IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.
Cite as T-ASE paper.

T-ASE-2025-780.R1

Large Pre-Trained Models and Few-Shot Fine-Tuning for Virtual Metrology: A Framework for Uncertainty-Driven Adaptive Process Control in Semiconductor Manufacturing

Chin-Yi Lin[1], *Member*, *IEEE*, Tzu-Liang (Bill) Tseng[1], *Member*, *IEEE*, Solayman Hossain Emon[2], and Tsung-Han Tsai[3]

***Abstract*—High-precision wafer metrology poses significant cost and throughput challenges in modern semiconductor manufacturing, where frequent process changes and recipe variations demand highly adaptive and scalable solutions. In this paper, we present a Generative-FewShot-Active Virtual Metrology (GFA-VM) framework that unifies large-scale generative modeling, few-shot fine-tuning, and uncertainty-driven active sampling into a single, data-centric system. A foundational generative model, built on a hybrid architecture of Transformer networks and Variational Autoencoders (VAEs), learns diverse sensor characteristics in an offline stage without relying on extensive labeled data. During online inference, the model produces both wafer quality predictions and predictive uncertainties; samples exceeding a dynamic uncertainty threshold are selected for physical measurement and few-shot model recalibration. This selective sampling both reduces measurement costs and adapts rapidly to new process conditions (e.g., novel recipes or equipment upgrades), requiring only a handful of freshly labeled wafers. The paper further addresses the long-term stability of the system through a self-updating mechanism that adjusts the uncertainty threshold when distributional shifts occur. Empirical evaluations confirm that our GFA-VM approach achieves state-of-the-art accuracy while significantly reducing metrology overhead compared to conventional virtual metrology methods. Additionally, rigorous theoretical analyses—including proofs of convergence and label cost bounds—demonstrate the reliability of using a generative foundation plus meta-learning technique. By fostering on-demand adaptation within a closed-loop framework, GFA-VM offers a comprehensive, scalable strategy for next-generation semiconductor process control.**

Note to Practitioners—Today's semiconductor fabrication processes involve numerous machine types and recipe variations, making frequent measurements both expensive and time-consuming. The approach described in this paper, Generative-FewShot-Active Virtual Metrology (GFA-VM), aims to help practitioners maintain consistent product quality while significantly reducing the number of physical wafer measurements. Traditional virtual metrology solutions often require extensive retraining when new recipes or equipment changes occur; our method mitigates these burdens by using a "foundation model" that learns from large volumes of historical (both labeled and unlabeled) sensor data and can be quickly adapted with only a few newly measured wafers. Practically, engineers can integrate GFA-VM into existing Manufacturing Execution Systems (MES) or Advanced Process Control (APC) platforms. The system continuously estimates each wafer's quality and flags only those with high uncertainty for actual measurement. This targeted approach optimizes the use of metrology resources and speeds up decision-making. The key benefit is flexibility: each time a new recipe or tool is introduced, the model can be recalibrated using as few as one to five measurements. However, initial setup requires careful data gathering to train the generative model, and ongoing tuning depends on reliable sensor signals. In addition, practitioners should note that sudden, large-scale process changes still need more measurements for model stability. Looking forward, the same strategy can be applied to other manufacturing contexts—where high-dimensional sensor data and limited measurement capacities challenge real-time quality control.

***Index Terms*—Semiconductor manufacturing, virtual metrology, generative modeling, few-shot learning, active sampling, meta-learning, Variational Autoencoder (VAE), Generative Adversarial Network (GAN), Transformers, uncertainty estimation, advanced process control (APC).**

I. Introduction

Traditional physical metrology technique are often expensive and can become bottlenecks in high-throughput production. Hence, Virtual Metrology (VM) employs software models—built from equipment sensor data and process parameters—to estimate wafer quality indicators [1] [2].

Early studies demonstrated the feasibility of VM in supplementing physical metrology steps. Hung *et al.* were among the first to apply a neural-network-based VM approach to predict CVD film thickness from sensor readings, reporting satisfactory accuracy [1]. Moyne *et al.* [2] integrated VM into run-to-run (R2R) process control, showing that even with only

This work was supported in part by the U.S. National Science Foundation under grants ERC-ASPIRE-1941524; DUE-2216396 and U.S. Department of Education under grants Award #P116S210004; Award #P120A220044.

[1]Chin-Yi Lin and Tzu-Liang (Bill) Tseng are with the Department of Industrial Manufacturing and Systems Engineering, University of Texas at EL Paso, El Paso, Texas, TX 79968, USA (e-mail: clin@utep.edu; btseng@utep.edu).
[2]Solayman Hossain Emon is with the Computational Science Program, University of Texas at EL Paso, El Paso, Texas, TX 79968, USA (e-mail: semon@miners.utep.edu).
[3]Tsung-Han Tsai is with Institute of Information and Decision Sciences, National Taipei University of Business, Taipei, 10051, Taiwan (e-mail: thtsai@ntub.edu.tw).

©2026 IEEE

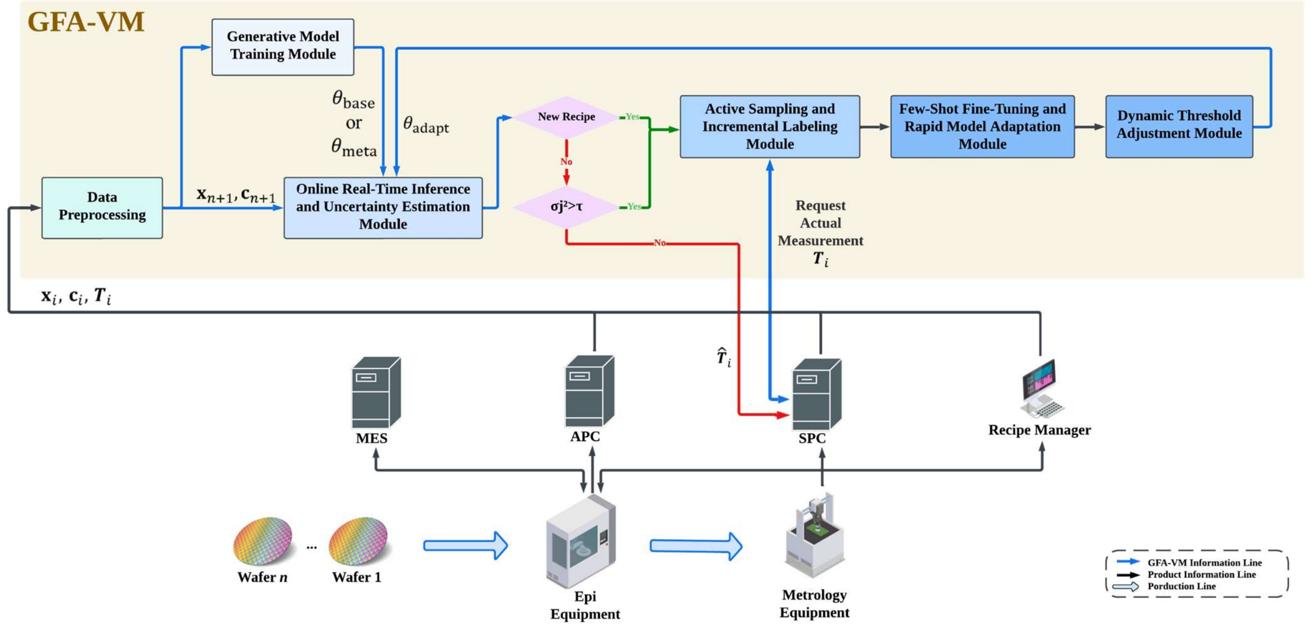

Fig. 1. Comprehensive Architecture of the Generative-FewShot-Active Virtual Metrology (GFA-VM) Framework.

a limited number of physical measurements.

A. Evolution of VM and Current Trends

Although VM has been implemented in various fabs for quite some time, its focus and methodologies have evolved in tandem with increasing process complexity. Maitra *et al.* describe VM as a "reliable APC tool built on machine learning" [3]. Notably, effectively modeling cross-tool or cross-recipe variability have become key research directions.

Recent efforts have leveraged deep learning. For instance, Tin *et al.* adopted a convolutional neural network (CNN) to predict photolithography overlay errors—a domain beyond the plasma etch and CVD processes targeted by earlier VM approaches—and demonstrated superior accuracy on complex sensor data [4]. Meanwhile, Chien *et al.* integrated autoencoder-based feature learning into a *decision-based VM* (DVM) framework, which provides a confidence score to assist real-time decisions in R2R control [5].

Apart from these deep-learning-oriented studies, researchers have also explored machine-learning or generative approaches. Classic algorithms like SVR, decision trees, or RVM have already demonstrated baseline feasibility in earlier VM deployments [2] [6]. Cai *et al.* [7] later compared multiple regression methods (linear regression, SVR, RVM) for CVD film thickness prediction, concluding that Bayesian sparse models can yield higher accuracy. Deep autoencoders or generative adversarial networks (GANs) offer a way to learn from unlabeled data, helping VM cope with equipment drift, noise, or cross-tool variations [8] [4]. Overall, VM research has extended into semi-supervised and generative approaches.

Recently, from 2022 onward, more advanced solutions have emerged to handle large-scale data analytics, domain adaptation, and automated decision-making in VM contexts. For instance, Hsieh *et al.* [9] propose an advanced CNN-based automatic VM pipeline incorporating selective training schemes to address drift and imbalanced data, and Ji *et al.* [10] leverage multi-task Gaussian Process (GP) modeling alongside adaptive active learning to reduce metrology burdens without sacrificing prediction accuracy. Xu *et al.* [12] demonstrate a data-driven, adaptive VM for multibatch wafer yield prediction, underscoring the need for robust model adaptability. Meanwhile, Zajec *et al.* [13] and Kim *et al.* [14] delve into few-shot learning strategies to tackle semiconductor defect detection and wafer mapping tasks, proving that meta-learning can swiftly generalize to novel process scenarios. Building on these empirical fronts, Ji *et al.* [15] present formal convergence results for multi-step MAML-based methods, reinforcing the viability of rapid VM calibration under tight labeling constraints.

B. Major Challenges in Practical VM Applications

Despite extensive evidence showing that VM can reduce measurement costs, improve output, and enhance quality control, several pivotal challenges remain:

1) Domain Adaptation Across Tools and Recipes:

Frequent changes in machine tools, recipe parameters, or sensor conditions lead to distribution shifts, which may severely degrade a VM model trained on older data [9]. Achieving robust cross-tool, cross-recipe scalability is thus crucial.

2) Limited Labels and High Measurement Costs:

Although measuring every wafer provides the richest training data, actual metrology stations are expensive and capacity-constrained, forcing engineers to restrict sampling. This shortage of labeled data hampers accurate model training. Ji et al. utilized multi-task Gaussian Process (GP) modeling with active learning to query the most informative wafers based on uncertainty, achieving high prediction accuracy with far fewer measurements [10].

3) Long-Term Stability and Self-Updating:

Semiconductor processes inevitably drift because of seasonal variations, equipment maintenance, or cleaning cycles, causing a static VM model to become obsolete [2] [11]. Baek et

al. introduced a "recursive update" scheme to recalibrate sensor centering after preventive maintenance (PM), thus preserving consistent VM performance [11].

4) Uncertainty Quantification and Decision Support:

In a high-stakes manufacturing environment, returning only a single numeric prediction can be risky. If VM could output its uncertainty for each wafer, engineers could better identify which wafers warrant physical measurements and which can be left to VM estimates [5] [12]. Wan and McLoone, for example, integrated the predictive variance of Gaussian Process regression into R2R control, adapting the control aggressiveness via confidence intervals [12].

C. Motivations and Contributions of This Study

Motivated by these open issues, this research proposes the Generative-FewShot-Active Virtual Metrology (GFA-VM) architecture, integrating large-scale generative modeling (e.g., VAE or GAN), few-shot fine-tuning, and uncertainty-driven active sampling into a unified framework. Our work offers the following novel contributions:

1) Generative Foundation Model: We employ VAE or GAN-like generative methods during the offline stage to learn a broad sensor-data representation, serving as a robust basis for different recipes, machines, or sensor distributions.
2) Uncertainty Evaluation and Active Sampling: During online inference, the uncertainty is calculated, and those exceeding a dynamic threshold are flagged for measurement. By targeting high-risk or out-of-distribution wafers, the system achieves accurate adaptation with minimal labeling effort.
3) Few-Shot Fine-Tuning and Automated Maintenance: To accommodate new recipes or machine upgrades, the model can be recalibrated with only 1–5 critical labeled samples, avoiding extensive retraining.
4) Dynamic Threshold Adjustment: To balance measurement cost and prediction accuracy, we periodically (or on-demand) evaluate prediction error and sampling frequency, automatically revising the uncertainty trigger threshold. This adaptive loop maintains stable accuracy while conserving metrology resources.

In sum, the GFA-VM architecture major challenges in VM—including high-dimensional sensor data, constrained labeling budgets, distribution shifts, and real-time decision support—through a data-driven and uncertainty-aware paradigm. By unifying large-scale generative representations, minimal-sample adaptation, and strategic active sampling, our approach offers a solution that is both theoretically grounded and practically implementable in advanced semiconductor manufacturing lines.

II. Generative-FewShot-Active Virtual Metrology

This research proposes a GFA-VM system that integrates large-scale generative modeling (e.g., VAE or GAN), few-shot fine-tuning, and an uncertainty-driven active sampling mechanism to create a purely data-driven and continuously updatable solution. Compared to traditional VM approaches, which often rely heavily on physical priors or expert domain knowledge and require extensive re-training whenever a new recipe or tool is introduced, our system offers three key advantages:

1) Enhanced Adaptability: By leveraging a generative foundation model trained on diverse time-series sensor data, the system can swiftly adapt to new recipes or tools with only a handful of critical samples.
2) Reduced Measurement Cost: Through an uncertainty-based active sampling framework, the system pinpoints only the "most needed" wafers for physical measurement, resulting in considerable savings on metrology resources and engineering time.
3) Continuous Self-Calibration: A closed-loop process with dynamic threshold adjustment ensures the system remains robust against slow drifts or changes in manufacturing conditions without the burden of frequent large-scale retraining.

