

SimuScan-Agent: language-model orchestration of synthetic-data-driven autonomous atomic force microscopy

Ruben Millan-Solsona

solsonarm@ornl.gov

Oak Ridge National Laboratory <https://orcid.org/0000-0003-0912-7246>

Marti Checa

Oak Ridge National Lab <https://orcid.org/0000-0003-2607-6866>

Yongtao Liu

Oak Ridge National Laboratory <https://orcid.org/0000-0003-0152-1783>

Bernadeta Srijanto

Oak Ridge National Laboratory

Rama K. Vasudevan

Oak Ridge National Laboratory (ORNL) <https://orcid.org/0000-0003-4692-8579>

Scott Retterer

Oak Ridge National Laboratory

Liam Collins

Oak Ridge National Lab

Article

Keywords:

Posted Date: October 21st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11435524/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: There is **NO** Competing Interest.

SimuScan-Agent: language-model orchestration of synthetic-data-driven autonomous atomic force microscopy.

Ruben Millan-Solsona^{1*}, Marti Checa¹, Yongtao Liu¹, Bernadeta Srijanto¹, Rama K. Vasudevan¹,
Scott Retterer^{1,2}, Liam Collins¹

¹Center for Nanophase Materials Sciences, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, USA.

²Biosciences Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, USA.

*Correspondence to: solsonarm@ornl.gov

Abstract: Atomic force microscopy (AFM) enables nanoscale characterization, but large-area investigations remain slow and require coordination of data preparation, model training and instrument operation. Here we introduce SimuScan-Agent, an LLM-orchestrated framework that translates natural-language objectives into adaptive AFM workflows. Combining multimodal morphological sources with physics-informed simulation, it generates annotated synthetic datasets without manual per-image labeling, trains and deploys segmentation and representation-learning models, and guides acquisition using online detections and class-specific sampling objectives. In simulations, probabilistic active exploration reduced measurements and sampled area by approximately 71% relative to raster scanning. Experimentally, SimuScan-Agent explored a ~ 10 mm² region by sampling 17.2% of available positions, estimating class-specific spatial distributions and assembling a morphological library to identify representative and atypical nanostructures. By linking synthetic-data generation, machine learning and adaptive acquisition through natural-language

interaction, the framework lowers barriers to goal-directed nanoscale studies and offers a foundation for more intelligent, autonomous AFM systems that adapt experimental decisions to what they learn during measurement.

1. Introduction

Atomic force microscopy (AFM)¹ is a versatile platform for nanoscale characterization, providing topographic and functional measurements in materials², nanostructures³, and biological systems⁴ and in environments ranging from vacuum to physiological liquids^{5,6}. Beyond morphology, AFM can probe local mechanical⁷, electrical^{8,9}, magnetic¹⁰, or electrochemical¹¹ properties, making it a powerful multimodal platform for studying heterogeneous materials and interfaces.

Importantly, data are often analyzed only after the experiment has finished, meaning that decisions about whether a region is informative, whether a rare population has been sampled sufficiently, or whether enough data have been collected are made retrospectively rather than during the experiment. This separation between acquisition and analysis can lead to substantial oversampling of uninformative regions or, conversely, to experiments that end before statistically representative information has been obtained. The challenge is particularly acute for large, heterogeneous, or sparsely populated samples, in which scientifically relevant features may occupy only a small fraction of the accessible surface and exhaustive raster scanning becomes inefficient. More generally, conventional AFM workflows have limited ability to use information obtained during acquisition to decide what should be measured next or when the experimental objective has been satisfied. Thus, the limitation is not simply that AFM is slow, but that acquisition is typically decoupled from real-time assessment of what has been learned and what remains to be measured.

Machine learning (ML) offers a route toward more scalable and autonomous AFM¹²⁻¹⁴ by enabling automated feature detection¹⁵, image-quality assessment, parameter optimization, data analysis, uncertainty estimation, and adaptive measurement selection¹⁶⁻²¹. Realizing these capabilities requires coordinating multiple specialized models for perception, analysis, control, and decision-making. This creates two interconnected challenges: generating representative

training data and integrating models and experimental tools into workflows that are reproducible, easy to execute, and do not require deep expertise in ML or programming.

The first challenge is particularly important for supervised ML, which requires sufficiently large and representative well-labeled datasets. The need for large, diverse, and accurately annotated AFM datasets has recently been emphasized by the bioAFM community, together with efforts to establish standardized and accessible data repositories²². Experimental AFM data are time-consuming to acquire, while manual annotation is labor-intensive, requires specialized expertise, and can introduce subjective bias. Training datasets must capture not only sample variability but also measurement-specific effects that distinguish AFM images from conventional images, including tip-sample convolution, instrumental noise, scanner distortions, substrate roughness, contamination, and other acquisition artifacts. General-purpose image generators can produce visually plausible images from natural-language prompts, while task-specific diffusion methods have also generated paired microscopy images and annotations.^{23,24} However, visual plausibility alone does not guarantee explicit control over AFM image-formation physics or reliable reference labels for quantitative model training. Nevertheless, generative models can provide a flexible means of proposing object morphologies that can subsequently be incorporated into physics-informed simulation workflows.

We recently developed SimuScan²⁵, a physics-based synthetic AFM data-generation framework that incorporates key aspects of image formation and produces precisely labeled datasets without the need for manual annotation. Models trained predominantly on these synthetic data have been shown to transfer to experimental AFM images²⁵, providing a practical solution to the training-data bottleneck. However, moving from synthetic data to an autonomous experiment still requires substantial AFM knowledge, ML and programming expertise to coordinate simulation, model training and evaluation, analysis, and instrument control across multiple tools and interfaces. Thus, translating an experimental objective into a reproducible

workflow that orchestrates data generation, model training and deployment, measurement, and analysis remains an unresolved impediment.

Large language models (LLMs) offer a potential interface for bridging this gap by translating natural-language objectives into structured configurations and tool calls^{22,26,27}. However, reliable scientific automation requires instrument operations, model execution, and quantitative analysis to remain constrained, reproducible, and traceable. This motivates an approach in which the LLM does not act as the experimental controller itself, but rather as an orchestration layer that interprets user intent and coordinates specialized tools that can be independently verified.

Accordingly, here we develop SimuScan-Agent, an LLM-orchestrated framework that connects physics-based synthetic data generation, ML model training and evaluation, and adaptive experimental AFM exploration within a modular and traceable workflow. SimuScan-Agent translates experimental objectives expressed in natural language into structured actions, delegating execution to specialized modules for simulation, model training, analysis, and instrument control. The framework also incorporates complementary approaches for defining object morphologies, including programmatic generation, generative image models, and retrieval of reference structures from scientific literature. These morphological references are integrated into SimuScan, where image formation and annotations remain explicitly controlled. We then integrate these capabilities through a standardized tool architecture in which the agent orchestrates and controls the workflow by delegating each action to specialized, independently verified modules for simulation, machine learning, analysis, and instrument control.

We demonstrate the complete framework through an end-to-end multiclass AFM experiment in which models trained on precisely labeled synthetic data are deployed during adaptive exploration of a nanostructured sample. Detected objects continuously update class-specific spatial probability maps, while a Bayesian strategy selects subsequent acquisition locations according to target probability, uncertainty, scanner-movement cost, and class-

specific sampling objectives. The resulting measurements reconstruct spatial distributions from partial sampling. At the same time, they build a morphological library in which learned representations identify representative and potentially anomalous structures. SimuScan-Agent therefore connects synthetic-data-driven AFM from model development to goal-directed experimental exploration, narrowing the gap between an experimental question and an adaptive nanoscale measurement workflow.

2. Results

SimuScan-Agent: LLM-orchestrated synthetic-data and AFM workflows

SimuScan-Agent organizes the workflow into specialized modules coordinated through an LLM-based orchestration layer (Fig. 1). The user interacts with the system via natural-language prompts and a graphical interface to inspect reference objects, preview synthetic data, and examine model-training and evaluation results. From these instructions, the agent translates user objectives into structured actions, selects appropriate tools, and coordinates their execution. The LLM acts as a high-level coordination layer rather than a direct microscope controller; simulation, model training, analysis, and acquisition remain implemented in specialized modules. The operational core comprises four connected modules: Multimodal Morphological Sources, which provides morphological inputs through procedural generation (PyGenerator), generative models, and experimental or literature-derived images; Simulator, dedicated to generating synthetic AFM images and annotations; Trainers, responsible for training and evaluating detection, segmentation, and representation-learning models; and Exploration, which executes automated large-area AFM exploration strategies and selects new measurement regions. This architecture provides an integrated framework that enables users to define experimental objectives through natural-language interaction and coordinate synthetic-data generation, model training and deployment, and autonomous, decision-driven AFM exploration, without requiring task-specific programming.

Multimodal routes for morphological library expansion

SimuScan-Agent expands the morphological library used for synthetic AFM data generation through three complementary routes (Fig. 2). First, PyGenerator translates natural-language descriptions into parameterized Python functions compatible with SimuScan (Fig. 2a–e). As demonstrated with four Gaussian peaks arranged in a circle with alternating heights, the agent generates executable geometry definitions and visual previews, enabling users to inspect and approve new object shapes without manually writing code.

Second, a general-purpose generative image model provides visual references for complex morphologies that are difficult to describe through simple geometric functions (Fig. 2f–i). Examples include circular DNA structures, bacilli with flagella, nanoparticle clusters, and DNA-origami structures. Through the interface, users can select object classes and specify their relative abundances or combinations for subsequent dataset generation. These images serve as morphological references rather than directly labeled AFM training data. The limitations of directly generating AFM-like images and segmentation masks using a general-purpose image model are examined in Supplementary Section S1.

