A Technical Appendices and Supplementary Material

A.1 Related Works

Our work is situated at the intersection of foundation models for scientific discovery [18, 19] and domain adaptation techniques. While many studies explore either parameter-efficient fine-tuning (PEFT) techniques like LoRA [9] or in-context learning (ICL) with retrieval augmentation [20], a systematic comparison of these strategies in specialized, low-data scientific domains has been lacking.

A.2 The PtychoBench Dataset

This section provides a detailed description of the PtychoBench benchmark, which was curated to facilitate a reproducible and quantitative study of AI agents for ptychographic analysis.

A.2.1 Data Curation and Partitioning

The dataset was constructed from an initial pool of 394 ptychographic reconstructions of experimental data acquired at the Advanced Photon Source, each annotated by a domain expert. After cleaning, the primary dataset consists of 391 samples. This full dataset was first partitioned into a training set of 312 samples and a test set of 79 samples. This primary 80/20 partition serves as the basis for the Artifact Detection task. Subsequently, the dataset for the Parameter Recommendation task was derived by creating subsets from these existing partitions. We filtered both the training and test sets to include only those instances which contained an expert's free-text recommendation label (caption_param). This resulted in a final training set of 91 samples and a test set of 44 samples for the recommendation task. See Table 3 for sample data field.

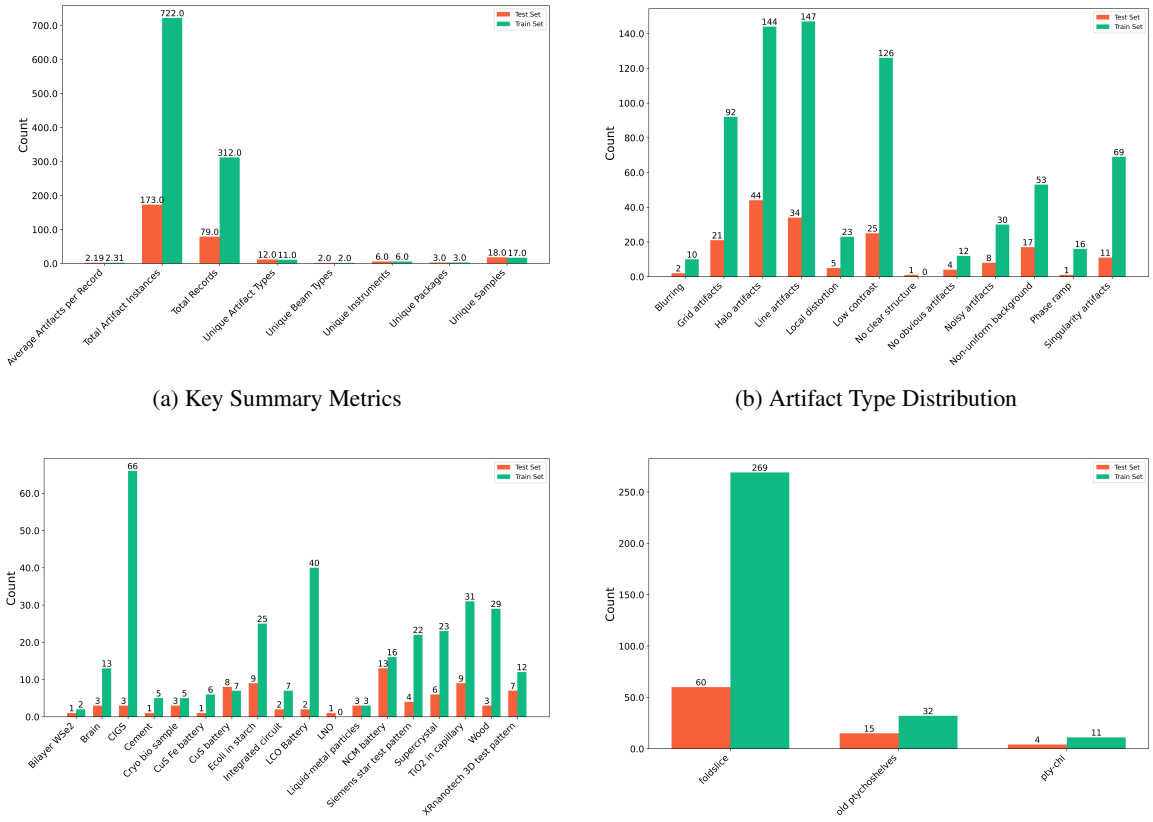
Table 3: **Dataset Field Descriptions and Examples**: The table presents the structure and annotation schema of the PtychoBench dataset, detailing each field's purpose and usage in the VLM and LLM training and evaluation tasks.