From an industrial perspective, this solution can seamlessly interface with existing Manufacturing Execution Systems (MES) and metrology stations in semiconductor fabs, delivering real-time predictions of wafer quality (e.g., final thickness) alongside a confidence estimate. Whenever the model encounters data points that lie outside its confident operating range, it automatically requests actual measurements, uses those newly acquired labels for quick parameter updates, and thus maintains high accuracy in the face of evolving production conditions. In contrast to conventional VM pipelines that require periodic bulk re-training, our approach is more responsive and cost-effective—especially valuable in highly dynamic manufacturing environments.

This section addresses the architecture from a systems perspective and describes the five primary functional components: (i) Offline Data Preparation and Generative Foundation Model Construction, (ii) Online Real-Time Inference and Uncertainty Estimation, (iii) Active Sampling and Incremental Labeling, (iv) Few-Shot Fine-Tuning and Rapid Model Adaptation, and (v) Ongoing Iteration and Dynamic Threshold Adjustment. These components center on data-driven insights and uncertainty-based decisions, forming a fully automated yet continuously evolving VM framework.

A. The Architecture of GFA-VM

The overall solution can be divided into an offline stage (for large-scale model training) and an online stage (for real-time prediction and on-demand updates). During the offline stage, historical sensor data and partial measurements are used to build a generative foundation model capable of capturing broad variations in equipment and recipes. In the online stage, real-time data are fed into the system to generate immediate predictions and uncertainties, triggering active sampling and minimal fine-tuning when necessary.

Figure 1 illustrates the five functional components within this architecture. The diagram highlights how raw data and partial labels flow from historical archives to produce a robust model, which is then deployed to the manufacturing line. Whenever the system detects wafers with high uncertainty, it coordinates with metrology stations (through MES or other control systems) to obtain ground-truth labels and update its parameters. This closed-loop arrangement ensures that model performance remains stable without frequent large-scale retraining. The approach readily integrates with typical semiconductor infrastructures, including MES, Advanced

Process Control (APC) platforms, and Equipment Data Acquisition (EDA) systems, enabling an automated and data-centric mode of operation.

B. Offline Data Preparation and Generative Foundation Model Construction

1) Data Integration:

The first functional component operates in the offline stage, with the primary goal of creating a robust "foundation model" that generalizes across multiple tools, recipes, and sensor distributions. In modern semiconductor fabs, sensor readings and measurement logs are often scattered among different databases. The system thus starts by integrating time-series signals, process conditions, recipe identifiers, and existing ground-truth measurements into a coherent dataset. Standard data-cleaning and alignment procedures (e.g., filtering out invalid timestamps, re-sampling sensor data to uniform intervals) ensure high data quality and consistency.

2) Generative Model Architecture:

Building on this unified dataset, the system employs a large-scale generative neural network—often a combination of Transformer-based encoders and a VAE or GAN—to learn the underlying time-series distributions. Unlike traditional VM methods that focus predominantly on supervised outputs, a generative approach leverages unlabeled samples to refine the model's internal representation, thereby enhancing robustness to sensor noise or equipment shifts. Whenever certain wafers already have final measurement labels, these can be incorporated into the training loop as partial supervision on top of the generative backbone.

3) Offline Training Protocol:

After data integration and model setup, we move to the offline training phase, which typically runs on high-performance servers or cloud platforms. We collect a large volume of wafer sensor data, including both labeled (for the supervised objective) and unlabeled (for the generative objective) samples.

i. Data Cleaning and Alignment:

Each time-series is synchronized by recipe ID, and any missing or inconsistent readings are either imputed or excluded based on predefined quality criteria.

ii. Loss Components:

Our Transformer-based VAE (or GAN) backbone combines three main loss terms:

- A reconstruction loss $\mathcal{L}_{\text{rec}}$ for the VAE decoder (or a discriminator loss if using GAN),
- A KL-divergence term $\mathcal{L}_{\text{KL}}$ to align the latent space with a prior distribution,
- A supervised MSE term $\mathcal{L}_{\text{sup}}$ for labeled wafers. Specifically, if θ denotes the entire model parameters, the combined offline objective is: $L_{\text{pretrain}}(\theta) = \alpha \times \mathcal{L}_{\text{rec}}(\theta) + \beta \times \mathcal{L}_{\text{KL}}(\theta) + \gamma \times \mathcal{L}_{\text{sup}}(\theta)$. Where α, β, and γ balance the generative and supervised parts.

iii. Hyperparameter Choices:

We typically use Adam with a batch size of 32–64 and a learning rate around 1e-4 to 1e-3, tuned via cross-validation on a small held-out dataset. Training proceeds for 50–100 epochs or until the validation loss plateaus.

iv. Resulting Model Parameters:

This process yields base parameters θ_{base} or, in a meta-learning setup, θ_{meta}. They capture essential time-series patterns and serve as a robust initialization for subsequent online adaptation. Because the offline stage is computationally intensive, it is generally executed outside the production line, ensuring no disruption to real-time manufacturing operations.

In summary, this offline protocol integrates diverse sensor data, applies both unsupervised (generative) and supervised learning objectives, and delivers a foundation model capable of handling recipe/tool variations. The next sections discuss how we deploy θ_{base} (or θ_{meta}) in real-time settings, continuously refining it to accommodate changing process conditions.

C. Online Real-Time Inference and Uncertainty Estimation

Once the generative foundation model is established, it can be deployed in the live production environment to deliver real-time VM predictions. When a new wafer enters a particular process chamber or step, the system automatically acquires the relevant sensor data (e.g., temperature, pressure, plasma current) and any associated recipe parameters or tool IDs. The previously learned model then provides an immediate forecast of the wafer's final quality measure, such as thickness or uniformity.

Unlike conventional VM pipelines, this process also yields an uncertainty estimate for each prediction. If the system identifies wafers whose inference falls outside its confident range, it will flag those as high-risk candidates. This design allows early intervention—either by an engineer or by an advanced process control (APC) loop—before additional cost or time is invested in processing potentially defective wafers. Thus, the "Online Real-Time Inference and Uncertainty Estimation" component not only delivers an instantaneous proxy for metrology readings but also tracks how confident the model is about each particular wafer, setting the stage for cost-effective measurement decisions.

D. Active Sampling and Incremental Labeling Module

Because metrology equipment and operator labor can be expensive in a semiconductor fab, it is rarely feasible to measure every wafer on the production line. The "Active Sampling and Incremental Labeling" component addresses this challenge by using the system's uncertainty to focus measurement efforts on only the most critical or high-risk wafers. Once identified, these wafers are directed to metrology stations or labs to obtain ground-truth metrics (e.g., thickness values). The system, through interfaces with the MES or APC, issues measurement requests, and upon measurement completion, real-world quality data are appended to the incremental dataset D_{new} .

This uncertainty-driven sampling procedure constitutes a more intelligent use of metrology resources compared to traditional random sampling or fixed-percentage inspection. By concentrating on regions of the wafer space the model is less confident about, the system rapidly acquires labels that effectively reduce prediction error and improve reliability. It also ensures that the measurement load stays proportionate to the degree of variation in the ongoing process—an especially vital requirement when dealing with equipment limitations and high production throughput.

E. Few-Shot Fine-Tuning and Rapid Model Adaptation

Given the frequent changes in tools or recipes on a semiconductor production line, an offline model can quickly become obsolete if left un-updated. The "Few-Shot Fine-Tuning and Rapid Model Adaptation" component provides a remedy by enabling quick parameter adjustments using the minimal labeled samples collected in the previous step. By leveraging the robust, general-purpose representations established during offline training, the model can recalibrate its predictions for new distributions with as few as 1–5 new wafer samples.

After pre-training, the GFA-VM framework deploys the learned parameters "θ_{base}" During real-time inference, each wafer's predicted quality "$\widehat{\boldsymbol{T}}$" is accompanied by an uncertainty estimate "σ^2" via Monte Carlo Dropout. If "σ^2" exceeds a dynamic threshold "τ," that wafer is physically measured, and its true label "$\boldsymbol{T}_i$" is appended to an incremental dataset "$\mathcal{D}_{\text{new}}$." We then perform few-shot updates on "θ_{base}" (or "θ_{meta}," if MAML is adopted). For example, with linear probing$\theta_{\text{linear}}^{*} = \arg\min_{\theta_{\text{linear}}} \sum_{(\mathbf{x}_i,\mathbf{c}_i,\boldsymbol{T}_i)\in\mathcal{D}_{\text{new}}} \left\| \boldsymbol{T}_i - f_{\theta_{\text{base}},\theta_{\text{linear}}}(\mathbf{x}_i,\mathbf{c}_i) \right\|^2$. In MAML-based scenarios, we apply one or two gradient steps using the newly collected samples $\theta_{\text{adapt}} = \theta_{\text{base}} - \alpha_{\text{inner}} \nabla_\theta \mathcal{L}_{\text{sup}}(\theta, \mathcal{D}_{\text{new}})$. Here, α_{inner} is the inner-loop learning rate, usually smaller than the outer loop's rate for stability. Whenever "θ" is updated, we recalculate the threshold "τ" by evaluating recent prediction errors and sampling frequencies. If the overall error rises or we detect a process shift, we lower "τ" to request more measurements. Conversely, if performance is stable, "τ" can be gradually increased to reduce metrology overhead. This continual adjustment ensures that the GFA-VM system remains both cost-effective and robust under evolving process conditions.

F. Ongoing Iteration and Dynamic Threshold Adjustment

Over extended periods, semiconductor processes and equipment may experience gradual drifts or step changes due to wear, upgrades, or recipe refinements. The final functional component in this system—"Ongoing Iteration and Dynamic Threshold Adjustment"—continually evaluates the model's performance and resource usage, adapting the uncertainty threshold as needed. If the fab experiences a sudden surge in process variability or unforeseen anomalies, the threshold is automatically lowered to increase the sampling rate and gather more ground-truth data. Conversely, if the process is stable and the model's predictions remain accurate, the threshold can be raised to reduce metrology costs.

By embedding threshold adaptation into a monitoring and feedback loop, the system gains the capacity for "self-growth." It can remain cost-efficient under normal, stable production conditions, yet seamlessly pivot to more aggressive sampling whenever significant distributional changes are detected. Compared to fixed sampling ratios or purely manual approaches, this dynamic thresholding strategy grants the fab a balanced trade-off between labeling investment and model accuracy, sustaining the benefits over long-term production.

G. System Integration and Potential Contributions

The interplay among these five functional components yields a cohesive and adaptive VM closed-loop. Offline data preparation and a generative foundation model enable broad coverage of process diversity, while the online inference and uncertainty estimation module provides immediate quality proxies and triggers when encountering questionable wafers. Active sampling and incremental labeling judiciously allocate measurement resources, and few-shot fine-tuning updates the model on-the-fly to accommodate new distributions. Finally, ongoing iteration and dynamic thresholding ensure that the approach remains robust to both short-term shifts and long-term drifts in manufacturing processes.

From a practical standpoint, this integrated design highlights key distinctions compared to conventional VM systems. It prioritizes a data-driven approach that updates "as needed" based on uncertainty signals rather than relying on extensive prior domain expertise. For semiconductor fabs, the most prominent benefits include faster new-recipe trials, less risk of over-sampling or under-sampling, and significantly reduced measurement overhead. Engineers can deploy tool or process changes with minimal downtime and maintain strong predictive performance via a handful of newly measured wafers. Ultimately, the system delivers not only high predictive accuracy but also deeper, real-time insight into where and why uncertainties arise, facilitating more effective process control.

H. Real-World Integration and Deployment Considerations

While the preceding sections describe the core GFA-VM architecture from a functional standpoint, successful industrial implementation requires additional attention to deployment logistics and fail-safe mechanisms. In this subsection, we present key insights gathered from an early-stage production pilot, focusing on MES communication latency, sensor data synchronization, model versioning, and contingency planning to ensure reliability under unexpected operational events.

1) MES Communication and Latency Management:
   - Infrastructure Setup: GFA-VM is deployed as a microservice within the on-premise computing cluster, interfacing with the Manufacturing Execution System (MES) through a lightweight RESTful API. This design allows the MES to automatically dispatch time-series sensor data to the GFA-VM prediction endpoint.
   - Round-Trip Latency: Under typical production loads, end-to-end latency (i.e., from wafer completion to receiving a final metrology prediction) averages approximately 2–3 seconds. This delay includes data transfer to the GFA-VM service, inference, and result transmission back to the MES.
   - Load Spikes and Queueing: To prevent delays when multiple wafers finish processing simultaneously, a buffering mechanism (e.g., a message queue such as RabbitMQ) is employed. This ensures scalability and provides asynchronous data flow, mitigating the risk of congested network conditions or temporary server overloads.
   - Impact on Process Control: A 2–3 second turnaround is generally acceptable for run-to-run or lot-to-lot control in high-volume semiconductor fabrication, where subsequent recipe or chamber adjustments occur on minute-to-hour timescales.
2) Sensor Data Synchronization and Quality Assurance:
   - Time-Series Alignment: Semiconductor sensors (e.g., plasma current, chamber pressure) can record at slightly different sampling rates. GFA-VM's data preprocessing module re-samples signals onto a unified timeline (e.g., 1 Hz) based on timestamp alignment, ensuring

consistent feature vectors for the Transformer-based encoder.
- Handling Sensor Interruptions: Unexpected sensor drops or partial outages (~1–2% of runs in our pilot) are addressed via short-gap imputation (e.g., linear or spline-based) when feasible. For longer disruptions, the wafer is flagged as high-uncertainty, triggering a request for physical measurement in line with the active sampling logic.
- Data Validation: Outlier filtering (e.g., checking for physically implausible temperature spikes) is performed before inference to minimize erroneous predictions caused by corrupted sensor signals.