Third, the retrieval-augmented generation (RAG) component enables the identification and extraction of morphological references from scientific literature (Fig. 2j–m). In the illustrated example, the agent locates a publication containing triangular DNA-origami structures and displays the corresponding figure²⁸. The user then selects the relevant region through a dedicated graphical interface, allowing the extracted morphology to be incorporated into the SimuScan object library without requiring a predefined geometric model.

Together, these demonstrations establish complementary routes for introducing procedural, generated, and literature-derived morphologies into the simulation workflow. Following user review and incorporation into SimuScan, the selected references are processed through its physics-informed image-formation pipeline, which controls object placement,

transformations, and AFM-associated artifacts while producing simulator-derived masks, annotations, and traceable metadata. This separation enables the expansion of morphological diversity without relying on generated or experimental reference images as quantitative ground truth, providing labeled synthetic datasets for subsequent model training and experimental AFM analysis.

Multiclass probabilistic active exploration

We evaluated multiclass probabilistic active exploration through simulations designed to locate two spatially distributed object populations, one abundant and one rare, while minimizing AFM measurements and sampled area. Across 100 randomly generated distributions on a 20×20 grid, we compared five acquisition strategies: raster scanning, random sparse sampling, standard active search, active search with 10% uncertainty-driven exploration, and active search with a reduced movement penalty. Representative trajectories and cumulative object detections are shown in **Fig. S2.1**, while **Fig. S2.2** summarizes the Monte Carlo results.

All three active-search strategies required fewer measurements than raster scanning and random sparse sampling. Raster acquisition required a median of 172.5 [126–215] images (interquartile range, IQR), sampling an average of 43% of the grid, whereas random sampling required 96 [68–146.5] images and covered 30%. Standard active search required only 38 [27–57] images, sampling 13% of the grid and reducing the mean number of measurements and sampled area by approximately 71% relative to raster acquisition. Reducing the movement penalty further decreased the required measurements to 33 [26–50.2] images (12% of the grid). Introducing 10% uncertainty-driven exploration required 43.5 [29–66.5] images (15%), retaining substantial efficiency gains while promoting exploration of less-characterized regions. Complete statistics and scanner displacement comparisons are provided in **Table S2**.

These results demonstrate that adaptive measurement selection substantially outperforms non-adaptive sparse sampling and conventional raster acquisition, while allowing

the balance between acquisition efficiency, scanner movement, and spatial exploration to be adjusted.

Experimental demonstration of the complete SimuScan-Agent workflow

After computationally validating the acquisition strategy, we next tested the complete SimuScan-Agent workflow using a fabricated library of nanostructures containing four geometric classes: ellipses, hexagons, squares, and triangles. The structures had known nominal positions and class abundances, providing a reference for evaluating object detection and the reconstructed spatial distributions. Because fabrication can modify or remove individual structures, the predefined arrangement represents the nominal design, whereas AFM measurements reveal the structures and morphologies actually present on the sample (Fig. S3).

The experimental demonstration was organized around three linked objectives. First, we evaluated whether synthetic data generated through natural-language interaction with SimuScan-Agent could be used directly to train recognition, segmentation, and metric-learning models for generating morphological *embeddings* (Fig. 3). Second, we deployed these models during adaptive AFM acquisition to efficiently locate objects of interest and reconstruct their class-specific spatial distributions from partial sampling (Fig. 4). Third, we organized the detected objects into a morphological library to characterize within-class variability and identify potentially unusual structures (Fig. 5). Together, these stages connect synthetic data generation and model training with autonomous acquisition, spatial inference, and morphological analysis. Supplementary Videos 1–3, available through the associated Zenodo repository (see Video Availability), illustrate the corresponding stages of the workflow, from synthetic data generation and model training to morphological analysis and autonomous AFM exploration.

To train the models used during exploration, SimuScan generated 4,761 simulator-labeled synthetic AFM images containing 25,560 instances of the four classes of interest

without manual annotation (Fig. 3a). Variations in object size, orientation, and position, together with changes in the background and AFM image-formation artifacts, were incorporated during synthesis, providing statistical variability that would be difficult to achieve with experimental, manually labeled data.

The resulting YOLO instance-segmentation model achieved precision, recall, and mAP50 values of approximately 0.96-0.99 on the held-out test set, with mAP50-95 values of approximately 0.92 for bounding boxes and 0.85 for masks (Fig. 3b). Most residual confusion occurred between partially cropped squares and triangles, consistent with contour truncation and tip-convolution effects. In parallel, a Triplet Network model²⁹ learned continuous morphological representations using anchor, positive, and negative samples to constrain relative distances within the latent space. The training and validation losses decreased consistently and converged to similar values, indicating stable representation learning without a pronounced train-validation divergence. The resulting embeddings separated the four morphological classes into distinct regions of the latent space while preserving within-class variation (Fig. 3c). Objects located near the centroid of each class showed the expected square, triangular, elliptical, and hexagonal morphologies, confirming that these centroid-associated objects provide representative examples of the corresponding classes. These trained models were then transferred to the automated exploration workflow for online object detection and subsequent construction of the experimental morphological library.

Adaptive multiclass acquisition and spatial reconstruction

The trained models were then deployed during automated exploration of a region of approximately 10 mm². Rather than following a raster trajectory, the microscope adaptively selected successive positions based on the information accumulated during acquisition (Fig. 4a). Exploration ended after 579 images, corresponding to only 17.2% of the available positions,

while online predictions were continuously transferred to the exploration module to update the spatial models (Fig. 4b).

The class-weighted acquisition objective progressively shifted sampling toward the less-represented populations. Ellipses exceeded the target of 100 objects after approximately 130 images, whereas hexagons and squares required nearly 500 positions to approach their targets (Fig. 4c). Only eight triangles were found, below the target of ten, so the rarest population remained below its requested target when the run ended. The acquisition was terminated because an additional experimental time limit of approximately 24 h was reached before all class-specific targets were satisfied. This illustrates an important advantage of multiclass exploration: acquisition does not stop simply because the abundant populations are sufficiently represented.

Class-specific spatial distributions were reconstructed online using Gaussian processes (GP) and compared with the nominal sample design used as a spatial reference (Fig. 4d,e). Spatial agreement was quantified using the overlapping coefficient (OVL) (See the methods section.), computed as the intersection between the normalized reconstructed and design-derived spatial distributions. For structured and sufficiently populated classes, reconstruction improved rapidly as coverage increased. For example, the spatial overlap for squares increased from approximately 60% after 2% of positions were sampled to 81% after 17.2% were sampled. The reconstruction curves for squares and hexagons began to plateau after approximately 10% of the grid had been sampled. Beyond this point, improvements became progressively smaller as the separation between sampled regions approached the characteristic correlation length of the spatial distribution³⁰. Because the nominal design does not account for fabrication defects, missing structures, or other deviations in the fabricated sample, residual disagreement should not be interpreted solely as reconstruction error. It may also reflect the inherent smoothing of the GP reconstruction and the nonuniform sampling introduced by the quota-based adaptive strategy.

Reconstruction was less robust for nearly uniform or extremely sparse populations. The nearly uniform ellipse distribution was sensitive to small sampling fluctuations, whereas the very small number of detected triangles prevented a stable spatial estimate from being obtained. For triangles, the initially uniform spatial estimate closely matched the nearly uniform nominal reference distribution, yielding an OVL close to 100% even at very low sampling coverage. As additional measurements were incorporated, local fluctuations in the reconstructed map slightly reduced the overlap, without providing sufficient information to reliably resolve the sparse triangle population. Thus, the initially high OVL primarily reflects agreement between two nearly uniform distributions rather than successful spatial reconstruction. These results indicate that adaptive partial sampling is most effective when the target population is sufficiently represented and exhibits spatial structure above the level of sampling and detection noise.

From adaptive acquisition to an object-level morphological library

Importantly, the outcome of adaptive exploration is not limited to the acquired images or reconstructed spatial maps. Each detected structure is retained in an object-level morphological library together with its class identity, spatial location, and quantitative descriptors, allowing the sampled population to be analyzed and queried beyond the acquisition itself. Classical morphological descriptors provide a first description of this variability (Fig. 5a). In the present dataset, object area clearly distinguishes several of the fabricated geometries, reflecting their nominal size differences. In contrast, eccentricity shows substantial overlap between populations, while RMS roughness is concentrated over a relatively narrow range for most objects, with long tails associated with a smaller number of structures. Thus, although individual descriptors can discriminate particular properties or classes, no single scalar parameter provides a sufficiently rich description of the morphological variability present within each population.

To capture this variability more comprehensively, the detected objects were embedded using a triplet network trained on synthetic data, and the resulting latent representations were visualized using a two-dimensional UMAP projection (Fig. 5b). The four geometric classes occupy distinct regions of the embedding while retaining substantial internal structure, indicating that the learned representation preserves both class identity and within-class morphological variability. Objects located in the highest-density region of each class provide representative examples of the nominal ellipse, hexagon, square, and triangle morphologies. Importantly, the distributions in latent space are neither necessarily spherical nor uniformly populated, making a simple distance from the class centroid an incomplete measure of how unusual an individual object is.

