



**ANNUAL**  
CONFERENCE & EXPO 2026

## **Risk-Aware Decision Making for Metal Additive Manufacturing via Conditional Diffusion Models**

**Presenter:** Donghyun Ko\*, Jacob Peloquin, PhD\*

\* Edward P. Fitts Department of Industrial and Systems Engineering, NC State University, Raleigh, NC, USA

**Keywords:** Additive manufacturing (AM); Laser powder bed fusion (LPBF); Conditional diffusion models (CDM); CNN-based defect prediction; Real-time decision support; Peregrine dataset

[iise.org/annual](https://iise.org/annual)

[#IISEANNUAL2026](https://twitter.com/IISEANNUAL2026)

# Motivation & Background

Laser Powder Bed Fusion (**LPBF**) is a leading technology for metal additive manufacturing, enabling complex geometries and high-performance components.



## 1. Why LPBF?

- Enables complex geometries
- High material efficiency
- Produces high-performance metal components



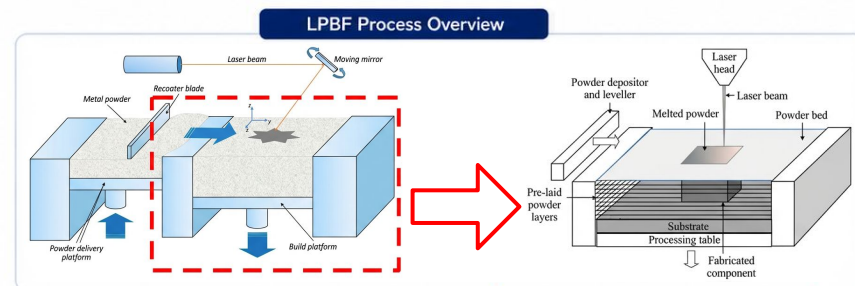
## 2. Key Challenge

- High-dimensional process parameters (e.g., laser power, scan speed, hatch spacing, etc.)
- Expensive and time-consuming to evaluate all possibilities

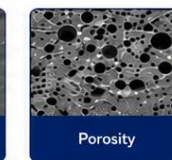


## 3. Quality Variability

- Defects: Recoater issues, incomplete spreading, swelling, debris, porosity, etc.
- Consequences: Part rejection, rework



## Example Defects in LPBF Builds

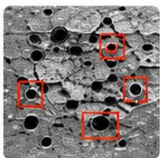


## Example LPBF Products



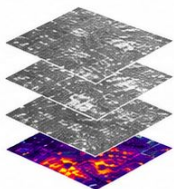
# Research Gap & Proposing a framework

## 1 What prior studies focused on



### CNN-based defect detection

Existing LPBF studies focus on defect detection from observed layer images.



### Generative AI in AM

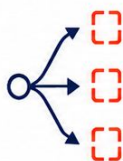
Generative AI in AM has mostly focused on generating realistic layer images and evaluating whether the generated layers preserve real structure, texture, and other characteristics.

## 2 Research gap



### Lack of decision-oriented integration

Existing studies rarely connect layer forecasting and defect evaluation in a single real-time pipeline for manufacturing decisions.



### What we're proposing

A closed-loop framework that (1) forecasts plausible future layers, (2) evaluates defect risk from those predicted layers, and (3) supports real-time defect detection and process control on the production line.

## 3 Our work



### ORNL Peregrine dataset

Multimodal, co-registered LPBF digital twin data



### Conditional print forecasting

Predict multiple plausible next-layer realizations

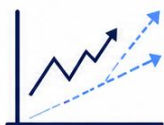


### CNN-based, real-time manufacturing decisions

Enable action during real-time operation



## Framework objective



- 1 Replace costly physical experimentation with probabilistic **print forecasting**.



- 2 Enable distribution-aware defect estimation.

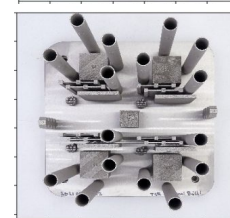
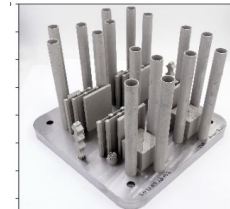
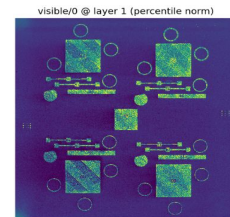


- 3 Support real-time defect detection and **process control**.

# Dataset

- ❖ This work uses the Peregrine digital twin dataset from Oak Ridge National Laboratory (DOI: <https://doi.org/10.13139/ORNLNCCS/2008021>)
  - Consist of 5 builds in that all modalities are **spatially** (“pixel-wise”) and **temporally** (“layer-wise”) aligned;
    - Post-melt images
    - Post-recoat images
    - Part ID map
    - Anomaly segmentation mask (12)
    - Laser scan path
    - Temporal in-build signals
    - Part-level process parameters
- ❖ Build 1
  - Co-registered *in-situ* and *ex-situ* dataset
  - Total # of layers = 3575, Total # of parts = 39, Printer: Concept Laser M2
  - Material: SS316L (TruForm 316-3, Praxair, Batch 30)
  - High-level tree structure: parts, reference images, samples, scans, slices, temporal

Build ID	Printer	AM Process	Material	# of Parts	# of Samples	# of Layers
Build 1	Concept Laser M2 (ORNL1)	L-PBF	SS316L (TruForm 316-3, Lot 30)	39	3,296	3,575
Build 2	EOS M290	L-PBF	SS316L	30	~1,500	~3,000
Build 3	AddUp FormUp 350 (dual-laser)	L-PBF	SS316L	25	~1,200	~2,800
Build 4	ExOne M-Flex	Binder Jetting (BJ)	SS316L	20	~800	n/a (no layerwise imaging)
Build 5	Concept Laser M2 (variant)	L-PBF	SS316L or 17-4PH	35	~1,000	~3,000



