split for seen classes and unseen classes in the dataset.

In the testing/implementation phase, the defect classes can include seen and/or unseen classes. This study will focus on two scenarios : (1) all classes in the testing data are unseen to demonstrate the potential of the ontology-guided attribute learning, and (2) a mixture of seen and unseen classes in the testing data that may lead to lower classification accuracy due to more candidate classes included.

Feature extraction was performed to obtain $\boldsymbol{\theta}(\mathbf{x})$ from defect images. This study uses Img2Vec python library for image embedding based on a ResNet50 model and its weights pre-trained on the ImageNet dataset (Deng et al. (2009)). Supplementary S5 outlines the details of the pre-trained network used for feature extraction and why its suitable for the proposed methodology.

This study also employed the ”BERT BASE Uncased” variant of the BERT model to convert sentences generated from multiple walks into contextualized embedding vectors ($\boldsymbol{\phi}$) of size 768. The tokenization process was performed using the open-source python library ”TransformerWordEmbeddings.” The BERT BASE model used in the study is the smaller variant of the BERT model, designed for lowercase sentences. The ”BERT BASE Uncased” variant consists of 12 transformer blocks, each containing a multi-head self-attention mechanism, 768 hidden layers, and 12 attention heads. To embed defect classes, the model takes a sentence as input, tokenizes it into individual words, and generates a vector embedding for each token. The embedding of each defect class token captures the meaning and context of the sentence that describes the class.

3.3 *Results and discussion*

One outcome of this study is to build an ontology of defects that promotes the sharing of defect attributes. Since not all features are equally relevant to learning the classifier, the algorithm should learn to re-structure the defect ontology and fine-tune the representation of attributes shared between seen and unseen classes. This section presents the results on (1) the optimized tuning of the defect ontology structure, (2) the ZSL accuracy of the defect classification achieved by ontology-guided attribute learning, (3) the analysis of how ontology exploration rate in the optimization impacts testing accuracy, (4) the robustness of the methodology against noises in the data, and (5) comparative study against baseline transfer learning methodologies.

3.3.1 *Optimized tuning of the defect ontology structure*

This section presents the optimized tuning of the ontology that can facilitate the sharing of attributes between seen and unseen classes and its effect on defect classification performance. Figure 8 illustrates an example of optimized tuning of a 2D ontology structure in a tabular format when certifying processes with ten possible defect classes (assuming six seen classes in the training and four unseen classes in the blue cells with bold text present in the testing data). A light yellow cell with “x” label indicates that a particular attribute is selected for the corresponding defect class in the ontology configuration, and a grey cell means that the corresponding candidate attribute is not selected.

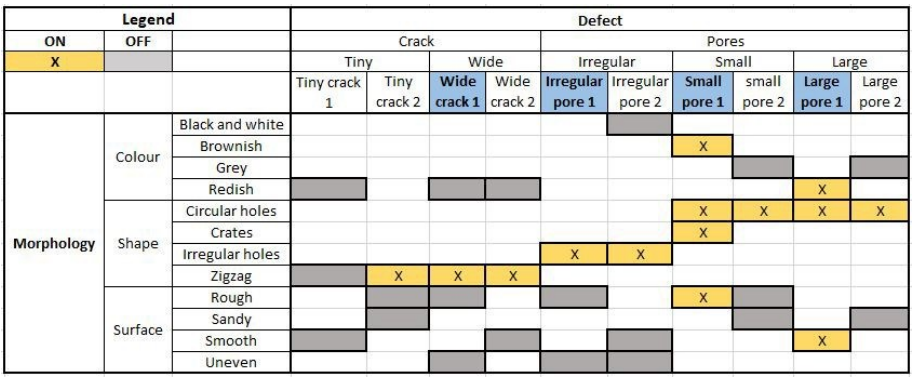

Figure 8: An example of the optimized ontology through attribute selection.

Figure 8 illustrates how the morphological information in the ontology enables knowledge sharing and transferability between seen and unseen classes. For example, even though Wide crack-1 is an unseen class, the attribute learner extrapolates how it might appear because it shares "zigzag-shaped" morphology with seen classes Tiny crack-2 and Wide crack-2. Similarly, Irregular pore-1 is unseen but shares an "irregular pore" attribute with Irregular pore-2. In addition, both Small pore-1 and Large pore-1 are unseen classes, each with a unique set of attributes. They both share an attribute "circular holes," which is the shape of the defect with the seen class Small pore-2. The tuned ontology in Figure 8 is a key enabler in performing ZSL of unseen defects by generating appropriate class embeddings for seen and unseen classes.

