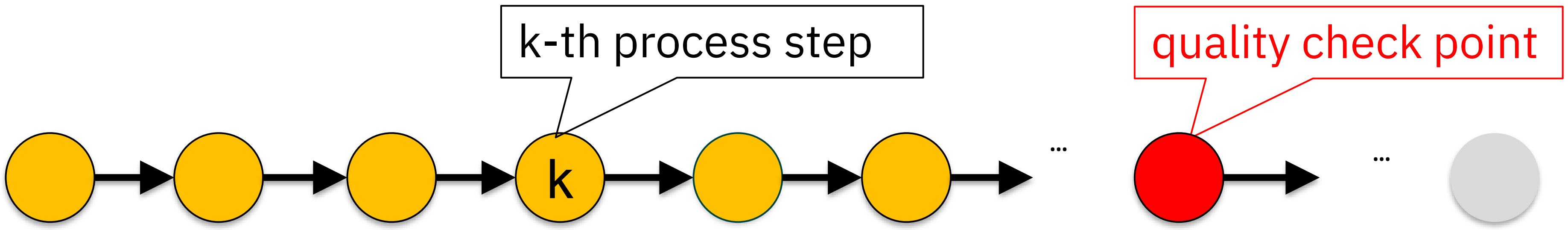
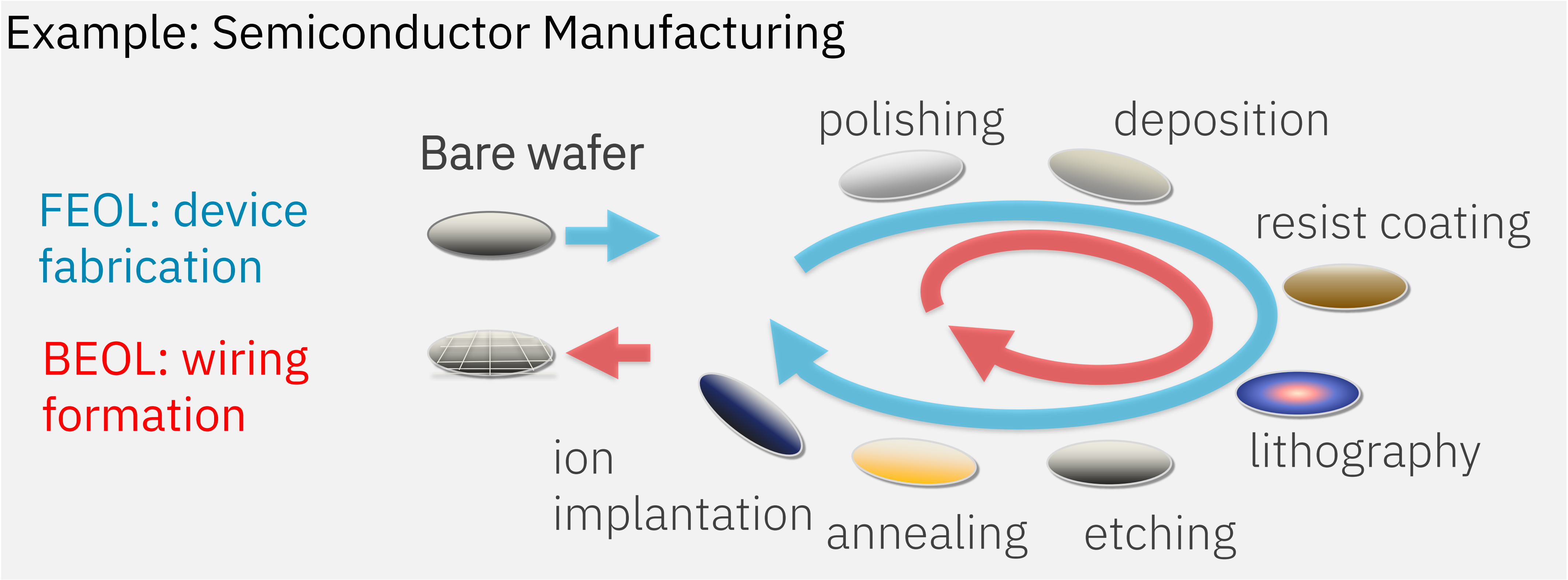
Tsuyoshi (Ide-san) Ide