We therefore used the mean distance to the five nearest neighbors in the learned feature space (kNN, $k=5$) as an exploratory measure of local morphological atypicality (Fig. 5b). This local metric does not assume a particular cluster geometry and provides a ranking of objects according to how sparsely represented their morphology is within the corresponding class. Figure 5c shows representative low-atypicality objects selected from the most densely populated region of each class, providing examples of the dominant morphologies and their normal within-class variability. In contrast, Fig. 5d shows the eight highest-ranked atypical objects from each sufficiently populated class, revealing a substantially broader range of morphological variability. These examples include apparent acquisition or processing artifacts, but also structures exhibiting shape distortions, localized protrusions, surface contamination, incomplete features, or deviations from the nominal geometry that are consistent with fabrication defects. This illustrates how the embedding-based ranking can help identify morphologically unusual structures that may warrant further inspection and targeted remeasurement.

Only eight triangular objects were detected in the region explored (Fig. 5e). Although these objects can be embedded and characterized, the population is too sparse to establish a

meaningful within-class atypicality ranking, and they are therefore shown separately rather than interpreted in terms of novelty.

Together, these results demonstrate how adaptive AFM acquisition can generate a spatially indexed, object-level morphological library that enables the retrieval, characterization, and ranking of representative and atypical structures based on learned morphological representations.

3. Discussion

SimuScan-Agent demonstrates how physics-informed synthetic data, machine learning and language-model orchestration can be integrated into a unified workflow for autonomous AFM exploration. By connecting model development to adaptive acquisition and morphological analysis, the framework moves beyond automated imaging towards experiments in which measurement locations are selected according to scientific objectives and accumulated observations. This approach reduced the number of measurements by approximately 71% relative to raster scanning in simulations and enabled experimental exploration of a $\sim 10 \text{ mm}^2$ region by sampling only 17.2% of available positions.

The significance of these results extends beyond acquisition efficiency. Although previous closed-loop scanning-probe approaches have demonstrated adaptive measurements and autonomous optimization³¹, and LLM-based agents have enabled AFM automation²⁶, many approaches remain focused on individual scan regions or specific experimental tasks. SimuScan-Agent extends decision-driven acquisition across larger spatial scales, opening opportunities to investigate dispersed populations, rare structures and long-range spatial heterogeneity while retaining nanoscale resolution at sampled locations, a combination that is often impractical with conventional AFM workflows. Moreover, the resulting object-level morphological library connects spatial exploration with quantitative characterization, allowing

representative and atypical structures to be identified beyond conventional morphological descriptors.

Beyond these spatial and morphological capabilities, a further contribution is making physics-informed synthetic-data generation and model deployment more accessible to researchers without specialized programming or machine-learning expertise. Combining diverse morphological sources with controlled AFM image formation facilitates the construction of annotated training datasets, while LLM orchestration connects previously separate computational and experimental operations. This modular architecture could also be transferred to other microscopy techniques, including optical and electron microscopy, where adaptive acquisition and agent-based automation are advancing^{32,33}.

Nevertheless, several challenges remain. Synthetic-data generation still requires iterative user evaluation, and correspondence between simulated variability and experimental morphology critically affects model performance. Excessive synthetic diversity may improve generalization but compromise sensitivity to specific experimental features, emphasizing the need for experimental validation. Spatial reconstruction also becomes less reliable for extremely sparse or nearly uniform populations, while morphological atypicality depends on the chosen representation, reference population and ranking criterion.

A particularly promising direction is closing the loop between spatial exploration and morphological learning, allowing newly identified atypical structures and underrepresented morphologies to influence subsequent acquisition decisions. Further developments could incorporate experimental feedback to refine synthetic datasets and provide agents with structured scientific knowledge and experimental histories to improve their autonomy. Ultimately, these advances could contribute to a new generation of more intelligent AFM systems capable not only of autonomously acquiring data, but also of adapting their experimental strategies to what they learn about the sample.

4. Methods

SimuScan-Agent software implementation and orchestration

SimuScan-Agent was implemented in Python 3.11, with dependencies managed using uv and pinned through a lockfile to ensure reproducibility. The computational environment included PyTorch 2.6.0 and torchvision 0.21.0 with CUDA 12.4 support, together with NumPy, SciPy, scikit-image, OpenCV, scikit-learn, trimesh, Matplotlib, and PyQt6. Model training and inference were performed on an NVIDIA RTX 2000 Ada GPU with 16 GB of memory.

Communication between the LLM and the individual workflow components was implemented using the Model Context Protocol (MCP)^{34,35}. A FastMCP 3.2 server exposed 83 typed tools covering simulator configuration, synthetic-data generation, model training and evaluation, inference, reference-image management, document retrieval, and microscope automation. The agent launched the MCP server as a subprocess and communicated with it through standard input/output using the official MCP Python SDK (v1.26). At session initialization, available tool schemas were retrieved from the server and converted to the function-calling format used by the OpenAI API (OpenAI Python SDK 2.29). Tool calls and their outputs were iteratively returned to the model until no additional action was requested.

Computationally intensive operations, including dataset generation, neural-network training, batch inference, and AFM exploration, were executed outside the MCP server process. For these operations, the MCP tools validate and normalize the requested parameters, whereas execution is performed locally by the agent process with access to the GPU and graphical interface. This separation keeps the model-facing interface declarative while preventing arbitrary code execution by the LLM. Simulator parameters are maintained in a structured “*SimuScanConfig*” object and serialized to YAML, allowing complete configurations to be inspected, modified through typed tools, saved, and subsequently reproduced.

The agent loop runs asynchronously and independently of the PyQt6 graphical-interface thread. Communication with the interface is handled through thread-safe Qt signals, enabling live visualization of generated data, model-training metrics, and experimental progress. Each session records the complete conversation history, tool calls and arguments, tool outputs, execution logs, generated figures, and saved configuration files, providing a traceable record of the workflow.

For retrieval-augmented generation, documentation and literature files are automatically indexed from a local document repository. PDF text is extracted using PyMuPDF, divided into overlapping text chunks, embedded using `text-embedding-3-small`, and stored in a persistent ChromaDB vector database using cosine similarity. The index is updated incrementally when new or modified documents are detected, and the five most similar text fragments are retrieved during document queries. Additional implementation details and the source code required to reproduce the workflow are provided in the associated code repository (see Code Availability).

Physics-informed synthetic AFM data generation

Training data for the experimental demonstration were generated using SimuScan 2.0, an updated implementation of the previously reported SimuScan framework [Ref. 25]. SimuScan explicitly defines object geometry, spatial distributions, and reference annotations before simulating AFM image formation, allowing simulator-derived masks and class labels to remain registered with the generated images without manual annotation. Controlled variability can be introduced in object size, orientation, position, and morphology, together with AFM-associated image-formation effects such as tip-sample convolution, substrate/background variations, noise, and scan artifacts.

SimuScan 2.0 extends the original framework by supporting multiple sources for defining object morphology. In addition to the original parameterized geometries constructed from line and segment definitions, objects can be imported from STL meshes or derived from

image-based references. The object library can also be expanded through SimuScan-Agent using PyGenerator, which translates natural-language morphological descriptions into parameterized object generators, or by using reference morphologies proposed by generative image models. Regardless of the source, the resulting object representation is incorporated into the same SimuScan image-formation pipeline, preserving explicit control over simulation parameters and simulator-derived annotations.

For the four-class nanostructure experiment, 4,761 synthetic AFM images containing 25,560 instances of ellipses, hexagons, squares, and triangles were generated and used for model training. The complete synthetic dataset is publicly available at the link provided in Code Availability section, together with the configuration file used for generation. This file records the parameter ranges and distributions defining object geometry, population variability, AFM tip parameters, background properties, and simulated acquisition artifacts, enabling the dataset-generation conditions to be reproduced.

Machine-learning models

The synthetic datasets generated with SimuScan 2.0 were used to train two complementary models: an instance-segmentation model for online object identification during AFM exploration and a metric-learning model for morphological representation. Simulator-derived annotations were converted from Common Objects in Context (COCO) format to the corresponding training formats without manual relabeling.

Instance segmentation was performed using Ultralytics YOLO v8.4.36 with the YOLOv8m-seg architecture, initialized from COCO-pretrained weights. The model was trained for 50 epochs using 512×512 pixel inputs and a batch size of 32. Data augmentation was restricted to transformations considered compatible with AFM imaging. Horizontal and vertical flips were applied with a probability of 0.5, together with a small intensity variation ($\text{hsv}_v = 0.1$) and label smoothing of 0.05. Geometric augmentations that could alter lateral dimensions

or object morphology, including rotation, translation, scaling, shear, perspective transformation, mosaic, and mixup, were disabled. During experimental exploration, model predictions were filtered using a configurable confidence threshold before detected objects were passed to the adaptive-exploration and morphological-analysis modules.

Morphological representations were obtained using a triplet-network model trained from the simulator-derived class labels. Individual objects were cropped using their segmentation masks with a 15% margin, aligned to a canonical orientation using singular-value decomposition of the mask coordinates, and resized to 64×64 pixels while preserving aspect ratio. The encoder consisted of four stride-2 convolutional blocks with 16, 32, 64, and 128 channels, respectively, followed by BatchNorm and ReLU activations. The resulting feature representation was passed through fully connected layers of dimensions 2048, 256, 64, and 32, and the final 32-dimensional embedding was L2-normalized.

The network was trained using TripletMarginLoss with a margin of 0.5 and Euclidean distance. Anchor and positive samples were selected from the same synthetic class, whereas negative samples belonged to a different class. Optimization used Adam with an initial learning rate of (10^{-3}) , weight decay of (10^{-5}) , cosine-annealing learning-rate scheduling, 40 epochs, and a batch size of 64.