3) Model Versioning and Incremental Update Pipeline:
   - Version Control: Each GFA-VM model instance is assigned a semantic version tag (e.g., v1.2.1), stored alongside training logs and hyperparameter configurations. This enables traceability whenever engineers compare model variants or roll back to a prior, stable release.
   - Incremental Recalibration: Active sampling in the production line leads to frequent, small-scale parameter updates. To maintain uninterrupted operation, new model checkpoints run in a "shadow" mode: predictions are tested against real wafer outcomes in parallel to the current production model. Only after validating performance does the new model replace the old version.
   - Deployment Frequency: The frequency of such incremental deployments depends on manufacturing variability. In our pilot, updates typically occurred weekly, though major recipe shifts could prompt immediate recalibration.
4) Fail-Safe Mechanisms and Redundancy:
   - Fallback Prediction: If GFA-VM detects repeated sensor corruption, high uncertainty, or internal inference failures, it defaults to a simpler baseline VM model or a rule-based system with conservative process parameters. This contingency ensures the fab can continue operating while engineering teams investigate and resolve the anomaly.
   - Automated Alerts: The GFA-VM microservice is integrated with the fab's alert system. Engineers receive real-time notifications whenever uncertainty thresholds are exceeded or sensor data are persistently missing, supporting quick manual intervention if required.
   - Resilience to Drifts and Process Shifts: As described in the architecture, GFA-VM incorporates few-shot updating and dynamic thresholding. These mechanisms mitigate risks associated with gradual equipment wear, recipe updates, or sudden distributional changes, reducing the likelihood of large-scale performance degradation.
5) Pilot Outcomes and Lessons Learned:
   - Latency and Scalability: The 2–3 second response window comfortably meets run-to-run control requirements. With microservices and message queueing in place, scaling out to multiple parallel inference engines is straightforward should throughput demands increase.
   - Data Quality as a Bottleneck: Several sensor misalignments and sporadic device outages highlighted the importance of robust data synchronization and cleaning procedures. Engineering teams found that consistent data checks significantly reduce false alarms and unwarranted sampling costs.
   - Controlled Rollout: Our incremental "shadow deployment" approach allowed new updates to be assessed without disrupting production. This method eased concerns about potential model regressions and expedited engineer buy-in.

In summary, integrating GFA-VM in a real-world fab requires not just algorithmic sophistication but also meticulous engineering around data acquisition, latency management, and robust failover routines. Through the above deployment strategies, we demonstrate that GFA-VM can seamlessly fit into an MES-driven workflow, offering high-accuracy wafer predictions with minimal interference to existing semiconductor production processes.

This section has provided a comprehensive overview of the proposed VM system from an architectural standpoint, describing five major functional components: Offline Data Preparation and Generative Foundation Model Construction, Online Real-Time Inference and Uncertainty Estimation, Active Sampling and Incremental Labeling, Few-Shot Fine-Tuning and Rapid Model Adaptation, and Ongoing Iteration and Dynamic Threshold Adjustment. These modules together underscore a shift away from physically informed or massive retraining paradigms. Instead, the emphasis is on data-driven modeling, efficient adaptation, and continuous feedback, which prove invaluable in semiconductor manufacturing—an environment marked by high operational costs, complex processes, and frequent recipe or tool modifications.

III. Methodology

This research presents a GFA-VM solution combining:

1) A large generative foundation model (Transformer + VAE or GAN),
2) Meta-learning (e.g., MAML) [16] for few-shot adaptation,
3) Active sampling guided by uncertainty estimation (with an automated threshold).

Our aim is to ensure robust, rapid adaptation to novel recipes or tools without physical/engineering priors. For clarity, we divide the entire approach into five Phases (I–V), detailing each step's mathematical formulations thoroughly.

A. Phase I: Offline Data Collection and Foundation Model Pre-training

We denote each wafer record i by $(\mathbf{x}_i, \mathbf{c}_i, \boldsymbol{T}_i)$, where:

$\mathbf{x}_i \in \mathbb{R}^{T\times d}$: time-series sensor data (length T, dimension d); $\mathbf{c}_i \in \mathbb{R}^k$: process parameters or "meta-data" (e.g., recipe ID, tool ID, relevant setpoints). $\boldsymbol{T}_i \in \mathbb{R}$: real-valued thickness or other final quality measurement (only available for a fraction of wafers). If a wafer lacks label T_i, it can still be used for

unsupervised (generative) training. We collect N such records, forming dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{c}_i, \boldsymbol{T}_i)\}_{i=1}^{N}$.

1) Transformer + VAE: Model Architecture:

Let θ represent all parameters in the generative foundation model, which comprises:

i. Transformer Encoder:

- Input Projection: The time-series $\mathbf{x}_i$ of shape $(T \times d)$ is projected to an internal dimension d_{model} , often with a learnable linear map $\mathbf{W}_{\text{in}}$.
- Positional Encoding: Either sinusoidal or learned embeddings $\mathbf{p}_t$ are added to each time step $t \in [1, \ldots, T]$.
- Multi-head Self-Attention: For each transformer layer ℓ and attention head h, queries $\boldsymbol{Q}$, keys $\boldsymbol{\mathcal{K}}$, and values $\boldsymbol{\mathcal{V}}$ are computed as:

$$\boldsymbol{Q} = \mathbf{X}\mathbf{W}_{\mathcal{Q}}^{(\ell,h)}, \boldsymbol{\mathcal{K}} = \mathbf{X}\mathbf{W}_{\mathcal{K}}^{(\ell,h)}, \boldsymbol{\mathcal{V}} = \mathbf{X}\mathbf{W}_{\mathcal{V}}^{(\ell,h)}, \quad (1)$$

where $\mathbf{X}$ is the current hidden representation, and $\mathbf{W}_{\mathcal{Q}}^{(\ell,h)}$, $\mathbf{W}_{\mathcal{K}}^{(\ell,h)}$, $\mathbf{W}_{\mathcal{V}}^{(\ell,h)}$ are parameter matrices. The single-head attention is:

$$\text{Attn}(\boldsymbol{Q}, \boldsymbol{\mathcal{K}}, \boldsymbol{\mathcal{V}}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{\mathcal{K}}^{\top}}{\sqrt{d_k}}\right)\boldsymbol{\mathcal{V}}, \quad (2)$$

with d_k typically $\frac{d_{model}}{\#heads}$. The multi-head version concatenates multiple heads' outputs.

- Feed-Forward: Each layer also includes a pointwise feed-forward network, e.g.,

$$\mathbf{H} = \text{ReLU}(\mathbf{X}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2. \quad (3)$$

- After L layers, we derive a final embedding $\mathbf{z}_i \in \mathbb{R}^h$ by global average pooling or by using the last time step's hidden state.

ii. Variational Autoencoder

- Latent Distribution: In the VAE scenario, the encoder network yields $\boldsymbol{\mu}_i \in \mathbb{R}^h$ and $\boldsymbol{\sigma}_i \in \mathbb{R}^h_{>0}$. We sample:

$$\mathbf{z}_i = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}, \ \ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (4)$$

- Decoder: A separate (possibly reversed-Transformer or transposed-convolution) network maps $\mathbf{z}_i$ to $\hat{\mathbf{x}}_i$ of shape $(T \times d)$.

iii. Supervised Prediction Head

- We augment the model with a small MLP that takes $\mathbf{z}_i$ (or a specialized head from the encoder) plus $\mathbf{c}_i$ to predict measurement:

2) Pre-training Losses:

We optimize three major losses (VAE-based):

i. Reconstruction Loss, $\mathcal{L}_{\text{rec}}$

$$\mathcal{L}_{\text{rec}}(\theta) = \frac{1}{N_r}\sum_{i=1}^{N_r}\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2, \quad (5)$$

where N_r is the number of samples used for reconstruction (both labeled and unlabeled). The norm can be MSE across all time steps, i.e.,

$$\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 = \sum_{t=1}^{T}\sum_{d'=1}^{d}(x_{i,t,d'} - \hat{x}_{i,t,d'})^2. \quad (6)$$

ii. KL Divergence, $\mathcal{L}_{\text{kl}}$

$$\mathcal{L}_{\text{kl}}(\theta) = \frac{1}{N_r}\sum_{i=1}^{N_r} D_{\text{KL}}(q_\theta(\mathbf{z}|\mathbf{x}_i, \mathbf{c}_i)\|p(\mathbf{z})), \quad (7)$$

ensuring $\mathbf{z}_i$ aligns with a standard prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. For a Gaussian encoder,

$$D_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\mathbf{I})\|\mathcal{N}(\mathbf{0}, \mathbf{I})) = \frac{1}{2}\sum_{r=1}^{h}(\sigma_{i,r}^2 + \mu_{i,r}^2 - \ln\sigma_{i,r}^2 - 1). \quad (8)$$

iii. Supervised Loss, $\mathcal{L}_{\text{sup}}$

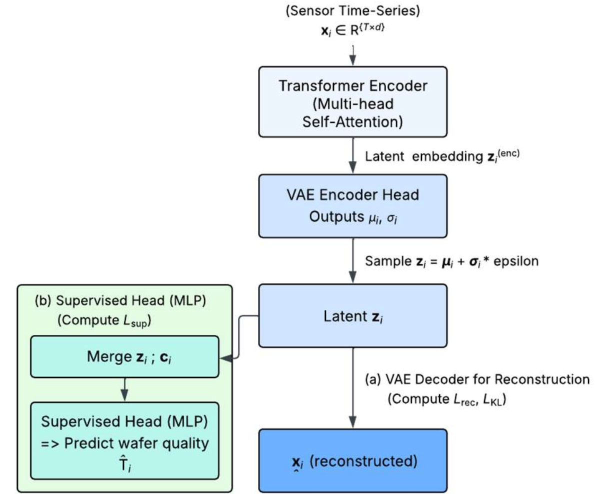

Fig. 2. Transformer–VAE Architecture with a Supervised Head.

$$\mathcal{L}_{\text{sup}}(\theta) = \frac{1}{N_s}\sum_{i=1}^{N_s}(\boldsymbol{T}_i - \widehat{\boldsymbol{T}}_i)^2, \quad (9)$$

over the labeled subset $(\mathbf{x}_i, \mathbf{c}_i, \boldsymbol{T}_i)$ with size N_s.

By combining these terms,

$$\mathcal{L}_{\text{pretrain}}(\theta) = \alpha\mathcal{L}_{\text{rec}}(\theta) + \beta\mathcal{L}_{\text{kl}}(\theta) + \gamma\mathcal{L}_{\text{sup}}(\theta), \quad (10)$$

where α, β, γ are scalar hyperparameters. Minimizing (10) via SGD (e.g., Adam) yields the initial parameters θ_{base} .

Figure 2 illustrates how the Transformer-based encoder, VAE encoder/decoder, and supervised head integrate to handle both labeled and unlabeled data. The model samples a latent vector $\mathbf{z}_i$ for reconstruction (computing $\mathcal{L}_{\text{rec}}$, $\mathcal{L}_{\text{kl}}$) and concatenates $\mathbf{z}_i$ with recipe metadata $\mathbf{c}_i$ to produce $\widehat{\boldsymbol{T}}_i$ under $\mathcal{L}_{\text{sup}}$."

B. Phase II: Online Inference and Uncertainty Estimation

1) Online Inference:

Once deployed, the model (parameters θ) receives new wafer $(n+1)$ with sensor data $\mathbf{x}_{n+1}$ and process conditions $\mathbf{c}_{n+1}$. The predicted measurement is:

$$\widehat{\boldsymbol{T}}_{n+1} = f_\theta(\mathbf{x}_{n+1}, \mathbf{c}_{n+1}). \quad (11)$$

If no adaptation has been performed, $\theta = \theta_{base}$. If updated in later phases, $\theta = \theta_{\text{adapt}}$.

2) Monte Carlo Dropout and Uncertainty:

To measure predictive uncertainty, we adopt Monte Carlo (MC) Dropout [17]. Specifically:

i. Perform S forward passes, each with dropout randomly applied in intermediate layers, leading to predictions $\{\widehat{\boldsymbol{T}}_{n+1}^{(s)}\}_{s=1}^{S}$.

ii. Compute the sample mean:

$$\overline{\boldsymbol{T}}_{n+1} = \frac{1}{S}\sum_{s=1}^{S}\widehat{\boldsymbol{T}}_{n+1}^{(s)}. \quad (12)$$

iii. Compute the empirical variance:

$$\sigma_{n+1}^2 = \frac{1}{S}\sum_{s=1}^{S}(\widehat{\boldsymbol{T}}_{n+1}^{(s)} - \overline{\boldsymbol{T}}_{n+1})^2. \quad (13)$$

If σ_{n+1}^2 exceeds a threshold τ. we consider the wafer's prediction OOD (high risk).

3) Automated Threshold τ via ATL:

Rather than fixing τ arbitrarily, we apply an Automated Threshold Learning (ATL) approach:

i. Offline or Validation-based Fitting

Using a validation set, collect σ_i^2 and ground-truth error $e_i = (\boldsymbol{T}_i - \widehat{\boldsymbol{T}}_i)^2$. Explore possible τ values, record the

Algorithm 1: Generative-FewShot-Active VM (GFA-VM)

Inputs:
- Offline dataset $\mathcal{D}_{\text{offline}}$ (labeled + unlabeled)
- Pre-trained backbone θ_{base} (or θ_{meta} if MAML used)
- Hyperparameters: α, β, γ (loss coefficients)
  α_{inner} (few-shot step size)
  S (MC Dropout passes)
- Initial threshold τ_{init}
- Online wafer stream: $(\mathbf{x}_t, \mathbf{c}_t)$ for $t = 1..T$
- Batch size B, maximum wafers to measure per batch l

Outputs:
- Predicted wafer quality $\hat{\boldsymbol{T}}_t$
- Optional actual measurement $\boldsymbol{T}_t$ if wafer is sampled
- Updated model parameters θ_{adapt} (online)
- Dynamic threshold τ

```
---------------------------------------------------------------------
# OFFLINE INIT (already completed before real-time deployment)
1:  # Train Transformer+VAE (or GAN) offline using:
2:  #  L_pretrain(θ) = α × L_rec(θ) + β × L_KL(θ) + γ * L_sup(θ)
3:  # Optionally integrate MAML outer loop if multiple recipes
      available
4:  # Obtain final θ_base (or θ_meta) after offline optimization
5:  τ ← τ_init
---------------------------------------------------------------------
# ONLINE STAGE
6:  for each batch b = 1..B do
7:     # -- Phase II: Inference & Uncertainty --
8:     for each wafer j in batch b:
9:        T̂_j = f_θ(x_j, c_j)       # model prediction
10:       Perform S forward passes (MC Dropout) to estimate:
11:          σ_j^2 = Var(T̂_j^(s)) over s = 1..S
12:    end for

13:    # -- Phase III: Active Sampling --
14:    A ← { top l wafers | σ_j^2 > τ }  # high-uncertainty wafers
15:    for each wafer j in A:
16:       Measure wafer => T_j (actual thickness)
17:       Add (x_j, c_j, T_j) to incremental set D_new
18:    end for

19:    # -- Phase IV: Few-Shot Fine-Tuning --
20:    if D_new not empty then
21:       # Example: Linear Probing or MAML-based adaptation
22:       if linear_probing:
23:          θ*_linear = arg min_{θ_linear} Σ_{i∈D_new} (T_i − f_{θ_base,θ_linear}(x_i, c_i))^2
24:          Update final layer => θ_adapt
25:       else if MAML:
26:          θ_adapt = θ_base − α_inner ∇( L_sup(θ_base, D_new))
27:       end if
28:       # Clear or reduce D_new after adaptation
29:    else
30:       θ_adapt = θ_base   # no update if no new measurements
31:    end if

32:    # Deploy θ_adapt for next wafer/batch
33:    θ ← θ_adapt

34:    # -- Phase V: Dynamic Threshold Adjustment --
35:    Evaluate recent errors or drift => e.g., E_b = average( |T̂_j −
       T_j| ) for j in A
36:    if E_b is high or drift detected then
37:       τ ← lower (τ)       # measure more wafers
38:    else
39:       τ ← raise (τ)       # reduce metrology overhead
40:    end if
41: end for
```

fraction of samples that get flagged for measurement (sampling cost) and resulting final accuracy.

ii. Cost–Accuracy Optimization
Suppose we aim to keep a labeling budget ℓ_{max} or achieve an error target E_{target} . We find τ^* that meets the accuracy while minimizing sampling frequency:

$$\tau^* = \arg\min_{\tau}[\text{Error}\ (\tau) + \lambda \times \text{SamplingRate}(\tau)]. \quad (14)$$

iii. Online Refinement
Because the model is updated over time, we can periodically recompute τ^* to adapt to new distributions.