Partial Trajectory Analysis for Diagnosing Sequential Manufacturing Processes

Head of Data Science, IBM Semiconductors,
IBM T. J. Watson Research Center

Target task: cross-process defect attribution

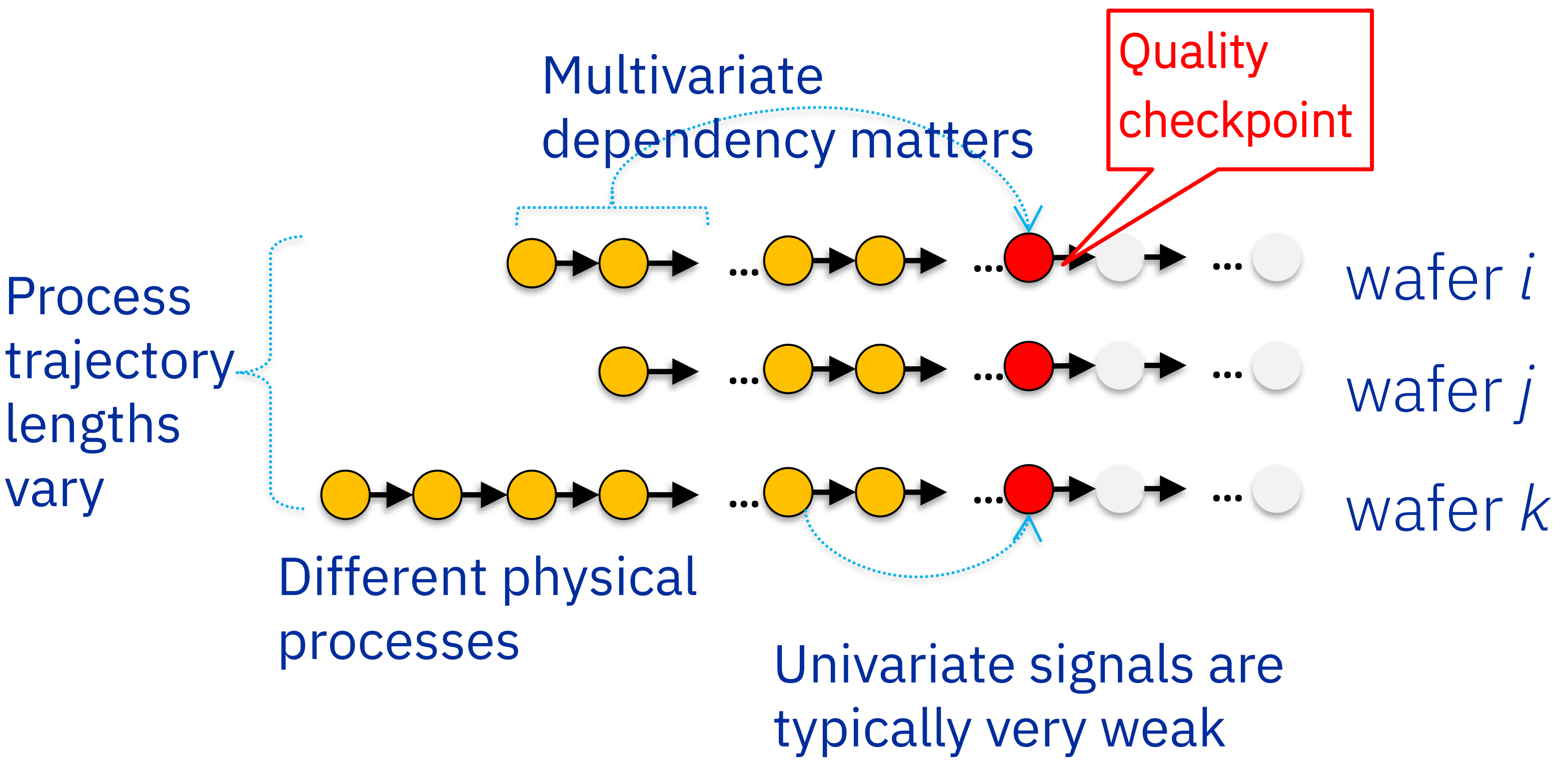




© 2025 IBM Corporation

Target task: cross-process defect attribution

Problem: Given a wafer quality metric value, compute the **attribution score** for each of the upstream process steps.



- In current practice…
  - The only viable approach is to run as many wafers as possible under varying conditions.
  - Then, relatively simple statistical analysis is applied.
  - This approach requires significant domain expertise to decide on parameter choices.
  - This semi-manual approach is reaching its limits as technology nodes advance.

