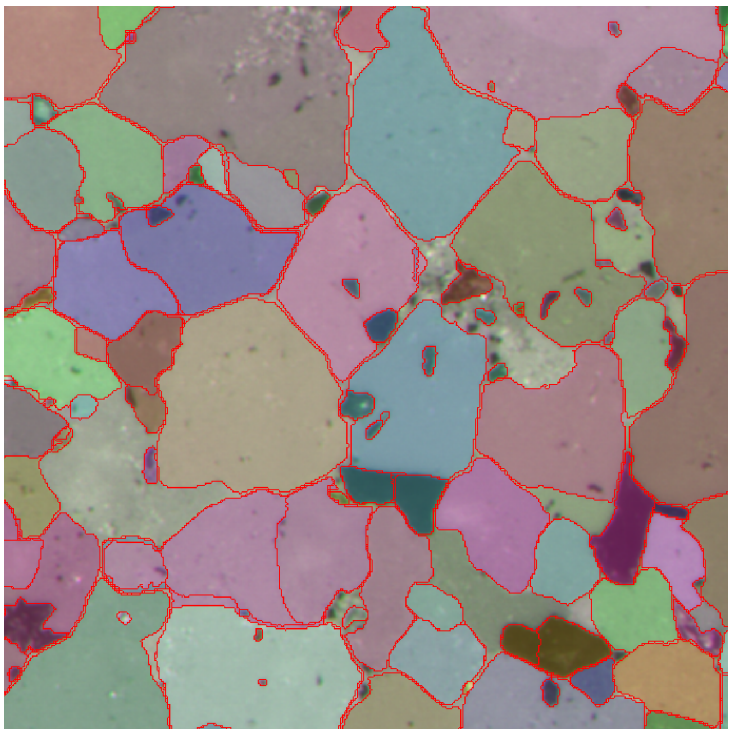
# Fine-tuning SAM for Grain Segmentation in Reflected-light Microscopy

**Auteur :** Campo, Adriaan
**Promoteur(s) :** Pirard, Eric
**Faculté :** Faculté des Sciences appliquées
**Diplôme :** Master en ingénieur civil des mines et géologue, à finalité spécialisée en géologie de l'ingénieur et de l'environnement
**Année académique :** 2025-2026
**URI/URL :** http://hdl.handle.net/2268.2/25195

UNIVERSITE DE LIEGE
FACULTE DES SCIENCES APPLIQUEES

# Fine-tuning SAM for Grain Segmentation in Reflected-light Microscopy



Thesis presented to obtain the degree of:

**Ingénieur Civil des Mines et Géologue**

Author: **Adriaan Campo**
Thesis supervisor: **Prof. Dr. Eric Pirard**
Academic year: **2025 – 2026**

# 1 Remerciements

Je tiens à exprimer ma sincère gratitude à mon promoteur, le Prof. Dr. Eric Pirard, pour m'avoir donné l'opportunité de réaliser ce projet et pour ses conseils, sa perspicacité et son soutien tout au long de son développement.

Je souhaite adresser mes remerciements tout particuliers au Dr. Matteo Ruth, dont l'aide a été essentielle à l'aboutissement de ce travail. Son expertise en apprentissage automatique, sa disponibilité constante et son soutien continu ont été précieux pour la mise en place, la structuration et la réalisation réussie de cette recherche.

Je remercie également le Dr. Hassan Bouzahzah pour avoir fourni les données utilisées dans ce travail et pour son aide précieuse lors de la réalisation du projet.

Je tiens aussi à remercier l'ensemble de mes professeurs qui ont contribué à mon développement académique et personnel au cours des deux dernières années, et en particulier le Prof. Dr. Brouyère pour m'avoir admis dans ce programme. Leur engagement et leur enseignement ont joué un rôle fondamental dans la construction de mes connaissances et de ma manière de penser.

Sur un plan plus personnel, je souhaite remercier mes parents, Elli et Hubert, ainsi que ma partenaire, Wilma, pour leur soutien constant, leurs encouragements et leur confiance tout au long de ce parcours exigeant et parfois atypique. Enfin, je suis profondément reconnaissant envers mes amis pour leur amitié, leur soutien et la joie qu'ils apportent à ma vie.

# 2 Abstract

Automated grain segmentation in reflected-light microscopy (RLM) is a key step in quantitative texture analysis and automated mineralogy, yet it remains challenging due to weak and heterogeneous contrast and strong domain variability, among other factors. Classical image-processing approaches often lack robustness, while supervised deep-learning methods require large, domain-specific annotated datasets and frequently generalize poorly across materials. This work investigates the applicability of the Segment Anything Model (SAM), a promptable vision foundation model, for grain segmentation in RLM images and evaluates whether parameter-efficient fine-tuning (PEFT) can improve its performance.