# Dataset (Cont')

```
(root)
├── /parts
│   ├── /reference_images
│   ├── /samples
│   ├── /scans
│   ├── /slices
│   └── /temporal
├── parts/ (group, children=2)
│   ├── process_parameters/ (group, children=8)
│   │   ├── hatch_spacing (dataset, shape=(40), dtype=float32)
│   │   ├── laser_beam_power (dataset, shape=(40), dtype=float32)
│   │   ├── laser_beam_speed (dataset, shape=(40), dtype=float32)
│   │   ├── laser_module (dataset, shape=(40), dtype=float32)
│   │   ├── laser_spot_size (dataset, shape=(40), dtype=float32)
│   │   ├── parameter_set (dataset, shape=(40), dtype=uint8)
│   │   ├── scan_rotation (dataset, shape=(40), dtype=float32)
│   │   └── stripe_width (dataset, shape=(40), dtype=float32)
│   └── test_results/ (group, children=0)
├── reference_images/ (group, children=6)
│   ├── DSC04467A (dataset, shape=(4080, 3045, 3), dtype=uint8)
│   ├── DSC04468A (dataset, shape=(3765, 3499, 3), dtype=uint8)
│   ├── DSC04469A (dataset, shape=(4276, 3550, 3), dtype=uint8)
│   ├── DSC04470A (dataset, shape=(3512, 3568, 3), dtype=uint8)
│   ├── DSC04471A (dataset, shape=(3580, 3575, 3), dtype=uint8)
│   └── thumbnail (dataset, shape=(1842, 1842), dtype=uint8)
├── samples/ (group, children=1)
│   └── test_results/ (group, children=4)
│       ├── total_elongation (dataset, shape=(3297,), dtype=float32)
│       ├── ultimate_tensile_strength (dataset, shape=(3297,), dtype=float32)
│       ├── uniform_elongation (dataset, shape=(3297,), dtype=float32)
│       └── yield_strength (dataset, shape=(3297,), dtype=float32)
├── scans/ (group, children=3575)
│   ├── 0 (dataset, shape=(512127, 5), dtype=float32)
│   ├── 1 (dataset, shape=(606111, 5), dtype=float32)
│   ├── ...
│   └── 3575 (dataset, shape=(606111, 5), dtype=float32)
```

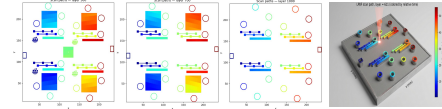
```
├── slices/ (group, children=4)
│   ├── camera_data/ (group, children=1)
│   │   ├── visible/ (group, children=2)
│   │   │   ├── 0 (dataset, shape=(3575, 1842, 1842), dtype=float32)
│   │   │   └── 1 (dataset, shape=(3575, 1842, 1842), dtype=float32)
│   │   ├── part_ids (dataset, shape=(3575, 1842, 1842), dtype=uint32)
│   │   ├── sample_ids (dataset, shape=(3575, 1842, 1842), dtype=uint32)
│   │   └── segmentation_results/ (group, children=12)
│   │       ├── 0 (dataset, shape=(3575, 1842, 1842), dtype=bool)
│   │       ├── 1 (dataset, shape=(3575, 1842, 1842), dtype=bool)
│   │       ├── 10 (dataset, shape=(3575, 1842, 1842), dtype=bool)
│   │       ├── 11 (dataset, shape=(3575, 1842, 1842), dtype=bool)
│   │       ├── 2 (dataset, shape=(3575, 1842, 1842), dtype=bool)
│   │       ├── 3 (dataset, shape=(3575, 1842, 1842), dtype=bool)
│   │       ├── 4 (dataset, shape=(3575, 1842, 1842), dtype=bool)
│   │       ├── 5 (dataset, shape=(3575, 1842, 1842), dtype=bool)
│   │       ├── 6 (dataset, shape=(3575, 1842, 1842), dtype=bool)
│   │       ├── 7 (dataset, shape=(3575, 1842, 1842), dtype=bool)
│   │       ├── 8 (dataset, shape=(3575, 1842, 1842), dtype=bool)
│   │       └── 9 (dataset, shape=(3575, 1842, 1842), dtype=bool)
│   └── temporal/ (group, children=19)
│       ├── absolute_image_capture_timestamp (dataset, shape=(3575,), dtype=float32)
│       ├── actual_ventilator_low_rate (dataset, shape=(3576,), dtype=float32)
│       ├── bottom_chamber_temperature (dataset, shape=(3576,), dtype=float32)
│       ├── bottom_flow_rate (dataset, shape=(3576,), dtype=float32)
│       ├── bottom_flow_temperature (dataset, shape=(3576,), dtype=float32)
│       ├── build_chamber_position (dataset, shape=(3576,), dtype=float32)
│       ├── build_plate_temperature (dataset, shape=(3576,), dtype=float32)
│       ├── build_time (dataset, shape=(3576,), dtype=float32)
│       ├── gas_loop_oxygen (dataset, shape=(3576,), dtype=float32)
│       ├── glass_scale_temperature (dataset, shape=(3576,), dtype=float32)
│       ├── laser_rail_temperature (dataset, shape=(3576,), dtype=float32)
│       ├── layer_times (dataset, shape=(3576,), dtype=float32)
│       ├── module_oxygen (dataset, shape=(3576,), dtype=float32)
│       ├── powder_chamber_position (dataset, shape=(3576,), dtype=float32)
│       ├── target_ventilator_flow_rate (dataset, shape=(3576,), dtype=float32)
│       ├── top_chamber_temperature (dataset, shape=(3576,), dtype=float32)
│       ├── top_flow_rate (dataset, shape=(3576,), dtype=float32)
│       ├── top_flow_temperature (dataset, shape=(3576,), dtype=float32)
│       └── ventilator_speed (dataset, shape=(3576,), dtype=float32)
```

**1 Process parameters**

- hatch spacing, laser beam power/speed, laser spot size, scan rotations, and etc.

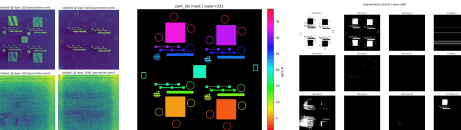
**2 Scans**

- laser trajectory for each layer
- (1~2 columns: 'x' start/end, 3~4 columns: 'y' start/end, 5th col: relative time)



**3 Slice-level images**

- visible: post melt / post recoat images
- part\_ids: pixel-wise 2D mask for indexing parts
- segmentation: 2D mask for 12 defects



**4 Temporal**

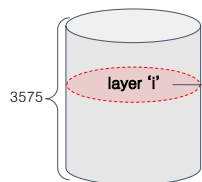
- flow rate, temperature, oxygen-related signals, chamber conditions, build time, and ventilator-related measurements

# Dataset (Cont')

## ❖ Build 1 (Cont')

### ➤ Slices: post\_melt images, post\_recoat images, part\_ids, segmentation\_results

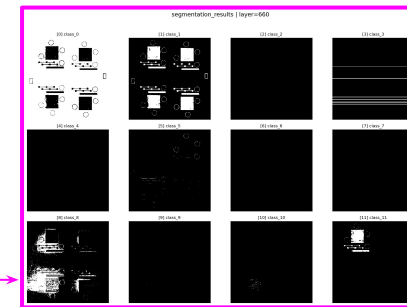
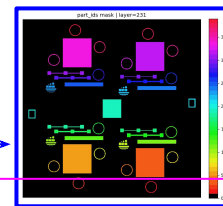
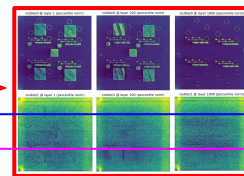
- All parts start at the first layer( $z_0$ ), but the final height ( $z_{max}$ ) differs by part
- Parts are assigned a unique ID based on their locations at each layer (2D-array mask)
- 'Segmentation\_results' include 1842×1842 binary mask of 12 classes for all layers
  - [0] Powder: Areas of the powder bed which contain no detected anomalies or printed parts
  - [1] Printed: Printed regions with no detected anomalies
  - [2] Recoater Hopping: Ripples which occur if the recoater impacts parts below the surface
  - [3] Recoater Streaking: Occurs if the recoater is damaged or dragging a large particle
  - [4] Incomplete Spreading: Occurs if insufficient powder is spread across the powder bed
  - [5] Swelling: A distortion / warping of the printed material that protrudes above the powder
  - [6] Debris: A catch-all class for small-to-medium disturbances in the powder bed
  - [7] Super-Elevation: A lack of powder coverage over top of a printed region
  - [8] Spatter: Ejecta from the weld pool which has landed back on the powder bed
  - [9] Misprint: Printed material detected outside of the intended part geometries
  - [10] Over Melting: Regions melted with high energy density process parameters
  - [11] Under Melting: Regions melted with low energy density process parameters