Field Name	Data Type	Description	Example Value	Usage
`id`	Integer	Unique sample identifier	2	Both tasks
`captioning`	String	Path to ptychographic reconstruction image	`/data/upload/1/af707e7b-...png`	VLM Training
`beam_choice`	Categorical	Type of radiation beam used	X-ray	LLM Training
`instrument_ choice`	Categorical	Specific beamline/instrument identifier	APS-2IDE-XFM	LLM Training
`sample_text`	Categorical	Material/sample type being imaged	Integrated circuit	SSFS retrieval, LLM Training
`package_ choice`	Categorical	Reconstruction software package	pty-chi	LLM Training
`artifact_ choice`	List	Expert-annotated visual artifacts present	[*"Local distortion"*]	**VLM Target**
`caption_obj`	String	Expert description of visual artifacts/quality	*"Lines look zig-zag with discontinuities..."*	LLM Training
`caption_ param`	String	Expert parameter recommendations	*"Increase iterations or use smaller patterns..."*	**LLM Target**
`recon_path_ text`	String	Full file path to reconstruction data	`/mnt/micdata1/2ide/2025-1/...`	Metadata
`annotator`	Integer	Expert annotator identifier	1	Quality control
`annotation_ id`	Integer	Annotation session identifier	3	Quality control
`created_at`	Timestamp	Initial annotation timestamp	2025-05-14T20:10:35Z	Metadata
`updated_at`	Timestamp	Last modification timestamp	2025-06-02T19:22:01Z	Metadata
`lead_time`	Float	Annotation time in seconds	180.11	Quality metrics

A.2.2 Task Definitions

The PtychoBench benchmark is designed for two sequential tasks central to autonomous characterization:

- **Artifact Detection**: A multi-modal, multi-label classification task where a Vision-Language Model (VLM) takes a ptychographic image as input and identifies a set of visual artifacts or defects.
- **Parameter Recommendation**: A text-only, multi-label classification task where a Language Model (LLM) takes a textual description of observed artifacts and experimental conditions as input and recommends corrective actions.



(a) Key Summary Metrics

(b) Artifact Type Distribution

(c) Sample Type Distribution

(d) Software Package Distribution

Figure 2: Statistical distributions of the PtychoBench dataset across the training and test sets. The summary plot (a) is followed by detailed breakdowns for artifact types (b), sample types (c), and software packages (d).

A.2.3 Dataset Composition and Characteristics

The PtychoBench is characterized by its diversity and complexity, reflecting real-world experimental conditions. The full dataset spans 18 distinct sample types and 6 unique instruments. The Artifact Detection task presents a challenging multi-label problem. The 79 samples in the test set contain a total of 173 annotated artifact instances, averaging 2.19 artifacts per image. Across the entire dataset, the most prevalent of the 12 possible artifact types are Halo artifacts and Line artifacts. The Parameter Recommendation task is similarly complex, drawing from a rich context of metadata and observed issues. A complete statistical breakdown of the dataset composition, including the distributions for all categorical metadata as provided, is detailed in Figure 1 and Table 3.

Table 4: **Task-Specific Data Usage**. The table details how different fields from the PtychoBench dataset are utilized in the VLM artifact detection task and LLM parameter recommendation task.

Task	Input Features	Target Labels	Context Selection	Sample Size
VLM Artifact Detection	captioning (image) + caption_obj (description) [sample_text, instrument_choice, beam_choice]	**artifact_choice** (multi-label)	sample_text matching	**391**
LLM Parameter Recommendation	prompt + context [beam_choice, instrument_choice, sample_text, package_choice, caption_obj (artifact description)]	**caption_param** (text gen)	sample_text matching	**135**

A.3 Task Formulation and Prompt Engineering

The prompts for both tasks were structured to emulate a real-world interaction with an expert AI agent, providing clear context and instructions. The data was formatted into a conversational structure with USER and ASSISTANT roles.

A.3.1 Task 1: LLM-based Parameter Recommendation

This task is a text-only, multi-label classification problem simulating the decision-making step that follows artifact diagnosis.

Prompt Template: The prompt provides a rich textual context constructed by populating the template below with a sample's specific metadata.

Parameter Recommendation Prompt

SAMPLE INFORMATION:

Package Choice: package
Instrument Used: instrument
Sample Type: sample
Beam Type: beam
Expert Description: caption_obj
Observed Artifacts: artifacts
Current Reconstruction Parameters:
param_info

Based on this sample information, what specific parameter adjustments are needed to improve image quality?

