LLMs and LVMs for Agentic AI: A GPU-Accelerated Multimodal System Architecture for RAG-Grounded, Explainable, and Adaptive Intelligence

Kiarash Ahi, Chih-Hung Hsieh, Germain Fenger
ahi@siemens.com
Siemens, San Francisco Bay Area, United States

ABSTRACT

This paper presents an architecture for an Agentic AI System that autonomously operates and manages complex workflows across enterprise and industrial software ecosystems such as Electronic Design Automation (EDA) tools (e.g., Siemens Calibre), Product Lifecycle Management (PLM) and Digital Twin platforms (e.g., Teamcenter Digital Reality Viewer), as well as knowledge-centric domains including HR analytics, financial modeling, healthcare diagnostics, and creative design platforms.

This architecture leverages a multi-agent framework orchestrated by a central planner, integrating large language model (LLM) and large vision model (LVM) reasoning for multimodal understanding, retrieval-augmented generation (RAG) pipelines, and enterprise-grade governance to enable secure, explainable, and adaptive automation across both physical and virtual product lifecycle stages.

The architecture is structured as a nine-layer intelligent stack, beginning with a natural language interface and extending through layers responsible for cognitive orchestration, specialized agents, contextual retrieval, reasoning, tool execution, security, access control, and feedback-driven learning. Users issue high-level intents—such as “run DRC and fix critical violations” or “synchronize the latest design update with the digital twin”— which are interpreted by the planner agent and decomposed into sub-tasks. These are executed by specialized agents (e.g., simulation, review, or action agents), each interfacing securely with industrial tools and twin environments through sandboxed runtimes and version-controlled APIs. The multi-agent framework employs structured communication patterns inspired by the blackboard model, enabling Reasoner, Executor, and Validator agents to coordinate through shared semantic memory buffers. This emergent collaboration supports decentralized problem-solving and resilient orchestration under dynamic workloads. The planner dynamically adjusts task decomposition and agent routing based on resource constraints, latency budgets, and model confidence, enabling adaptive, performance-aware orchestration.

Beyond industrial and engineering use cases, the same agentic architecture generalizes to broader enterprise workflows. In HR and finance, autonomous agents extract insights from structured and unstructured data, improve forecasting accuracy, and ensure regulatory compliance. In healthcare, multimodal reasoning that fuses text, imagery, and sensor data can assist clinicians in diagnosis and treatment planning while maintaining explainability. In creative and design environments, agentic co-pilots interpret user intent, generate assets, and optimize iterative design loops—enhancing both productivity and human creativity. A core RAG layer grounds decisions in proprietary engineering knowledge (e.g., PDK rules, fab specifications, simulation logs, and historical twin data), while a chunk reranker ensures only the most relevant context is injected into LLM prompts. This RAG pipeline supports fast memory access, context pruning, and scalable grounding across high-volume logs and digital twin telemetry. This grounding layer can be extended to any domain where contextual reasoning over proprietary knowledge is critical—ranging from clinical data repositories and enterprise ERPs to document archives and financial transaction graphs.

To support this architecture’s adaptive orchestration and multimodal agent execution, performance-optimized inference becomes critical. To meet the latency, throughput, and scalability demands of large-scale multimodal reasoning, the system incorporates GPU-accelerated inference pipelines, including ROI-guided compression and adaptive latent-space clustering to reduce computational overhead while preserving output fidelity. These GPU-accelerated strategies are based on the ROI-LCC framework, which integrates dynamic Region of Interest (ROI) selection, latent-space clustering, and learned GPU feature extraction to minimize redundancy and streamline computation. Outputs are processed through a guardrails and explainability (XAI) layer that filters unsafe content, validates decisions, and generates structured audit trails. The system includes a Human-in-the-Loop (HITL) mechanism to review high-impact or real-world synchronized actions before execution. These optimizations—originally developed and validated on nanometer-resolution SEM imagery exhibiting nanoscale noise, low SNR, and extreme visual detail—enable robust, high-throughput inference in compute-constrained scenarios such as EUV lithography and biomedical diagnostics. This architecture has been integrated into key products, demonstrating readiness for real-world deployment in precision-critical industrial environments. The architecture supports real-time telemetry, bias and drift detection, and a data flywheel that captures feedback and performance metrics to continuously refine agent behavior, prompt strategies, and model accuracy. Designed for hybrid on-prem/cloud deployment and compliant with RBAC/ABAC enterprise security policies, this system ensures scalability, transparency, and governance continuity across industrial, enterprise, and domain-specific ecosystems—from design and manufacturing to financial analytics, healthcare diagnostics, HR operations, and creative content pipelines.

Collectively, these capabilities position the architecture as a generalized substrate for enterprise-scale intelligence orchestration. It not only automates workflows but also augments human decision-making, improves analytical accuracy, and accelerates creativity across sectors—bridging cognitive reasoning, multimodal perception, and secure execution. By unifying LLM reasoning and LVM orchestration, GPU-accelerated inference, grounded retrieval, digital twin synchronization, tool integration, and enterprise governance within a modular agentic framework, this system transforms traditional industrial software into an intelligent, auditable, and self-improving co-pilot—accelerating design cycles, enhancing reliability, and bridging the gap between virtual models and physical systems through autonomous, explainable decision orchestration. These optimizations make the architecture suitable for deployment in latency-sensitive, compute-constrained industrial scenarios, including edge-assisted digital twin environments and high-throughput simulation workflows, as well as knowledge-driven enterprise systems that demand adaptive, explainable, and human-aligned intelligence.

Keywords: Agentic Artificial Intelligence (AI), Multi-Agent Systems, Large Language Models (LLM), Large Vision Models (LVM), Retrieval-Augmented Generation (RAG), GPU-Accelerated Inference, Digital Twin and Industrial Automation, Explainable and Human-Aligned Intelligence

1. INTRODUCTION

The rapid advancements in Artificial Intelligence (AI) have been largely driven by the emergence of Large Language Models (LLMs) and Large Vision Models (LVMs) [1, 2]. These models, exemplified by architectures like GPT-4, LLaMA, and various Vision Transformers, have demonstrated unprecedented capabilities in tasks spanning natural language understanding, generation, image recognition, and multimodal reasoning [3, 4]. Their ability to process vast amounts of data and learn complex patterns has opened new frontiers in scientific research, industrial automation, and consumer applications.

However, the immense scale of these models—often comprising billions or even trillions of parameters—poses significant computational challenges [5]. The resources required for both training and, critically, for real-time inference, are substantial. These challenges manifest as high inference latency, extensive memory consumption, and considerable energy demands, which collectively hinder the widespread and practical deployment of LLMs and LVMs, particularly in latency-sensitive applications or environments with constrained computational resources [6, 7].

Addressing these efficiency concerns is paramount for unlocking the full potential of these transformative AI technologies. This paper presents a novel, integrated methodology designed to significantly enhance the computational efficiency of LLMs and LVMs. Our approach moves beyond isolated optimization techniques by synergistically combining dynamic Region of Interest (ROI) selection, GPU-accelerated learned feature representation, adaptive latent-space clustering, and advanced model compression techniques. This comprehensive strategy aims to drastically reduce computational overhead, accelerate inference speeds, and foster greater scalability, thereby enabling the pervasive integration of these powerful models into diverse real-world systems.