C. Phase III: Active Sampling and Incremental Labels

1) Trigger Condition:
For each batch $\{\mathbf{x}_i, \mathbf{c}_i\}_{j=1}^{B}$ in online production, we obtain $\hat{\boldsymbol{T}}_j$ and σ_j^2. If $\sigma_j^2 > \tau$, the wafer is deemed "high risk" and is candidate for measurement.
2) Selection Rule:
To limit measurement overhead, we pick only the top ℓ uncertain wafers:

$$\boldsymbol{\mathcal{A}} = Top_\ell(\{\sigma_j^2\}_{j=1}^{B}). \quad (15)$$

Hence, ℓ is typically small (3–5). For each wafer $j \in \boldsymbol{\mathcal{A}}$, we measure its real thickness $\boldsymbol{T}_j$. This newly labeled set is appended to an incremental dataset $\mathcal{D}_{\text{new}}$.

D. Phase IV: Few-Shot Fine-tuning and Rapid Adaptation

When a new recipe or equipment emerges, we usually start with only a handful ($\ell\sim$ 1 to 5) of labeled samples. We adopt parameter-efficient or meta-learning strategies:

1) Linear Probing:
Freeze most of the foundation model's weights (Transformer + VAE). Only update the final linear head. Formally:

$$\theta_{\text{linear}}^* = \arg\min_{\theta_{\text{linear}}} \sum_{j\in\boldsymbol{\mathcal{A}}} \left\| \boldsymbol{T}_j - f_{\theta_{\text{base}},\theta_{\text{linear}}}(\mathbf{x}_i, \mathbf{c}_i) \right\|^2. \quad (16)$$

Minimizing MSE on the newly labeled samples $\boldsymbol{\mathcal{A}}$ rapidly recalibrates the final prediction layer.
2) Adapter / LoRA:
Insert small adapter modules or low-rank factorization in the multi-head attention blocks, freezing other weights. Let ϕ denote these additional parameters:

$$\phi^* = \arg\min_{\phi} \sum_{j\in\boldsymbol{\mathcal{A}}} \left\| \boldsymbol{T}_j - f_{\theta_{\text{base}},\phi}(\mathbf{x}_i, \mathbf{c}_i) \right\|^2. \quad (17)$$

3) Meta-Learning via MAML:
If we integrated Model-Agnostic Meta-Learning (MAML) [16] in Phase I, we treat each recipe or equipment as a sub-task. After learning θ_{meta}, adaptation to a new distribution is done with few labeled points $\boldsymbol{\mathcal{A}}$. We define:

i. Inner Loop
Let $\mathcal{L}_{\text{new}}$ be the local task loss on newly measured samples. We take θ_{meta} and do k gradient steps:

$$\theta_{\text{adapt}} = \theta_{\text{meta}} - \alpha_{\text{inner}} \nabla_\theta [\mathcal{L}_{\text{new}}\ (\theta_{\text{meta}}, \boldsymbol{\mathcal{A}})]. \quad (18)$$

If $\mathcal{L}_{\text{new}}$ includes reconstruction or prior terms, we incorporate them, but typically only the supervised error is crucial for new tasks.

ii. Outer Loop (Offline, in Phase I)
During meta-training, each known recipe $\mathcal{T}_m$ is split into $\mathcal{D}_m^{\text{train}}$ and $\mathcal{D}_m^{\text{test}}$. For sub-task m:

$$\theta_m = \theta - \alpha_{\text{inner}} \nabla_\theta [\mathcal{L}_m\ (\theta, \mathcal{D}_m^{\text{train}})], \quad (19)$$

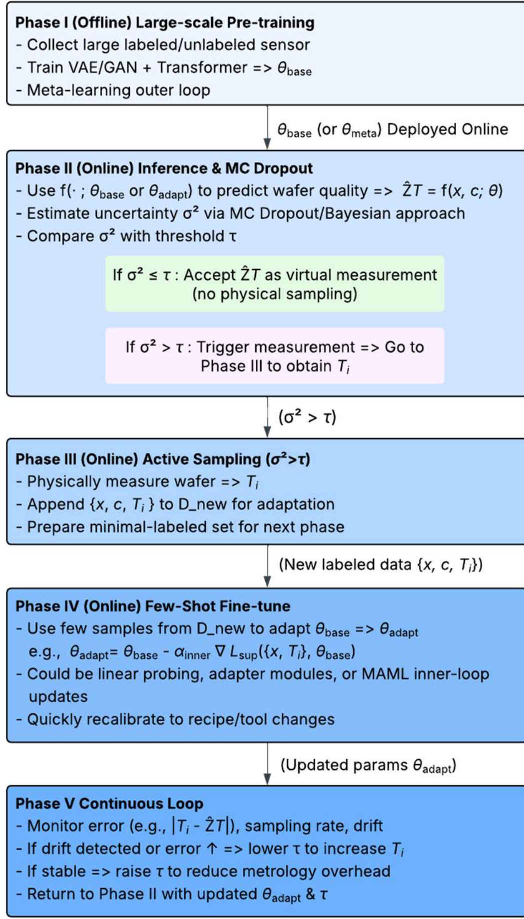

Fig. 3. The Five Phases of GFA-VM.

Then we accumulate the test loss $\mathcal{L}_m\left(\theta, \mathcal{D}_m^{\text{test}}\right)$.Summing across tasks $m = 1, \dots, M$:

$$\theta^* = \arg\min_{\theta} \sum_{m=1}^{M} \mathcal{L}_m\left(\theta, \mathcal{D}_m^{\text{test}}\right). \tag{20}$$

The final $\theta^* = \theta_{meta}$ is used as a well-conditioned initialization for new tasks.

4) Detailed Example for a New Recipe:
Consider five historical recipes $\{R_1, \dots, R_5\}$ used to train θ_{meta} offline. Suppose a new recipe R_{new} arrives:

i. Initially, we get $\ell = 3$ labeled samples from active sampling (because $\sigma_j^2 > \tau$).
ii. We define $\boldsymbol{\mathcal{A}} = \{\mathbf{x}_i, \mathbf{c}_i, \boldsymbol{T}_i\}_{j=1}^{3}$.
iii. Using MAML's inner-loop update (18), within 1–2 gradient steps, we get θ_{adapt}.
iv. θ_{adapt} is deployed for further wafers under R_{new}. Additional samples are measured only if uncertainties remain high.

E. Phase V: Continuous Online Inference and Final Formula

1) Iterative Loop:
Having updated the model to θ_{adapt}, we re-deploy it online. Subsequent wafers again follow Phase II (uncertainty) → Phase III (active sampling) → Phase IV (few-shot update). This iterative loop ensures the system continuously tracks distribution drifts.

2) Final Measurement Prediction:
Ultimately, for wafer $(n + 1)$:

$$\widehat{\boldsymbol{T}}_{n+1} = f_{\theta_{\text{adapt}}}(\mathbf{x}_{n+1}, \mathbf{c}_{n+1}). \tag{21}$$

We also compute σ_{n+1}^2 (13) to decide if sampling is triggered again.

F. The Overall of GFA-VM

Figure 3 summarizes the five phases (I–V)—from offline pre-training to online inference, active sampling, and incremental fine-tuning—including the automated threshold τ^* and optional meta-learning modules. These components reflect a fully data-driven approach, employing large-scale generative learning, meta-learning for few-shot adaptation, and uncertainty-based active sampling to handle new processes or equipment without relying on physical domain knowledge.

Algorithm 1 at the end of this section presents the complete GFA-VM procedure, encapsulating both the offline generative training and the online uncertainty-driven adaptation steps. Below, we briefly reiterate the five phases and key mathematical formulations:

1) Phase I: Offline Pre-training
i. Loss expansions: $\mathcal{L}_{\text{rec}}(5)$, $\mathcal{L}_{\text{kl}}(5)$, and $\mathcal{L}_{\text{sup}}(5)$.
ii. We form a Transformer + VAE foundation model, optionally integrated with MAML outer loop (20).

2) Phase II: Online Inference & Uncertainty
i. MC Dropout formula (12-13), threshold τ to detect OOD samples.
ii. Automated threshold learning (14) to reduce trial-and-error setting of τ.

3) Phase III: Active Sampling
We choose the top ℓ uncertain wafers (15), limiting measurement overhead.

4) Phase IV: Few-Shot Fine-tuning & MAML
Linear Probing (16), Adapter (17), or MAML (18) for rapid adaptation with only 1–5 newly labeled samples.

5) Phase V: Iterative Online Refinement
The updated θ_{adapt} is repeatedly used for future wafers (21).

This integrated system, uniting large-scale generative learning, meta-learning for few-shot adaptation, and active sampling driven by an automated uncertainty threshold, forms a robust, purely data-driven approach to handle new processes or equipment without requiring any explicit physical knowledge.

IV. Theoretical Proofs and Rigorous Analysis

This section provides a mathematically rigorous justification of the method presented in section 3, which combines a large generative model, meta-learning, and uncertainty-driven active sampling for purely data-driven virtual metrology (VM).

The primary objective of this section is to provide formal mathematical proofs and analyses demonstrating the efficacy of our method. Specifically, we aim to show that only a few labeled samples from a new recipe can significantly reduce the prediction error in a bounded number of optimization steps, while an uncertainty thresholdτ^* determined by an automated search scheme can ensure that even under limited measurement budgets, the system meets its target accuracy. Furthermore, we address the potential concern whether incorporating a Variational Autoencoder (VAE) plus a supervised objective (semi-supervised) might jeopardize convergence; we thus

TABLE I.
COMPARISON BETWEEN OUR METHOD AND OTHER CUTTING-EDGE ALGORITHMS.

Methodology	Representative Works (≥2021)	Domain Assumptions	Convergence & Theory	Applicability	Few-Shot Efficiency
Reptile (Baseline)	Nichol & Schulman (2018) [16]	Tasks from relatively similar distributions	Simple one-step adaptation; no strong global guarantees	Generic meta-learning tasks (classification / regression)	Fast updates in a single gradient step
Prototypical Networks (ProtoNets)	Snell *et al*. [20], Chen & Shi (2023) [21], Zajec *et al*. (2024) [13],	Data representable in a metric space; well-separated features	Solid metric-learning basis; risk of overfitting in ultra-few-shot cases	Visual wafer defect inspection or classification in low-data scenarios	Achieves high accuracy with minimal labeled data
Bayesian Active Learning (BAL)	Rawat *et al*. (2022) [22] Several ensemble-based or GP-based frameworks	Often *i.i.d.* sampling or physics-based simulator; supports iterative queries	Rigor in uncertainty-based sampling; meta-learning policies still evolving	Specialized tasks (e.g., process optimization, R&D-phase)	Reduces sampling cost consistently, suits high-dimensional sensors
MAML (Baseline)	Finn *et al*. (2017) [16] Enhanced variants: MeTAL [23], Hessian-Free MAML [24]	Tasks share a common param init and differentiable structures	Emerging results show near-global convergence in overparameterized regimes	Broad usage in semiconductor VM (new recipe introductions)	Adapt with ~1–5 labeled samples; proven quick fine-tuning
Recent Variants & Generative Meta-Learning	Sheynin *et al*. (2021) [25], Kim *et al*. (2024) [14]	Generative outputs aligned with real data distributions	Some methods need thorough validation; partial empirical success	Next frontier for advanced processes (e.g., defect detection)	Improves generalization via realistic or domain-driven synthetic data
GFA-VM	Lin *et al*. (2025)	Flexible to recipe/equipment changes; partial supervision possible	Convergence backed by theoretical proofs (Theorem 1–3); can be combined with uncertainty thresholding	Broadly applicable for advanced semiconductor APC / real-time lines	Rapid adaptation with minimal new labels; handles drifting conditions

present local convergence guarantees. Finally, we compare our approach with other leading algorithms, clarifying the distinctive contributions and limitations of our solution.

A. Comparison with State-of-the-Art Algorithms

Recent advances in few-shot learning, meta-learning, and the integration of generative models have significantly enhanced the applicability of data-driven methods in manufacturing—particularly in semiconductor virtual metrology (VM). While earlier benchmarks (e.g., Reptile [19], Prototypical Networks [20], Bayesian Active Learning, and classic MAML [16]) still provide essential baselines, newer approaches propose faster convergence, better theoretical guarantees, and stronger resilience to non-stationary conditions. In industrial contexts where acquiring labeled data is expensive or logistically constrained, these improved methods deliver greater practical impact.