POSSIBLE PARAMETER UPDATES:
A) Change number of batches to 1
B) Disable momentum acceleration
C) Disable multimodal update
D) Disable position correction
E) Enable momentum acceleration
F) Enable multimodal update
G) Enable position correction
H) Increase batch size
I) Increase number of OPR modes
J) Increase number of probe modes
K) Increase the number of iterations
L) No changes needed
M) Recenter diffraction patterns
N) Reduce batch size
O) Reduce diffraction pattern size by factor of 2
P) Reduce or disable regularization
Q) Try other diffraction pattern orientations
R) Turn off affine constraint
S) Turn off variable probe correction (set OPR modes to 0)
T) Turn on variable probe correction (set OPR modes to 1)
U) Use Gaussian noise model
V) Use compact batch selection scheme
W) Use multislice model
X) Use sparse batch selection scheme

Your task:
1. Select all correct options (A, B, C, ...) and list them as a comma-separated list inside <answer></answer> tags (e.g., <answer>E, J, V</answer>).
2. The <answer></answer> tags must contain ONLY the letters. Do not include any explanations or other text.

Ground-Truth Generation: The ground-truth labels for the assistant response were generated via a complex procedure. The expert's original free-text advice (the caption_param field) was programmatically mapped to the corresponding alphabetic choices (A-X) using a curated set of regular expression patterns. This allowed us to convert natural language instructions (e.g., "Increase the number of iterations to 2000") into a standardized, multi-label format (e.g., <answer>K</answer>).

A.3.2 Task 2: VLM-based Artifact Detection

This task is formulated as a multi-modal, multi-label classification problem where a VLM identifies visual artifacts in a ptychographic image.

Prompt Template: The user prompt consists of an image paired with the following text instruction:

Artifact Detection Prompt

You are an expert in X-ray ptychography image analysis. What artifacts are visible?

Possible artifacts include: Grid artifacts, Halo artifacts, Line artifacts, Local distortion, Low contrast, No clear structure, Noisy artifacts, Non-uniform background, Phase ramp, Singularity artifacts, Blurring.

Format your response as: <answer>[Comma-separated list of artifacts, or "No obvious artifacts" if the image quality is good]</answer> Provide your answers now:

Ground-Truth Generation: The assistant response in the training data contains the expert-annotated list of artifacts (from the artifact_choice field), formatted within <answer> tags as requested by the prompt.

A.4 Model and Training Details

This section provides the specific identifiers, libraries, and hyperparameters used for all experiments.

A.4.1 Models

All open-weight models were used in their full-precision versions. The specific Hugging Face identifiers used are:

- **VLMs:** meta-llama/Llama-3.2-11B-vision and meta-llama/Llama-3.2-90B-vision.
- **LLMs:** meta-llama/Meta-Llama-3.1-8B-Instruct and meta-llama/Meta-Llama-3.1-70B-Instruct.
- **Library**: Both the LLM and VLM training processes utilized the Unsloth library [23].
- **Proprietary Baseline:** OpenAI's `gpt-4o` model was accessed via their API.
- **Vision Baseline:** The Meta AI `DINOv3` model was used as a fixed feature extractor for the traditional computer vision baseline.

Table 5: **SFT Hyperparameters for Vision-Language Models (VLMs)**. Training configuration used for supervised fine-tuning of Llama 3.2-Vision models on the artifact detection task.

Hyperparameter	Value
SFT Epoch (11B model)	50
SFT Epoch (90B model)	20
`learning_rate`	2e-4
`lora_r`	16
`lora_alpha`	16
`lora_dropout`	0
`bias`	"none"
`optim`	"adamw_8bit"
`weight_decay`	0.01
`max_seq_length`	2048
`per_device_train_batch_size`	1
`gradient_accumulation_steps`	8

Table 6: **SFT Hyperparameters for Language Models (LLMs)**. Training configuration used for supervised fine-tuning of Llama 3.1 models on the parameter recommendation task.

Hyperparameter	Value
num_epochs	50
learning_rate	2e-4
lora_r	16
lora_alpha	16
target_modules	["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj"]
lora_dropout	0
bias	"none"
optim	"adamw_8bit"
weight_decay	0.01
max_seq_length	2048
per_device_train_batch_size	4
gradient_accumulation_steps	8

A.4.2 Supervised Fine-Tuning (SFT) Protocol

We employed parameter-efficient fine-tuning (PEFT) using Low-Rank Adaptation (LoRA). The loss function for our multi-label tasks was Binary Cross-Entropy with Logits Loss. The detailed hyperparameters are provided in Table 5 and Table 6.