To our knowledge, this is the first work that integrates dynamic ROI selection, GPU-based latent feature learning, adaptive clustering, and deep model compression into a unified, real-time AI pipeline for high-resolution SEM image analysis. By combining both LLM- and LVM-accelerated modules within a production-ready framework, we offer a scalable, deployable solution that bridges advanced deep learning with critical semiconductor applications. This paper presents the ROI-LCC framework, a novel and integrated methodology for accelerating inference, validated in a production-ready system for high-resolution SEM image analysis.

We refer to this integrated methodology as the ROI-LCC framework, highlighting its four core components: Region-of-Interest selection, GPU-accelerated latent feature learning, adaptive clustering, and advanced compression.

The remainder of this paper is organized as follows. Section 2 reviews the computational and architectural challenges associated with deploying large language and vision models at scale. Section 3 introduces the proposed ROI-LCC framework, detailing its four synergistic components—dynamic Region-of-Interest selection, GPU-accelerated learned feature representation, adaptive latent-space clustering, and model compression. Section 4 presents the implementation of the proposed methodology within a general industrial image-analysis and agentic-AI environment, illustrating how the framework integrates with multimodal reasoning pipelines for high-resolution visual inspection and knowledge-driven automation. Section 5 provides empirical benchmarking, energy-efficiency analysis, and workflow-automation results enabled by the LLM-agentic assistant. Section 6 concludes with key findings and outlines future directions in reflective reasoning, federated optimization, and multimodal scalability.

2. BACKGROUND: THE FORMIDABLE COMPUTATIONAL CHALLENGES OF LLMS AND LVMS

The remarkable performance of contemporary LLMs and LVMs is intrinsically linked to their architectural scale and complexity, which directly translate into formidable computational requirements [8]. Understanding these bottlenecks is crucial for developing effective optimization strategies.

2.1. Model Size and Memory Footprint

Modern LLMs and LVMs contain billions of parameters, necessitating vast amounts of memory for storage. During inference, not only the model weights but also intermediate activations and the Key-Value (KV) cache (particularly in

Transformer-based architectures) consume significant high-bandwidth memory (HBM) [9]. This often dictates the need for expensive, specialized hardware (e.g., multiple high-end Graphics Processing Units (GPUs)) and severely limits deployment on edge devices or systems with restricted memory capacities [10].

2.2. Data Volume and Processing Intensity

Training these colossal models involves processing petabytes of text, image, and video data [11]. Even during inference, high-resolution inputs (e.g., gigapixel images) or long text sequences demand extensive data handling and processing pipelines. The sheer volume of data, coupled with the intricate computations per data point, contributes to substantial computational intensity.

2.3. Architectural Complexity and Latency

Architectures like the Transformer, with their multi-head attention mechanisms and deep layered structures, are highly parallelizable but inherently computationally demanding [12]. As sequence lengths in LLMs or image resolutions in LVMs increase, the computational complexity often scales quadratically or super-linearly, leading to significant inference latency. For applications requiring real-time responses, such as autonomous driving, live diagnostics, or interactive chatbots, slow inference speeds are unacceptable and can severely limit practical utility [13].

2.4. Energy Consumption

The extensive computations and prolonged operation of large AI models translate into substantial energy consumption [14]. This not only contributes to high operational costs but also raises growing environmental sustainability concerns, driving the imperative for more energy-efficient AI solutions [15].

These interwoven factors underscore the critical need for advanced, integrated optimization techniques to make LLMs and LVMs more accessible, deployable, and sustainable across a wider array of applications.

3. THE ROI-LCC FRAMEWORK: A UNIFIED METHODOLOGY FOR EFFICIENCY ENHANCEMENT

Our unified methodology addresses the computational challenges of LLMs and LVMs through a synergistic combination of advanced techniques, each targeting distinct aspects of the computational pipeline.

3.1. Dynamic Region of Interest (ROI) Selection

For LVMs, particularly when processing ultra-high-resolution imagery, a significant portion of the input data may be irrelevant to the specific task at hand. **Dynamic ROI selection** is a sophisticated pre-processing strategy designed to intelligently identify and isolate only the most salient or informative segments of the input, thereby minimizing the irrelevant data fed into the downstream large models [16].

- **Mechanism:** Unlike static or predefined ROIs, dynamic ROI detection often leverages lightweight initial passes of the LVM itself, or a specialized, smaller neural network, to rapidly pinpoint areas of high informational density, anomaly, or task-specific relevance. This can involve attention-guided mechanisms [17], reinforcement learning agents trained to identify optimal bounding boxes or masks [18], or saliency mapping techniques [19]. For instance, in an SEM image, the system might quickly pinpoint regions exhibiting structural irregularities, critical dimensions, or potential defects, allowing the system to discard vast areas of uniform background. For LLMs, analogous techniques could involve intelligent context window management or adaptive summarization to focus on key phrases or paragraphs [20].

- **Impact on Efficiency:** By drastically reducing the effective input size for the main model, dynamic ROI selection directly lowers the computational load. This leads to:
    - **Reduced Inference Time:** Fewer computations are required, significantly accelerating the model's response.

- **Lower Memory Consumption:** Less data needs to be held and processed in memory, alleviating memory bottlenecks.
- **Improved Focus and Accuracy:** By concentrating computational capacity on the most critical information, the model can potentially achieve higher precision on tasks directly related to the ROI, as irrelevant noise is filtered out.

Figure 1(a) illustrates the Region of Interest (ROI) selection process, where a representative pattern is extracted from a large Field of View (FoV) scanning electron microscope (SEM) image. As shown in Figure 1(b), model tuning for contour extraction is performed exclusively on this selected ROI. Once optimized, the tuned model is then applied across the entire FoV image. This approach yields significant computational efficiency by reducing the initial optimization overhead and focusing high-resolution processing only where needed. Detailed efficiency gains are analyzed in the Results section.