For visualization, the resulting 32-dimensional embeddings were projected into two dimensions using UMAP (umap-learn 0.5.12; n_neighbors = 15, min_dist = 0.1, random_state = 42). The UMAP projection was used only for visualization; quantitative morphological atypicality was calculated directly in the original 32-dimensional embedding space.

Nanostructure test sample

The experimental test sample consisted of four classes of lithographically defined nanostructures -ellipses, squares, hexagons, and triangles- distributed over a nominal $3 \times 3 \text{ mm}^2$ area. The mask layout was designed using KLayout. A custom Python script was used to

programmatically generate and place the different object classes according to the prescribed spatial distributions and to export the resulting mask geometry. The script used to generate the layout is provided with the source code associated with this work (see Code Availability).

The nominal design contained 3,600 periodically spaced ellipses arranged on a 50 μm -pitch square lattice, together with 1,000 squares, 1,000 hexagons, and 100 triangles. All nanostructures had characteristic lateral dimensions of approximately 4 μm . Squares and hexagons were distributed according to predefined class-specific spatial probability maps composed of multiple two-dimensional Gaussian regions, including both shared high-probability areas and regions preferentially enriched in one class. Triangles were distributed randomly across the accessible area. The complete nominal layout and class-specific spatial distributions are shown in Supplementary Fig. S3.

The fabricated sample was produced using standard micro/nanofabrication procedures. A combination of electron beam lithography (EBL) and plasma etching was used to fabricate the micro/nanostructured surface. A 4-inch silicon wafer coated with 100 nm of thermally grown silicon dioxide (SiO_2) was cleaned with acetone and isopropyl alcohol (IPA) before being spin-coated with ma-N 2405 electron beam resist (Micro Resist Technology GmbH, Germany) at 3000 rpm for 45 s. The resist was then baked on a hotplate at 95 °C for 1 min.

EBL was performed using a JEOL JBX8100-FS system operated at an accelerating voltage of 100 kV. A beam current of 30 nA and an exposure dose of 350 $\mu\text{C}/\text{cm}^2$ were used to define the desired pattern. Following exposure, the substrate was developed in Microposit MF-319 developer (less than 5% tetramethylammonium hydroxide, TMAH) for 1 min, rinsed with deionized (DI) water, and dried with nitrogen.

The patterned resist served as an etch mask for the 100 nm SiO_2 layer. Etching was performed using an inductively coupled plasma reactive ion etching (ICP-RIE) system (Oxford Plasmalab 100) with a gas mixture of 45 sccm C_4F_8 and 2 sccm O_2 at 20 °C for 40 s. After

etching, the remaining electron beam resist was removed by soaking the sample in acetone, followed by an IPA rinse, leaving micro/nanostructured SiO₂ pillars on the silicon substrate.

Automated AFM acquisition

AFM measurements were performed using a DriveAFM system (Nanosurf), equipped with a motorized stage that enables automated imaging over areas larger than 100 mm². Measurements were acquired in tapping mode using Multi75-G probes (BudgetSensors), with an image resolution of 256×256 pixels over a scan area of 50×50 μm² and a scan rate of 2.0 Hz.

The microscope was interfaced with the SimuScan-Agent Exploration module through the Nanosurf Python API, allowing the software to move the motorized sample stage to selected coordinates, initiate AFM acquisition, retrieve the resulting image, and transfer it to the trained machine-learning models for analysis.

The approximately 10 mm² exploration area was discretized into a regular grid of candidate AFM scan positions, with a 55 μm center-to-center spacing corresponding to a 10% gap between adjacent 50×50 μm² images. Following each acquisition, the YOLO instance-segmentation model identified the structures present in the image using a confidence threshold of 0.7 and returned their predicted classes, object counts, segmentation masks, and spatial information. These observations were passed to the active-exploration algorithm to update the class-specific occupancy models and select the next measurement location. Detected objects and their associated metadata were simultaneously stored in the morphological library.

The experimental acquisition continued until either the user-defined target number of objects for all classes was reached or the experimental time limit of approximately 24 h was exceeded.

Probabilistic active exploration

Adaptive acquisition was implemented as a multiclass probabilistic search in which a separate spatial occupancy model was maintained for each object class. After each AFM image, detected and non-detected objects updated class-specific Beta-distribution occupancy estimates, with evidence propagated spatially using a Gaussian kernel. The next measurement position was selected from a utility function combining predicted class occupancy, uncertainty, the remaining class-specific sampling deficit, and stage-movement cost. An optional uncertainty-driven exploration term and a maximum displacement constraint were used to promote sampling of poorly characterized regions while limiting excessive stage motion.

The complete mathematical formulation, including the Beta pseudo-count updates, spatial propagation, acquisition utility, exploration rule, stopping criteria, parameter definitions, and simulation settings, is provided in Supplementary Section S2.

Spatial reconstruction and overlap analysis

Class-specific spatial distributions were reconstructed using Gaussian-process (GP) regression. For each scanned grid cell and object class, the observation value was defined as the mean, over all frames centered at that cell, of the summed YOLO detection confidences assigned to that class. Separate GP models were fitted for each class using the two-dimensional stage coordinates of the measured cells.

A constant-amplitude radial-basis-function kernel with an additive white-noise term was used,

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\ell^2}\right) + \sigma_n^2 \delta_{\mathbf{x}\mathbf{x}'},$$

where (σ_f^2) is the signal variance, (ℓ) the spatial correlation length, and (σ_n^2) the noise variance. Hyperparameters were optimized independently for each class and coverage level by maximizing the log marginal likelihood using the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm with two optimization restarts. The correlation length was

constrained to 20-3000 μm , and the target field was standardized to zero mean and unit variance before fitting.

Predictions were evaluated over a 58×58 candidate grid comprising 3,364 possible scan positions. Each position corresponded to a $50 \times 50 \mu\text{m}^2$ AFM frame, with a center-to-center spacing of 55 μm along both axes, leaving a 5 μm gap between adjacent frames (10% of the scan size). The resulting exploration grid extended over approximately $3.2 \times 3.2 \text{ mm}^2$, corresponding to an area of approximately 10 mm^2 .

Agreement between the reconstructed spatial distribution and the corresponding nominal design distribution was quantified using the overlapping coefficient (OVL)³⁶. Both distributions were normalized to unit sum, yielding (p_i) and (q_i) at grid position (i) , and the overlap was calculated as

$$\text{OVL} = 100 \sum_i \min(p_i, q_i).$$

An OVL of 100% indicates identical normalized spatial distributions. Because both maps are normalized before comparison, OVL measures spatial agreement independently of differences in absolute confidence-weighted signal magnitude. OVL was calculated at successive coverage levels to quantify the evolution of the reconstruction during exploration.

Morphological library and atypicality analysis

Each experimentally detected object was stored together with its AFM image crop, predicted class, sample coordinates, segmentation mask, and a comprehensive set of classical morphological descriptors. These included size/shape descriptors obtained from the segmentation mask (area, perimeter, equivalent diameter, major- and minor-axis length, aspect ratio, eccentricity, orientation, solidity, compactness and extent) and topographic descriptors from the AFM height data (mean height, maximum height and root-mean-square (RMS) surface roughness). Figure 5 displays a representative subset of these descriptors (area, eccentricity and RMS surface roughness); the complete set is retained in the library for every object. In parallel,

each object crop was passed through the trained triplet network to obtain a learned 32-dimensional morphological embedding.

For visualization, the high-dimensional embeddings were projected into two dimensions using Uniform Manifold Approximation and Projection (UMAP; `n_neighbors = 15`, `min_dist = 0.1`, Euclidean distance metric, random seed = 42). Representative objects were selected from densely populated regions of each class embedding.

Local morphological atypicality was quantified in the original 32-dimensional learned feature space rather than in the two-dimensional UMAP projection. For each object i , the Euclidean distances to the five nearest objects belonging to the same predicted class were calculated, and the atypicality score was defined as

$$A_i = \frac{1}{5} \sum_{j \in N_5(i)} d(\mathbf{z}_i, \mathbf{z}_j)$$

where $\mathbf{z}_i$ is the learned embedding of object i , $N_5(i)$ denotes its five nearest same-class neighbors, and d is Euclidean distance in embedding space. Higher values therefore identify objects located in locally sparse regions of the learned morphological representation. Objects were ranked independently within each sufficiently populated class. Because only eight triangles were detected experimentally, this population was considered too small for a meaningful within-class kNN atypicality ranking and was analyzed separately.

Code Availability

The complete SimuScan-Agent software package developed for this work, including the graphical user interface (GUI), orchestration framework, MCP tools, SimuScan 2.0 integration, machine-learning workflows, morphological analysis, Gaussian-process reconstruction, and adaptive AFM exploration, is available at <https://github.com/Rmillansol/SimuScan-Agent-GUI>. SimuScan-Agent is designed to operate with GPT models through the OpenAI API, with the language model acting as the orchestration layer for the available tools.

The synthetic dataset used in this study, together with the corresponding SimuScan configuration file specifying the parameter ranges and distributions used for data generation, is available on Zenodo at <https://zenodo.org/records/23482159>.

The trained machine-learning models used in this work are available separately on Zenodo at <https://zenodo.org/records/23481955>.

The custom script used to generate the nanostructure mask layout in KLayout is available in a separate repository at <https://github.com/Rmillansol/probabilistic-gds-mask.git>.

The scripts used to generate and analyze the prompt-based AFM-like images and candidate segmentation masks with the generative image model are available in a separate repository at https://github.com/Rmillansol/Generative-AFM-like_workflow.