3575

slices:

- camera\_data/visible/0: 1842×1842 pixel (245×245 mm) **in-situ** image
- camera\_data/visible/1: 1842×1842 pixel **in-situ** image
- slices/part\_ids: 1842×1842 pixel mask (integer: 0 & 1~39)
- slices/sample\_ids: 1842×1842 pixel mask (integer: 0 & 1~3296)
- slices/segmentation\_results/0~11: 1842×1842 pixel mask (int: 0 & 1~11)



# Dataset (Cont')

## ❖ Build 1 (Cont')

➤ **Temporal:** contain information regarding 19 health indicators of printer itself (1-D array and indexed by the layer number)

- absolute\_image\_capture\_timestamp: UTC time stamp for each printed layer
- actual\_ventilator\_flow\_rate: Measured argon flow through the ventilator
- ⋮

(Temporal)

- top\_flow\_temperature: Temperature of the upper argon flow
- ventilator\_speed: Measured ventilator speed

➤ **Scan path:** 5-column vectors for each layer

- columns 0 ~ 1 store the start and end 'x' coordinates
- columns 2 ~ 3 store the start and end 'y' coordinates
- column 4 stores the relative time at which the vector was printed
- This can be converted into 2D mask, temporally and spatially aligned with layer-wise image data

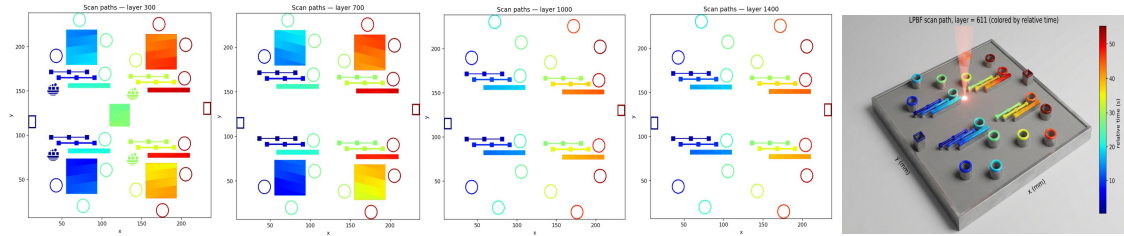
```

Temporal (group_children=14)
├─ absolute_image_capture_timestamp (dataset, shape=(3575,), dtype=float32)
├─ actual_ventilator_flow_rate (dataset, shape=(3576,), dtype=float32)
├─ bottom_chamber_temperature (dataset, shape=(3576,), dtype=float32)
├─ bottom_flow_rate (dataset, shape=(3576,), dtype=float32)
├─ bottom_flow_temperature (dataset, shape=(3576,), dtype=float32)
├─ build_chamber_position (dataset, shape=(3576,), dtype=float32)
├─ build_plate_temperature (dataset, shape=(3576,), dtype=float32)
├─ build_time (dataset, shape=(3576,), dtype=float32)
├─ gas_loop_oxygen (dataset, shape=(3576,), dtype=float32)
├─ glass_scale_temperature (dataset, shape=(3576,), dtype=float32)
├─ laser_rail_temperature (dataset, shape=(3576,), dtype=float32)
├─ layer_times (dataset, shape=(3576,), dtype=float32)
├─ module_oxygen (dataset, shape=(3576,), dtype=float32)
├─ powder_chamber_position (dataset, shape=(3576,), dtype=float32)
├─ target_ventilator_flow_rate (dataset, shape=(3576,), dtype=float32)
├─ top_chamber_temperature (dataset, shape=(3576,), dtype=float32)
├─ top_flow_rate (dataset, shape=(3576,), dtype=float32)
├─ top_flow_temperature (dataset, shape=(3576,), dtype=float32)
├─ ventilator_speed (dataset, shape=(3576,), dtype=float32)

```

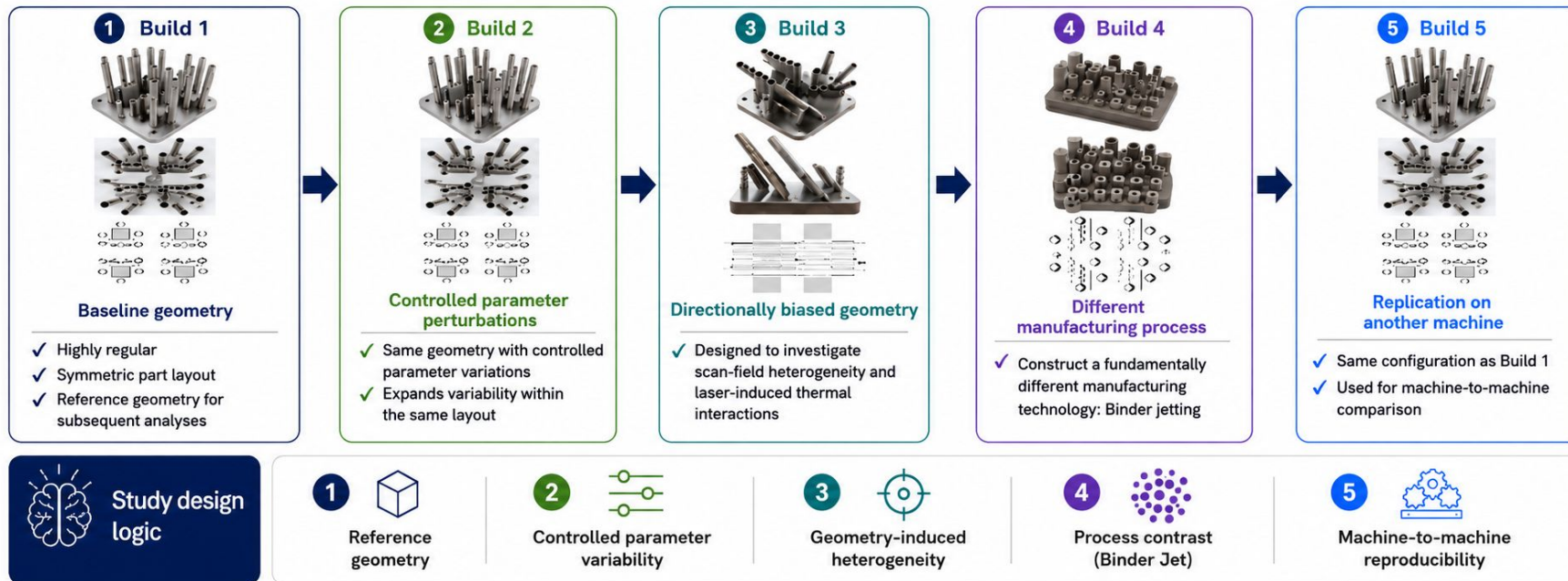
(Process\_parameters)