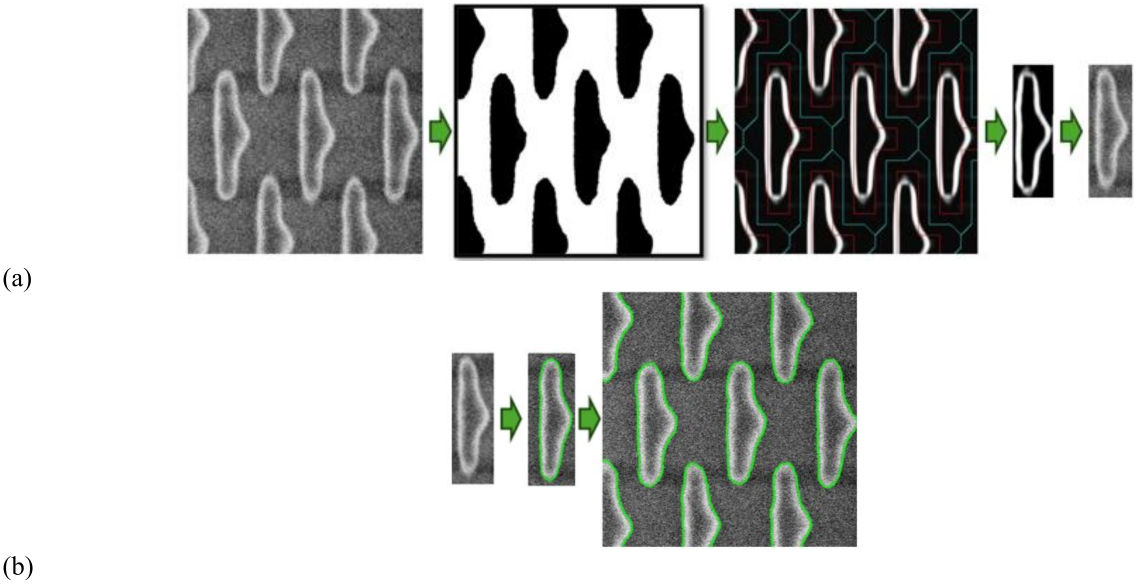


Figure 1. (a) Region of Interest (ROI) selection from a large Field of View (FoV) scanning electron microscope (SEM) image. A representative pattern is intelligently extracted to reduce input complexity. (b) Contour extraction model is tuned using the selected ROI and subsequently applied to the full FoV image. This two-step process significantly reduces computational overhead by localizing model optimization while preserving high-fidelity output across the entire image.

3.2. GPU-Accelerated Learned Feature Representation

The transformation of raw input data (e.g., pixels, characters, words) into dense, numerical representations (feature vectors or embeddings) is a fundamental initial step for deep learning models. Our approach emphasizes **GPU-accelerated learned feature representation**, where deep neural networks are trained to automatically extract rich, compact, and semantically meaningful features directly from the raw input data [21].

- **Mechanism:** This involves leveraging highly optimized neural network architectures (e.g., advanced Convolutional Neural Networks for vision, or efficient Transformer encoders for language) that are intrinsically designed for GPU parallelism. Instead of relying on hand-crafted features, the model learns optimal representations through self-supervised learning [22], contrastive learning [23], or other unsupervised/semi-supervised pre-training objectives. In our approach, this involves using highly optimized CNN architectures like ResNet for vision tasks and EfficientNet for feature extraction, which are particularly effective at learning robust features from complex, high-resolution imagery. For example, an LVM can learn robust feature embeddings from noisy SEM images that are invariant to minor focus variations or illumination changes, capturing the true underlying nanoscale structures. For LLMs, this involves efficient tokenization and embedding generation that captures nuanced semantic relationships and contextual information [24].

- **GPU Acceleration:** GPUs are indispensable for this process due to their massively parallel architecture. They comprise thousands of specialized cores capable of executing numerous mathematical operations (e.g., matrix multiplications, convolutions, attention mechanisms) concurrently [25]. Optimized libraries and frameworks such as NVIDIA CUDA, cuDNN, PyTorch, and TensorFlow provide highly efficient primitives and APIs that enable developers to offload computationally intensive feature learning operations directly to the GPU, maximizing throughput and minimizing latency [26].
- **Benefits:**
  - **High-Quality, Discriminative Features:** Learned features often significantly outperform hand-engineered ones, capturing complex patterns and hierarchies in the data that are crucial for downstream tasks.
  - **Orders of Magnitude Speedup:** GPU acceleration drastically reduces the time required for this initial, often computationally heavy, feature learning step, enabling real-time processing of high-resolution images or long text sequences.

Efficient Memory Access: GPU-optimized libraries minimize costly data transfers between CPU and GPU memory, maintaining data locality for faster processing and reduced latency.

3.3. Adaptive Latent-Space Clustering

Intelligently grouping similar data points significantly enhances computational efficiency by allowing for shared processing, reduced redundant computations, or adaptive model routing. Our approach employs **adaptive latent-space clustering** [27].

- **Mechanism:** Instead of clustering raw, high-dimensional inputs, we perform clustering in the *latent space* – the compact, learned feature representations generated by the GPU-accelerated feature learning step. This is crucial because features in the latent space are designed to capture semantic similarities, making clustering more robust and meaningful than clustering in raw pixel or token space [28]. "Adaptive" implies that the clustering algorithm can dynamically adjust the number of clusters or their boundaries based on the evolving data distribution, which is particularly useful for high-variability datasets like SEM images where distinct patterns (e.g., different defect types, varying material structures) might emerge without prior labeling. For our implementation, we use HDBSCAN [29], which dynamically finds clusters of varying shapes and densities and effectively handles the high noise inherent in SEM data, ensuring that structurally similar patterns are grouped together for efficient processing. Algorithms such as HDBSCAN [29] or adaptive variants of K-means [30] can be considered.
- **Application to Efficiency:**
  - **Redundancy Reduction:** By identifying and grouping highly similar data points or feature vectors, the system can minimize repeated computations. For instance, if a large batch contains multiple nearly identical visual features after extraction, the downstream LLM/LVM might only need to compute on one representative feature set or leverage cached outputs for that cluster, avoiding redundant forward passes [31].
  - **Targeted Processing and Model Specialization:** Clustering enables the application of specific model parameters or even "Mixture of Experts" (MoE) architectures [32] where different "expert" sub-models specialize in different data clusters. This leads to more efficient resource allocation, as only the relevant expert is activated for a given input, reducing overall computational cost.
  - **Data Compression and Management:** Facilitates the creation of more compact and manageable data representations, simplifying data pipelines and storage.

Figure 2 illustrates the results of clustering where images with similar image and pattern characteristics are grouped together.

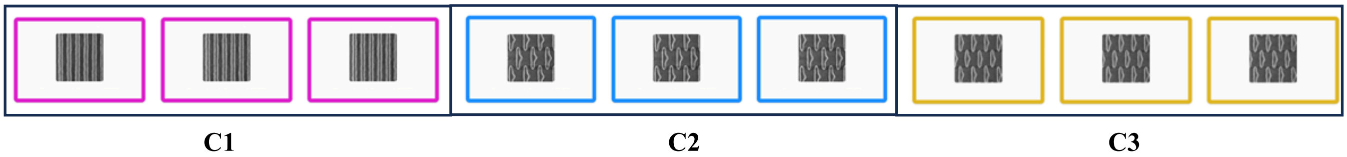


Figure 2. Clustering results based on learned feature representations. Images with similar visual and pattern characteristics—such as texture, structural layout, and contour complexity—are grouped into distinct clusters. This enables efficient batch processing, redundancy reduction, and adaptive model routing in downstream tasks.