3.3.2 *Application of the ontology: Zero-shot defect classification*

The ontology-guided attribute classifiers are trained and tested under different hypothetical scenarios with different splits between seen vs. unseen classes. Results in Table 2 (column 3) show that the algorithm successfully identified defects with unseen classes only. The methodology was tested for a range of scenarios from one unseen class to five unseen classes out of ten. It can be seen that the ZSL based on the tuned defect ontology achieved an accuracy of 100% when one unseen class is mixed with seen classes in the testing data (column 4). The accuracy decreased as more unseen classes are included for both scenarios. It should be noted that the inclusion of more classes in the training data may potentially contribute to the defect ontology, thus improving the learning accuracy. The case study

includes ten classes in total, limiting the accuracy when the percentage of unseen classes increases in the testing data. However, an accuracy of 0.66% (worst result) is still far better than a random selection of one class out of ten, even if 50% of the classes are unseen.

Table 2: Comparison of our model to existing ZSL methods

No. of seen classes	No. of un seen classes	Proposed method accuracy for unseen classes	Proposed method accuracy mixed seen and unseen classes	ALE accuracy for unseen classes only	ALE Accuracy for mixed seen and unseen classes
9	1	**1**	**1**	**1**	**0.88**
8	2	**1**	**0.9**	**0.5**	**0.75**
7	3	**0.87**	**0.86**	**0.33**	**0.5**
6	4	**0.8**	**0.76**	**0.25**	**0.66**
5	5	**0.64**	**0.66**	**0.4**	**0.52**

In Table 2 (columns 5 and 6), the ontology-guided attribute learning was compared against a recent ZSL method called ALE Akata et al. (2015) in computer vision. It performs image classification based on the classes embedded by contextualized language processing models. In a ZSL situation, the ALE outperforms the conventional Direct Attribute Prediction Lampert et al. (2009) baseline approaches on the Animals With Attributes Lampert et al. (2009), and Caltech-UCSD-Birds Wah et al. (2011) datasets. The results in Table 2 show that the proposed ontology-guided attribute learning outperforms ALE in both testing scenarios, demonstrating the importance of class embedding of defects. When the defect classes are embedded by applying general language models to the text dataset in ALE, the ZSL learning accuracy achieved by ALE can sometimes be as worse as a random experiment (flat accuracy across all classes). With the class embedding improved by the proposed ontology-guided optimization, the performance is significantly improved.

Figure 9 shows a confusion matrix for the true and predicted labels when only three unseen classes are present in the testing data, and Figure 20 in Supplementary S9 illustrates the scenario when seen and unseen classes are mixed in the testing data. Both achieve an accuracy above 0.72. The three unseen class ZSL task only involves the classification of three unseen classes, while the mixture of seen and unseen classes includes all ten classes in the testing dataset. Therefore, the classification in the unseen-only scenario is less challenging than that in the mixed-class scenario. The lower accuracy obtained in the mixed class scenario is expected because the algorithm is presented with images from all ten classes, making the decision harder. However, this study does not intend to compare the results

between seen-only classes.

The results demonstrate the potential of the proposed ontology-guided attribute learning in accelerating process certification based on very few measurements from a new process of interest by leveraging the data from different processes and literature. This study is among the first attempts to deal with zero observations/measurements of the defects from the target process of interest. This learning challenge is beyond the capability of popular classification algorithms, transfer learning, or semi-supervised learning approaches that still anticipate measurements in the training data from the target process.

		Predicted label		
		Irregular pore -1	Large pore -1	Small pore -1
True label	Irregular pore -1	3	0	2
	Large pore -1	0	5	0
	Small pore -1	0	0	5

Class	Total number of samples	Samples classsfied to the class	Accuracy	Precision	Recall	F1 Score
Irregular pore -1	5	3	86.67%	1	0.6	0.75
Large pore -1	5	5	100%	1	1	1
Small pore -1	5	7	86.67	0.71	1	0.83

Figure 9: Confusion matrix, accuracy, precision, recall, and F1 score of ZSL to classify three unseen classes only

3.3.3 *Effect of ontology exploration rate*

The ontology exploration rate $0 \leq Th_0 \leq 1$ in the two-stage optimization can determine how likely the ontology would be re-structured to import new information. Compared to the tuning of class embedding, the ontology update implies a more drastic change in searching for new information. This section discusses the effect of the ontology exploration rate on the ZSL accuracy so that an appropriate range of Th_0 can be determined. All other parameters, such as epochs and cut-off values for terminating the iterations, were kept constant at the optimal level previously discovered.