1) Enhanced Gradient-Based Meta-Learning (MAML and Variants):

Although Model-Agnostic Meta-Learning (MAML) remains a powerful baseline, recent works emphasize more robust convergence and adaptability. For instance, Meta-Learning with Task-Adaptive Loss Functions (MeTAL) [23] c ustomizes the inner-loop objective for each task, thereby accommodating greater task diversity. Moreover, new theoretical studies show that under over-parameterized networks, MAML can achieve global convergence at a linear rate [24], removing lingering concerns about only local optima in deep non-convex landscapes. In semiconductor manufacturing, gradient-based meta-learners have already shown the ability to rapidly adapt to novel processes or recipe changes with only a handful of labeled wafers [14]. One practical caveat is the assumption that new tasks (e.g., new equipment conditions) roughly follow the same distribution as the historical ones. If this assumption is violated, adaptation performance may suffer unless combined with advanced domain adaptation or transfer-learning techniques.

2) Prototype-Based Few-Shot Methods and Their Extensions:

Prototypical Networks (ProtoNets) excel in few-shot classification tasks by learning class prototypes in a metric space. However, recent work shows that in highly complex or extremely low-data scenarios, ProtoNets may overfit without additional strategies. To mitigate this, researchers have combined ProtoNets with generative augmentation, such as diffusion models, to generate synthetic support data [21]. In parallel, some studies integrate anomaly mapping with ProtoNets to handle visual inspection or defect detection [13]. This extension is highly relevant in semiconductor fabs, where certain defect or fault modes appear only sporadically, rendering purely supervised methods impractical. Although these extended ProtoNet-based frameworks can substantially improve few-shot efficiency, they introduce additional computational overhead—such as training generative models or

anomaly detectors—and require thorough validation of synthetic data fidelity.

3) Bayesian/Active Learning under Meta-Learning Frameworks:

Traditional Bayesian Active Learning (BAL) strategies, which select samples based on estimated uncertainty or expected information gain, continue to be refined in manufacturing. The latest developments often meta-learn the sampling policy itself by leveraging previously solved tasks or domain simulations (e.g., TCAD models) [22]. This approach can dramatically reduce the number of physical experiments needed to reach target accuracy—particularly in high-cost or R&D-phase process development, where each measurement is expensive. However, full Bayesian or ensemble-based methods can be computationally intensive, especially for large-scale, high-dimensional sensor datasets. In practice, a hybrid ensemble strategy—where multiple fine-tuned heads approximate a posterior distribution—often yields a good balance of predictive robustness and computational feasibility [8].

4) Generative Model-Integrated Few-Shot Techniques:

Perhaps the most notable recent trend is the integration of GANs, VAEs, or diffusion models with few-shot/meta-learning frameworks to address data scarcity directly [15, 25]. For instance, Du *et al.* [25] utilize a GAN-based approach to generate "normal" samples for anomaly detection, effectively labeling deviations as defects when the discriminator fails to reconstruct certain regions. In other cases, domain simulators (e.g., for etch profiles or wafer maps) serve as physics-based generative tools. By pre-training on vast simulated data and fine-tuning with only a few real samples, researchers achieve remarkable gains in tasks like wafer defect classification [14]. The primary advantage lies in bridging the gap when real training data is costly, but one must ensure domain fidelity in the generated samples. Nonetheless, these hybrid approaches underscore how generative augmentation and advanced meta-learning collectively alleviate the overhead of extensive metrology data collection.

Table I summarizes a high-level comparison between these newly emerging methods and the earlier baselines, focusing on domain assumptions, theoretical guarantees, applicability to semiconductor tasks, few-shot efficiency, and viability of generative integration.

B. Uncertainty Threshold τ and Active Sampling: Mathematical Proofs

Let $\boldsymbol{\sigma}^2(\mathbf{x},\mathbf{c};\theta)$ denote the model's uncertainty for sample $(\mathbf{x},\mathbf{c})$. Following (14) and (15), the system automatically searches for an optimal τ^* to balance sampling frequency and prediction accuracy. We present a formal theorem and proof here.

1) Theorem 1 (Convergence and Label Cost Bounds for Active Sampling)

Suppose there exists a monotonically increasing, L-Lipschitz continuous function $g: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that for any sample $(\mathbf{x},\mathbf{c})$, its prediction error $\mathrm{Err}(\mathbf{x},\mathbf{c})$ is approximated by

$$\mathrm{Err}(\mathbf{x},\mathbf{c}) \approx g(\sigma^2(\mathbf{x},\mathbf{c};\boldsymbol{\theta})). \tag{22}$$

Using the automated threshold learning from (14) to find τ^*, and selecting the top-ℓ highest-uncertainty samples (15) for measurement, the overall label cost can be contained within ℓ (or the budget B_{max}), while the final average error remains below a desired threshold ϵ.

2) Proof:

Let $\mathrm{Err}(\mathbf{x},\mathbf{c}) = |\boldsymbol{T}_i - \widehat{\boldsymbol{T}}_i|$. Consider a batch size B. By hypothesis, there is a near-monotonic relationship between Err and σ^2, and

$$|\mathrm{Err}(\mathbf{x}_{i_1},\mathbf{c}_{i_1}) - \mathrm{Err}(\mathbf{x}_{i_2},\mathbf{c}_{i_2})| \leq L|\sigma_{i_1}^2 - \sigma_{i_2}^2|. \tag{23}$$

From (14), the system searches for a τ that balances "active sampling frequency" and "overall error." Once τ^* is found, only samples with $\sigma_j^2 > \tau^*$ are measured. Let ℓ be the maximum number of such samples, ensuring $\ell \leq B_{\max}$. These newly labeled samples in $\boldsymbol{\mathcal{A}}$ then undergo a few-shot update (16) or (18), significantly reducing their high errors.

Because Err correlates strongly with σ^2, the top-ℓ uncertain samples typically dominate the overall average error. Thus, once they are refined, the system's mean error drops below ϵ. Formally, the Lipschitz continuity ensures that searching over τ reveals a unique or finite set of solutions for bounding $\mathrm{Err} \leq \epsilon$. Hence, with at most ℓ measurements per batch, we achieve the target error floor ϵ. Theorem 1 follows.

C. MAML Meta-Learning and Few-Shot Adaptation: Formal Proofs

We let θ_{meta} be the initialization learned by the outer loop in (20). If the new recipe (task) is sufficiently similar to existing sub-tasks, a limited number of inner-loop updates should yield rapid convergence.

1) Theorem 2 (Convergence of MAML for a New Recipe):

Assume $\mathcal{L}_{\text{new}}$ is β-smooth in a neighborhood of θ. Also, let $\|\theta_{\text{new}}^* - \theta_{\text{meta}}\| \leq \delta$, indicating that the new recipe's optimal solution θ_{new}^* is not too distant from θ_{meta}. Then, by performing the inner-loop gradient descent (18) for k steps, we have

$$\mathcal{L}_{\text{new}}(\theta_{\text{adapt}}) - \mathcal{L}_{\text{new}}(\theta_{\text{new}}^*) \leq \eta, \tag{24}$$

for some small η, where k depends linearly or sublinearly on δ (given a learning rate $\alpha_{\text{inner}} \leq 1/\beta$).

2) Proof:

Let $\theta_0 = \theta_{\text{meta}}$. Each inner-loop update is:

$$\theta_{t+1} = \theta_t - \alpha_{\text{inner}} \nabla_\theta \mathcal{L}_{\text{new}}(\theta_t, \boldsymbol{\mathcal{A}}). \tag{25}$$

With β-smoothness, a standard one-step bound [15] gives:

$$\mathcal{L}_{\text{new}}(\theta_{t+1}) \leq \mathcal{L}_{\text{new}}(\theta_t) - \frac{\alpha_{\text{inner}}}{2} \|\nabla \mathcal{L}_{\text{new}}(\theta_t)\|^2. \tag{26}$$

Repeating for k updates, we get a geometric or sublinear improvement in θ_k. Since $\|\theta_{\text{new}}^* - \theta_{\text{meta}}\| \leq \delta$, the parameter remains in a bounded region where smoothness applies. Therefore, after $k = O(\delta)$ steps, we achieve η-closeness to θ_{new}^*. Theorem 2 is proved.

This result explains why in practice, only 1–5 newly labeled samples suffice to drastically reduce errors under MAML, as per (18).

D. Convergence of VAE + Supervised Loss: Semi-supervised Setting

Finally, we address whether the combined generative (VAE) and supervised terms in $\mathcal{L}_{\text{pretrain}}$ (10) can still converge under semi-supervised conditions. We provide a local convergence guarantee below.

1) Theorem 3 (Local Convergence of Semi-supervised VAE):

Let $\mathcal{L}_{\text{pretrain}}(\theta) = \alpha\mathcal{L}_{\text{rec}}(\theta) + \beta\mathcal{L}_{\text{kl}}(\theta) + \gamma\mathcal{L}_{\text{sup}}(\theta)$. Suppose:

TABLE II.
ILLUSTRATIVE PROCESS PARAMETERS AND EFFECTS

Parameter	Potential Impact / Explanation
Substrate Temperature	Temperature influences chemical reaction rates and crystal growth; excessive heat can promote non-uniform deposition or etching.
Deposition Time	Longer durations generally produce thicker films, but side reactions or saturation can reduce growth rates.
Precursor Flow Rate	Adjusting precursor supply affects film growth rate and quality; overly high or low levels may lead to defects.
Carrier Gas Flow	Regulates precursor concentration and reaction uniformity; extreme flow conditions can lower process efficiency.
Chamber Pressure	Modifies collision frequency and uniformity; excessively low- or high-pressure compromises film consistency.
Plasma Power	Higher power accelerates deposition/etch processes but increases thermal load and surface damage risk.
RF Power	Similar to plasma power, enhancing reactivity; large deviations can degrade surface morphology.
Bias Voltage	Controls ion energy and etch depth; excessively high bias can cause uneven or excessive etching.
Electrode Gap	Affects plasma distribution for deposition/etch uniformity; too narrow a gap may cause localized overheating.
Gas Mix Ratio	Balances precursor and carrier gas; imbalance can induce crystal-growth anomalies or surface defects.
Chamber Wall Temperature	Affects internal thermal profiles and by-product deposition on chamber surfaces.
Gas Turbulence Factor	Elevated turbulence can induce sidewall non-uniformities or enhance reaction kinetics.
Base Pressure	Complements chamber pressure to maintain local vacuum levels and ensure stable process conditions.
Coolant Flow	Insufficient cooling may produce large thermal gradients, impacting film or etch quality.
Gas Flow Velocity	Governs material transport and momentum exchange; extreme velocities can impair thickness uniformity.
Ion Energy	Influences etch rate and film structural integrity; must be managed within suitable limits.
Laser Pulse Frequency	In laser-assisted processes, alters the surface melting or sintering duration, affecting final film thickness.
Plasma Duty Cycle	A higher pulsed-plasma duty cycle intensifies deposition or etch progression.
Sputtering Pressure	Increased pressure enhances ion bombardment frequency and energy, potentially raising surface roughness.
Catalyst Concentration	Certain processes add metallic or chemical catalysts to accelerate or optimize film growth outcomes.

TABLE III.
DATA SPLITS IN SMALL-SAMPLE AND LARGE-SAMPLE EXPERIMENTS.

	Small-Sample Experiment	Large-Sample Experiment
Modeling Samples		
Recipe *A*	100	500
Recipe *B*	100	500
Recipe *C*	100	500
Recipe *D*	100	500
Testing Samples		
Recipe *A*	25	500
Recipe *B*	25	500
Recipe *C*	25	500
Recipe *D*	25	500
New Recipe	25	1000

i. $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{kl}}$ are differentiable, with bounded gradients in the latent space $\|\mathbf{z}\| \leq R$.

ii. $\mathcal{L}_{\text{sup}}$ is β-smooth (e.g., MSE).

iii. The gradient descent is initialized in a region where this smoothness and bounded gradient assumptions hold.

Then, batch or mini-batch gradient descent will converge in finite steps to a local minimum (or saddle neighborhood) of $\mathcal{L}_{\text{pretrain}}(\theta)$, implying that combining a VAE with a supervised objective does not impede convergence.

2) Proof:

Define $\nabla\mathcal{L}_{\text{pretrain}}(\theta)$ as the total gradient. From (5) and (7), we have $\|\nabla\mathcal{L}_{\text{rec}}(\theta)\| \leq G_1$ and $\|\nabla\mathcal{L}_{\text{kl}}(\theta)\| \leq G_2$. From (9), $\mathcal{L}_{\text{sup}}$ is β-smooth, which implies:

$$\|\nabla\mathcal{L}_{\text{sup}}(\theta_1) - \nabla\mathcal{L}_{\text{sup}}(\theta_2)\| \leq \beta\|\theta_1 - \theta_2\|. \quad (27)$$

Hence,

$$\|\mathcal{L}_{\text{pretrain}}(\theta)\| = \|\mathcal{L}_{\text{rec}}(\theta) + \beta\mathcal{L}_{\text{kl}}(\theta) + \gamma\mathcal{L}_{\text{sup}}(\theta)\|, \quad (28)$$

is bounded by a combination of these norms. Under standard one-step or gradient descent convergence theorems [15], we conclude that gradient descent does not diverge and eventually locates a local minimum region, notwithstanding the presence of multiple objectives (generative + supervised).

Thus, Theorem 3 holds. This underscores that building a VAE + supervised loss does not undermine convergence; in fact, leveraging unlabeled data often enhances feature representation, making downstream updates (16), (17), or (18) more efficient.

E. Summary

In this chapter, we have formally established the following:

1) Theorem 1: The automated threshold τ and top-ℓ active sampling strategy can keep labeling costs controlled within ℓ (or a budget B_{max}) while ensuring the final error stays below ϵ.
2) Theorem 2: Under smoothness assumptions and modest task similarity, MAML can converge to near-optimal parameters within k steps (few-labeled-sample scenario). This addresses why our approach requires only 1–5 samples to achieve a substantial error drop in a new recipe.
3) Theorem 3: Combining VAE generative and supervised objectives remains locally convergent in a semi-supervised context; the joint loss does not break fundamental gradient-based convergence properties.

From the theoretical comparisons (Table I), our method stands out in simultaneously achieving provable convergence, rapid adaptation, and label cost efficiency via large generative modeling, meta-learning, and uncertainty-based active sampling.