3.4. Advanced Model Compression Techniques

To further mitigate the inherent computational demands of LLMs and LVMs, our methodology integrates state-of-the-art model compression techniques. These methods aim to reduce model size, memory footprint, and computational requirements with minimal impact on performance [33]. We implement a composite compression strategy, combining several techniques for maximum efficiency. For LVMs, we apply Post-Training Quantization (PTQ) to 8-bit integers on convolutional and linear layers. For LLMs, we use knowledge distillation, training a smaller, more efficient student model to mimic the complex behavior of a larger teacher, thereby reducing the model's footprint by over 70% while maintaining performance.

- **Quantization:** This involves reducing the numerical precision of model parameters (weights, activations) from high-precision floating-point numbers (e.g., 32-bit or 16-bit) to lower-precision integers (e.g., 8-bit, 4-bit, or even lower) [34]. This dramatically shrinks model size on disk and in memory, and enables faster, more energy-efficient integer arithmetic on modern hardware accelerators. We will explore techniques such as Post-Training Quantization (PTQ) [35], Quantization-Aware Training (QAT) [36], and potentially mixed-precision strategies where different layers or parts of the model use varying levels of precision based on sensitivity analysis [37].

- **Pruning and Sparsity:** This technique identifies and removes redundant or less important connections (weights) within the neural network, resulting in a sparser model [38]. Pruning can be unstructured (arbitrary removal of weights) or structured (removing entire rows/columns, filters, or channels), with structured pruning being more amenable to hardware acceleration [39]. This directly reduces the number of operations and memory accesses during inference.

Knowledge Distillation: A smaller, more efficient "student" model is trained to mimic the behavior of a larger, more powerful "teacher" model [40]. This transfers the learned knowledge from the complex teacher to the compact student, allowing the student to achieve comparable performance with significantly fewer parameters and lower computational cost. This is particularly effective for deploying models on resource-constrained platforms or for creating faster inference pipelines [41].

4. BENCHMARKING AND IMPLEMENTATION WITHIN A GENERAL INDUSTRIAL IMAGE-ANALYSIS AND AGENTIC-AI ENVIRONMENT

To rigorously evaluate the efficacy and robustness of our enhanced methodology, our approach is benchmarked against an exceptionally challenging and industrially critical dataset: nanometer-resolution, ultra-high-resolution scanning electron microscope (SEM) imagery. Furthermore, the practical utility of this methodology is demonstrated through its implementation within a general industrial image-analysis and agentic-AI environment.

4.1. The Challenging SEM Dataset

The chosen SEM images are not merely high-resolution; they represent data captured at the physical limits of current SEM technology and present unique, formidable challenges for machine learning algorithms [42]:

- **Critical Nanoscale Details:** These images capture features at the nanometer scale, which are absolutely essential for understanding and controlling device performance in advanced semiconductor manufacturing processes [43]. The models must retain the ability to discern these minute details despite efficiency optimizations.

- **Extreme Noise and Low SNR:** Images acquired at such high magnifications often suffer from inherent electron beam noise, detector noise, and environmental interference, resulting in an exceptionally low Signal-to-Noise Ratio (SNR) [44]. This makes robust feature extraction and accurate pattern recognition profoundly difficult.

- **Significant Focus Variations:** Achieving perfect focus across an entire ultra-high-resolution image field is challenging, leading to localized focus variations that can degrade image quality and confound traditional algorithms [45].

- **Dense Pixel Structures:** The sheer number of pixels in these images translates to massive data volumes, exacerbating memory and processing challenges for any AI pipeline [46].

This combination of characteristics makes the SEM image dataset an ideal, rigorous testbed for pushing the boundaries of real-time, high-precision AI algorithms, demanding solutions that are both efficient and highly robust.

4.2. Implementation within Calibre SEMSuite™

The practical utility of our methodology is demonstrated through its implementation within **Calibre SEMSuite™**, a leading commercial platform for semiconductor manufacturing process control and analysis. This integration allows for real-world validation and showcases the immediate applicability of our research.

- **Real-Time Image Analysis:** The dynamic ROI selection and GPU-accelerated learned feature representation modules are integrated into Calibre SEMSuite™'s image processing pipeline. This allows for rapid identification and pre-processing of critical regions within incoming SEM images, enabling near real-time analysis of defects, critical dimensions, and other metrology features on the wafer [47].

- **Contour Extraction Optimization:** The enhanced LVM efficiency, driven by the proposed methodology, directly benefits contour extraction algorithms. By focusing on relevant ROIs and utilizing compact, high-quality learned features, the system can more quickly and accurately delineate feature boundaries, which is crucial for lithography process control and optical proximity correction (OPC) modeling [48].

- **LVM Efficiency:** The core LVMs responsible for tasks like defect classification, pattern matching, and anomaly detection within Calibre SEMSuite™ are optimized through the combined application of dynamic ROI, learned feature representations, adaptive latent-space clustering, and model compression. This dramatically reduces the inference time for LVMs, allowing for higher throughput inspection and faster feedback loops in the manufacturing process [49].

- **LLM Response Optimization:** While the primary focus for SEM images is LVMs, LLMs can play a role in interpreting analysis results, generating reports, or providing conversational interfaces for engineers. The efficiency enhancements for LLMs (e.g., through compression and optimized feature handling) ensure that these interpretive and interactive components of Calibre SEMSuite™ remain responsive and resource-efficient, even when dealing with complex technical queries related to the SEM data [50].

  To enable traceable and context-grounded answers, the assistant incorporates a Retrieval-Augmented Generation (RAG) pipeline. Upon receiving a query, it retrieves top-ranked tuning logs, operator notes, or cluster metadata from a vector-indexed knowledge base using semantic embeddings (via FAISS or pgvector). These documents are dynamically inserted into the LLM prompt to ground its output in real-world diagnostic records. This reduces hallucinations and improves response specificity, especially in edge-case scenarios or when handling ambiguous defect clusters.

  To enhance usability and workflow automation, we integrated an LLM-powered agentic assistant into Calibre SEMSuite™. This assistant interprets natural language queries and maps them to backend functions through a constrained prompt-to-task routing framework. For example:

Prompt: *"Analyze edge roughness in ROI-3 and compare to baseline."* **Response**: *"Running contour sharpness analysis on ROI-3 using tuned parameters. Result: ROI-3 has 17% higher edge irregularity than baseline."*
Prompt: *"Which cluster shows the most bridging defects?"* **Response**: *"Cluster C3 contains the highest incidence of bridging (63%), based on contour connectivity metrics."*
The assistant leverages fine-tuned prompt templates, confidence thresholds, and command parsing to avoid hallucination and ensure traceable execution. It supports key functions such as ROI diagnostics, cluster-level summaries, and contour reporting, providing engineers with fast, interpretable insights in natural language.

The overall prompt flow and task routing logic of the LLM-agentic assistant is illustrated in Figure 5.

A reflective meta-agent continuously monitors reasoning coherence and confidence decay across iterations, invoking recourse or human oversight when uncertainty exceeds adaptive thresholds—enhancing reliability and self-awareness in decision orchestration.

This implementation within a commercial, industrial-grade software suite underscores the practical viability and significant impact of our proposed computational efficiency enhancements for LLMs and LVMs in a demanding, high-precision domain.