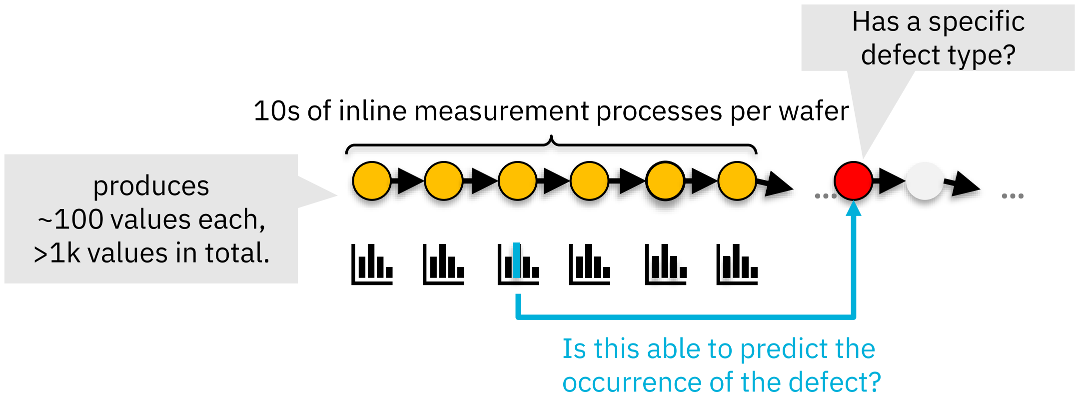
© 2025 IBM Corporation

Univeriate correlation is often not informative

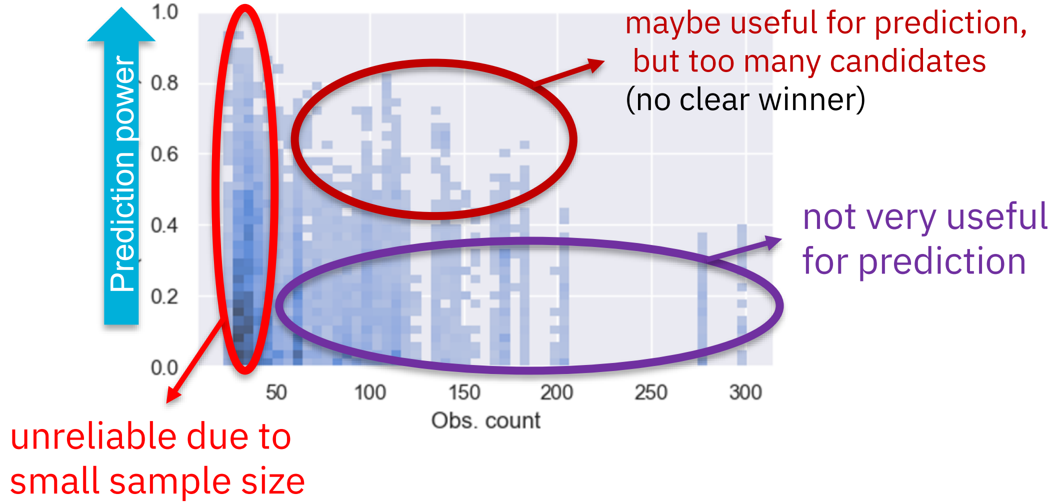
- In test production, 10s of inline measurements produces 1K+ measurement numbers for each wafer.

- Univariate analysis typically does not give strong signal.
  - Trained a univariate binary classifier to distinguish between good and bad wafers.
  - Accuracy tends to decrease as the observation count increases.

- Process engineers usually focus on a limited number of measurement items based on prior knowledge.