A high exploration rate means a high frequency of ontology re-structuring to generate very different embeddings, thus searching along more directions to seek optimal solutions. By contrast, the optimization on a lower exploration rate relies more on tuning class embedding and initial ontology. Figure 10 shows the 95% confidence interval (CI) of average accuracy estimated from multiple samples for each Th_0. The results indicate that a lower rate of ontology exploration (0.1-0.4) resulted in lower average accuracy. As Th_0 increases,

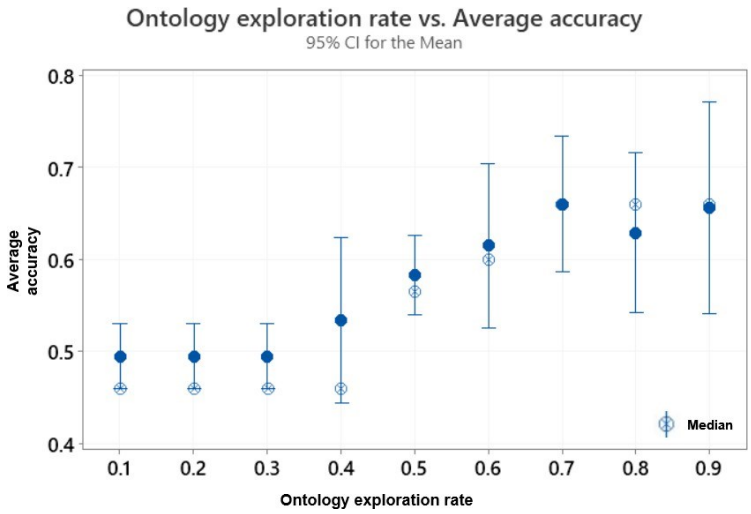

Figure 10: The effect of ontology exploration rate on the performance of the classifier.

the average accuracy increases; however, it does not exhibit a statistical difference when $Th_0 \geq 0.5$. The median accuracy is relatively close when $Th_0 \geq 0.7$. It is also noticed that the highest Th_0 may also lead to larger variability in the average accuracy. Therefore, a value around 0.7 is recommended for Th_0 in this case. In practice, the selection of ontology exploration can be affected by the selection of the initial ontology. An initial ontology capturing more common attributes between seen and unseen classes may be expected to work better with a lower value of Th_0. As such, a mid-range value for Th_0 is recommended considering the balance of all possible scenarios.

3.3.4 *Robustness against data noises*

A robustness study was conducted by adding noises at different levels to existing images. Gaussian white noise variance ranging up to 1.2 was added to every image. Figure 11 (lower panels) shows an example of a noise-free reference image against four levels of added noises. The added noises can reduce any common features among classes and images. Also, added noises reduce visibility by removing fine details and making images blurry.

The noisy data include seven training classes and three unseen testing classes and each run was repeated ten times. Boxplot in Figure 11 shows the interquartile range (IQR) and the median of classifier accuracy at each noise level for testing only unseen classes (upper left panel) and a mix of seen and unseen classes (upper right panel). In general, the accuracy of