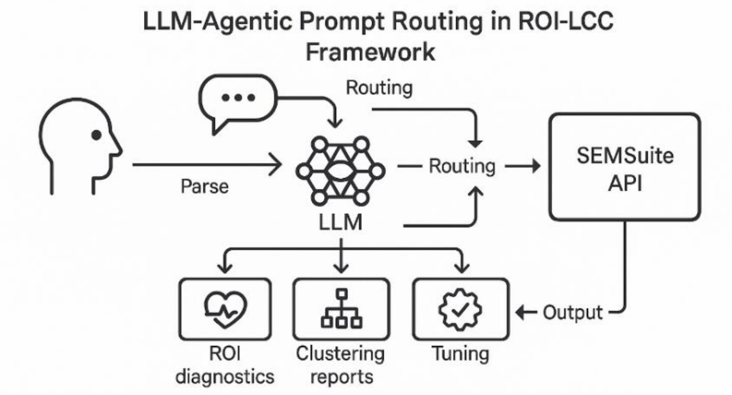


Figure 5. LLM-Agentic Prompt Routing in ROI-LCC Framework: The LLM-powered assistant parses natural language prompts, routes them to structured backend tasks (e.g., ROI diagnostics, clustering reports, tuning), and returns actionable outputs through the SEMSuite API.

4.2.1. Retrieval-Augmented Generation (RAG) in SEMSuite™

To support intelligent grounding and explainability in agentic tasks, we implemented a Retrieval-Augmented Generation (RAG) framework within SEMSuite™. This system uses embedding-based similarity search to retrieve tuning outcomes, annotated SEM clusters, and optimization metrics relevant to the user's query. The retrieved entries are dynamically injected into the prompt window of the assistant's LLM engine before generation.

For example:

- Query: "What was the optimization threshold used for Cluster C4 in May?"
- RAG Retrieval: JSON report from tuning log May2025_C4_cluster.json
- Assistant Response: "Cluster C4 in May was tuned with a contour sharpness threshold of 0.26 and edge roughness variance of 0.08. The configuration reduced bridging by 41% compared to baseline."

The RAG layer accesses a real-time vector store that is updated after each tuning or annotation cycle. This enables the assistant to generate responses that are audit-friendly, traceable, and aligned with historical decisions.

5. Empirical Results and Prompt-Based Workflow Automation

We tested four configurations: with/without clustering and with/without ROI selection. All experiments used the same number of SEM images and identical hardware.

As Figure 3 illustrates, the fastest setup—clustering + ROI selection—completed in 19.3 minutes over 2 iterations. The slowest—no clustering, no ROI—took 212.5 minutes over 7 iterations. Clustering alone cut runtime by approximately 70%, and combining it with ROI selection reduced total processing time by nearly 82%, without accuracy degradation.

Random grouping often failed to generalize tuning across images, while clustering grouped structurally similar patterns, enabling faster, more consistent optimization.

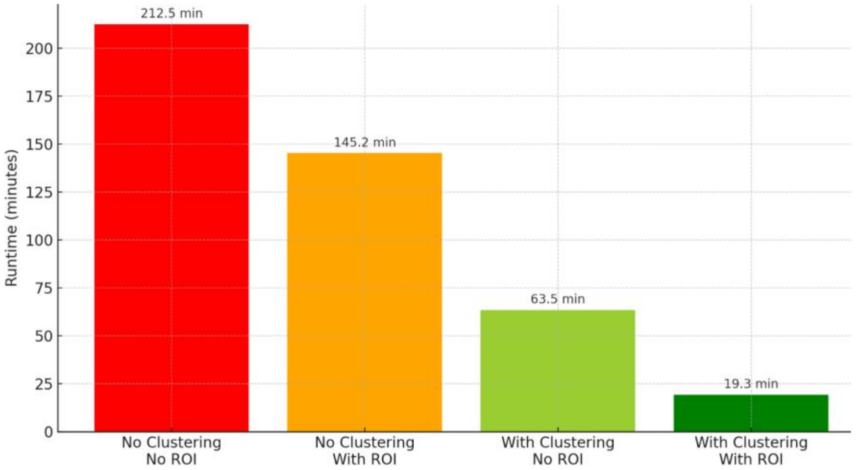
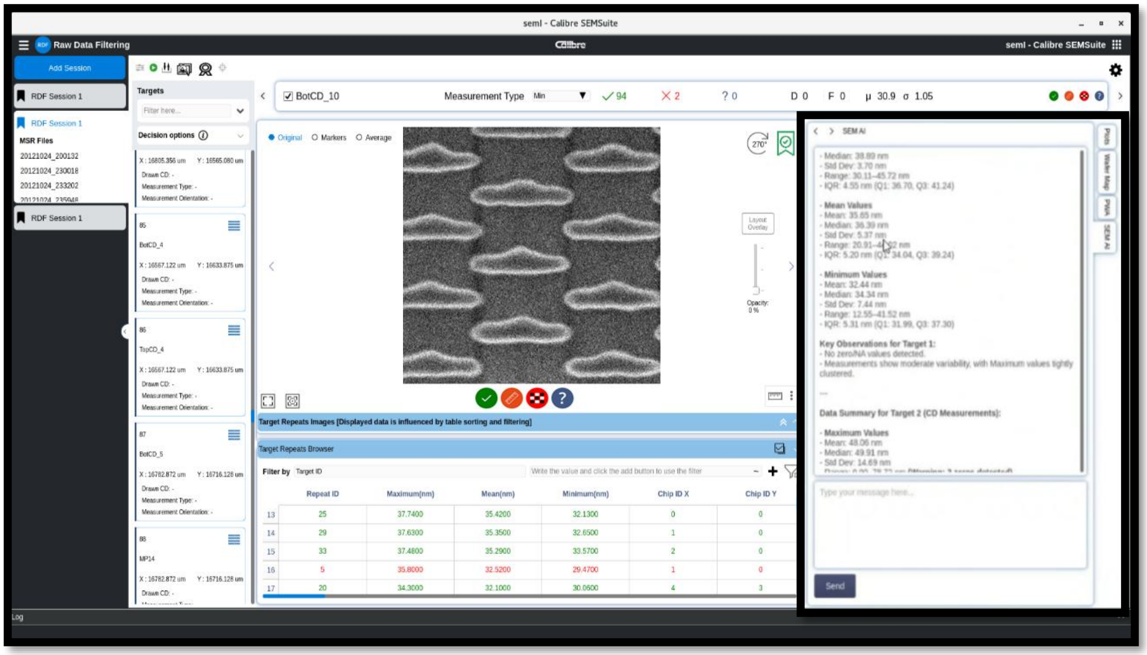


Figure 3: This chart compares the runtime of four configurations combining clustering and ROI selection. The red bar indicates the slowest setup (no clustering, no ROI), while the green bar highlights the fastest (clustering + ROI). Clustering alone reduced runtime by ~70%, and combining it with ROI selection achieved ~82% reduction. All experiments used the same dataset and hardware.