A.4.3 In-Context Learning (ICL) Protocol

For the Sample-Specific Few-Shot (SSFS) strategy, examples were selected from the training set based on an exact match of the sample type metadata field with the test sample. If fewer than k examples with a matching sample type were available, the remaining slots in the prompt were filled by randomly sampling from the rest of the training set, ensuring the test sample itself was excluded.

A.5 Results

A.5.1 Comprehensive Results with Statistical Validation

The main paper presents the mean and standard deviation for our experimental results in Tables 1 and 2. To provide a complete picture of statistical significance and address the variability from a limited test set, this section provides the full 95% confidence intervals (CIs) for all VLM and LLM experiments. These CIs were computed via bootstrapping with 10,000 resamples. The tables allow for a direct comparison of the performance ranges between different models and strategies, directly supporting the statistical conclusions drawn in the main text.

Table 7: VLM Performance on Artifact Detection (95% Confidence Interval)

Model	Fewshot Strategy	0-shot[2]	1-shot	3-shot	5-shot	7-shot
SFT Llama 3.2-Vision 11B	RFS	[0.412 - 0.573]	[0.251 - 0.392] ↓	[0.239 - 0.399]	[0.308 - 0.468]	[0.315 - 0.475]
	SSFS		**[0.458 - 0.631]**	**[0.547 - 0.720]**	**[0.581 - 0.748]**	**[0.636 - 0.784]**
Base Llama 3.2-Vision 11B	RFS	[0.168 - 0.266]	[0.188 - 0.320]	[0.183 - 0.321]	[0.248 - 0.377]	[0.270 - 0.412]
	SSFS		**[0.512 - 0.696]**	**[0.535 - 0.715]**	**[0.610 - 0.773]**	**[0.615 - 0.769]**
SFT Llama 3.2-Vision 90B	RFS	[0.348 - 0.509]	[0.264 - 0.403] ↓	[0.337 - 0.479]	[0.326 - 0.473]	[0.377 - 0.516]
	SSFS		**[0.500 - 0.668]**	**[0.504 - 0.683]**	**[0.589 - 0.749]**	**[0.655 - 0.796]**
Base Llama 3.2-Vision 90B	RFS	[0.000 - 0.050]	[0.189 - 0.322]	[0.109 - 0.248]	[0.040 - 0.147]	[0.083 - 0.229]
	SSFS		**[0.514 - 0.700]**	**[0.525 - 0.723]**	**[0.525 - 0.713]**	**[0.626 - 0.779]**
GPT-4o	RFS	[0.126 - 0.226]	[0.117 - 0.209]	[0.137 - 0.242]	[0.189 - 0.308]	[0.206 - 0.357]
	SSFS		**[0.262 - 0.410]**	**[0.441 - 0.614]**	**[0.546 - 0.720]**	**[0.571 - 0.739]**

Table 8: LLM Performance on Parameter Recommendation (95% Confidence Intervals)

Model	Fewshot Strategy	0-shot	1-shot	3-shot	5-shot	7-shot
SFT Llama 3.1 8B	RFS	[0.370 - 0.572]	[0.247 - 0.389] ↓	[0.328 - 0.502]	[0.394 - 0.598]	[0.477 - 0.630]
	SSFS		**[0.225 - 0.417]**	**[0.479 - 0.722]**	**[0.549 - 0.770]**	**[0.587 - 0.797]**
Base Llama 3.1 8B	RFS	[0.051 - 0.136]	[0.114 - 0.258]	[0.205 - 0.337]	[0.359 - 0.546]	[0.297 - 0.481]
	SSFS		**[0.337 - 0.483]**	**[0.491 - 0.715]**	**[0.533 - 0.792]**	**[0.593 - 0.814]**
SFT Llama 3.1 70B	RFS	[0.734 - 0.928]	[0.364 - 0.545] ↓	[0.411 - 0.573]	[0.427 - 0.621]	[0.496 - 0.689]
	SSFS		**[0.437 - 0.599]**	**[0.508 - 0.703]**	**[0.593 - 0.764]**	**[0.597 - 0.781]**
Base Llama 3.1 70B	RFS	[0.189 - 0.266]	[0.193 - 0.343]	[0.273 - 0.452]	[0.436 - 0.652]	[0.463 - 0.684]
	SSFS		**[0.446 - 0.634]**	**[0.664 - 0.820]**	**[0.697 - 0.861]**	**[0.755 - 0.923]**
GPT-4o	RFS	[0.255 - 0.373]	[0.269 - 0.418]	[0.240 - 0.415]	[0.423 - 0.613]	[0.456 - 0.650]
	SSFS		**[0.521 - 0.709]**	**[0.614 - 0.796]**	**[0.671 - 0.851]**	**[0.701 - 0.854]**

[2]0-shot CIs from SSFS and RFS bootstrap runs differed negligibly; for simplicity, a single representative interval is reported.