Defect occurrence prediction with single measurement values

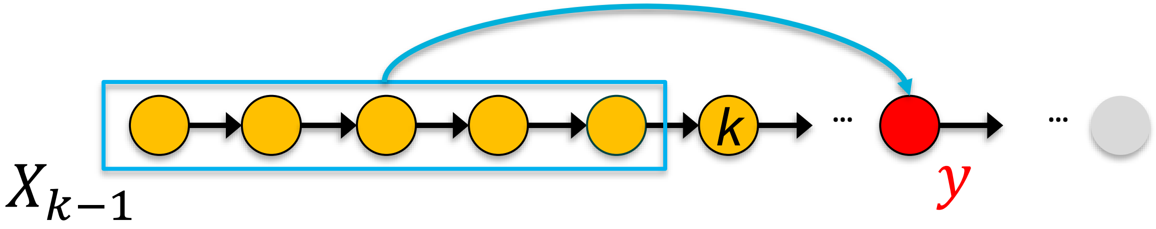


© 2025 IBM Corporation

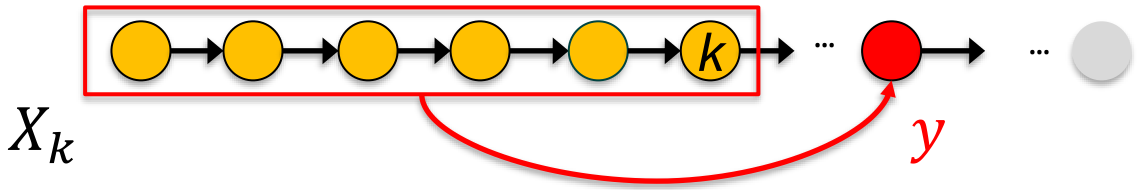
Partial trajectory analysis (PTA) approach: Key idea

- PTA computes the attribution score of the *k*-th process by evaluating the impact of *k*'s "participation" in the process trajectory.

A: Prediction using a partial trajectory **not** including *k*



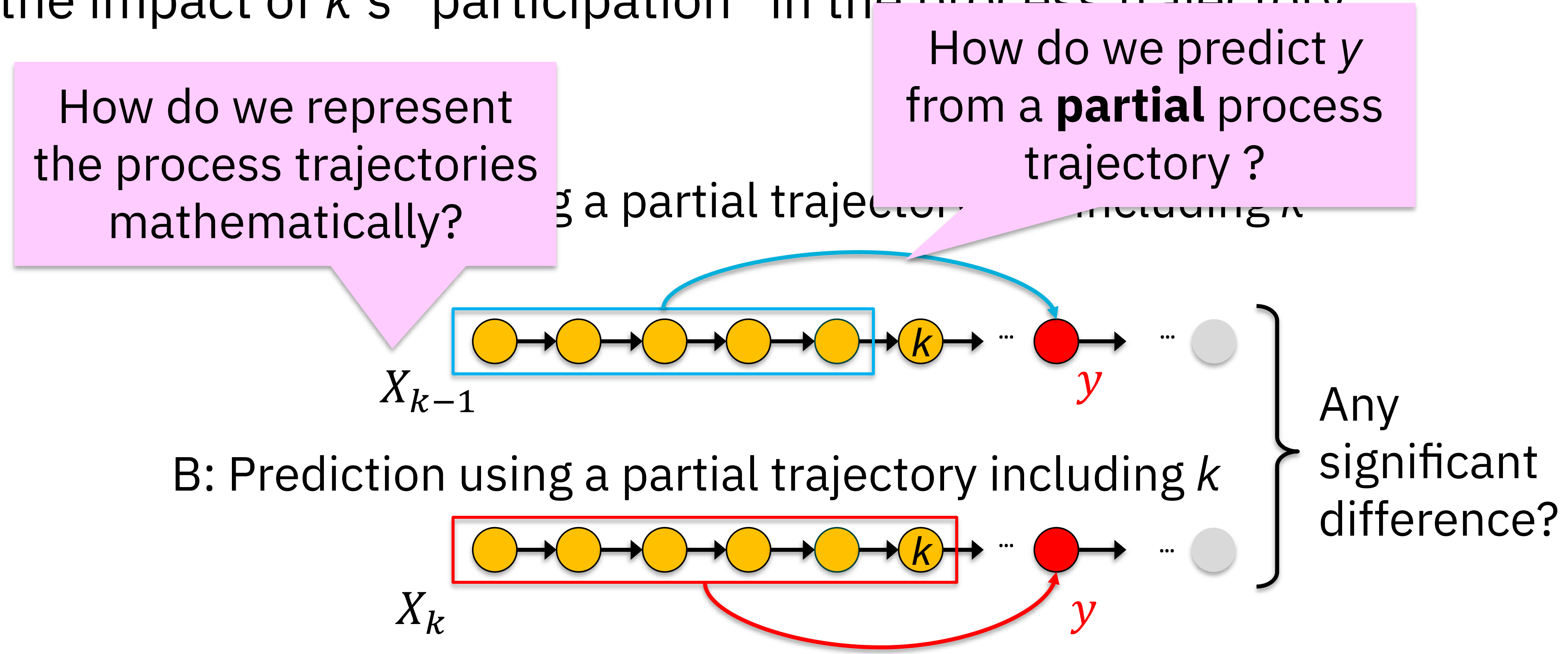
B: Prediction using a partial trajectory including *k*



Any significant difference?