The SimuScan-Agent repository also includes the software-environment specification and dependency lockfile used in this study, together with installation instructions and example configuration files. Direct microscope operation requires the Nanosurf Python API and compatible instrument hardware. User-specific API credentials, instrument-access credentials, proprietary vendor software, and third-party components distributed under separate licenses are not included in the repositories.

Video Availability

Three demonstration videos illustrating the SimuScan-Agent workflow, including synthetic AFM data generation and model training, morphological analysis, and autonomous AFM exploration, are publicly available on Zenodo at <https://zenodo.org/records/23481112>. The videos provide supplementary visual documentation of the framework and its experimental operation.

Acknowledgements

This work was supported by the U.S. Department of Energy, Office of Science FWP ERKCZ64, Structure Guided Design of Materials to Optimize the Abiotic-Biotic Material

Interface, as part of the Biopreparedness Research Virtual Environment (BRaVE) initiative. AFM measurements, sample preparation, and image analysis were conducted as part of a user project at the Center for Nanophase Materials Sciences (CNMS), which is a US Department of Energy, Office of Science User Facility at Oak Ridge National Laboratory.

Author contributions

R.M.-S., M.C., and L.C. conceived the study. R.M.-S. led the methodology and software development, experimental investigation, formal analysis, data curation, and visualization. M.C. and L.C. contributed to data visualization, validation and formal analysis. R.M.-S., M.C., Y.L., R.K.V., S.R. and L.C. contributed to the interpretation and scientific discussion of the results. R.M.-S., Y.L., and R.K.V. contributed to the agent architecture. B.S. contributed to the fabrication and preparation of the nanostructured test sample and provided experimental resources. L.C. supervised the study. R.M.-S. and L.C. wrote the original manuscript draft, and all authors reviewed and edited the manuscript. L.C. and S.R. acquired funding supporting the work.

Competing interests

The authors declare no competing interests.

Received: ((will be filled in by the editorial staff))

Revised: ((will be filled in by the editorial staff))

Published online: ((will be filled in by the editorial staff))

Tables and figures

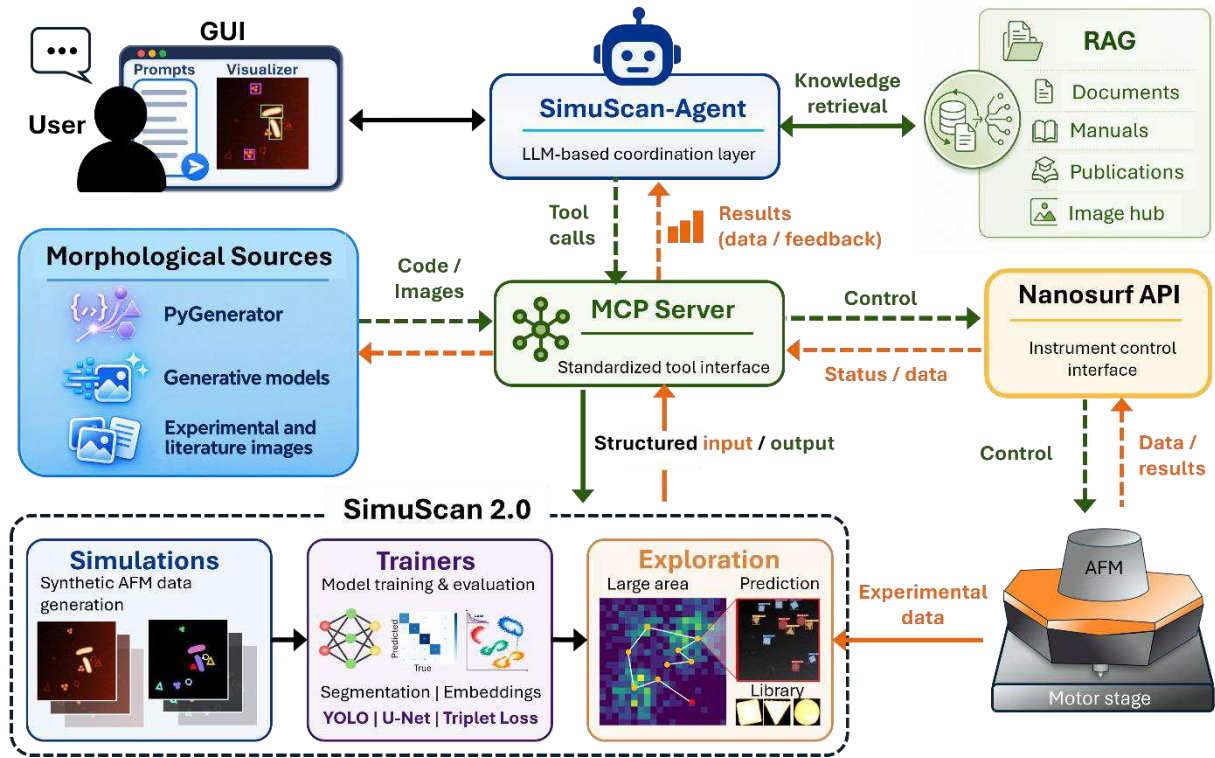
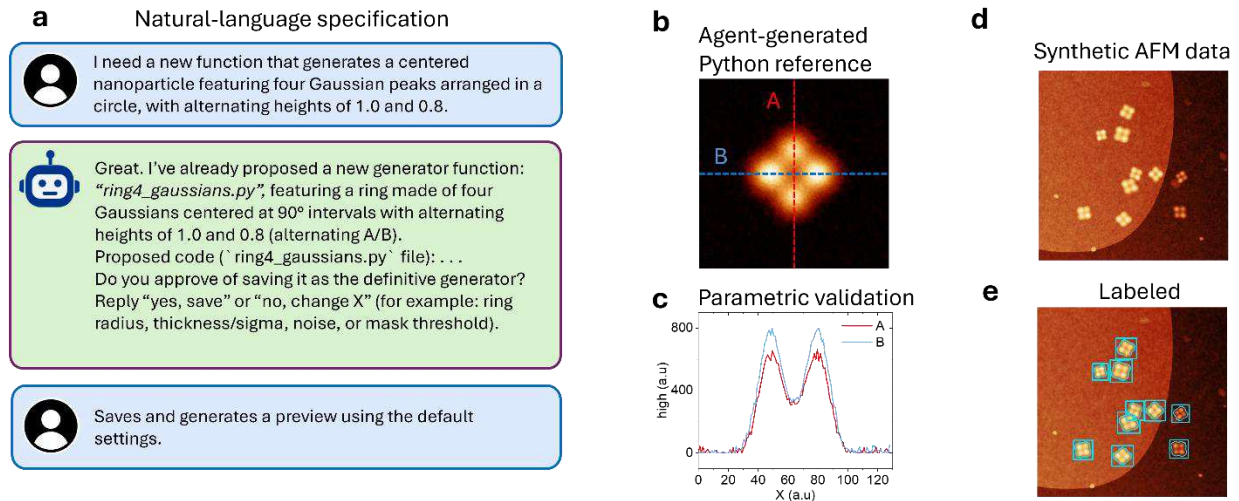
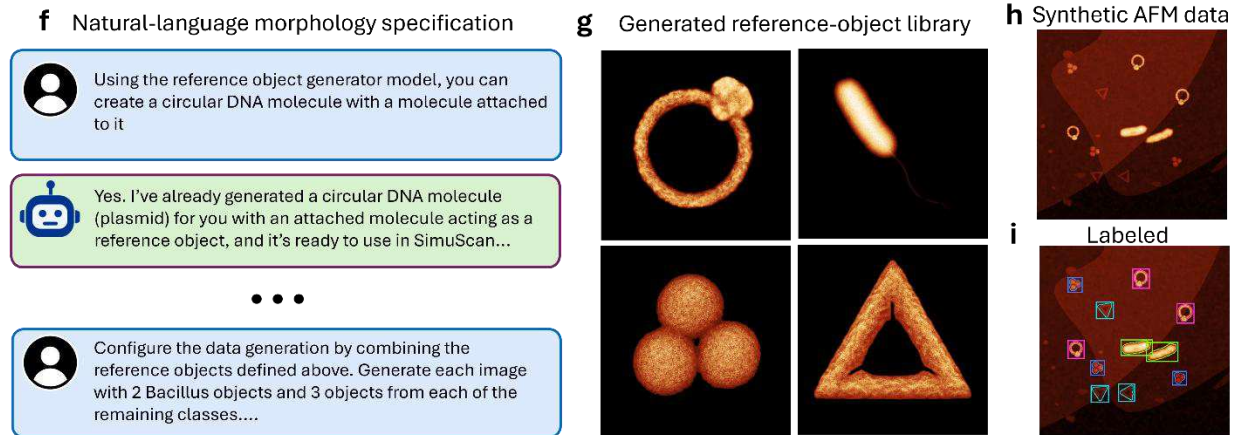


Figure 1 | Overall architecture of the SimuScan-Agent framework. The figure illustrates the overall architecture of the system. The user interacts with SimuScan-Agent through natural-language prompts. The agent, represented as an LLM-based coordination layer, communicates bidirectionally with an MCP server, which provides a standardized interface for invoking four functional modules: Multimodal Morphological Sources, which provides object geometries and reference images through procedural generation (PyGenerator), generative models, and experimental or literature-derived images; Simulator, which generates labeled synthetic AFM images and masks from the morphological inputs; Trainers, responsible for training and evaluating models for detection, segmentation, and extraction of latent representations (embeddings); and Exploration, which performs autonomous, adaptive AFM exploration. The synthetic images and corresponding annotations generated by the Simulator are supplied to the Trainers module, where models are trained and evaluated. The resulting trained models are then deployed in Exploration to analyze experimental AFM images. Through the instrument-control interface, Exploration initiates measurements, receives the acquired images and analysis results, and autonomously selects subsequent measurement regions based on accumulated observations and user-defined objectives. Arrows indicate the flow of instructions, morphological inputs, synthetic data, trained models, and experimental results among the components.