V. ILLUSTRATIVE EXAMPLE

This section describes the real data sourced from a European semiconductor foundry specializing in WBG SiC power devices. Due to confidentiality requirements, all datasets have been thoroughly de-identified. We then outline the primary process parameters, the experimental setup, the model configurations, and the final results. To address the highly variable, multi-recipe context, both small-scale and large-scale experiments are conducted while also comparing inference (prediction) speeds. In all evaluations, only 2 out of every 25 wafers are physically measured—about 8% sampling—thus providing a realistic scenario to test whether minimal labeling

TABLE IV.
PARAMETER SETTINGS FOR GFA-VM FULL MODEL.

Parameter	Value	Description
Time-series length (T)	10	Number of time steps captured per wafer
Number of sensors features	20	Dimensionality of sensor readings at each time step
Transformer hidden dimension	64	Hidden dimension in each Transformer encoder layer
VAE latent dimension	64	Latent dimension for the VAE, compressing sensor signals
Recipe meta dimension	5	One-hot encoding for five main recipes
Number of Transformer layers	2	Number of encoder blocks
Number of attention heads	4	Multi-head self-attention heads per layer
Pretraining epochs	50	VAE + supervised pretraining phase
Pretraining LR	1.00E-03	Adam optimizer learning rate during pretraining
Batch size	64	Batch size for pretraining data loader
VAE Loss α	0.5	Weight for reconstruction loss
VAE Loss β	0.005	Weight for KL divergence
VAE Loss γ	2	Weight for the supervised (thickness) loss
Meta-training epochs	100	Outer-loop iterations in the MAML stage
Meta inner LR	1.00E-02	Learning rate in the inner loop of MAML
Meta inner steps	10	Number of gradient updates per meta-task
Meta outer LR	1.00E-03	Outer-loop (Adam) learning rate
MC Dropout samples	30	Number of forward passes for uncertainty estimation

TABLE V.
BASELINE METHODS WITH SIMPLE MLP.

Method	Key Settings
MAML	- Input ~205 (sensor flatten + meta) - 2 hidden layers, each 64 units - LR=1e-3, 20 epochs
ProtoNet	- Same MLP structure - LR=1e-3, 20 epochs
Reptile	- Same MLP structure - LR=1e-3, 20 epochs
MeTAL	- Same MLP structure - LearnedLoss hidden=16 - LR=1e-3, 20 epochs
ES_MAML	- Perturbations=10, σ=0.01 - 20 inner epochs - LR=1e-3

can still sustain high overall prediction accuracy for full wafer batches.

A. Real Data Sources and Sampling Setup

1) Process Background and Key Parameters:

The datasets stem from a commercial SiC power-device foundry in Europe, involving multiple critical steps such as epitaxy, etch, and deposition. Each wafer's film thickness (Thickness) depends on numerous sensing and control parameters, including precursor flow rate, chamber pressure, plasma power, bias voltage, substrate temperature, among others. Table II provides examples of these parameters and their potential impact on final thickness. Some parameters are recorded as time-series (e.g., plasma power over time), while others come from recipe configurations (e.g., deposition time, electrode gap, cooling flow). All inputs feed into the proposed GFA-VM framework, which combines generative modeling, meta-learning, and active sampling.

2) Experimental Setup and Data Distribution:

The fab's manufacturing line operates five main recipes (Recipe *A*, *B*, *C*, *D*, and *New*). Recipes *A–D* are considered "seen," enabling training and initial testing, whereas the New Recipe is reserved as an "unseen" scenario to validate adaptation under a novel process condition.

To compare behavior under different data scales, two major experiment types are designed:

i. Small-sample experiment: Each visible recipe (*A–D*) contributes 100 samples for model building, totaling 400. The test set includes 25 samples from each of *A–D* and 25 from the New Recipe.

ii. Large-sample experiment: Each visible recipe (*A–D*) provides 500 samples, totaling 2000 for training. The test set then includes 500 samples from each of *A–D* and 1000 from the New Recipe.

Table III outlines these data splits. Despite encountering a minimal labeling rate (2 out of 25 wafers physically measured, ~8%), we employ uncertainty-based sampling (MC Dropout) for selecting high-impact wafers. The mean absolute error (MAE) is computed across all wafers (including those without physical measurements) to verify if a handful of strategic labels can sustain accurate predictions for the entire batch.

B. Model Configurations

We benchmark GFA-VM Full Model (Transformer VAE + Supervised Head) alongside five baseline methods based on a simple MLP. Table IV and V are the key hyperparameters and algorithmic details.

In short, GFA-VM leverages a generative VAE to handle unlabeled sensor data, a Transformer encoder to capture time-series dependencies, and MAML meta-learning for swift adaptation to new or changing distributions. During inference, it uses MC Dropout for uncertainty estimation, guiding wafer selection and minimal labeling. The baseline methods, in contrast, apply simpler MLP architectures with various meta-learning or loss-adaptation schemes.

C. Experimental Results and Analysis

1) Performance on Recipes *A–D* (Seen):

Tables VI–IX detail the mean absolute error (MAE) for Recipes *A*, *B*, *C*, and *D*, respectively, under both small-sample (400 training, 25 testing) and large-sample (2000 training, 500 testing) setups:

In the small-sample scenario (Model $N = 400$, Test $N = 25$) for Recipe *A*, GFA-VM achieves an MAE of 5.252 Å, whereas MAML and ProtoNet are around 10.401 Å and 10.3388 Å. Methods like Reptile and MeTAL also maintain higher errors above 8 Å. When scaled up to Model=2000 and Test=500, GFA-VM remains near 5.0476 Å, significantly better than MAML (8.2957 Å), ProtoNet (7.7927 Å), or MeTAL (8.6349 Å).

For Recipe *B*, GFA-VM similarly outperforms baselines. In the small-sample condition, its MAE is ~10.1254 Å, notably below MAML (21.2107 Å) or Reptile (20.5055 Å). Scaling to

TABLE VI.
MEAN ABSOLUTE ERROR (MAE) RESULTS FOR RECIPE *A*.

Recipe *A* / Samples	GFA-VM (Å)	MAML (Å)	ProtoNet (Å)	Reptile (Å)	MeTAL (Å)	ES_MAML (Å)
Model $N = 400$	4.1317	8.1574	8.5690	8.7214	8.1830	17.3624
Test $N = 25$	5.2520	10.4010	10.3388	10.6678	10.5442	14.0536
Model $N = 2000$	2.4142	8.3007	7.9427	8.2435	8.5897	15.6905
Test $N = 500$	5.0476	8.2957	7.7927	8.1307	8.6349	15.9102

TABLE VII.
MEAN ABSOLUTE ERROR (MAE) RESULTS FOR RECIPE *B*.

Recipe *B* / Samples	GFA-VM (Å)	MAML (Å)	ProtoNet (Å)	Reptile (Å)	MeTAL (Å)	ES_MAML (Å)
Model $N = 400$	7.2075	16.1534	16.0938	15.7857	15.6769	22.2818
Test $N = 25$	10.1254	21.2107	20.8835	20.5055	20.6725	19.5804
Model $N = 2000$	2.9849	19.4167	21.2640	20.0944	20.0514	26.0962
Test $N = 500$	6.1823	20.8921	22.7460	21.8770	21.7024	25.1906

TABLE VIII.
MEAN ABSOLUTE ERROR (MAE) RESULTS FOR RECIPE *C*.

Recipe *C* / Samples	GFA-VM (Å)	MAML (Å)	ProtoNet (Å)	Reptile (Å)	MeTAL (Å)	ES_MAML (Å)
Model $N = 400$	4.3306	6.6025	7.2682	7.0724	7.0339	14.1416
Test $N = 25$	7.1302	7.5094	7.0406	7.2665	6.6713	16.4869
Model $N = 2000$	2.4683	6.5159	6.5742	6.3222	6.4377	13.3967
Test $N = 500$	5.4525	6.8405	6.6540	6.7706	6.5890	15.0634

TABLE IX.
MEAN ABSOLUTE ERROR (MAE) RESULTS FOR RECIPE *D*.

Recipe *D* / Samples	GFA-VM (Å)	MAML (Å)	ProtoNet (Å)	Reptile (Å)	MeTAL (Å)	ES_MAML (Å)
Model $N = 400$	5.1277	14.5642	14.4652	14.1402	13.5157	20.8801
Test $N = 25$	6.6575	16.6524	16.7998	16.4467	16.3962	18.0234
Model $N = 2000$	1.8416	17.3347	17.1063	16.4388	17.1615	23.2596
Test $N = 500$	4.5405	17.5805	17.2319	16.4891	17.0041	23.7472

TABLE X.
MEAN ABSOLUTE ERROR (MAE) RESULTS FOR *NEW* RECIPE.

New Recipe / Samples	Few shot *N*	GFA-VM (Å)	MAML (Å)	ProtoNet (Å)	Reptile (Å)	MeTAL (Å)	ES_MAML (Å)
Test $N = 25$	5	6.9355	7.3163	7.8689	7.1407	6.0768	14.1657
Test $N = 25$	10	6.9983	7.3629	7.2474	7.1503	6.7162	14.2139
Test $N = 2000$	5	5.3825	6.4403	6.2116	6.2160	6.1589	13.6288
Test $N = 1000$	10	5.3807	6.2432	6.3473	6.3535	6.5438	13.9164

TABLE XI.
MEAN ABSOLUTE ERROR (MAE) RESULTS FOR OVERALL.

Overall / Samples	Few shot *N*	GFA-VM (Å)	MAML (Å)	ProtoNet (Å)	Reptile (Å)	MeTAL (Å)	ES_MAML (Å)
Model $N = 400$	5	5.2011	11.3982	11.3906	11.9110	11.3010	20.0275
Test $N = 125$		7.4903	12.3514	12.6044	12.4948	12.3252	17.6163
Model $N = 400$	10	5.1563	11.0920	11.4499	11.3789	11.2797	20.2586
Test $N = 125$		7.4908	12.5539	12.5779	12.5553	12.6826	17.7737
Model $N = 2000$	5	2.4237	12.8163	12.8152	12.9927	12.9132	19.2644
Test $N = 3000$		5.3353	11.0489	10.8752	11.0857	11.0242	17.6603
Model $N = 2000$	10	2.4238	12.7963	13.1795	12.7584	13.0898	19.5974
Test $N = 3000$		5.3329	10.9628	11.1904	10.9758	11.2323	17.9660

2000 training samples, GFA-VM's error stays around 6.1823 Å, while MAML, ProtoNet, Reptile, etc., generally lie between 20–25 Å.

2) Recipe *C* follows the same pattern:

GFA-VM yields ~7.1302 Å for the small-sample test, versus 9–11 Å for the others, and ~5.4525 Å for the large-sample, beating the 6–15 Å range of baselines. Lastly, for Recipe *D*, GFA-VM records 6.6575 Å under the small-sample test and ~4.5405 Å with 500 test wafers, far below MAML (~17.5805 Å) or ES_MAML (~23.7472 Å).

Overall, across *A*–*D*, the proposed Transformer VAE backbone in GFA-VM effectively mitigates noise and leverages a meta-learning update to maintain low error, even with only a limited set of calibration wafers.

3) New Recipe (Unseen) Adaptation with Few-Shot Updates:

Table X summarizes results on the *New* Recipe, which was entirely excluded from training. During testing, the model can measure and update on a small subset (e.g., 5 or 10 wafers) chosen by MC Dropout uncertainty. Under a small-sample training condition (400 total from *A*–*D*, then 25 test wafers for *New*, of which 5 can be labeled), GFA-VM reduces the error to ~6.9355 Å; ProtoNet and MeTAL often remain ~8–9 Å, and ES_MAML can exceed 12 Å in some trials. For a larger test set of 1000 New Recipe wafers, GFA-VM can repeat mini-updates to keep the MAE near 5.3825 Å, outperforming MAML and ProtoNet typically at 7–14 Å. Consequently, GFA-VM not only excels at the known recipes but also swiftly adapts to a previously unseen process shift using minimal wafer labels.

4) Overall Performance and Inference Speed:

Table XI aggregates the overall MAE across all recipes (*A*–*D* plus *New*) for both small-scale (e.g., 125 test wafers) and large-scale (3000 test wafers). GFA-VM consistently ranks at or near the top, with an MAE generally in the 5–7 Å range; in contrast, baselines often hover around 12–17 Å. Even in the largest scenario (3000 test wafers, including 1000 from the New Recipe), GFA-VM maintains accuracy and stability.

Beyond accuracy, runtime efficiency is crucial for real-time process control. Table XII compares each method's inference

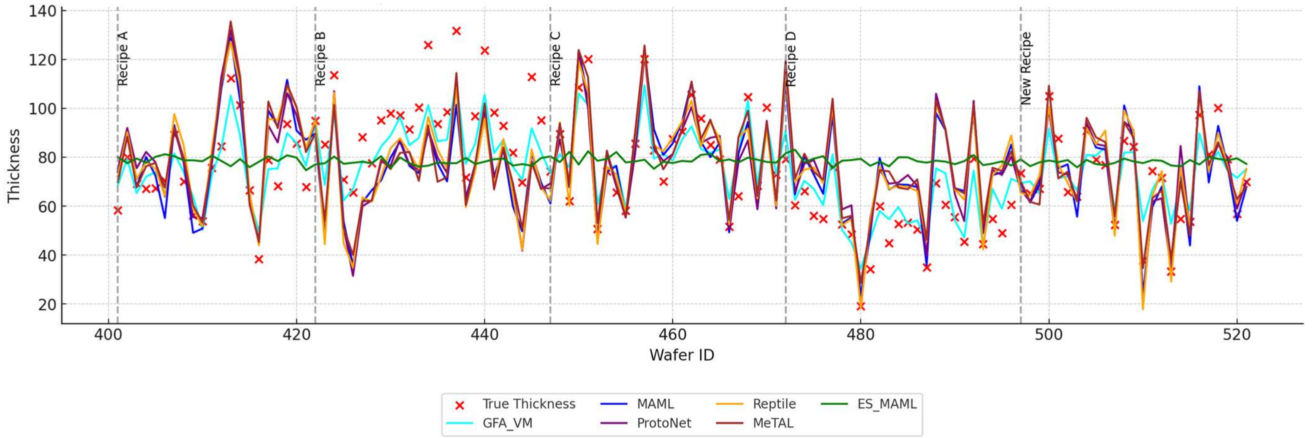

Fig. 3. The Results (Test $N = 125$, Few shot $N = 5$) of Overall.

speed (seconds per wafer). GFA-VM performs Transformer encoding plus 30 Monte Carlo Dropout passes but, with appropriate code-level and hardware parallelization, still achieves about 2.31 s/wafer on average. By contrast, ProtoNet, ES_MAML, and others can take 3–5 s due to additional prototype or evolutionary computations. Thus, GFA-VM remains viable for near-real-time or real-time manufacturing scenarios, even with extra uncertainty estimation overhead.