part_id	latch_speed	laser_beam_power	laser_beam_speed	laser_module	laser_spot_size	parameter_set	scan_ratio	strip_width
1	90	370	1350	2	130	0	67	10
2	90	370	1350	1	130	0	67	10
3	90	370	1350	2	130	0	67	10
4	90	370	1350	2	130	0	67	10
5	90	370	1350	1	130	0	67	10
6	90	370	1350	1	130	0	67	10
7	90	370	1350	2	130	0	67	10
8	90	370	1350	1	130	0	67	10
9	90	370	1350	2	130	0	67	10
10	90	370	1350	1	130	0	67	10
11	90	370	1350	2	130	0	67	10
12	90	370	1350	1	130	0	67	10
13	90	370	1350	2	130	0	67	10
14	90	370	1350	2	130	0	67	10
15	90	370	1350	1	130	0	67	10
16	90	370	1350	1	130	0	67	10
17	90	370	1350	2	130	0	67	10
18	90	370	1350	1	130	0	67	10
19	90	370	1350	1	130	0	67	10
20	90	370	1350	2	130	0	67	10
21	90	370	1350	1	130	0	67	10
22	90	370	1350	2	130	0	67	10
23	90	370	1350	2	130	0	67	10
24	90	370	1350	1	130	0	67	10
25	90	370	1350	2	130	0	67	10
26	90	370	1350	1	130	0	67	10
27	90	370	1350	1	130	0	67	10
28	90	370	1350	2	130	0	67	10
29	90	370	1350	2	130	0	67	10
30	90	370	1350	1	130	0	67	10
31	90	370	1350	2	130	0	67	10
32	90	370	1350	1	130	0	67	10
33	90	370	1350	1	130	0	67	10
34	90	370	1350	2	130	0	67	10
35	90	370	1350	2	130	0	67	10
36	90	370	1350	1	130	0	67	10
37	90	370	1350	2	130	0	67	10
38	90	370	1350	1	130	0	67	10
39	90	370	1350	2	130	0	67	10



# Dataset (Cont')

 Builds 1–5 share the same HDF5 hierarchy



# Dataset (Cont')

## ❖ The raw Peregrine HDF5 data stores required modalities in heterogeneous formats:

- Images (e.g., post melt, post recoat, part-ID maps, and segmentation maps)
- Scan paths, process parameters, temporal signals

## ❖ Without reconstruction, training would require:

- Repeated HDF5 traversal; on-the-fly shape handling
- Increases I/O overhead

## ❖ Dataset Refine Principles

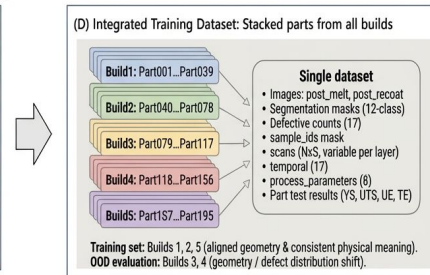
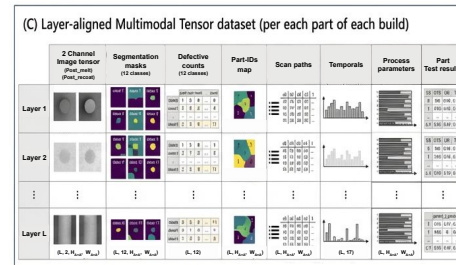
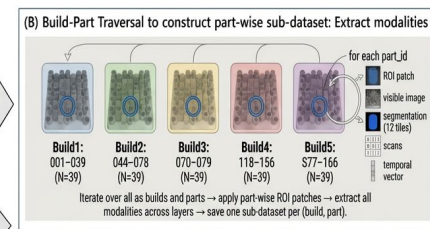
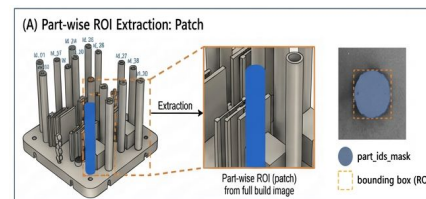
- Layer-aligned organization
- Unified-shape inputs
  - Scan paths (5D vector) are converted into pixel-aligned heatmaps
  - Process\_para (8D vector) are converted into pixel-aligned masks
  - Temporals (17D vector) are used as an input vector in isolation
  - Part-ID map is not used as input data, but as spatial encoding tool
- Efficient multimodal access

### Core Indices

- $b \in \mathcal{B}$ : build index
- $\ell \in \mathcal{N}$ : layer index
- $k$ : context length
- $i$ : last observed layer in the context
- $\ell_i = i + 1$ : target layer

### Core Variables

- $X_\ell^{(b)} \in \mathbb{R}^{H \times W}$ : post-melt image
- $R_\ell^{(b)} \in \mathbb{R}^{H \times W}$ : post-recoat image
- $S_\ell^{(b)} \in \mathbb{R}^{H \times W}$ : scan heatmap
- $\tau_\ell^{(b)} \in \mathbb{R}^{17}$ : temporal feature vector
- $P_\ell^{(b)} \in \mathbb{Z}^{H \times W}$ : part-ID map
- $\pi_p^{(b)} \in \mathbb{R}^8$ : process parameter vector for part  $p$