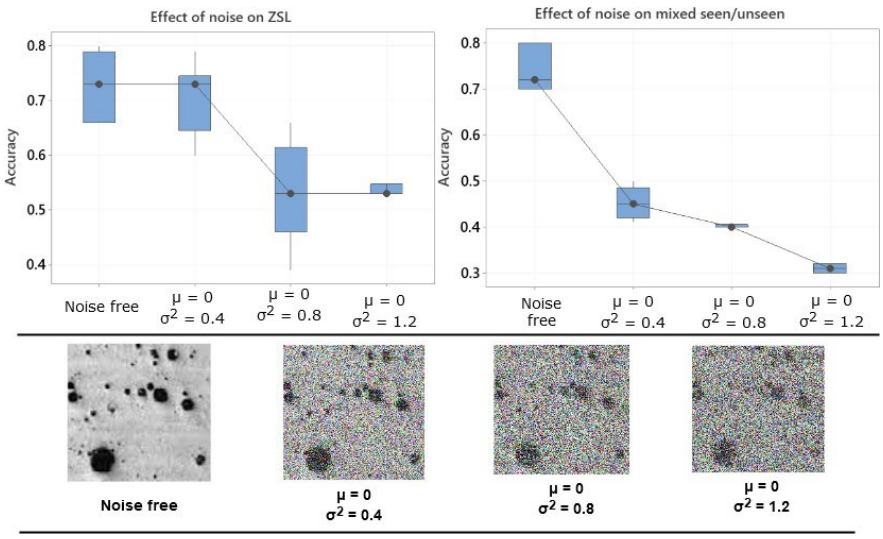

Figure 11: Effect of noise on accuracy for unseen classes only and mixed seen/unseen classes.

the proposed methodology decreases as the images become increasingly blurred. We notice that the algorithm's performance for mixed classes of seen and unseen classes exhibits a rapid decline in performance due to the bias toward the seen classes during training. This pattern occurred because the number of classes to be classified could significantly affect the accuracy consistency as the noise level increases. Thus, the accuracy in the classification of ten seen-unseen classes shows low variability (very narrow IQR boxplot), indicating that the performance is becoming consistently worse. By contrast, the accuracy in the classification of three unseen classes exhibits larger variability (wide IQR).

3.3.5 *Comparative study against transfer learning*

The proposed methodology was also compared against three benchmark methods under a transfer learning framework. It should be noted that traditional transfer learning is incapable of dealing with ZSL. Thus, this comparison was made for one-shot and two-shot learning, i.e., classes with only one or two samples observed in training data. In this study, three popular pre-trained CNN's for image classification were employed, including AlexNet, VGG, and ResNet50. Supplementary S6 outlines details of the transfer learning networks.

To implement transfer learning, a pre-trained CNN network was downloaded from an open-source repository (Falbel (2022)). Then, the last fully-connected and softmax layers

were replaced with a new custom fully connected layer containing ten output neurons. During training, the pre-trained weights were frozen and not updated, but the added final layer was trained using the AM defect data. In this research, transfer learning techniques such as AlexNet, ResNet50, and VGG were compared with the proposed method for one-shot and two-shot learning scenarios where only one or two samples of a class were present in the training data. The box plots of accuracy for the one-shot learning tests are shown in the left panel of Figure 12, and the results of two-shot learning experiments are reported in the right panel. For both cases, the proposed method statistically outperformed the other transfer learning methods, with ResNet being the worst.

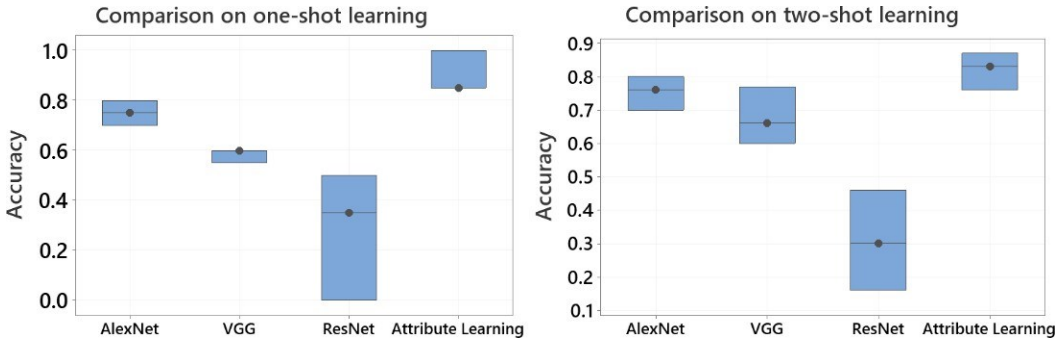

Figure 12: Comparison of attribute learning against three transfer learning methods during one-shot (left panel) and two-shot (right panel) experiments

4 Conclusion

The scarcity of measurement data during new printing process development may prevent scale-up production or commercialization within a short time. This study aims to address the challenge of limited measurements and experiments to be conducted for new printing process development. By leveraging published data and literature, this research establishes an ontology-guided attribute learning methodology to identify defects that are completely unobserved in training data. The method tackles the zero-shot learning (ZSL) challenge by mimicking how humans learn a new concept by recognizing attribute combinations. The proposed algorithm learns the attributes of a defect based on developing/learning a defect ontology and how attributes are shared across seen and unseen classes.