Partial trajectory analysis (PTA) approach: Key idea

- PTA computes the attribution score of the *k*-th process by evaluating the impact of *k*'s "participation" in the process trajectory

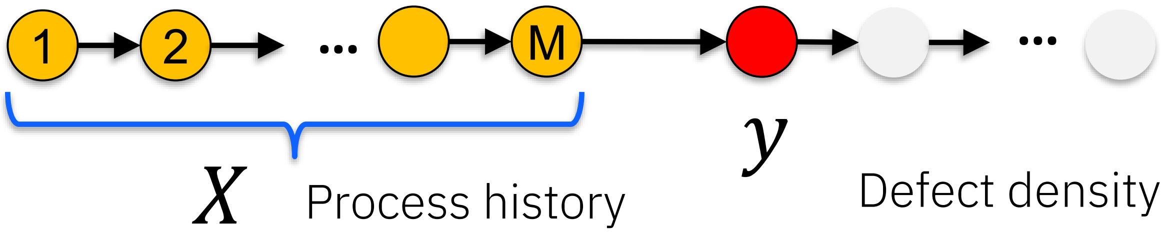


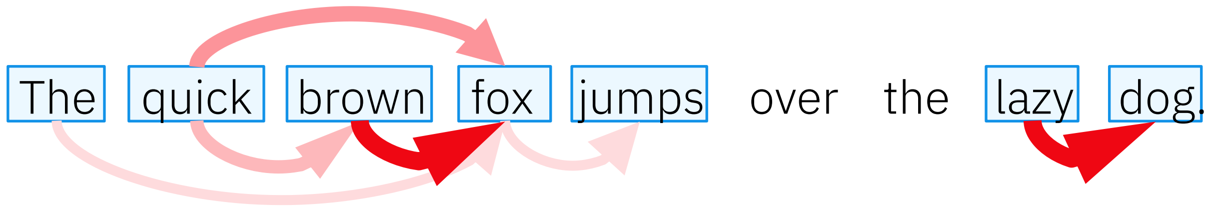
© 2025 IBM Corporation

Data assumption and process embedding ("proc2vec")

- Data $D = \{(X^{(n)}, y^{(n)}) \mid n = 1, \dots, N\}$
  - $X^{(n)}$: process trajectory $(\boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_{L^{(n)}}^{(n)})$. Each process is assumed to have a vector representation (embedding).
  - $y^{(n)}$: Product badness such as defect density (a real number).

- How do we get the embeddings?
  - Create a synthetic word for each
    - ✓ "Process token"=eqp_id$\oplus$ recipe_id$\oplus \cdots \oplus$tool_trace
  - Employ a transformer-like approach
    - ✓ [Miyaguchi + ASMC 25]