Programmatic reference-object generation



Generative models for reference-object creation



RAG-assisted reference-object creation from scientific literature

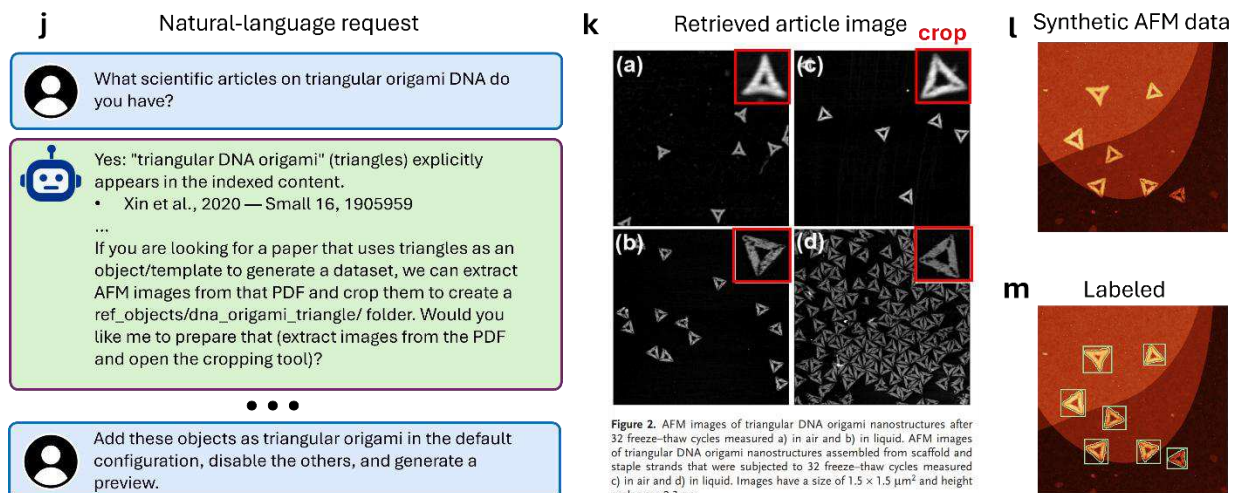
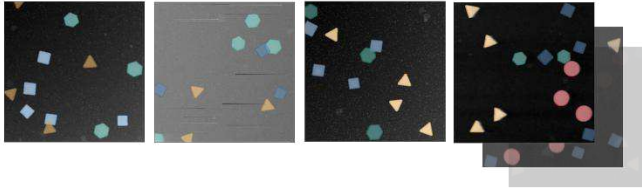


Figure 2. AFM images of triangular DNA origami nanostructures after 32 freeze-thaw cycles measured a) in air and b) in liquid. AFM images of triangular DNA origami nanostructures assembled from scaffold and staple strands that were subjected to 32 freeze-thaw cycles measured c) in air and d) in liquid. Images have a size of $1.5 \times 1.5 \mu\text{m}^2$ and height scales are 2.3 nm.

Figure 2 | Creation and incorporation of custom reference objects through natural language in SimuScan-Agent. Three complementary routes are shown through which the agent can expand the morphological library used by SimuScan and subsequently configure synthetic AFM data generation. (a-e) Programmatic generation of reference objects. (a) User-

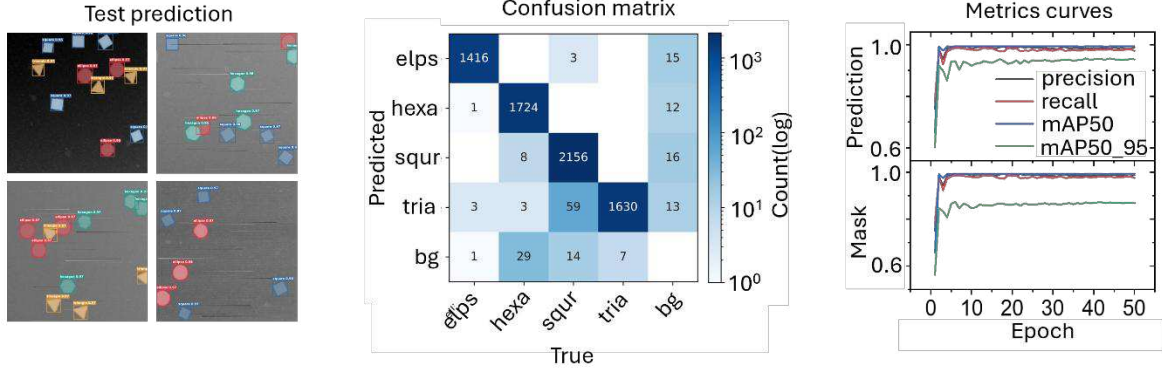
agent conversation in which the user requests a new geometry and the agent generates a parameterized Python function to describe it. **(b)** Preview of the resulting reference object, composed of four Gaussian peaks distributed circularly with alternating heights. **(c)** Cross-sectional profiles of the object along the directions indicated in (b), used to verify its geometry and parameterization before adding it to the library. **(d)** Example synthetic AFM image generated by SimuScan using the new reference object. **(e)** Corresponding simulator-derived annotations, generated automatically from the known geometry and available for training or evaluating machine-learning models. **(f-i)** Creation of reference objects using generative models. **(f)** Natural-language interaction used to request new morphologies and subsequently specify their combination and abundance in the dataset. **(g)** Representative library of objects generated from these instructions, including circular DNA with an associated structure, a bacillus with a flagellum, clusters of spherical nanoparticles, and a triangular DNA-origami structure. **(h)** Multiclass synthetic AFM image generated after incorporating these objects into the SimuScan workflow. **(i)** Simulator-derived labels associated with the objects present in the synthetic image, generated deterministically for use in training and validation. **(j-m)** Creation of reference objects from the scientific literature. **(j)** Natural-language query through which the agent uses retrieval-augmented generation (RAG) to locate articles containing the requested morphology. **(k)** AFM image retrieved from a scientific article³⁷ (Reproduced from Xin et al., Small 2020, 16, 1905959, under the terms of the Creative Commons Attribution 4.0 License) , with representative regions selected as morphological references. **(l)** Synthetic AFM image generated after incorporating into SimuScan the morphology derived from the example retrieved from the literature. **(m)** Corresponding ground-truth annotations for the generated synthetic objects. Across all three routes, external sources are used to define or parameterize object morphology before AFM image formation; geometry, labels, tip-sample interaction, substrate, noise, acquisition artifacts, and metadata remain under the explicit control of SimuScan.

a Synthetic labeled AFM frames



- Generation time ~ 2h
- 4 classes (ellipse, hexagon, square, triangle)
- 25560 instances in 4761 images
- Train = 3433
- Val = 904
- Test = 424

b YOLO training results



c Triplet training results

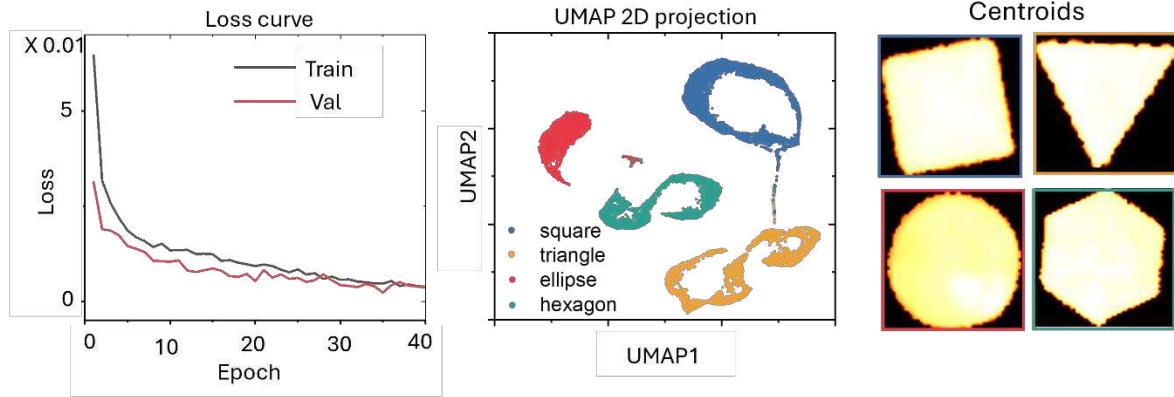


Figure 3 | Synthetic data generation and training of segmentation and morphological-representation models. (a) Representative examples of synthetic AFM images with annotations generated exactly from the known object geometry. The dataset contains four morphological classes (ellipses, hexagons, squares, and triangles) and was split approximately 70% for training, 20% for validation, and 10% for testing, with a total of 25,560 instances. (b) Training and evaluation of the YOLO instance-segmentation model. From left to right, representative predictions on the test set, the confusion matrix for the four classes and background, and the evolution of precision, *recall*, mAP50, and mAP50–95 during training are shown for both object detection (*Prediction*) and segmentation masks (*Mask*). (c) Learning of a morphological representation space using a model trained with triplet loss. From left to right, the evolution of training and validation losses, the two-dimensional UMAP projection of the learned *embeddings*, colored according to morphological class, and examples of the objects located closest to the centroid of each class in this representation space are shown.