The proposed attribute learning first establishes an initial defect ontology to facilitate

uniform knowledge representation and attribute sharing, representing the morphology, materials, and process parameters related to defects. This procedure is guided by engineering knowledge from published data and literature and facilitated by an annotation tool, *owl*, to build a defect ontology. An algorithm for exploring the ontology structure is developed and combined with a pre-trained BERT, thus parsing the ontology into defect class embedding (attributes). By compatibility mapping, the attributes are correlated to the features extracted from input data, such as microscopic images by Img2vec embedding. In the training phase, attribute learning involves a joint decision-making problem by maximizing the compatibility mapping between the input features and defect attribute combinations extracted from the defect ontology. The compatibility mapping is found by stochastic gradient descent in stage (1) optimization. In stage (2), a metaheuristics was developed to explore and exploit ontology structures and class embedding. The testing phase finds the most compatible classes with testing images based on the trained mapping and class embedding.

A case study is conducted on classifying defects for a direct-ink-writing of nanocomposites. A simplified defect ontology based on morphologies was established and tuned to facilitate attribute extraction for attribute learning. The shared attributes between seen and unseen classes of defects resulted in better classification performance when identifying unseen defect classes. The results show an accuracy of 0.76, 0.86, and 0.9 achieved when testing four, three, and two unseen classes, respectively, when both seen and unseen testing samples are mixed. The study also discusses the relationship between the ontology exploration rate and testing accuracy to help identify an appropriate value for it. The results from the case study show that accuracy is low at lower exploration rates. A steady increase in performance has been recorded as the ontology exploration rate increases. This increase in performance is because higher rates result in a frequent change of attribute combinations, which increases the probability of finding the most optimal one. Furthermore, analyses were also carried out to investigate the robustness of the method by adding noises of different levels to every image in the dataset.

The proposed methodology's accuracy decreases with increasingly blurred images. Performance decline is observed in the algorithm when dealing with mixed classes of seen and unseen classes compared to tests containing only unseen classes. This decline is due to bias toward seen classes during training and the impact of increased class numbers on accuracy

consistency as noise levels increase.

Future research directions include: (1) improving and refining defect ontology by incorporating more process parameters and their value ranges, such as printing speed and temperature, to facilitate the discovery of process-defect relationships; (2) Using ontology-guided attribute learning to establish process windows for new process qualification given limited measurements; (3) developing a pipeline of algorithms based on a search engine, adaptively scraping published data online by topic modeling, such as latent Dirichlet allocation (LDA), to extract most relevant attributes for class embedding in ZSL; (4) leveraging the emerging Generative Pre-trained Transformers (GPT) or other large language models to improve the attribute embedding based on their models pre-trained from massive literature data (refer to Supplementary S10); and (5) developing a self-supervised learning framework for ZSL where defect knowledge is organized under ontology and labels are generated without human supervision before ontology fine-tuning and attribute learning.

5 Data availability statement

The data and code used in this study is here (https://github.com/FAMU-FSU-IME/Ontology-guided-Attribute-Learning.git).

References

Akata, Z., F. Perronnin, Z. Harchaoui, and C. Schmid (2015). Label-embedding for image classification. *IEEE transactions on pattern analysis and machine intelligence 38* (7), 1425–1438.

Bappy, M. M., C. Liu, L. Bian, and W. Tian (2022). Morphological dynamics-based anomaly detection towards in situ layer-wise certification for directed energy deposition processes. *Journal of Manufacturing Science and Engineering 144* (11), 111007.

Bartlett, J. L., F. M. Heim, Y. V. Murty, and X. Li (2018). In situ defect detection in selective laser melting via full-field infrared thermography. *Additive Manufacturing 24*, 595–605.

Chen, Y., X. Peng, L. Kong, G. Dong, A. Remani, and R. Leach (2021). Defect inspection technologies for additive manufacturing. *International Journal of Extreme Manufacturing 3* (2), 022002.

Cheng, L., F. Tsung, and A. Wang (2017). A statistical transfer learning perspective for modeling shape deviations in additive manufacturing. *IEEE Robotics and Automation Letters 2* (4), 1988–1993.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Du, W., Q. Bai, Y. Wang, and B. Zhang (2018). Eddy current detection of subsurface defects for additive/subtractive hybrid manufacturing. *The International Journal of Advanced Manufacturing Technology 95*(9), 3185–3195.