First, we benchmark the zero-shot performance of SAM with ViT-B and ViT-H backbones against a fully supervised DeepLabV3+ model on two open-source ore microscopy datasets (iron ore and copper ore). Quantitative evaluation using pixel-wise metrics (IoU, Dice, precision, and recall) shows that SAM achieves segmentation accuracy comparable to DeepLabV3+ without any task-specific training, while demonstrating superior cross-dataset generalization. In contrast, DeepLabV3+ exhibits severe performance degradation when applied outside its training domain.

Second, we apply SAM to polished dolomite sections. For this dataset, SAM is compared with classical stereological measurements, different preprocessing strategies are tested, and we show that SAM yields quantitatively consistent textural descriptors. Moreover, we demonstrate that fine-tuning SAM improves both mask-level and full-field segmentation accuracy while updating only a small fraction of the model parameters.

Third, we apply SAM to industrial datasets of glass and silica grains from solar panels, which are challenging because the reflectivity of glass is close to that of the resin. The out-of-the-box versions of SAM can segment both glass and silica grains without difficulty.

Overall, the study shows that

- **(i)** SAM is a strong starting point for segmenting unseen RLM data,
- **(ii)** preprocessing can be critical for achieving good segmentation results with SAM, and
- **(iii)** small, targeted fine-tuning is an effective and practical path for improving foundation models in reflected-light microscopy segmentation tasks.

Finally, we release a Colab-ready codebase with a clear repository structure to support reproducibility.

# 3 Résumé

La segmentation automatique des grains en microscopie en lumière réfléchie (RLM) constitue une étape clé de l'analyse quantitative des textures et de la minéralogie automatisée, mais elle demeure difficile en raison, entre autres, d'un contraste faible et hétérogène et d'une forte variabilité entre domaines. Les approches classiques de traitement d'images manquent souvent de robustesse, tandis que les méthodes supervisées basées sur l'apprentissage profond nécessitent de grands jeux de données annotés spécifiques au domaine et généralisent fréquemment mal d'un matériau à l'autre. Ce travail étudie l'applicabilité du Segment Anything Model (SAM), un modèle fondationnel de vision guidé par des invites (promptable), pour la segmentation des grains dans des images RLM, et évalue si un ajustement fin économe en paramètres (parameter-efficient fine-tuning, PEFT) peut améliorer ses performances.

Tout d'abord, nous évaluons les performances zero-shot de SAM avec des architectures ViT-B et ViT-H par rapport à un modèle DeepLabV3+ entièrement supervisé sur deux jeux de données open source de microscopie des minerais (minerai de fer et minerai de cuivre). L'évaluation quantitative à l'aide de métriques pixel à pixel (IoU, Dice, précision et rappel) montre que SAM atteint une précision de segmentation comparable à celle de DeepLabV3+ sans aucun entraînement spécifique à la tâche, tout en démontrant une meilleure capacité de généralisation inter-jeux de données. À l'inverse, DeepLabV3+ présente une forte dégradation de performance lorsqu'il est appliqué en dehors de son domaine d'entraînement.

Dans un deuxième temps, nous appliquons SAM à des sections polies de dolomite. Pour ce jeu de données, SAM est comparé à des mesures stéréologiques classiques, différentes stratégies de prétraitement sont testées, et nous montrons que SAM fournit des descripteurs texturaux quantitativement cohérents. De plus, nous démontrons que l'ajustement fin de SAM améliore à la fois la précision de segmentation au niveau des masques et sur l'ensemble du champ, tout en ne mettant à jour qu'une petite fraction des paramètres du modèle.

Troisièmement, nous appliquons SAM à des jeux de données industriels de grains de verre et de silice issus de panneaux solaires, qui sont particulièrement difficiles car la réflectivité du verre est proche de celle de la résine. Les versions prêtes à l'emploi de SAM permettent de segmenter sans difficulté à la fois les grains de verre et de silice.

Dans l'ensemble, cette étude montre que

- **(i)** SAM constitue un excellent point de départ pour la segmentation de données RLM inédites,
- **(ii)** le prétraitement peut être déterminant pour obtenir de bons résultats de segmentation avec SAM, et
- **(iii)** un ajustement fin léger et ciblé représente une voie efficace et pratique pour améliorer les modèles fondationnels dans les tâches de segmentation en microscopie en lumière réfléchie.