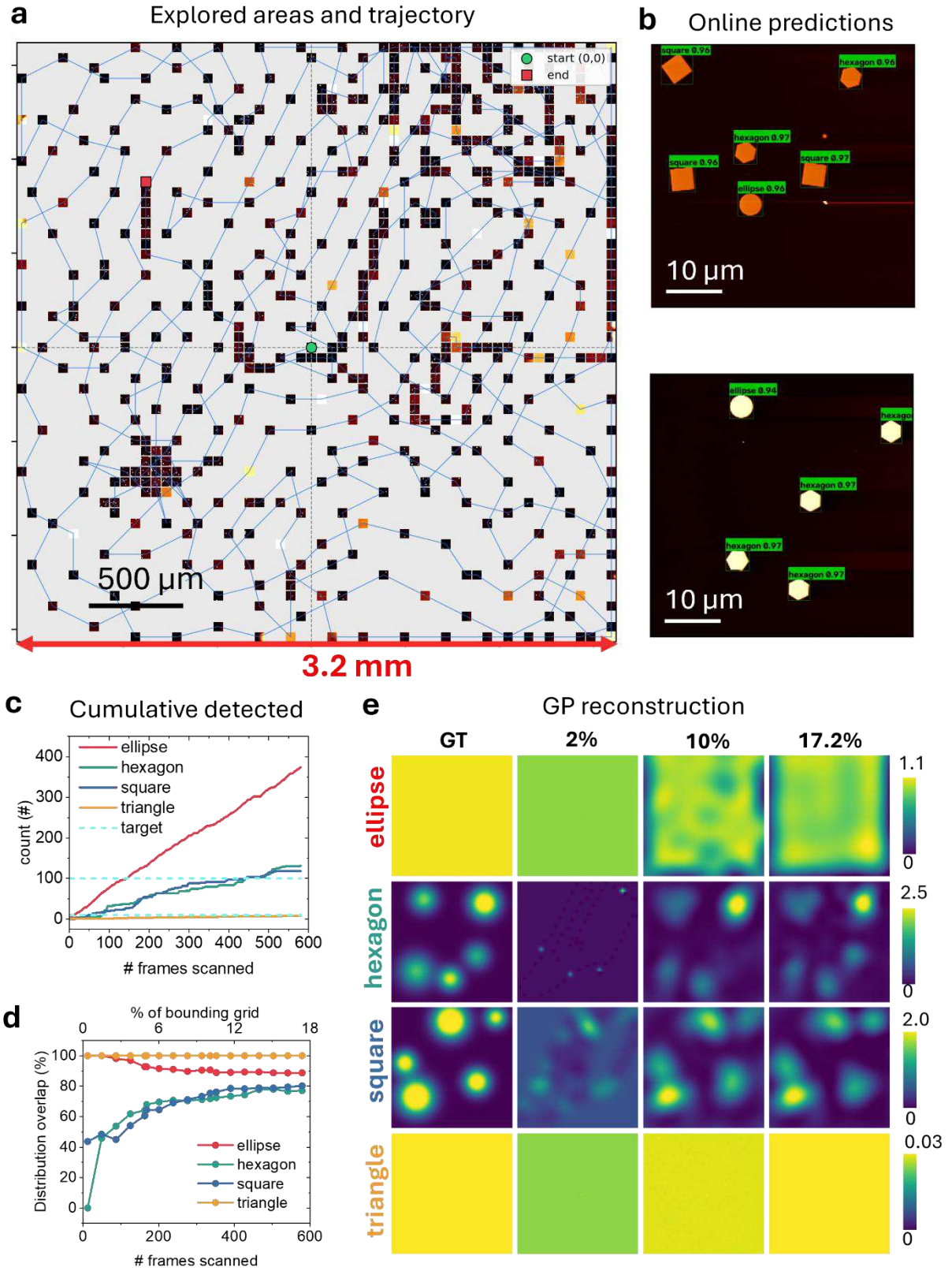
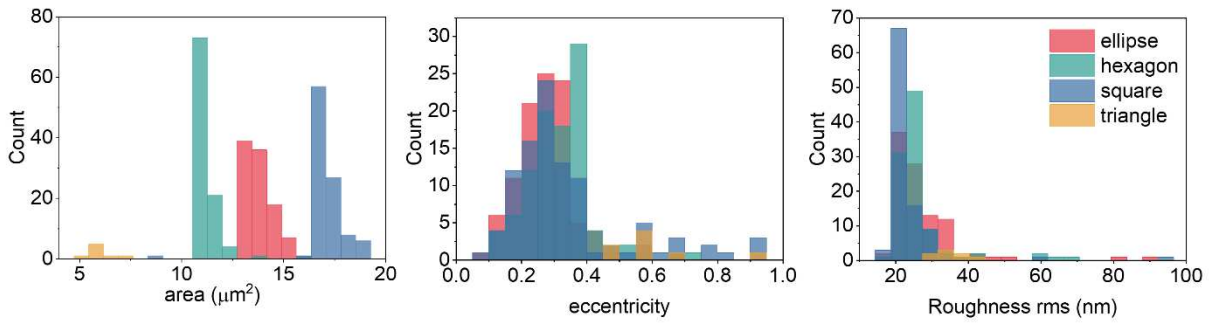


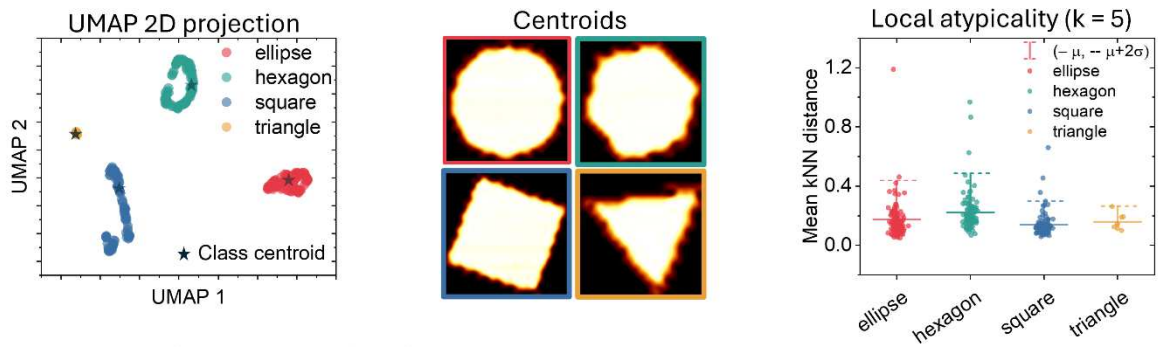
Figure 4 | Automated multiclass AFM exploration and reconstruction of nanostructure spatial distributions. (a) Explored positions and trajectory followed during automated

acquisition over an area of approximately $3.2 \times 3.2 \text{ mm}^2$. The initial and final exploration positions are indicated. **(b)** Examples of in-situ predictions made by the segmentation model during AFM acquisition. **(c)** Cumulative number of ellipses, hexagons, squares, and triangles detected as a function of the number of acquired images. Dashed lines indicate the targets defined for the different classes. **(d)** Evolution of the overlap between the reconstructed maps and the nominal design reference as a function of the number of acquired images and the fraction of candidate scan positions sampled. For triangles ($n = 8$), the initially uniform spatial estimate yields a near-100% OVL against the nearly uniform nominal reference, even at very low coverage. Subsequent fluctuations slightly reduce the overlap; therefore, the high OVL should not be interpreted as evidence of reliable spatial reconstruction. **(e)** Nominal design reference maps and spatial distributions reconstructed using Gaussian processes after sampling 2% (67 images), 10% (336 images), and 17.2% (579 images) of the candidate scan positions.

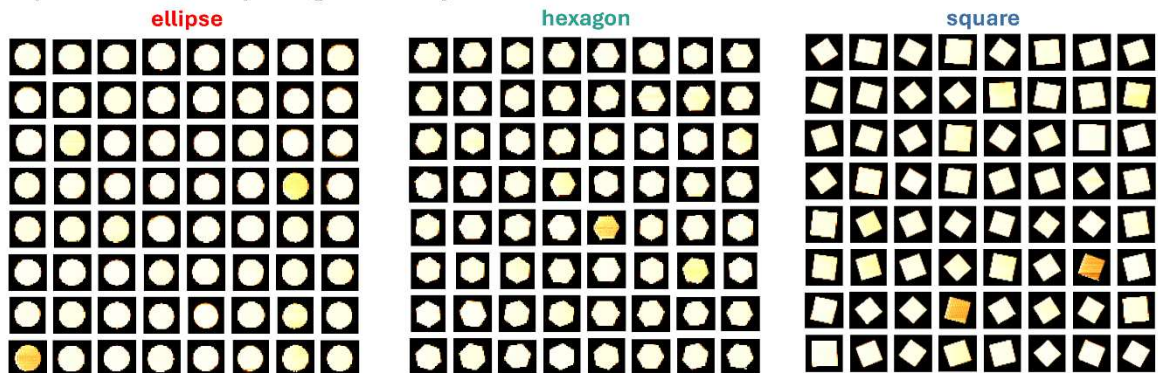
a Classical morphological descriptors



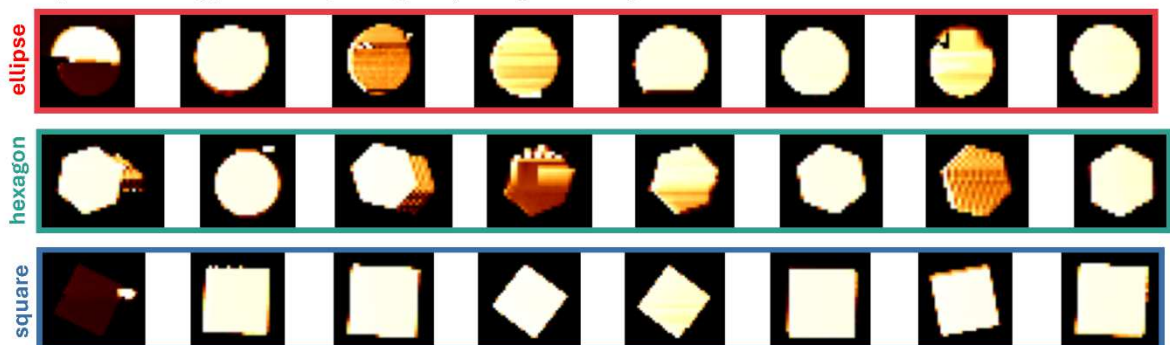
b Learned morphological representation and atypicality