Esfahani, M. N., M. M. Bappy, L. Bian, and W. Tian (2022). In-situ layer-wise certification for direct laser deposition processes based on thermal image series analysis. *Journal of Manufacturing Processes 75*, 895–902.

Falbel, D. (2022). *torchvision: Models, Datasets and Transformations for Images*. https://torchvision.mlverse.org, https://github.com/mlverse/torchvision.

He, K., X. Zhang, S. Ren, and J. Sun (2016, June). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jin, Y., S. Joe Qin, and Q. Huang (2016). Offline predictive control of out-of-plane shape deformation for additive manufacturing. *Journal of Manufacturing Science and Engineering 138*(12).

Khanzadeh, M., S. Chowdhury, M. A. Tschopp, H. R. Doude, M. Marufuzzaman, and L. Bian (2019). In-situ monitoring of melt pool images for porosity prediction in directed energy deposition processes. *IISE Transactions 51*(5), 437–455.

Lampert, C. H., H. Nickisch, and S. Harmeling (2009). Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 951–958. IEEE.

Lampert, C. H., H. Nickisch, and S. Harmeling (2013). Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence 36*(3), 453–465.

Luan, H. and Q. Huang (2016). Prescriptive modeling and compensation of in-plane shape deformation for 3-d printed freeform products. *IEEE Transactions on Automation Science and Engineering 14*(1), 73–82.

Millon, C., A. Vanhoye, A.-F. Obaton, and J.-D. Penot (2018). Development of laser ultrasonics inspection for online monitoring of additive manufacturing. *Welding in the World 62*(3), 653–661.

Pandiyan, V., R. Drissi-Daoudi, S. Shevchik, G. Masinelli, T. Le-Quang, R. Logé, and K. Wasmer (2022). Deep transfer learning of additive manufacturing mechanisms across materials in metal-based laser powder bed fusion process. *Journal of Materials Processing Technology 303*, 117531.

Sabbaghi, A. and Q. Huang (2016). Predictive model building across different process conditions and shapes in 3d printing. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 774–779. IEEE.

Sabbaghi, A., Q. Huang, et al. (2018). Model transfer across additive manufacturing processes via mean effect equivalence of lurking variables. *Annals of Applied Statistics 12* (4), 2409–2429.

Scime, L. and J. Beuth (2018). A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process. *Additive Manufacturing 24*, 273–286.

Seifi, S. H., W. Tian, H. Doude, M. A. Tschopp, and L. Bian (2019, 06). Layer-Wise Modeling and Anomaly Detection for Laser-Based Additive Manufacturing. *Journal of Manufacturing Science and Engineering 141*(8). 081013.

Shan, X., P. Mao, H. Li, T. Geske, D. Bahadur, Y. Xin, S. Ramakrishnan, and Z. Yu (2019). 3d-printed photoactive semiconducting nanowire–polymer composites for light sensors. *ACS Applied Nano Materials 3*(2), 969–976.

Vora, H. D. and S. Sanyal (2020, 12). A comprehensive review: metrology in additive manufacturing and 3d printing technology. *Progress in Additive Manufacturing 5*, 319–353.

Wah, C., S. Branson, P. Perona, and S. Belongie (2011). Multiclass recognition and part localization with humans in the loop. In *2011 International Conference on Computer Vision*, pp. 2524–2531. IEEE.

Wang, A., S. Song, Q. Huang, and F. Tsung (2016). In-plane shape-deviation modeling and compensation for fused deposition modeling processes. *IEEE Transactions on Automation Science and Engineering 14*(2), 968–976.

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Wu, H., Z. Yu, and Y. Wang (2016). A new approach for online monitoring of additive manufacturing based on acoustic emission. In *International Manufacturing Science and Engineering Conference*, Volume 49910, pp. V003T08A013. American Society of Mechanical Engineers.

Xian, Y., Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele (2016). Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 69–77.

Xian, Y., B. Schiele, and Z. Akata (2017). Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4582–4591.

Ye, Z., C. Liu, W. Tian, and C. Kan (2021). In-situ point cloud fusion for layer-wise monitoring of additive manufacturing. *Journal of Manufacturing Systems 61*, 210–222.