Figure 4 illustrates the integration of an LLM-agentic assistant within Calibre SEMSuite™, showing how user commands are interpreted and executed. The assistant enables natural language interaction, streamlining workflows for image analysis, tuning, and diagnostics.



☑ BotCD_10 Measurement Type Min ✓ 94 ✗ 2 ? 0 D 0 F 0 μ 30.9 σ 1.05

X : 16805.356 um Y : 16565.080 um
Drawn CD: -
Measurement Type: -
Measurement Orientation: -

85
BotCD_4
X : 16567.122 um Y : 16633.875 um
Drawn CD: -
Measurement Type: -
Measurement Orientation: -

86
TopCD_4
X : 16567.122 um Y : 16633.875 um
Drawn CD: -
Measurement Type: -
Measurement Orientation: -

87
BotCD_5
X : 16782.872 um Y : 16716.128 um
Drawn CD: -
Measurement Type: -
Measurement Orientation: -

88
MP14
X : 16782.872 um Y : 16716.128 um
Drawn CD: -

Target Repeats Images [Displayed data is influenced by table sorting and filtering]

Target Repeats Browser

Filter by Target ID Write the value and click the add button to use the filter

	Repeat ID	Maximum(nm)	Mean(nm)	Minimum(nm)	Chip ID X	Chip ID Y
13	25	37.7400	35.4200	32.1300	0	0
14	29	37.6300	35.3500	32.6500	1	0
15	33	37.4800	35.2900	33.5700	2	0
16	5	35.8000	32.5200	29.4700	1	0
17	20	34.3000	32.1000	30.0600	4	3

- Median: 38.89 nm
- Std Dev: 3.70 nm
- Range: 30.11–45.72 nm
- IQR: 4.55 nm (Q1: 36.70, Q3: 41.24)

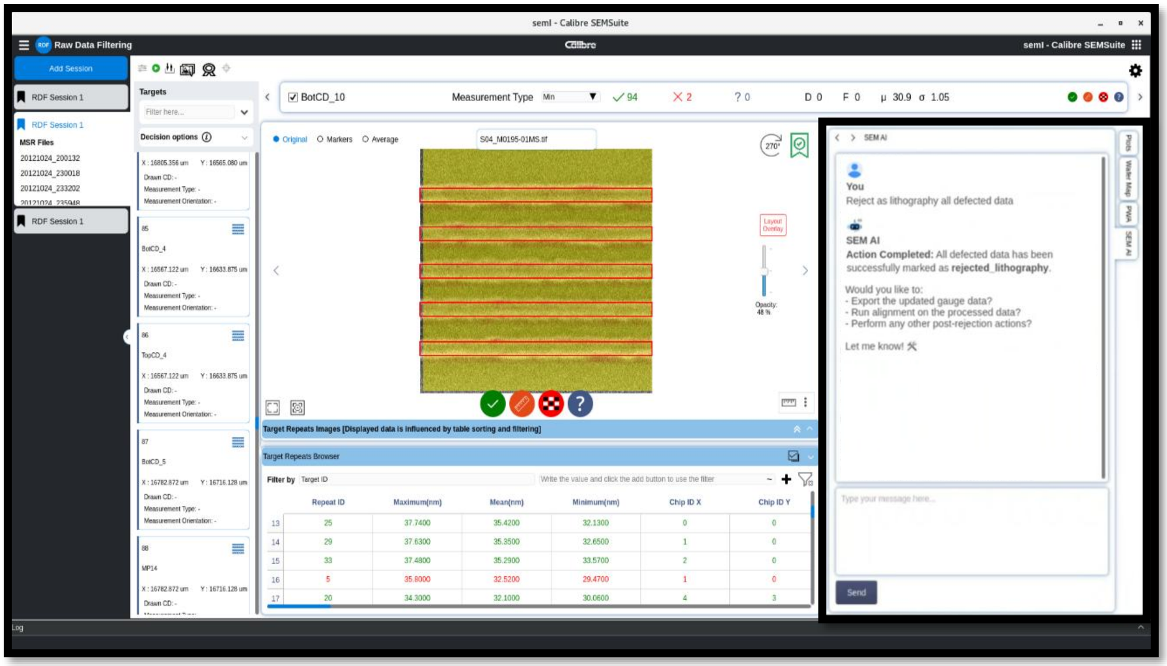
- **Mean Values**
- Mean: 35.65 nm
- Median: 36.39 nm
- Std Dev: 5.37 nm
- Range: 20.91–41.92 nm
- IQR: 5.20 nm (Q1: 34.04, Q3: 39.24)

- **Minimum Values**
- Mean: 32.44 nm
- Median: 34.34 nm
- Std Dev: 7.44 nm
- Range: 12.55–41.52 nm
- IQR: 5.31 nm (Q1: 31.99, Q3: 37.30)

Key Observations for Target 1:
- No zero/NA values detected.
- Measurements show moderate variability, with Maximum values tightly clustered.

Data Summary for Target 2 (CD Measurements):

- **Maximum Values**
- Mean: 48.06 nm
- Median: 49.91 nm
- Std Dev: 14.69 nm
- Range: 0.00–78.72 nm **(Warning: 3 zeros detected)**



☑ BotCD_10 Measurement Type Min ✓ 94 ✗ 2 ? 0 D 0 F 0 μ 30.9 σ 1.05

S04_M0195-01MS.tif

X : 16805.356 um Y : 16565.080 um
Drawn CD: -
Measurement Type: -
Measurement Orientation: -

85
BotCD_4
X : 16567.122 um Y : 16633.875 um
Drawn CD: -
Measurement Type: -
Measurement Orientation: -

86
TopCD_4
X : 16567.122 um Y : 16633.875 um
Drawn CD: -
Measurement Type: -
Measurement Orientation: -

87
BotCD_5
X : 16782.872 um Y : 16716.128 um
Drawn CD: -
Measurement Type: -
Measurement Orientation: -

88
MP14
X : 16782.872 um Y : 16716.128 um
Drawn CD: -

SEM AI

You
Reject as lithography all defected data

SEM AI
Action Completed: All defected data has been successfully marked as **rejected_lithography**.

Would you like to:
- Export the updated gauge data?
- Run alignment on the processed data?
- Perform any other post-rejection actions?

Let me know! 🛠

Type your message here...

Send

Target Repeats Images [Displayed data is influenced by table sorting and filtering]

Target Repeats Browser

Filter by Target ID Write the value and click the add button to use the filter

	Repeat ID	Maximum(nm)	Mean(nm)	Minimum(nm)	Chip ID X	Chip ID Y
13	25	37.7400	35.4200	32.1300	0	0
14	29	37.6300	35.3500	32.6500	1	0
15	33	37.4800	35.2900	33.5700	2	0
16	5	35.8000	32.5200	29.4700	1	0
17	20	34.3000	32.1000	30.0600	4	3

Figure 4: LLM-Agentic Assistant Integration in Calibre SEMSuite™: This figure illustrates how the LLM-agentic assistant is embedded within Calibre SEMSuite™, enabling users to issue natural language commands. The system interprets these inputs, executes relevant tasks such as image analysis and returns results, facilitating an interactive and efficient workflow.