Enfin, nous mettons à disposition une base de code compatible avec Colab, dotée d'une structure de dépôt claire afin de favoriser la reproductibilité.

# Contents

# List of Figures

# List of Tables

# 4 Introduction

## 4.1 State-of-the-art in Automated Mineralogy

Automated mineralogy (AM) is the computer-controlled identification, quantification, and textural mapping of minerals at scale, most commonly coupling backscattered-electron (BSE) imaging with energy-dispersive X-ray spectroscopy (SEM–EDS) to generate particle-level mineral maps, modal mineralogy, and liberation/association metrics without manual point counting [8]. In processing terms, AM formalizes the liberation spectrum (the joint distribution of mineral exposure and locking across particle sizes) which explicitly links ore texture to separation performance and circuit design [9].

Within SEM–EDS, the Mineral Liberation Analyser (MLA) established acquisition modes (BSE-only, sparse/area EDS, X-ray mapping) and workflow logic for mineral and texture quantification [10]. Quantitative Evaluation of Minerals by Scanning Electron Microscopy (QEMSCAN) extended automated SEM–EDS mineral mapping for process mineralogy and deportment studies [11]. In addition, the TESCAN Integrated Mineral Analyzer (TIMA) expanded highly automated mapping and intergrowth analysis, with robust image processing of mineral maps [12]. Finally, ZEISS Mineralogic provides a comparable SEM–EDS stack with integrated quantification and morphochemical outputs that has been deployed on iron-ore case studies [13]. Elemental imaging beyond SEM–EDS includes micro–X-ray fluorescence (XRF) raster mapping on drill-core surfaces with automated mineral calculation, providing quantitative mineral maps and AM-style descriptors, and demonstrating correlative workflows with infrared features and machine learning [14, 15].

Reflected-light microscopy (RLM) has been automated via multispectral, pixel-wise specular reflectance measurements in the visible–near-infrared (VNIR), enabling high-throughput classification of opaque minerals such as iron oxides [16] and sulfides [17] at lower cost and higher speed relative to scanning electron microscopy (SEM) for suitable ore types [18]. In this context, RLM can deliver:

- **(i)** modal mineralogy of opaque phases through reflectance-based supervised classification [16];
- **(ii)** grain size and shape of identified phases from segmented photomicrographs, including automated grain recognition and boundary detection under polarized reflected light [19]; and
- **(iii)** optical-property features (specular reflectance spectra and anisotropy/bireflectance patterns) used both as discriminants for mineral identification and as textural markers that complement morphological metrics [2].

Modern implementations measure multispectral reflectance and use machine-learning classifiers to produce mineral maps that, for ore types with diagnostic reflectance/anisotropy and high-quality polishing, approach the decision accuracy of SEM–EDS workflows at substantially lower operating costs and higher speeds [16].

Across platforms, AM extracts a consistent set of descriptors:

- **Modal mineralogy:** phase proportions per sample or stream, derived from classified pixels or segments [20]; these underpin element deportment, reconcile assays, and parameterize geometallurgical response models [21].

- **Liberation and mineral associations:** exposure classes (free, partially liberated, locked) and contact/edge-length tables between phases, computed directly from labeled mineral maps [10]. The mix of free versus locked particles strongly influences flotation: liberation class × size maps onto rate constants and selectivity, with coarse fractions often collision/attachment-limited and fines more susceptible to hydraulic entrainment and induction-time effects [22, 23].
- **Particle and grain size distributions:** whole particles and grains-in-particles, derived from segmentation and per-grain labeling [9]; recovery and selectivity vary strongly with size [23].
- **Shape and morphology:** aspect ratio, circularity, convexity, roundness, and perimeter/Fourier descriptors obtained from AM images [24]; these affect collision efficiencies, bubble attachment, and aggregation, thereby shifting flotation outcomes [25].
- **Textural descriptors:** intergrowth type, border-length density, inclusion/vein metrics, and phase-contact topology computed from mineral maps [26]; texture governs breakage–liberation pathways and mineral exposure at plant grind sizes, directly affecting separation behavior [27].