# Terminology

- ❖ Define core indices and variables, where  $H = W = 1842$

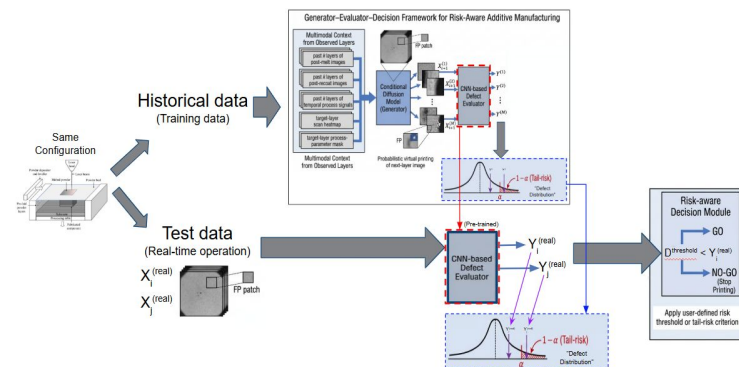
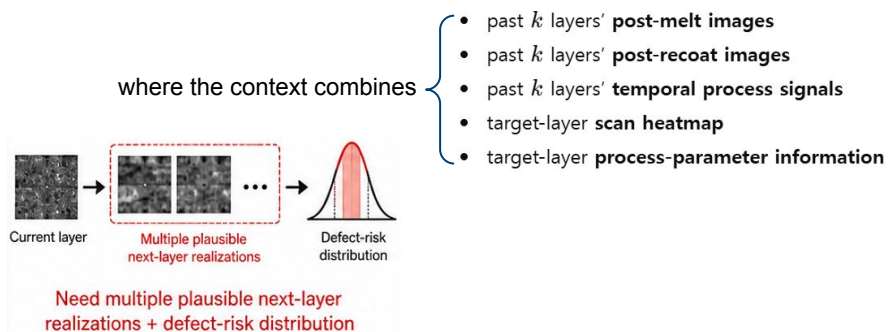
Core Indices	Core Variables
<ul style="list-style-type: none"> <li>• <math>b \in \mathcal{B}</math>: build index</li> <li>• <math>\ell \in \mathcal{N}</math>: layer index</li> <li>• <math>k</math>: context length</li> <li>• <math>i</math>: last observed layer in the context</li> <li>• <math>\ell_i = i + 1</math>: target layer</li> </ul>	<ul style="list-style-type: none"> <li>• <math>X_\ell^{(b)} \in \mathbb{R}^{H \times W}</math>: post-melt image</li> <li>• <math>R_\ell^{(b)} \in \mathbb{R}^{H \times W}</math>: post-recoat image</li> <li>• <math>S_\ell^{(b)} \in \mathbb{R}^{H \times W}</math>: scan heatmap</li> <li>• <math>\tau_\ell^{(b)} \in \mathbb{R}^{17}</math>: temporal feature vector</li> <li>• <math>P_\ell^{(b)} \in \mathbb{Z}^{H \times W}</math>: part-ID map</li> <li>• <math>\pi_p^{(b)} \in \mathbb{R}^8</math>: process parameter vector for part <math>p</math></li> </ul>

- ❖ Filter Patch (FP): For any modality  $Z_\ell^{(b)} \in \mathbb{R}^{H \times W}$  at layer ' $\ell$ ',  $Z_\ell^{(b)}[x_0, y_0] := Crop_{(x_0, y_0)}(Pad(Z_\ell^{(b)})) \in \mathbb{R}^{A \times A}$  with
  - Patch size  $A \times A$ , Padding size ' $p$ ', and Stride ' $s$ '
  - This ensures: <sup>1)</sup> full spatial coverage, <sup>2)</sup> uniform model input, <sup>3)</sup> scalable patch-based training
  - A FP location is considered valid if the cropped part-ID patch contains enough printed pixels:  $\#\{(u, v) : P_\ell^{(b)}[x_0, y_0](u, v) > 0\} \geq m_{\min}$
- ❖ Process\_parameter mask construction
  - For a target layer FP, we convert 'part-level 8D vectors' into a pixel-aligned 2D mask by  $M_{\ell_i}^{(b)}[x_0, y_0](:, u, v) = \begin{cases} \pi_p^{(b)}, & \text{if } p > 0 \\ 0, & \text{if } p = 0 \end{cases}$  by broadcasting 8D vector to the corresponding pixel aligned with 'part-ID map'
  - This creates a spatially aligned multi-channel tensor:  $M_{\ell_i}^{(b)}[x_0, y_0] \in \mathbb{R}^{8 \times A \times A}$

# Framework Overview

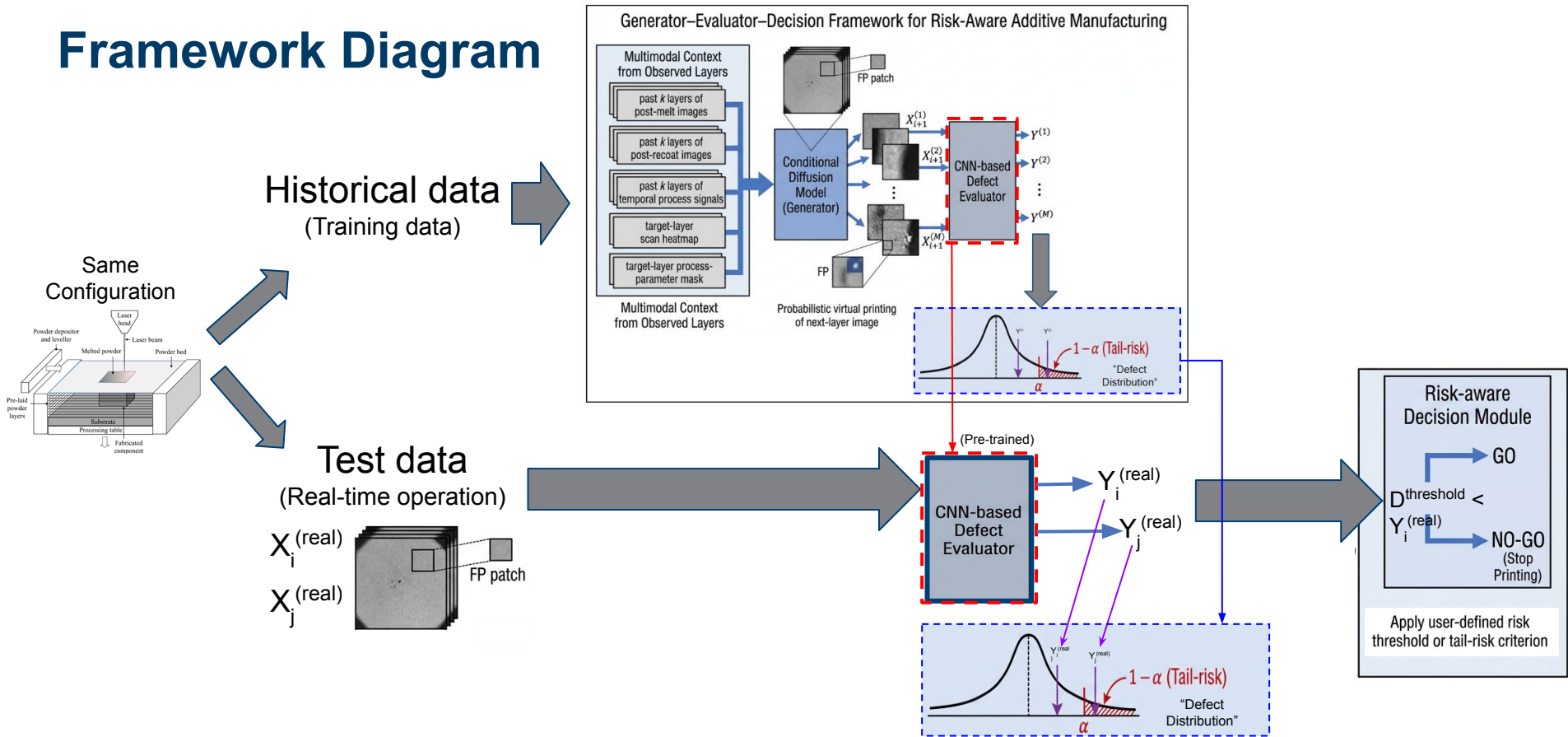
❖ We propose a ‘**Generator–Evaluator–Decision**’ framework, where