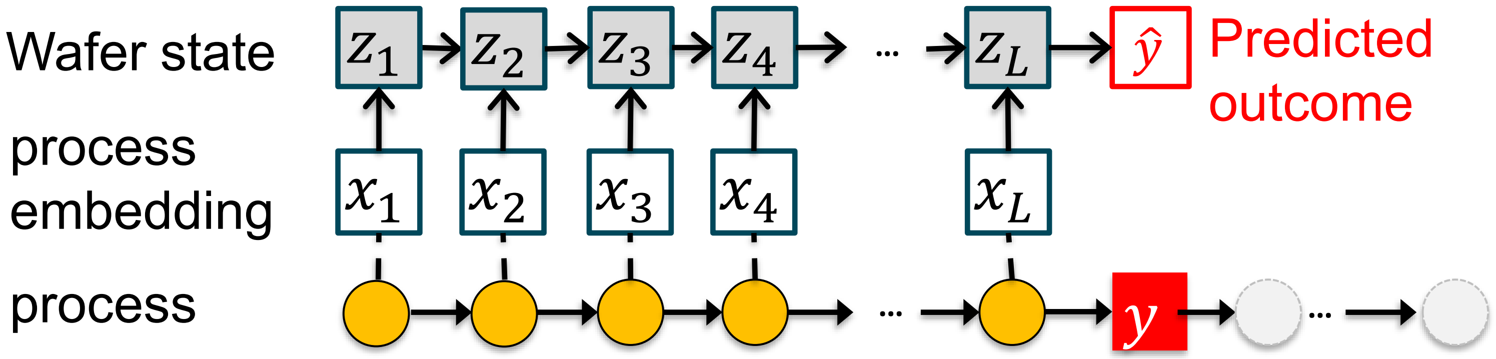




© 2025 IBM Corporation

Learning partial trajectory regression model

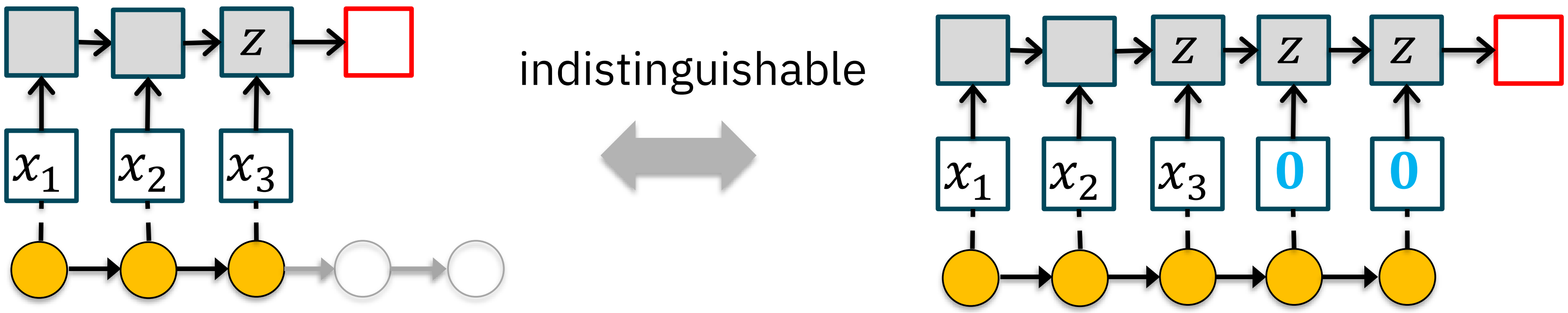
- Typically, a prediction function has a fixed number of input slots.
  - For 3 processes, it would be like $f(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$, a 3-slot function.
  - Hence, cannot handle process trajectories with different lengths.
- State-space model (or RNN) eliminates this limitation
  - Partial prediction by a length-k trajectory: $\hat{y} = f(\boldsymbol{z}_k = \mathrm{RNN}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_k))$
    - ✓ f : A parametric function with trainable parameters
    - ✓ $\boldsymbol{z}_k$: Latent state vector after observing $\boldsymbol{x}_k$



© 2025 IBM Corporation

Catch: You can’t simply zero off downstream processes

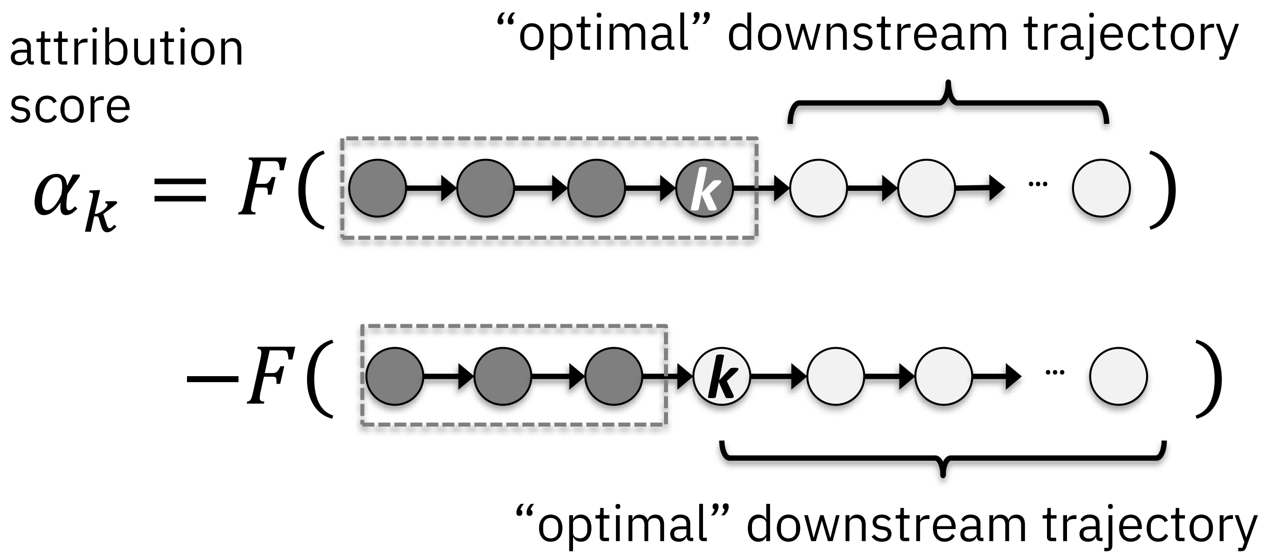
- The straightforward use of RNN introduces a significant bias.
- Example:
  - k=3 partial prediction is indistinguishable with k=5 partial prediction with zeroed-off input.
  - i.e., RNN’s partial trajectory prediction
    = full trajectory prediction with a zeroed-off process sequence