c Representative morphological library



d Top-ranked atypical morphologies (n = 8 per class)



e Triangle population



Figure 5 | Characterization of morphological variability and ranking of morphologically atypical objects in the experimental library. **(a)** Distributions of area, eccentricity, and RMS roughness for the detected objects, separated by class. **(b)** Learned morphological representation of the detected structures. Left, two-dimensional UMAP projection of the latent embeddings, showing class-dependent and internally structured regions of feature space; stars indicate the class centroids. Center, objects closest to the region of highest-density of each class, shown as representative examples of the dominant ellipse, hexagon, square, and triangle morphologies. Right, distribution of the mean distance to the five nearest neighbors in latent space (kNN, $k = 5$), used here as an exploratory measure of local morphological atypicality. **(c)** Representative morphological libraries containing 64 low-atypicality objects for each sufficiently populated class (ellipse, hexagon, and square), illustrating the morphological variability retained within the nominal populations. **(d)** Eight highest-ranked atypical objects from each of these classes. The selected examples include both apparent imaging or processing artifacts, such as scan-line and intensity discontinuities, and morphological deviations that may correspond to physical variations or fabrication defects, including localized protrusions and distorted or incomplete geometries. **(e)** Complete triangle population detected during the experiment ($n = 8$). Because of the limited population size, no within-class novelty ranking was interpreted for this class.

Bibliography

- 1 Binnig, G., Quate, C. F. & Gerber, C. Atomic force microscope. *Physical review letters* **56**, 930 (1986).
- 2 Loos, J. The art of SPM: Scanning probe microscopy in materials science. *Advanced Materials* **17**, 1821-1833 (2005).
- 3 Losic, D., Pillar, R. J., Dilger, T., Mitchell, J. G. & Voelcker, N. H. Atomic force microscopy (AFM) characterisation of the porous silica nanostructure of two centric diatoms. *Journal of Porous Materials* **14**, 61-69 (2007).
- 4 Muzio, M. D., Millan-Solsona, R., Borrell, J. H., Fumagalli, L. & Gomila, G. Cholesterol effect on the specific capacitance of submicrometric DOPC bilayer patches measured by in-liquid scanning dielectric microscopy. *Langmuir* **36**, 12963-12972 (2020).
- 5 Di Muzio, M. *et al.* Dielectric properties and lamellarity of single liposomes measured by in-liquid scanning dielectric microscopy. *Journal of nanobiotechnology* **19**, 167 (2021).
- 6 Millan-Solsona, R., Checa, M., Fumagalli, L. & Gomila, G. Mapping the capacitance of self-assembled monolayers at metal/electrolyte interfaces at the nanoscale by in-liquid scanning dielectric microscopy. *Nanoscale* **12**, 20658-20668 (2020).
- 7 Garcia, R. Nanomechanical mapping of soft materials with the atomic force microscope: methods, theory and applications. *Chemical Society Reviews* **49**, 5850-5884 (2020).
- 8 Dols-Perez, A., Gramse, G., Calò, A., Gomila, G. & Fumagalli, L. Nanoscale electric polarizability of ultrathin bilayers on insulating substrates by electrostatic force microscopy. *Nanoscale* **7**, 18327-18336 (2015).
- 9 Zahmatkeshsaredorahi, A., Millan-Solsona, R., Jakob, D. S., Collins, L. & Xu, X. G. Kelvin probe force microscopy under ambient conditions. *Nature Reviews Methods Primers* **5**, 53 (2025).
- 10 Grütter, P., Mamin, H. & Rugar, D. in *Scanning Tunneling Microscopy II: Further applications and related scanning techniques* 151-207 (Springer, 1992).
- 11 Collins, L. *et al.* Kelvin probe force microscopy in liquid using electrochemical force microscopy. *Beilstein journal of nanotechnology* **6**, 201-214 (2015).
- 12 Liu, Y. *et al.* AEcroscopyWave: Towards Self-Driving Characterization Platforms for Agentic AI. *arXiv preprint arXiv:2607.22975* (2026).
- 13 Narasimha, G., Bulanadi, R., Chowdhury, J., Vasudevan, R. & Liu, Y. Self-Driving Scanning Probe Microscopy: From Acceleration to Discovery and Manipulation. *Accounts of Chemical Research* (2026).
- 14 Kalinin, S. V. *et al.* Automated and autonomous experiments in electron and scanning probe microscopy. *ACS nano* **15**, 12604-12627 (2021).
- 15 Sotres, J., Boyd, H. & Gonzalez-Martinez, J. F. Enabling autonomous scanning probe microscopy imaging of single molecules with deep learning. *Nanoscale* **13**, 9193-9203 (2021).
- 16 Casuso, I. & Scheuring, S. Automated setpoint adjustment for biological contact mode atomic force microscopy imaging. *Nanotechnology* **21**, 035104 (2010).
- 17 Sadeghian, H. *et al.* Automated cantilever exchange and optical alignment for high-throughput parallel atomic force microscopy. *IEEE/ASME Transactions on Mechatronics* **22**, 2654-2661 (2017).
- 18 Yoo, R. Y. The story behind the first automatic atomic force microscope. *Microscopy Today* **30**, 40-45 (2022).

- 19 Liu, H. *et al.* Intelligent tuning method of PID parameters based on iterative learning control for atomic force microscopy. *Micron* **104**, 26-36 (2018).
- 20 Abramovitch, D. Y., Hoen, S. & Workman, R. Semi-automatic tuning of PID gains for atomic force microscopes. *Asian Journal of Control* **11**, 188-195 (2009).
- 21 Chowdhury, J. *et al.* Quality-controlled active learning via Gaussian processes for robust structure–property learning in autonomous microscopy. *npj Computational Materials* (2026).
- 22 Liu, Y., Checa, M. & Vasudevan, R. K. Synergizing human expertise and AI efficiency with language model for microscopy operation and automated experiment design. *Machine Learning: Science and Technology* **5**, 02LT01 (2024).
- 23 Saharia, C. *et al.* Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems* **35**, 36479-36494 (2022).
- 24 Eschweiler, D. *et al.* Denoising diffusion probabilistic models for generation of realistic fully-annotated microscopy image datasets. *PLOS Computational Biology* **20**, e1011890 (2024).
- 25 Millan-Solsona, R. *et al.* Synthetic data-driven deep learning for label-free autonomous atomic force microscopy. *Nature Communications* **17**, 3886 (2026).
- 26 Mandal, I. *et al.* Evaluating large language model agents for automation of atomic force microscopy. *Nature Communications* **16**, 9104 (2025).
- 27 Ayar, Z. *et al.* Agentic AI for operating scientific instruments for nanoscale characterization. *arXiv preprint arXiv:2608.26198* (2026).
- 28 Lewis, P. *et al.* Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* **33**, 9459-9474 (2020).
- 29 Hoffer, E. & Ailon, N. in *International workshop on similarity-based pattern recognition*. 84-92 (Springer).
- 30 Rasmussen, C. E. & Williams, C. Gaussian processes for machine learning the mit press. *Cambridge, MA* **32**, 68 (2006).
- 31 Ziatdinov, M., Liu, Y., Kelley, K., Vasudevan, R. & Kalinin, S. V. Bayesian active learning for scanning probe microscopy: From Gaussian processes to hypothesis learning. *ACS nano* **16**, 13492-13512 (2022).
- 32 Shi, Y. *et al.* Smart lattice light-sheet microscopy for imaging rare and complex cellular events. *Nature Methods* **21**, 301-310 (2024).
- 33 Wall, M. K., Pattison, A. J., Barnard, E. S., Ribet, S. M. & Ercius, P. TEM Agent: enhancing transmission electron microscopy with modern AI tools. *npj Computational Materials* (2026).
- 34 Hasan, M. M. *et al.* Model context protocol (mcp) at first glance: Studying the security and maintainability of mcp servers. *ACM Transactions on Software Engineering and Methodology* (2025).
- 35 Wang, Z. *et al.* in *International Conference on Learning Representations*. 97377-97425.
- 36 Inman, H. F. & Bradley Jr, E. L. The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. *Communications in Statistics-theory and Methods* **18**, 3851-3874 (1989).
- 37 Xin, Y. *et al.* Cryopreservation of DNA origami nanostructures. *Small* **16**, 1905959 (2020).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SimuScanAgentSINMI.pdf](#)