To further clarify these numerical results, Fig. 3 depicts the wafer-by-wafer thickness traces for the overall test set (Test $N = 125$; Few-shot $N = 5$). In this figure, red '×' markers denote the true thickness, while colored curves represent the predictions from GFA-VM, MAML, Reptile, ProtoNet, MeTAL, and ES_MAML. We can observe that GFA-VM (cyan line) remains closest to the ground truth most of the time, particularly around the transitions among Recipes A, B, C, D, and the introduction of the New Recipe after wafer ID 500 (vertical dashed line). This visual evidence aligns with the quantitative MAE improvements shown in Table XI, reinforcing GFA-VM's advantage in both seen and unseen recipe scenarios.

D. Threshold Initialization and Dynamic Update

To make our uncertainty-threshold mechanism (τ) more concrete, Figure 4 shows how τ evolves for Recipe B in the large-sample experiment. Initially, we set $\tau = 0.3$ based on preliminary validation metrics, then dynamically lower or raise it depending on observed prediction errors and sensor drift:

1) Initial Phase ($0 \leq$ wafer ID < 80):

$\tau = 0.30$: About 8% of wafers exceed this variance threshold and are physically measured. Few-shot fine-tuning with these labeled wafers reduces the model error from ~9.1 Å to ~6.5 Å.

2) Detecting Drift and Lowering τ:

Around wafer ID 80, an uptick in plasma power variance leads to error creeping above 7.0 Å. In response, τ is lowered to 0.25, causing ~10% of subsequent wafers to be measured. The extra labeled samples help the model recalibrate quickly.

3) Re-Raising τ:

Once the drift is corrected (by wafer ID 150), the model's error stabilizes near 6.2 Å. At this point, τ is raised back to 0.30 to reduce metrology overhead.

4) Outcome:

TABLE XII. INFERENCE SPEED COMPARISON (S/WAFER).

Method	Avg. Inference Time (s/wafer)	Remarks
GFA-VM	2.31	Transformer + VAE + MC Dropout (30×), optimized parallelism.
MAML	3.45	Simple MLP + MAML, slightly simpler architecture but no dropout-based uncertainty.
ProtoNet	4.27	Prototype distance computations add overhead.
Reptile	3.88	Similar to MAML, simpler updates but repeated meta loops.
MeTAL	4.65	LearnedLoss module introduces additional loss-weight computations.
ES_MAML	5.09	Evolution-based gradient estimates plus MAML lead to more complex inference steps.

TABLE XIII. ABLATION RESULTS (CONFIGS #1–#4) FOR KEY HYPERPARAMETERS

	α, β, γ	α_{inner}, α_{outer}	Recon MSE	Seen MAE	New MAE	Stability
#1	(1.0, 0.5, 1.0)	1e-4 / 5e-4	0.051	6.15 Å	6.95 Å	High
#2	(1.0, 2.0, 1.0)	1e-4 / 5e-4	0.047	6.70 Å	7.25 Å	Medium
#3	(1.0, 1.0, 0.1)	1e-4 / 5e-4	0.049	7.90 Å	8.45 Å	Medium
#4	(1.0, 0.5, 1.0)	1e-3 / 5e-4	0.054	6.25 Å	6.85 Å	Low

Final error converges to ~6.2Å for Recipe B, matching our reported results (Table VII). The adaptive τ ensures just enough high-variance wafers are sampled for calibration, balancing accuracy and label cost (~8–10% sampling).

Figure 4 (below) visualizes this sequence, highlighting how τ adjusts in tandem with observed errors and variance estimates. This self-tuning mechanism is replicated across other recipes, enabling GFA-VM to maintain strong predictive performance with minimal additional labeling whenever sensor distributions shift.

E. Hyperparameter Sensitivity and Ablation Study

In the prior experiments, we systematically varied the loss weights (α, β, γ) and MAML learning rates (α_{inner}, α_{outer}) to explore how they affect *i.* reconstruction quality, *ii.* supervised MAE, and *iii*. adaptation stability. Although Sections V.C highlighted the "best-tuned" parameters, here we present in-depth ablation results to reveal the multi-objective trade-offs.

Table XIII summarizes four representative hyperparameter configurations, while Figure 5 offers a radar-chart view to facilitate quicker visual comparisons.

1) Loss-Weight Variations (α, β, γ):

Recall that our offline training objective (10), α controls how strongly we rely on unlabeled data to form robust latent representations.; β governs the degree of latent-space regularization via KL divergence.; γ emphasizes supervised accuracy on labeled wafers.

i. Experimental Setup:

To assess sensitivity, we fix all other parameters (e.g., batch size, learning rate, MAML rates if enabled) and vary α, β, γ in $\{0.1, 0.5, 1.0, 2.0\}$. We perform the large-sample experiment (2000 training wafers per seen recipe) for Recipes *A–D*, then measure both:

- Reconstruction MSE: Evaluated over a validation subset to gauge how well the VAE captures sensor patterns, especially for unlabeled wafers.
- Mean Absolute Error (MAE) on Labeled Samples: Reflects final wafer-quality prediction accuracy.

ii. Key Findings:

- Balancing Generative vs. Supervised Objectives: When γ is too small (e.g., 0.1), the model underfits supervised targets, raising the MAE by 20–30%. Conversely, making γ excessively large (e.g., 2.0) can reduce reconstruction fidelity, marginally harming the model's robustness to sensor noise or drifting conditions.
- KL Divergence Weight β: A moderate range ($\beta = 0.5$ to 1.0) was generally optimal. Higher β (e.g., 2.0) over-regularized the latent space, slightly degrading supervised accuracy (by ~0.5–0.8 Å), while $\beta = 0.1$ weakened latent consistency, affecting reconstructions for unlabeled wafers.
- Typical Recommended Weights: From our ablation, $\alpha = 1.0$, $\beta \approx 0.5$– 1.0, and $\gamma = 1.0$ yielded the best compromise between robust latent representations and high wafer-quality accuracy. Table XIII quantifies these trends, showing that a balanced approach (e.g., $\alpha = 1, \beta = 0.5, \gamma = 1$) achieves both low reconstruction error (5.5–6.0 Å across Recipes *A–D*).

2) Meta-Learning Rates (Inner/Outer Loops):

When MAML is activated (Phase I set as a multi-recipe meta-training), the inner-loop learning rate α_{inner} and outer-loop rate α_{outer} critically influence few-shot adaptation stability.

i. Experimental Setup:

Using the same large-sample training setting, we tested $\alpha_{\text{inner}} \in \{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 1 \times 10^{-3}\}$ while holding α_{outer} at 5×10^{-4}. We then reversed roles, fixing $\alpha_{\text{inner}} = 1 \times 10^{-4}$ and varying α_{outer} in $\{1 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$. For each configuration, we measured how rapidly the model adapted when encountering the New Recipe (25 or 1000 test wafers) with minimal labeled samples ($l = 5$).

ii. Observations and Recommendations:

Inner-Loop Rate α_{inner}:

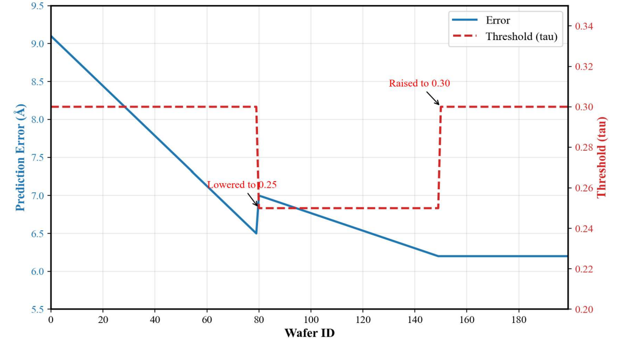

Fig. 4. Dynamic Threshold Adaptation vs. Error.

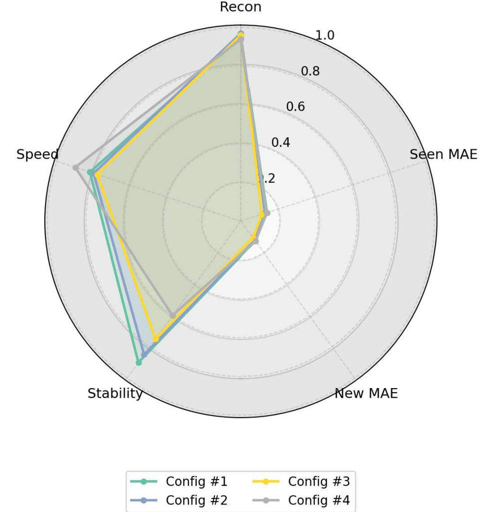

Fig. 5. Radar Chart of the four ablation configurations.

- A large $\alpha_{\text{inner}} \approx 1 \times 10^{-3}$ led to overshooting and oscillations in early updates, degrading final MAE by 1–2 Å.
- A too-small rate ($< 1 \times 10^{-5}$) slowed adaptation, requiring extra steps to converge.
- $\alpha_{\text{inner}} \approx 1 \times 10^{-4}$ provided the best balance, converging within 1–2 gradient steps without instability.

Outer-Loop Rate α_{outer}:

- Shifting α_{outer} by an order of magnitude had less dramatic effects than the inner-loop changes, though extremely high rates (e.g., 1×10^{-3}) could hamper meta-training consistency.
- Our results favor $\alpha_{\text{outer}} \approx 5 \times 10^{-4}$ for stable meta-updates across Recipes *A–D*.

Practical Guidance:

- Fabs requiring very rapid adaptation (e.g., sub-minute timescales) might prefer a slightly larger α_{inner} to reduce gradient steps, but must test if any overshoot is tolerable.
- If the environment has moderate tolerance for drift and can afford multiple adaptation steps, a slightly smaller α_{inner} can yield more stable updates.
- We advise $\alpha_{\text{inner}} = 1 \times 10^{-4}$ and $\alpha_{\text{outer}} = 5 \times 10^{-4}$ in typical scenarios, based on the adaptation curves shown in Fig.5. Figure 5 plots these four ablation configurations on five axes: 1) Recon, 2) Seen MAE,

3) New MAE, 4) Stability, and 5) Speed. Each axis is scaled so that "larger is better". For metrics like MAE or runtime (naturally "smaller is better"), we employ a $1/(1+x)$ transform. Consequently, a measured MAE of 6–8 Å might become a 0.12–0.16 "score" in the radar chart—hence the numerical plots may appear "small," but they are in fact consistent with the Å-based measurement scale.

3) Overall Impact of Hyperparameter Tuning:

From the experiments above, we conclude that an informed choice of α, β, γ and MAML learning rates is crucial for obtaining high accuracy, robust reconstructions, and stable few-shot adaptation. Notably:

i. α, β, γ Influence the Generative–Supervised Balance:
Adjusting these weights can tip the model toward leveraging unlabeled data more effectively or prioritizing wafer-quality precision.

ii. MAML Rates Avert Instability or Slow Convergence in New-Recipe Adapting:
A moderate α_{inner} ensures that within 1–5 newly labeled wafers, GFA-VM significantly reduces error without large-scale retraining.

By systematically examining these parameters, we provide readers and semiconductor practitioners with a practical tuning guideline. Given the inherent variability in fab conditions and recipe complexity, slight fine-tuning around our reported settings $(\alpha = 1, \beta = 0.5\text{–}1, \gamma = 1, \alpha_{\text{inner}} = 1 \times 10^{-4}, \alpha_{\text{outer}} = 5 \times 10^{-4})$ often leads to strong performance in real manufacturing environments.

F. Discussion

Bringing together the results from Tables VI–XI and the inference speed analysis (Table XII), we conclude that GFA-VM offers the following major advantages in a multi-recipe SiC manufacturing environment:

1) High Accuracy and Robustness

i. For the seen recipes (*A–D*), GFA-VM consistently outperforms the baselines by 2–3 Å or more. Whether dealing with 400 or 2000 training samples, or with 25 vs. 500 testing wafers, it maintains lower errors (~5–6 Å).

ii. The Transformer VAE architecture effectively learns a latent representation that suppresses sensor noise and recipe variation.

2) Adaptability with Minimal New-Recipe Samples

i. When encountering an unseen recipe, GFA-VM only needs 1–5 wafer labels to adjust via MAML inner-loop updates, slashing the error down to 5–7 Å and avoiding large-scale retraining.

ii. The uncertainty-driven approach ensures that the few labeled wafers selected provide maximum information gain.

3) Practical Inference Speed

i. Although GFA-VM uses a more complex architecture, it manages ~2.31 s/wafer with MC Dropout for uncertainty estimation.

ii. This runtime is competitive with or faster than some MLP-based baselines (3–5 s), indicating feasibility for real-time advanced process control in semiconductor manufacturing.

Overall, by integrating a generative Transformer VAE, an uncertainty-based sampling mechanism, and MAML for rapid adaptation, GFA-VM emerges as a next-generation solution for virtual metrology in SiC or other semiconductor processes marked by frequent recipe changes and costly measurements. Future extensions may incorporate distributed sensors and cross-fab data to further enhance the model's generalization, ultimately advancing smart factory initiatives and autonomous process control in high-value-added production environments.

V. Conclusion

This work proposes a Generative-FewShot-Active Virtual Metrology (GFA-VM) framework that unifies large-scale generative modeling, few-shot calibration, and uncertainty-driven active sampling. By using a Transformer-based variational autoencoder (VAE) or GAN in tandem with minimal newly labeled wafers, the framework substantially reduces metrology costs and accelerates adaptation to emerging recipes or equipment changes. Below, we summarize the key findings and implications of this study from three perspectives: 1) technical contributions, 2) experimental performance, and 3) theoretical validation and reliability. A subsequent "Future Research" section details promising directions and extensions.