Reflected-light methods yield pixel-level specular reflectance spectra for mineral identification [16], anisotropy and bireflectance textures under polarized illumination that aid in discriminating closely related oxides and sulfides and sharpen grain boundaries [2], and RLM-derived grain size/shape metrics that serve as texture quantifiers and liberation proxies [19]. Concretely, VNIR multispectral pipelines with supervised classification and curated reflectance libraries have demonstrated reliable automated ore identification and quantification [16]; sulfide-focused multispectral quantification has been validated on real ore images [17]; and polarized-light image analysis has shown automated hematite grain recognition and particle texture analysis using gray-level intercepts, strengthening boundary detection and anisotropy-aware features [2, 19]. Improvements are expected from broader, open reflectance databases and rigorously reproducible workflows to enhance cross-lab comparability (a limitation implicit in database-driven classification) [16]; from richer feature-fusion models that combine reflectance with polarization textures and reduce supervision in ambiguous spectral regimes [16, 17]; and from tighter quality assurance/quality control (QA/QC) around sample preparation and stereology, already recognized as critical for accurate size and texture extraction in automated measurements [28].

Next, we briefly describe RLM with emphasis on reflectance, bireflectance, and pleochroism, which contribute to the image contrast used in the automated segmentation pipeline.

## 4.2 Reflected-light Microscopy

RLM is a fundamental imaging technique in materials science and ore petrography, particularly for the study of opaque minerals, operationally defined as minerals that do not transmit light in standard petrographic thin sections [1]. This classification is gradational rather than absolute: minerals such as quartz remain transparent even when thick, whereas others, such as galena, absorb light completely even in very thin sections [1]. Most opaque minerals are oxides and sulfides and are commonly referred to as ore minerals due to their economic importance [1].

Opaque minerals are identified on polished samples using a petrographic microscope equipped with an incident-light illuminator (see Figure 1). The samples can be polished thin sections or epoxy-mounted rock or mineral fragments. Surfaces must be polished to a mirror finish using progressively finer abrasives, and no cover slip is used. Polished thin sections are especially versatile, as they can be examined by transmitted-light microscopy, RLM, and electron microprobe analysis [1].

Reflected-light images capture variations in reflectance and optical anisotropy, as well as crystallographic orientation and polishing quality (relief, scratches, and pull-outs). Apparent brightness and color may correlate with composition or phase but are also affected by illumination conditions, detector response, and surface preparation, leading to variable contrast if conditions are not standardized [2].