- Diffusion-based **Generator** replaces expensive physical experiments with probabilistic virtual printing;
  - Given ‘ $k$ ’ observed layers up to layer ‘ $i$ ’, we train a Conditional Diffusion Model (CDM) to generate the next-layer post-melt image,



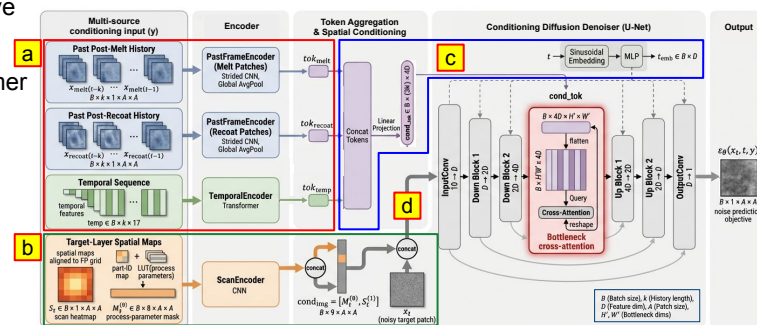
- Generated samples are evaluated using a CNN-based **Evaluator**, producing defect outcomes  $Y^{(m)}$  for each sample;
  - We train a CNN-based defect predictor that maps generated layer images (i.e., post melt + post recoat) to 12 defect mask outcomes
- Defect rates are aggregated to estimate ‘defect distributions’ of a certain configuration for risk-aware **Decision** making
  - Set the threshold,  $\alpha = 0.05$  (user-defined criteria)
  - If the predicted defect risk exceeds a user-defined threshold, a NO-GO decision is triggered