© 2025 IBM Corporation

Eliminating the bias of partial trajectory analysis: Potential Loss Analysis (PLA)

- PLA: The attribution score for k is evaluated, given an optimal downstream trajectory:
  - $\min_{\boldsymbol{x}_{k+1},\ldots,\boldsymbol{x}_L} F(\boldsymbol{z}_k, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_L)$



- This trajectory optimization problem can be solved by casting it as a reinforcement learning task.
  - → Next page

Details→

Ide & Miyaguchi, “Cross-process defect attribution suing potential loss analysis,” WSC 25, to appear; https://arxiv.org/abs/2508.00895

© 2025 IBM Corporation

Formalizing PLA as a reinforcement learning problem (1/2)

- $\min_{\boldsymbol{x}_{k+1},\ldots} F(\boldsymbol{z}_k, \boldsymbol{x}_{k+1}, \ldots) = \min_{\boldsymbol{x}_{k+1},\ldots} \mathbb{E}\left[\sum_{t=1}^{\infty} C(\boldsymbol{z}_{k+t}) \mid \boldsymbol{z}_k\right]$
  - terminal reward model: $C(\boldsymbol{z}) = \begin{cases} y(\boldsymbol{z}), & z \in \text{(terminal state)} \\ 0, & \text{otherwise} \end{cases}$
- Bellman equation
  - $F^*(\boldsymbol{z}) \equiv \min_{\boldsymbol{x}_1,\ldots} F(\boldsymbol{z}, \boldsymbol{x}_1, \ldots) = \min_{\boldsymbol{x}_1} \left\{ C(\boldsymbol{z}) + \sum_{\boldsymbol{z}_2} \underbrace{p(\boldsymbol{z}_2 \mid \boldsymbol{x}_1, \boldsymbol{z}_1)}_{\text{transition model (assumed deterministic)}} F^*(\boldsymbol{z}_2) \right\}$
- The optimization problem we solve (with $F^\theta(\boldsymbol{z})$ approximating $F^*(\boldsymbol{z})$):
  - $\max_\theta \sum_{\boldsymbol{z}} \underbrace{\rho(\boldsymbol{z})}_{\text{empirical density of } \boldsymbol{z} \text{ (under deterministic assumption)}} F^\theta(\boldsymbol{z}) \quad \text{s.t.} \quad F^\theta(\boldsymbol{z}) \leq C(\boldsymbol{z}) + F^*(\boldsymbol{z}'),\ \forall(\boldsymbol{z} \to \boldsymbol{z}'),$

Details→

Ide & Miyaguchi, "Cross-process defect attribution suing potential loss analysis," WSC 25, to appear; https://arxiv.org/abs/2508.00895

© 2025 IBM Corporation

Formalizing PLA as a reinforcement learning problem (2/2)

- Final objective function to be maximized
  - $$R(\theta \mid \mu) = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{\mu}{L^{(n)}}\sum_{t=1}^{L^{(n)}} F^{\theta}\left(z_t^{(n)}\right) - \frac{1}{2}\underbrace{\left\{y^{(n)} - F^{\theta}\left(z_{L^{(n)}}^{(n)}\right)\right\}^2}_{\text{squared loss}} - \frac{1}{2}\sum_{t=1}^{L^{(n)}-1}\left\{F^{\theta}\left(z_{t+1}^{(n)}\right) - F^{\theta}\left(z_t^{(n)}\right)\right\}^2\right]$$
  - This provides the partial prediction function and attribution model simultaneously.
- The time difference of the F function can be directly parameterized:
  - $G^{\theta}(\boldsymbol{z}_t, \boldsymbol{z}_{t-1}) \equiv F^{\theta}(\boldsymbol{z}_t) - F^{\theta}(\boldsymbol{z}_{t-1}) = \mathrm{ReLU}_{\theta}(\boldsymbol{z}_t \oplus \boldsymbol{z}_{t-1})$
  - This function provides the attribution score for the k-th process.