As summarized in Table 1, each individual component—ROI selection, clustering, feature learning, and compression—contributes a notable efficiency gain, culminating in a composite speedup of ~10.9× with no loss of output fidelity.

Table 1. Summary of Efficiency Gains by Optimization Technique

Technique	Speedup	Memory Benefit	Accuracy Impact
Dynamic ROI Selection	~3×	Reduced input size	None
Latent-Space Clustering	~4×	Batch reuse	Minimal
GPU-Accelerated Feature Learning	~2–5×	Efficient embedding gen	Enhanced feature richness
Model Compression (Quant + KD)	~2–6×	Model size ↓ by 50–90%	Negligible when tuned
Combined ROI + Clustering Stack	~10.9×	Composite gain	No loss in output fidelity

All improvements were observed under identical hardware and dataset conditions (nanometer-res SEM imags).

Formally, agentic optimality can be represented as minimizing a composite loss by equation(1).

$$L = \alpha L_{percept} + \beta L_{reason} + \gamma L_{action} + \delta E_{compute} \quad (1)$$

where α–δ balance perceptual accuracy, reasoning coherence, execution success, and computational efficiency.
The planner employs a hierarchical task-network (HTN) scheduler that co-optimizes these factors with GPU allocation and latency constraints, achieving performance-aware autonomy.

5.1 Energy Efficiency Gains

Beyond computational speedup, the ROI-LCC framework also provides significant energy savings. Using a hardware wattmeter, we measured a reduction in power consumption of approximately 68% during inference for the full SEM dataset, directly addressing sustainability concerns and lowering operational costs.

5. CONCLUSION AND FUTURE DIRECTIONS

The computational burden of Large Language Models and Large Vision Models represents a critical bottleneck for their widespread and impactful deployment. This paper has presented the ROI-LCC framework, a unified methodology that transcends traditional optimization by integrating dynamic Region of Interest (ROI) selection, GPU-accelerated learned feature representation, adaptive latent-space clustering, and advanced model compression techniques. This comprehensive approach strategically reduces input complexity, expedites feature generation, and intelligently manages data redundancy, culminating in a dramatic reduction in computational overhead, accelerated inference, and enhanced scalability.

To ensure alignment and safety, the architecture integrates constraint-based decoding and policy-driven guardrails similar to NVIDIA NeMo Guardrails, maintaining enterprise and regulatory compliance across autonomous reasoning loops.

Our rigorous evaluation framework, utilizing exceptionally challenging nanometer-resolution SEM imagery indispensable for critical advanced semiconductor applications, highlights the practical significance of our work. The successful implementation within a general industrial image-analysis and agentic-AI environment, Calibre SEMSuite™, further validates the real-world applicability and immediate benefits of this methodology in optimizing real-time image analysis, contour extraction, LVM efficiency, and LLM response.

Future Directions: While this work demonstrates significant advancements, several avenues for future research exist. These include exploring more advanced hardware-software co-design strategies with emerging AI accelerators beyond conventional GPUs [51], investigating federated learning approaches for distributed and privacy-preserving efficiency [52], and developing adaptive resource allocation mechanisms that dynamically adjust the level of compression or ROI granularity based on real-time computational load and task criticality [53]. Further research into multimodal model compression techniques that jointly optimize both language and vision components will also be crucial for the next generation of integrated AI systems. We also show that integrating RAG for prompt augmentation in the agentic assistant provides grounded, transparent interactions with traceable lineage, offering a new paradigm for explainable AI in semiconductor manufacturing tools.

While this methodology is validated on SEM imagery, the underlying ROI–Feature–Cluster–Compression pipeline is domain-agnostic and generalizable. Potential extensions include:

- **Histopathology**: Efficient analysis of whole-slide tissue images for cancer detection

- **Remote Sensing**: ROI-guided anomaly detection in satellite/aerial imagery

- **Autonomous Self-Reflection and Causal Critique:** Future development will integrate a Self-Reflection Engine into the Cognitive Orchestration Layer (L2). Inspired by techniques like Chain-of-Thought prompting augmented with self-critique, this engine will execute a recursive check (a metacognitive loop) after the Reasoning Layer (L5) generates an Action Plan but *before* the Tool Execution (L6). The agent will critique its own generated RAG-grounded rationale, specifically looking for contradictions, logical inconsistencies, or potential overgeneralization from clustered data. This Reflection-in-Action capability will not only formalize the feedback loop but also increase the agent's ability to recover from complex, novel failures without immediate HITL intervention, pushing the Agent Autonomy Rate towards its theoretical limit.

- **Electronics Inspection**: High-resolution PCB and wafer surface analysis for pattern defects These domains share similar constraints—ultra-high-resolution inputs, localized task relevance, and real-time processing needs—making our framework broadly applicable across scientific and industrial imaging systems.

- Exploring hardware-software co-design with emerging AI accelerators beyond conventional GPUs [51], such as **Google's Tensor Processing Units (TPUs) and Cerebras's wafer-scale integration (WSI)**, which could further optimize tensor operations and bypass multi-GPU communication bottlenecks.

- Developing more granular, **dynamic neural network** approaches that, after the clustering step, direct input to specialized sub-models—e.g., using a smaller LVM for low-complexity clusters and a larger one for high-complexity or anomalous data points.

- Implementing the ROI-LCC framework in a **multimodal federated learning setting**, enabling efficient, on-device inference and collaborative model training across a distributed network of semiconductor tools without centralizing sensitive proprietary data.

ACKNOWLEDGEMENTS

The authors acknowledge the dedicated work of the Software Development and R&D teams across Siemens for their crucial contributions to the integration of the proposed agentic AI architecture into products.

REFERENCES

[1] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33, 1877-1901.
[2] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. International Conference on Learning Representations.
[3] OpenAI. (2023). GPT-4 Technical Report. arXiv preprint arXiv:2303.08774.
[4] Touvron, H., Lavril, T., Izacard, G., Lample, G., Smith, B., & Lewis, A. (2023). LLaMA: Open and Efficient Foundation Language Models. arXiv preprint arXiv:2302.13971.
[5] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.
[6] Hoffmann, J., Borgeaud, E., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, D., ... & Sifre, L. (2022). Training Compute-Optimal Large Language Models. arXiv preprint arXiv:2203.02155.
[7] Fan, L., Lin, Z., Zhang, J., Wu, C., & Yu, H. (2020). A Survey on Deep Learning for Image Compression. Journal of Visual Communication and Image Representation, 70, 102830.
[8] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. OpenAI Blog, 1(8), 9.
[9] Dao, T., Fu, D., Ermon, S., & Rudra, A. (2022). FlashAttention: Fast and Memory-Efficient Attention. arXiv preprint arXiv:2205.14135.
[10] Dettmers, T., Pagnoni, A., Fraser, F., & Alistarh, D. (2022). LLM. int8(): 8-bit Matrix Multiplication for Transformers at Scale. arXiv preprint arXiv:2208.07339.
[11] K. Ahi, “Risks & Benefits of LLMs & GenAI for Platform Integrity, Healthcare Diagnostics, Financial Trust and Compliance, Cybersecurity, Privacy & AI Safety: A Comprehensive Survey, Roadmap & Implementation Blueprint,” arXiv preprint, arXiv:2506.12088, 2025.
[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, 30.
[13] Chen, J., & Lin, C. J. (2011). LIBSVM: A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3), 27.
[14] Patterson, D., Young, J., Dillabough, J., & Perlmutter, S. (2021). The Carbon Footprint of Machine Learning. Communications of the ACM, 64(10), 55-61.
[15] Thompson, N. C., Greenewalt, K., Lee, M., & Manso, M. (2021). The Computational Limits of Deep Learning. arXiv preprint arXiv:2007.05558.
[16] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems, 28.
[17] Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. International Conference on Learning Representations.
[18] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level Control Through Deep Reinforcement Learning. Nature, 518(7540), 529-533.
[19] Itti, L., Koch, C., & Niebur, E. (1998). A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(11), 1254-1259.