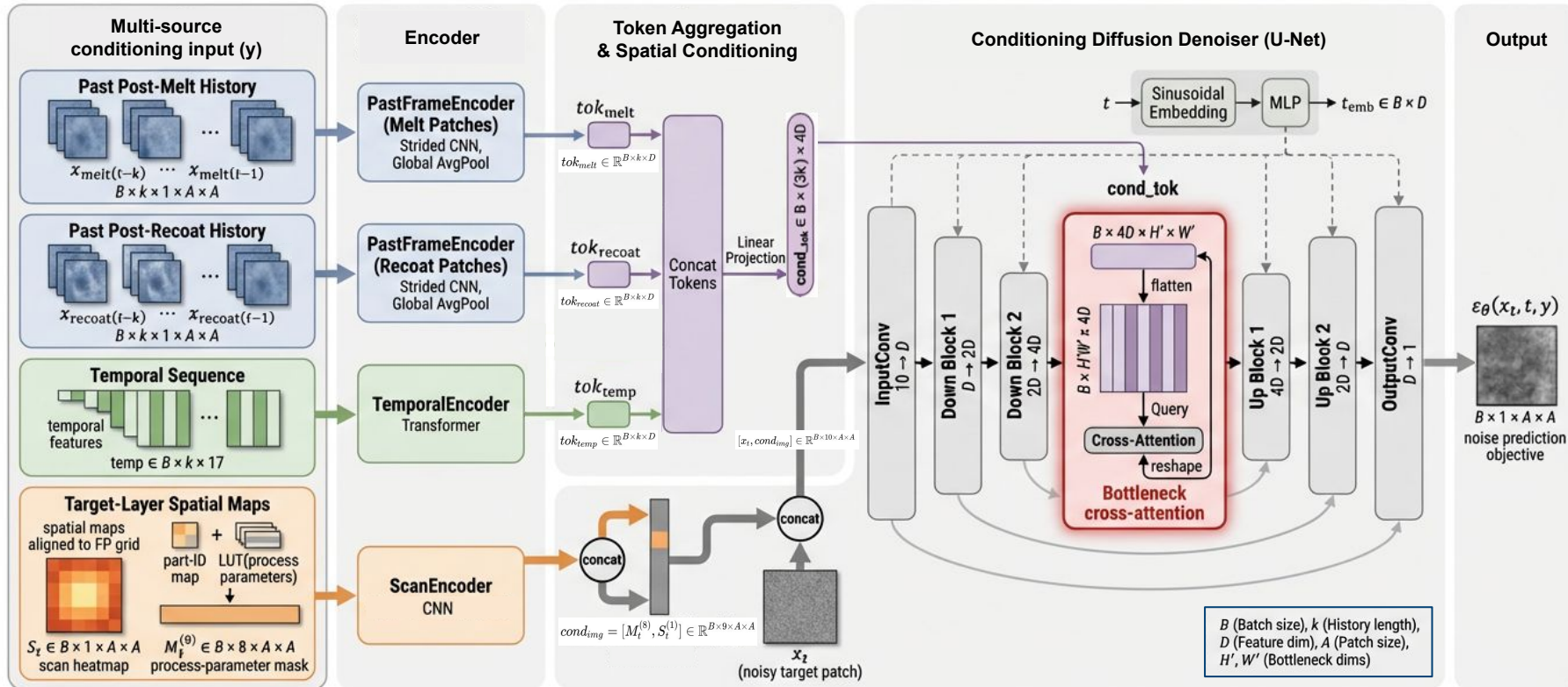
# Framework Diagram



# Methods: CDM-based Generator

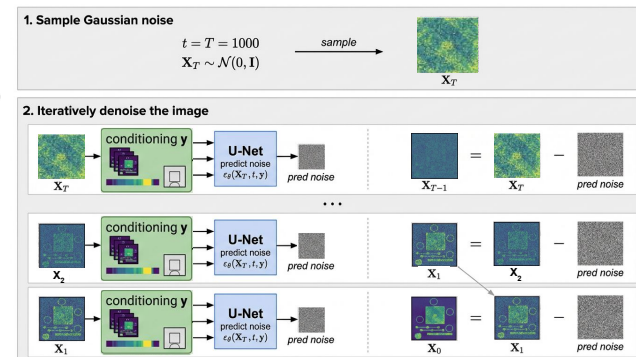
- ❖ Given a context window of ‘k’ observed layers up to layer ‘i’, the model generates the next-layer FP patch:
  - $p_{\theta}(X_{i+1}[x_0, y_0] | \text{context}_{i-k+1:i}(x_0, y_0))$ , where the conditioning context integrates multimodal information
    - past k layers’ post-melt images
    - past k layers’ post-recoat images
    - past k layers’ temporal process signals
    - target-layer scan heatmap
    - target-layer process-parameter information
- ❖ The model uses two types of conditioning information to generate the next-layer image
  - **a** Context ‘k’ history (e.g., previous layer images, temporal sequence): past observations are summarized into compact tokens
    - Processing all historical layers at full spatial resolution would be expensive
    - ‘Post-melt’ and ‘Post-recoat’ images are encoded by ‘CNN+GAP’
    - ‘Temporal layer-wise sequence’ is tokenized by a lightweight Transformer
  - **b** Two spatial maps aligned to the target layer are provided as tensor
    - **Scan heatmap**: spatial laser exposure pattern
    - **Process-parameter mask** (constructed from part-level process parameters and part-ID maps): Each pixel inherits the process parameter vector of the part it belongs to
    - These two maps are concatenated with the noisy diffusion input
- ❖ Architecture uses two complementary conditioning pathways
  - **c** Context tokens, capturing global spatial structure and temporal information, are projected to U-Net bottleneck directly
  - **d** Spatial maps, capturing local spatial structure and process information, are injected to U-Net as extra inputs





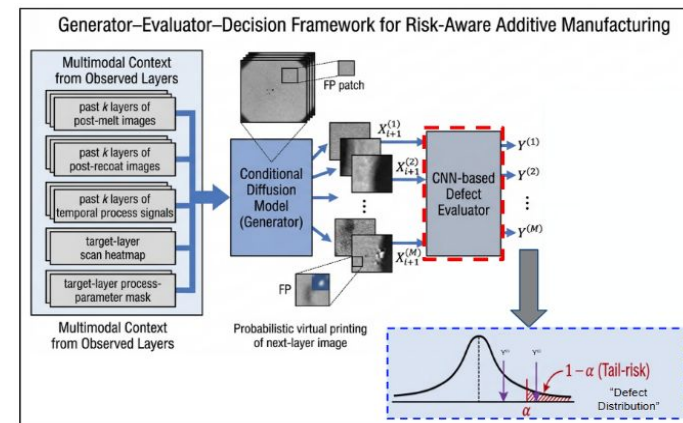
# Methods: CDM-based Generator (Cont')

- ❖ The CDM generates the next-layer post-melt image through an iterative denoising process
  - By the nature of diffusion model, the model starts from Gaussian noise and gradually reconstructs the layer image by removing noise instead of predicting the image directly
  - **Step 1.** Initialize with Gaussian noise (assumed that forward process totally corrupts the original image):  $\mathbf{X}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
    - 'T' is the number of diffusion steps (e.g., T=1000)
    - $\mathbf{X}_T$  represents a completely noisy image.
  - **Step 2.** At each diffusion step 't', the model predicts the noise using a conditional U-Net denoiser:  $\epsilon_{\theta}(\mathbf{X}_t, t, \mathbf{y})$ 
    - t: diffusion step where forward process adds noise at
    - y: multimodal conditioning context
  - **Step 3.** The predicted noise is removed from the current image:  $\mathbf{X}_{t-1} = \mathbf{X}_t - \epsilon_{\theta}(\mathbf{X}_t, t, \mathbf{y})$ 
    - This denoising step is repeated iteratively:  $\mathbf{X}_T \rightarrow \mathbf{X}_{T-1} \rightarrow \dots \rightarrow \mathbf{X}_1 \rightarrow \mathbf{X}_0$
    - As the steps progress, the image gradually becomes more structured
    - $\mathbf{X}_0$  represents the generated next-layer post-melt image (This image corresponds to a virtual realization of the next printed layer)
    - In this way, the model can generate multiple plausible realizations of the next-layer post-melt image:  $\mathbf{X}_0^{(1)}, \mathbf{X}_0^{(2)}, \dots, \mathbf{X}_0^{(M)}$



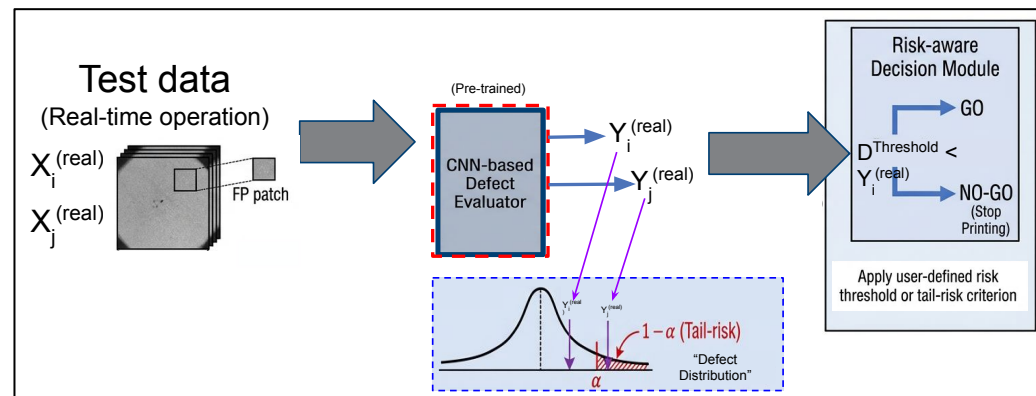
# Methods: CNN-based Defect Evaluator

- ❖ The evaluator maps generated samples to defect-related outcomes:  $Y^{(m)} = f_{\theta}(X^{(m)})$ 
  - $\theta$ : a neural network (e.g., CNN-based variations) parameter
  - Input:  $X^{(m)}$ , generated next-layer sample
    - Two images (generated post-melt + corresponding post-recoat) form a 2-channel input representation
  - Output:  $Y^{(m)}$ , predicted defect outcome (e.g., defect presence, defect severity, defect rate)
  
- ❖ Since the generator produces multiple stochastic samples, the evaluator outputs a set of predictions:
  - These predictions  $\{Y^{(1)}, Y^{(2)}, \dots, Y^{(m)}\}$  are aggregated to estimate the empirical defect distribution
  - This enables “distribution-level risk” estimation rather than single deterministic predictions



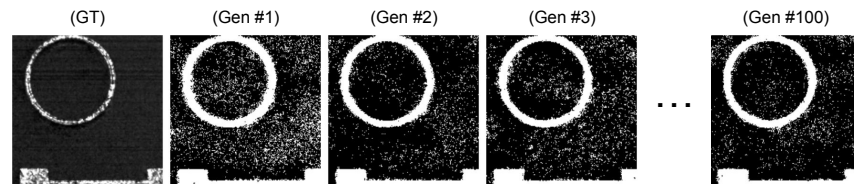
## Methods: Decision Engine (“Rule-based”)