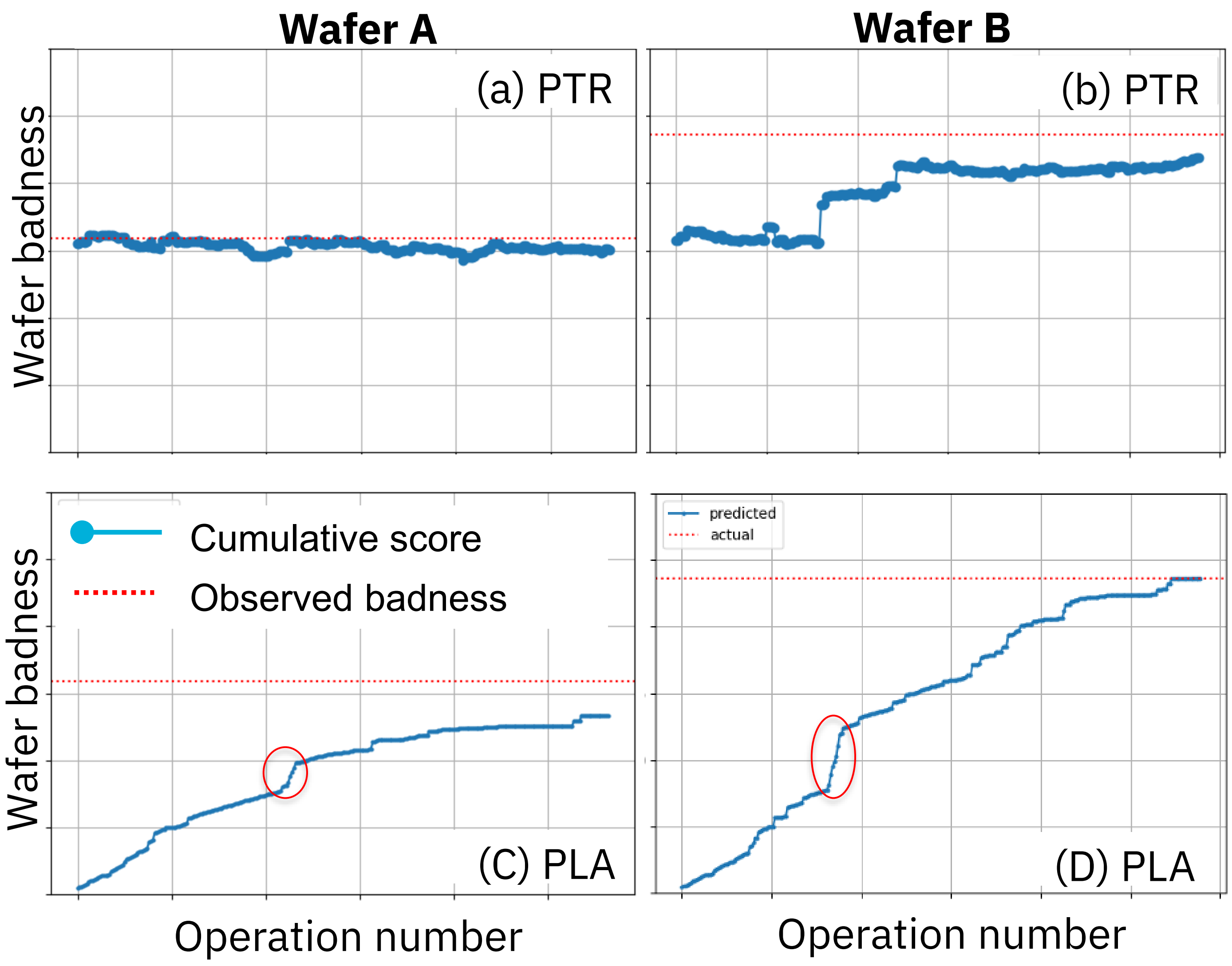
© 2025 IBM Corporation

2nm conductor process defect diagnosis example

- The graphs plot the cumulative attribution score.
  - Big jump = likely root cause
  - The model was trained with 727 wafers.

- PTR approach does not provide meaningful signals.

- PLA successfully detects likely root cause
  - In this case, they correspond to too long waiting hours at a certain piece of equipment.



© 2025 IBM Corporation

informs

ANNUAL
MEETING