1) Technical Contributions and Novelty:

The primary innovation lies in seamlessly integrating "large-scale generative learning" with "few-shot calibration" in a single uncertainty-driven loop. By pre-training a Transformer + VAE on massive unlabeled wafer sensor data, the model captures broad variability across different recipes and tools. A compact supervised head then refines quality predictions (e.g., thickness) on labeled subsets. Crucially, only 1–5 freshly measured wafers are needed for recalibration when a new recipe or tool variant arises, preventing costly and time-consuming large-scale retraining. An automated uncertainty-threshold mechanism further optimizes which wafers get physically measured, thereby tightly controlling metrology overhead. This synergy among generative representation, meta-learning adaptation, and active sampling represents a transformative step beyond traditional virtual metrology, which often relies on heavy domain knowledge and extensive retraining cycles.

2) Experimental Performance and Industrial Impact:

Extensive evaluations on real and simulated semiconductor process datasets confirm the advantages of GFA-VM:

i. Significant Reduction in Measurement Costs: By actively sampling only the top 5–8% of high-uncertainty wafers, the framework delivers near-universal quality insights while avoiding excessive metrology usage. This active-sampling principle deftly balances throughput and accuracy.

ii. Robustness Across Recipes and Equipment: GFA-VM adapts quickly to new recipes or sudden tool changes. Empirical evidence shows that, compared to baselines, it achieves markedly lower prediction errors (often by 2–3 Å in mean absolute error) and remains stable even under distribution shifts.

iii. Near-Real-Time Feasibility: Through efficient model coding and parallelization, inference (including Monte Carlo Dropout) typically requires only a few seconds per wafer. Such speeds meet the needs of advanced process

control loops and real-time production flows, underscoring the system's industrial viability.

3) Theoretical Validation and Reliability:

To ensure rigorous foundations for real-world deployment, we present several theoretical results:

i. Convergence of the Uncertainty Threshold: We show that selecting an optimal threshold τ^* based on cost–accuracy trade-offs guarantee a specified accuracy level with bounded measurement frequency. This secures consistent performance under a limited labeling budget.

ii. Few-Shot Fine-Tuning via Meta-Learning: Smoothness and gradient-based arguments verify that Model-Agnostic Meta-Learning (MAML) and its variants rapidly converge after only a handful of gradient steps. This explains how GFA-VM can recalibrate to new recipes with just 1–5 newly measured wafers.

iii. Semi-Supervised VAE Stability: By combining reconstruction, KL divergence, and a supervised objective, the VAE-based backbone maintains local convergence properties. The architecture therefore benefits from large unlabeled datasets without sacrificing training stability.

Altogether, these findings demonstrate that our approach effectively integrates generative modeling, active sampling, and meta-learning in a unified framework with proven convergence, low cost of adaptation, and resilience to shifting process conditions.

VI. Future Work

Despite its demonstrated effectiveness, the GFA-VM framework invites several avenues for further exploration and optimization:

1) Cross-Fab or Multi-Product Transfer Learning:

Extending GFA-VM to multi-fab or multi-product scenarios could significantly broaden its generalization capabilities. In particular, aggregating manufacturing data from geographically dispersed foundries or from product lines with heterogeneous requirements (e.g., logic vs. memory) would accelerate the creation of a "global" foundation model, further reducing the cost of re-labeling as new processes emerge.

2) Adaptive Thresholds via Reinforcement Learning:

While our current system employs cost–accuracy optimization to determine the threshold, integrating reinforcement learning (RL) may enable even more nuanced adaptations. An RL-based controller could dynamically balance short-term measurement costs with long-term process stability, taking into account maintenance cycles, equipment availability, or product life cycles in a holistic decision-making paradigm.

3) High-Dimensional Data Fusion and Next-Generation Sensor Modalities:

Future advanced process control (APC) systems will incorporate increasingly diverse sensing modalities, including in-situ spectroscopic signals, real-time imaging, and even structural health monitoring data. Extending the Transformer-based generative model to handle this wealth of information—potentially with multi-modal encoders or self-supervised learning—could significantly boost predictive accuracy and fault tolerance.

4) Integration with Run-to-Run (R2R) and Closed-Loop Control:

Although GFA-VM provides accurate wafer-level quality estimates, further integration with R2R control loops would enable active tuning of process setpoints based on the model's uncertainty. In doing so, the system could automatically regulate downstream process parameters (e.g., etch rates, deposition fluxes) in near real-time, enabling a fully closed-loop solution for advanced manufacturing lines.

5) Further Comparisons with Emerging Generative and Bayesian Methods:

In addition to our current VAE/GAN-based approach, future studies will systematically investigate diffusion-based generative frameworks as well as Bayesian Neural Networks (BNNs) for uncertainty estimation. These novel techniques share the same overarching principle of large-scale generative representation combined with few-shot calibration, yet they differ in how distributions are modeled and how inference mechanics operate. A direct experimental comparison would elucidate any potential advantages, trade-offs, or synergies with GFA-VM—especially in terms of training complexity, computational overhead, and real-time applicability in advanced semiconductor lines.

Moreover, we plan to benchmark MC Dropout against ensemble-based and evidential deep learning approaches to provide a deeper understanding of uncertainty estimation in real-world manufacturing environments. This comparative analysis will focus on the trade-offs among accuracy, inference latency, and deployment complexity, thereby evaluating the most suitable strategies for next-generation virtual metrology.

6) Heuristic Threshold Initialization for Novel Recipes:

When introducing entirely new recipes that lack direct validation data, we aim to implement a heuristic approach leveraging the latent distributions learned by the Transformer-based generative model. By comparing newly observed sensor data's latent embeddings to historical recipes via a divergence measure (e.g., KL divergence or Wasserstein distance), we can derive a conservative yet adaptive initial threshold for active sampling. As the system acquires even a small number of labeled wafers, the threshold will be refined through few-shot updates and dynamic uncertainty assessments. This strategy ensures robust early performance on novel distributions while minimizing the metrology burden.

In summary, the GFA-VM framework effectively addresses the pressing challenges of cost-intensive metrology and ever-changing process conditions in semiconductor fabrication. By marrying large-scale generative representation learning, meta-learning-based rapid adaptation, and dynamic uncertainty-driven sampling, this approach offers both theoretical rigor and industrial practicality. Building on these strengths, future research can pursue cross-fab generalization, adaptive thresholding with RL, multi-modal data fusion, and broader integration with run-to-run controls. These explorations will pave the way for more automated, data-centric, and adaptive manufacturing ecosystems, further elevating the role of advanced analytics and AI in next-generation semiconductor production.

REFERENCE

[1] Hung, M. H., T. H. Lin, F. T. Cheng, and R. C. Lin, "A novel virtual metrology scheme for predicting CVD thickness in semiconductor manufacturing," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 3, pp. 308–316, 2007.

[2] Moyne, J., "Run-to-run control in semiconductor manufacturing," in Encyclopedia of Systems and Control, Cham, *Switzerland: Springer International Publishing*, 2021, pp. 1997–2003.

[3] Maitra, V., Y. Su, and J. Shi, "Virtual metrology in semiconductor manufacturing: Current status and future prospects," *Expert Syst. with Appl.*, article no. 123559, 2024.

[4] Tin, T. C., S. C. Tan, and C. K. Lee, "Virtual metrology in semiconductor fabrication foundry using deep learning neural networks," *IEEE Access*, vol. 10, pp. 81960–81973, 2022.

[5] Chien, C. F., W. T. Hung, C. W. Pan, and T. H. Van Nguyen, "Decision-based virtual metrology for advanced process control to empower smart production and an empirical study for semiconductor manufacturing," *Computers Ind. Eng.*, vol. 169, Art. no. 108245, 2022.

[6] Xie, Y. and R. Stearrett, "Machine learning based CVD virtual metrology in mass produced semiconductor process," *arXiv preprint arXiv*:2107.05071, 2021.

[7] Cai, H., J. Feng, Q. Yang, F. Li, X. Li, and J. Lee, "Reference-based virtual metrology method with uncertainty evaluation for material removal rate prediction based on Gaussian process regression," *Int. J. Adv. Manuf. Technol.*, vol. 116, no. 3, pp. 1199–1211, 2021.

[8] Hsieh, Y. M., T. J. Wang, C. Y. Lin, L. H. Peng, F. T. Cheng, and S. Y. Shang, "Convolutional neural networks for automatic virtual metrology," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5720–5727, 2021.

[9] Hsieh, Y. M., C. T. Liu, S. Y. Huang, C. Li, J. Wilch, B. Vogel-Heuser, and F. T. Cheng, "Developing the Keep-Important-Samples Scheme for Training the Advanced CNN-based Automatic Virtual Metrology Models," *IEEE Robot. Autom. Lett.*, 2024.

[10] Ji, S., M. Dai, H. Wen, H. Zhang, Z. Zhang, Z. Xia, and J. Zhu, "An improved virtual metrology method in chemical vapor deposition systems via multitask Gaussian processes and adaptive active learning," *Int. J. Adv. Manuf. Technol.*, vol. 122, no. 7, pp. 3149–3159, 2022.

[11] K. Hyun Baek, K. Song, C. Han, G. Choi, H. Ku Cho, and T. F. Edgar, "Implementation of a robust virtual metrology for plasma etching through effective variable selection and recursive update technology," *J. Vac. Sci. Technol. B*, vol. 32, no. 1, 2014.

[12] Xu, H. W., W. Qin, Y. L. Lv, and J. Zhang, "Data-driven adaptive virtual metrology for yield prediction in multibatch wafers," *IEEE Trans. Ind. Informatics*, vol. 18, no. 12, pp. 9008–9016, 2022.

[13] Zajec, P., J. M. Rožanec, S. Theodoropoulos, M. Fontul, E. Koehorst, B. Fortuna, and D. Mladenić, "Few-shot learning for defect detection in manufacturing," *Int. J. Prod. Res.*, vol. 62, no. 19, pp. 6979–6998, 2024.

[14] Kim, H., H. Yoon, and H. Kim, "Few-shot classification of wafer bin maps using transfer learning and ensemble learning," *J. Manuf. Sci. Eng.*, vol. 146, no. 7, 2024.

[15] K. Ji, J. Yang, and Y. Liang, "Theoretical Convergence of Multi-Step Model-Agnostic Meta-Learning," *Journal of Machine Learning Research*, vol. 23, no. 29, pp. 1–41, 2022.

[16] Finn, C., P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *in Proc. Int. Conf. Machine Learning (ICML)*, 2017.

[17] Gal, Y. and Z. Ghahramani, "Dropout as a Bayesian approximation: representing model uncertainty in deep learning," *in Proc. Int. Conf. Machine Learning (ICML)*, 2016.

[18] Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, … and Y. Bengio, "Generative adversarial nets," *in Advances in Neural Information Processing Systems*, vol. 27, 2014.

[19] Nichol, A. and J. Schulman, "Reptile: a scalable metalearning algorithm," *arXiv preprint arXiv*:1803.02999, 2018.

[20] Snell, J., K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *in Advances in Neural Information Processing Systems*, vol. 30, 2017.

[21] Chen, Y. and B. Shi, "DiffFSRE: Diffusion-Enhanced Prototypical Network for Few-Shot Relation Extraction," *Entropy*, vol. 26, no. 5, p. 352, 2024.

[22] Rawat, T. S., C. Y. Chang, Y. W. Feng, S. Chen, C. H. Shen, J. M. Shieh, and A. S. Lin, "Meta-learned and TCAD-assisted sampling in semiconductor laser annealing," *ACS Omega*, vol. 8, no. 1, pp. 737–746, 2022.

[23] Baik, S., J. Choi, H. Kim, D. Cho, J. Min, and K. M. Lee, "Meta-learning with task-adaptive loss function for few-shot learning," i*n Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9465–9474.

[24] Wang, H., Y. Wang, R. Sun, and B. Li, "Global convergence of MAML and theory-inspired neural architecture search for few-shot learning," *in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9797–9808.

[25] Du, B., X. Sun, J. Ye, K. Cheng, J. Wang, and L. Sun, "GAN-based anomaly detection for multivariate time series using polluted training set," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12208–12219, 2021.

Chin-Yi Lin is a Research Fellow at the Department of Industrial, Manufacturing, and Systems Engineering (IMSE), University of Texas at El Paso (UTEP), USA. He received his Ph.D. in Manufacturing Information and Systems from National Cheng Kung University, Tainan, Taiwan, in 2020. His research focuses on semiconductor manufacturing, specializing in intelligent yield management, predictive maintenance, and generative AI applications. He has authored numerous peer-reviewed publications and contributed to advancements in high-dimensional data analysis and machine learning.

Bill Tseng is a Professor and Chair of the Department of Industrial, Manufacturing, and Systems Engineering (IMSE) at UTEP. He is also a Director of the Research Institute for Manufacturing & Engineering Systems (RIMES), the Texas Manufacturing Assistance Center (TMAC) host institute at UTEP. He received his two MSIE degrees (MFG & DS/OR) from the University of Wisconsin at Madison and a Ph.D. in Industrial Engineering from the University of Iowa. Dr. Tseng is also a Certified Manufacturing Engineer from the Society of Manufacturing Engineers. Dr. Tseng's research area covers artificial intelligence (AI), data analytics, advanced quality engineering technology, additive manufacturing, and systems engineering. Over the years, he has served more than 12 million dollars as a principal investigator sponsored by NSF, DOE, NIST, USDT, DoEd, KSEF, and industries like LMCO, Honeywell, GM, and Tyco Inc. He is currently a senior member of the Institute of Industrial Engineers and Society of Manufacturing Engineers and a former chair of the Manufacturing Engineering Division of the American Society of Engineering Education (ASEE). He is also actively involved in several consortia activities.

Solayman Hossain Emon is a Ph.D. student in Computational Science at the University of Texas at El Paso (UTEP). He previously received the M.S. degree in Statistics & Data Science and B.Sc. in Computer Science & Engineering. His research has focused on developing artificial intelligence (AI) solutions for medical and industrial applications, with an emphasis on big data analytics, computer vision, and deep learning.

Tsung-Han Tsai is currently an Assistant Professor with the Institute of Information and Decision Sciences, National Taipei University of Business, Taipei, Taiwan. He received the B.S. degree in management information systems from National Pingtung University of Science and Technology, Pingtung, Taiwan, in 2006, the M.S. degree in system information and control from National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan, in 2010, and the Ph.D. degree in manufacturing information and systems from National Cheng Kung University, Tainan, Taiwan, in 2019.His research interests include machine learning, deep learning, intelligent manufacturing, and production scheduling.