- ❖ The estimated defect distribution is used to support real-time manufacturing decisions
  - A user-defined risk threshold ( $D^{\text{threshold}}$ ) is applied to the predicted defect outcome ( $Y^{(m)}$ )
  - The decision rule evaluates the probability that the predicted defect ( $Y^{(m)}$ ) exceeds the acceptable limit
  - The printing process follows a simple rule-based policy such that:
    - **“Keep printing”** if predicted defect risk  $\leq D^{\text{threshold}}$
    - **“Stop printing”** if predicted defect risk  $> D^{\text{threshold}}$  (“early termination to avoid further production cost”)
  
- ❖ This decision framework enables:
  - real-time defect risk monitoring
  - early detection of high-risk builds
  - reduced material waste and production time

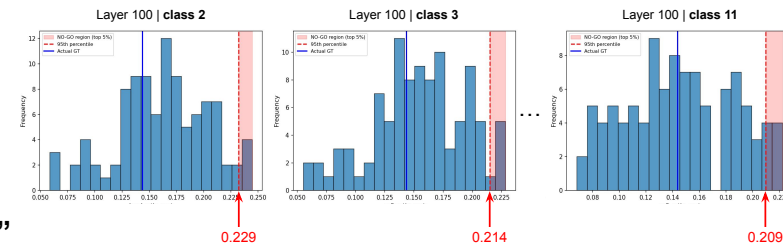
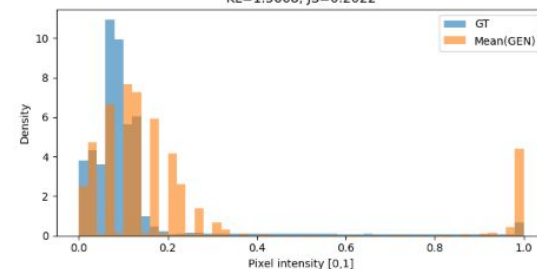


# Results & Evaluation

- ❖ High quality virtual printings generated by CDM
  - Structure and texture are close to GT
- ❖ Defect distribution estimated with GT in the mean-period
  - Evaluators take a tensor of [generated post-melt, post-recoat] as input
  - Output the defect rates for 12 classes at each patch
  - Decide “NO-GO” region in the defect distributions
    - Use a user-defined threshold ( $\alpha$ )
- ❖ Enable real time “GO/NO-GO” decision making
  - Real-time [post-melt, post-recoat] sent to the Evaluator
  - Evaluator estimates defect rates for those real time printed layer
  - Declare “NO-GO” once the estimated defect rates exceed ‘ $\alpha$ ’  
**(Ex. “If 100<sup>th</sup> real printing yields 2<sup>nd</sup> class defect rate of 0.230”**  
**⇒ I will stop the printing as this error will be propagated and make the process worse as it goes further!)**



Histogram | GT vs Mean(GEN) | GT#7 |  $i=425, t=426$   
 $KL=1.5668, JS=0.2022$



# Contributions

## ❖ A first unified “Generator–Evaluator–Decision” framework for risk-aware LPBF monitoring

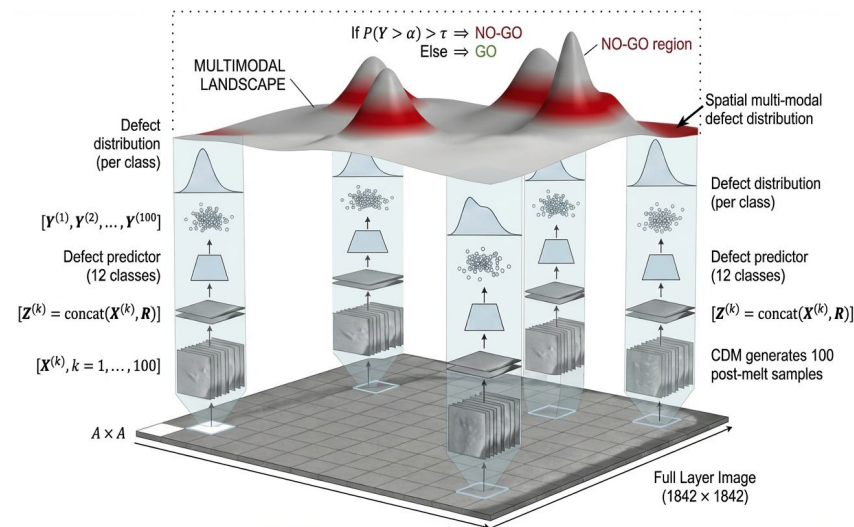
- Conditional virtual printing of next-layer post-melt images
- Defect-distribution estimation via stochastic sample evaluation

## ❖ A first multimodal CDM on the Peregrine dataset

- Fuses image history, temporal signals, scan paths, and process parameters in the architecture to maximize information usage
- Enables next-layer prediction under manufacturing context

## ❖ Decision-oriented uncertainty quantification

- Uses defect distributions instead of point predictions
- Supports real-time GO / NO-GO decisions under uncertainty



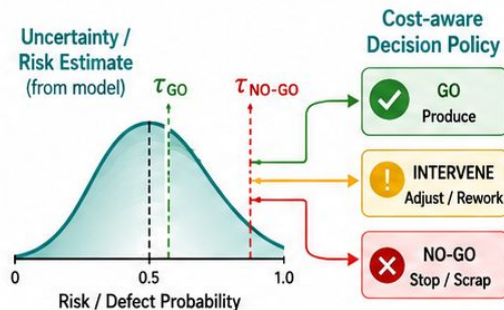
# Future Research

## 1 Generalization & Robustness



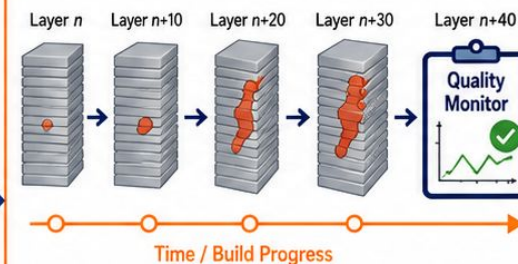
- Cross-build and cross-machine validation
- Robustness to geometry changes across layers
- Robustness to in-build parameter perturbations

## 2 Smarter Decision Policies



- Move beyond fixed GO / NO-GO rules
- Cost-aware stopping criteria
- Bayesian decision rules
- Confidence-based NO-GO region from Monte Carlo simulation

## 3 Error Propagation & Quality Control



- Track how defects evolve over time
- Study how propagated errors affect final quality
- Check whether defects remain within tolerance limits
- Develop practical control strategies



**ANNUAL**  
CONFERENCE & EXPO 2026

## Q & A

Any questions, or  
comments?

*Remember to complete your evaluation for this session within the app!*

[iise.org/annual](https://iise.org/annual) #IISEANNUAL2026