A.5.2 Per-Class F1-Score for Artifact Detection (SFT-90B + SSFS)

To provide a more granular analysis of model performance, especially given the inherent class imbalance of the dataset, we provide a per-class F1-score analysis for our best-performing visual model (SFT-90B with SSFS) in Table 9. The results show that the model's performance generally improves with more in-context examples (k) and is strongest on the most well-represented classes (e.g., "Halo artifacts," "Low contrast"). The model struggles with extremely rare classes (e.g., "No clear structure," "Phase ramp"), which have only one test sample. This imbalance is the primary driver for the gap between the Micro-F1 and Macro-F1 scores reported in the main paper.

Table 9: Per-Class F1-Scores for SFT-90B (SSFS) on Artifact Detection. N indicates the number of test samples (out of 79) that contain the artifact.

Artifact Class	**N**	**0-shot F1**	**1-shot F1**	**3-shot F1**	**5-shot F1**	**7-shot F1**
Halo artifacts	44	0.47	0.72	0.72	0.81	0.90
Line artifacts	34	0.53	0.64	0.64	0.60	0.67
Low contrast	25	0.52	0.68	0.67	0.84	0.88
Grid artifacts	21	0.40	0.48	0.50	0.56	0.67
Non-uniform background	17	0.49	0.64	0.60	0.72	0.75
Singularity artifacts	11	0.40	0.58	0.62	0.59	0.64
Noisy artifacts	8	0.29	0.38	0.46	0.80	0.63
Local distortion	5	0.00	0.00	0.00	0.22	0.00
No obvious artifacts	4	0.00	0.00	0.00	0.00	0.00
Blurring	2	0.00	0.00	0.00	0.00	0.00
No clear structure	1	0.00	0.00	0.00	0.00	0.00
Phase ramp	1	0.00	0.00	0.00	0.00	0.00
Overall (Micro F1)	**173**	**0.43**	**0.59**	**0.60**	**0.67**	**0.73**

A.5.3 Per-Sample-Type Performance on Parameter Recommendation

While the preceding analysis addresses class imbalance, this section provides a scientific breakdown of performance by material category (sample_type). We analyzed the best-performing LLM (Base Llama 3.1-70B with SSFS) to understand which ptychographic samples are more challenging for the model. The performance, as shown in Table 10, is highly dependent on the material being analyzed.

Table 10: Per-Sample-Type Micro-F1 Scores for Base Llama 3.1-70B with SSFS on the Parameter Recommendation task. N indicates the number of test samples for each type.

Sample Type	**N**	**0-shot F1**	**1-shot F1**	**3-shot F1**	**5-shot F1**	**7-shot F1**
Ecoli in starch	9	0.301	0.771	0.966	0.941	0.978
TiO2 in capillary	9	0.233	0.654	0.800	0.783	0.973
Siemens star test pattern	4	0.320	0.480	0.889	0.960	0.923
XRnanotech 3D test pattern	7	0.213	0.278	0.462	0.571	0.684
Supercrystal	6	0.108	0.222	0.429	0.417	0.500
Liquid-metal particles	3	0.250	0.714	0.889	1.000	1.000
Integrated circuit	2	0.167	0.000	0.286	0.667	0.889
NCM battery	1	0.250	0.750	0.750	1.000	1.000
Bilayer WSe2	1	0.200	0.500	0.333	0.000	0.000
CuS battery	1	0.000	0.000	0.500	0.667	0.500
LNO	1	0.000	0.000	0.000	0.000	0.000
Overall (Micro F1)	**44**	**0.229**	**0.544**	**0.747**	**0.785**	**0.848**

The results in Table 10 show a strong correlation between the number of available test samples (N) and model performance, particularly for sample types with very few instances. The model consistently achieves near-perfect scores on types with many examples (e.g., Ecoli in starch) but struggles on types with only a single test sample (e.g., LNO, CuS battery). This suggests that while ICL is highly effective, its performance is still sensitive to the diversity and representation of the underlying data distribution, which is a key area for future work in dataset augmentation and expansion.