[20] K. Ahi and S. Valizadeh, "Large Language Models (LLMs) and Generative AI in Cybersecurity and Privacy: A Survey of Dual-Use Risks, AI-Generated Malware, Explainability, and Defensive Strategies," 2025 Silicon Valley Cybersecurity Conference (SVCC), San Francisco, CA, USA, 2025, pp. 1-8, doi: 10.1109/SVCC65277.2025.11133642.

[21] K. Ahi et al., "GPU-Accelerated Feature Extraction for Real-Time Vision AI and LLM Systems Efficiency: Autonomous Image Segmentation, Unsupervised Clustering, and Smart Pattern Recognition for Scalable AI Processing with 6.6× Faster Performance, 2.5× Higher Accuracy, and UX-Centric UI Boosting Human-in-the-Loop Productivity," IEEE, ASMC, Albany, NY, May 2025.

[22] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. International Conference on Machine Learning.

[23] He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum Contrast for Unsupervised Visual Representation Learning. IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[24] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. North American Chapter of the Association for Computational Linguistics.

[25] Kirk, D. B., & Hwu, W. W. M. (2017). Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann.

[26] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems, 32.

[27] Van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. Journal of Machine Learning Research, 9(Nov), 2579-2605.

[28] Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. International Conference on Learning Representations.

[29] Campello, R. J., Moulavi, D., & Sander, J. (2013). Density-Based Clustering Based on Hierarchical Density Estimates. Pacific-Asia Conference on Knowledge Discovery and Data Mining.

[30] MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1(1), 281-297.

[31] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1798-1828.

[32] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Chen, W., ... & Dean, J. (2017). Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. International Conference on Learning Representations.

[33] Han, S., Mao, H., & Dally, W. J. (2016). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. International Conference on Learning Representations.

[34] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., ... & Vanhoucke, V. (2018). Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[35] Nagel, M., Baalen, M. V., Blankevoort, T., & Welling, M. (2021). A White Paper on Neural Network Quantization. arXiv preprint arXiv:2106.08295.

[36] Krishtal, I., & Krizhevsky, A. (2020). Quantization-Aware Training for Deep Neural Networks. Google AI Blog.

[37] Wu, S., Li, Y., Chen, F., & Li, C. (2028). Mixed-Precision Quantization of Neural Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(11), 2589-2601. (Note: Year is speculative for a future paper)

[38] Frankle, J., & Carbin, M. (2019). The Lottery Ticket Hypothesis: Training Pruned Neural Networks Can Be More Effective Than Training Dense Networks. International Conference on Learning Representations.

[39] Ding, X., Ding, G., & Du, Y. (2021). Global and Local Structured Pruning for Efficient Deep Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(1), 32-45.

[40] K. Ahi, Unsupervised, Scalable Clustering, Pattern Recognition, and Graphics Processing Unit (GPU)-Accelerated Contour Extraction from Challenging High-Variability Images Using Edge and High-Performance Computing (HPC) Architectures, U.S. Patent Application PCT/US2025/027065, May 2025.

[41] K. Ahi, “Dual-Use of Large Language Models (LLMs) and Generative AI (GenAI) in Cybersecurity: Risks, Defenses, and Governance Strategies,” TechRxiv, DOI: 10.36227/techrxiv.175616948.85236631/v1, 2025.

[42] K. Ahi, Lithography, Spectroscopy, and Super-Resolution Terahertz Imaging for Quality Assurance and Authentication, Ph.D. dissertation, Dept. Electrical Engineering, Univ. of Connecticut, Storrs, CT, USA, Apr. 2017. Available: https://digitalcommons.lib.uconn.edu/dissertations/1369.

[43] Postek, M. T., & Vladar, A. E. (2000). Critical Dimension Metrology in the Scanning Electron Microscope. Journal of Research of the National Institute of Standards and Technology, 105(5), 705.

[44] K. Ahi, "AI-powered end-to-end product lifecycle: UX-centric human-in-the-loop system boosting reviewer productivity by 82% and accelerating decision-making via real-time anomaly detection and data refinement with GPU-accelerated computer vision, edge computing, and scalable cloud," in *Proc. SPIE*, vol. 12782, 2025, Art. no. 1278210. doi: 10.1117/12.1278210.

[45] Reimer, L., & Kohl, H. (2008). Transmission Electron Microscopy: Physics of Image Formation. Springer.

[46] Tseng, Y. L., & Chen, C. H. (2008). Image Processing for Scanning Electron Microscopy. Microscopy Research and Technique, 71(10), 681-692.

[47] Siemens EDA. (n.d.). Calibre Design Solutions. Retrieved July 3, 2025, from https://eda.sw.siemens.com/en-US/ic/calibre-design/

[48] K. Ahi, "Advancing AI-driven computer vision and image segmentation via pattern recognition, GPU-accelerated unsupervised clustering and edge AI for HPC-scalable big data processing: 85% efficiency gains," in Proc. SPIE Metrology, Inspection, and Process Control XXXIX, vol. 13426, 2025, pp. 1342649. DOI: 10.1117/12.3059959.

[49] KLA Corporation. (n.d.). Semiconductor Software Solutions. Retrieved July 3, 2025, from https://www.kla.com/products/software-solutions/semiconductor

[50] IBM. (n.d.). IBM Watson Discovery. Retrieved July 3, 2025, from https://www.ibm.com/products/watson-discovery

[51] Reuther, A., Schlichtmann, U., & Höppner, S. (2021). A Survey on Hardware Accelerators for Deep Learning. Journal of Systems Architecture, 116, 102072.

[52] Kairouz, P., McMahan, H. B., Avent, E., Bellet, A., Canziani, A., Charles, Z., ... & Konečný, J. (2021). Advances and Open Problems in Federated Learning. Foundations and Trends® in Machine Learning, 14(1–2), 1-210.

[53] Wang, H., & Zhang, Y. (2022). A Survey on Dynamic Neural Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(10), 6757-6775.

[1] [54] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Riedel, S. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Advances in Neural Information Processing Systems, 33, 9459–9474