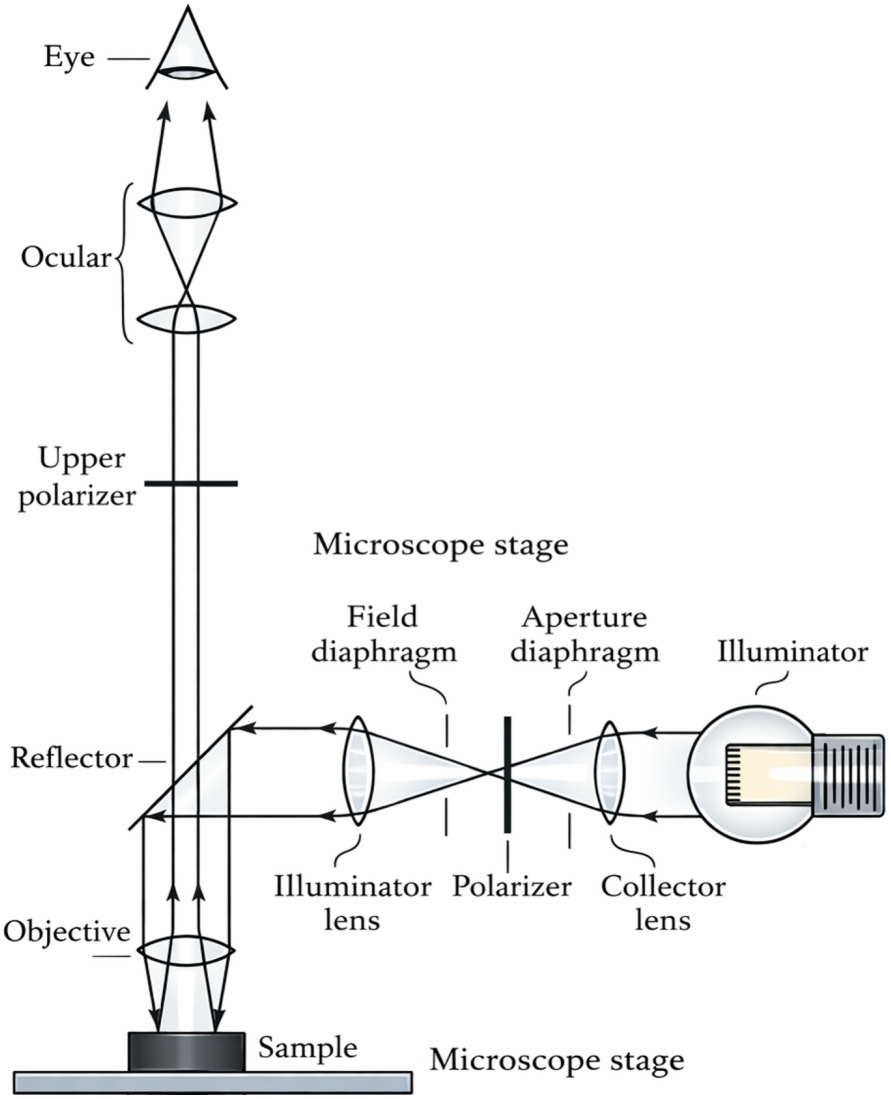



Figure 1: Essential elements of a reflected-light microscope. The reflector is usually a half-silvered mirror that allows some light to be reflected, some to pass. Light from the illuminator is directed through the objective onto the sample by the reflector. Light reflected from the sample passes up through the microscope to the ocular and then to the observer's eye. The polarizer usually provides E–W vibrating light. The upper polarizer (analyzer) is oriented N–S or may be rotated 10–20°. The two diaphragms are manipulated to control the size of the light beam reaching the sample. Figure adapted from [1].

Despite these limitations, RLM is widely used because it is simple, accessible, and well suited for large-area imaging, automated stage mapping, and mosaicking. With well-controlled preparation, illumination, and white-balance calibration, it supports automated image-analysis workflows such as grain detection, segmentation, and quantitative characterization [1]. In this work, we use monochrome images in which segmentation contrast is governed by reflectance and, in anisotropic minerals, by bireflectance (orientation-dependent reflectance).

### 4.2.1 Reflectance

Reflectance ($R$) is defined as the proportion of incident light that is reflected from a mineral surface and is perceived microscopically as the brightness of the mineral in plane polarized light. Reflectance, also termed reflectivity, for normally incident light is given by

$$R = \frac{I}{I_0} = \frac{(n - n_0)^2 + k^2}{(n + n_0)^2 + k^2} \tag{4.1}$$

where $I_0$ is the intensity of the incident light, $I$ is the intensity of the reflected light, $n_0$ is the refractive index of the immersion medium between the sample and the objective lens, $n$ is the refractive index of the mineral, and $k$ is its absorption coefficient. Although opaque minerals strongly absorb light, a refractive index can still be defined because a small amount of light penetrates the surface and propagates briefly within the mineral before being absorbed [1].

Both the refractive index and the absorption coefficient vary with wavelength, a phenomenon known as dispersion. Consequently, reflectance is also wavelength dependent, as predicted by Equation 4.1. Reflectance further depends on the refractive index of the immersion medium, which is typically air ($n = 1$). Equation 4.1 also shows that reflectance increases with increasing refractive index and increasing absorption coefficient of the mineral. Common silicate minerals such as quartz and feldspar, with low absorption coefficients and refractive indices near 1.5, have low reflectance values of approximately 5% in air. By contrast, sphalerite, with a refractive index of about 2.8 and an absorption coefficient near 0.01, has a mean reflectance of roughly 17% in air, whereas galena, with a refractive index of about 4.3 and an absorption coefficient of approximately 1.7, exhibits a much higher mean reflectance of around 43% in air [1].

### 4.2.2 Bireflectance

Bireflectance is an optical property characteristic of anisotropic opaque minerals and is defined as the difference in reflectance between two orthogonally vibrating components of reflected light. Minerals crystallizing in the isometric system are optically isotropic and display constant reflectance regardless of orientation or stage rotation, provided that the incident light is normal to the surface. Minerals of lower symmetry (including those in the hexagonal, tetragonal, orthorhombic, monoclinic, and triclinic systems) are optically anisotropic and bireflectant. In these minerals, reflected light is generally split into two orthogonally vibrating components of different intensity [1].

As the microscope stage is rotated, the apparent brightness of a bireflectant mineral varies, appearing brightest when the vibration direction associated with higher reflectance is parallel to the vibration direction of the incident light and dimmest when rotated by 90°. The magnitude of bireflectance depends on crystallographic orientation and is typically

greatest in orientations that would also exhibit maximum birefringence in transmitted light [1].

In uniaxial minerals, reflected-light behavior can be described by the ordinary reflectance ($R_o$) and extraordinary reflectance ($R_e$). The ordinary reflectance remains constant for all orientations, whereas the extraordinary reflectance varies with section orientation. Intermediate extraordinary values may be denoted $R'_e$, and the reflectance pair may alternatively be labeled $R_1$ and $R_2$, with $R_1 < R_2$. Maximum bireflectance in uniaxial minerals occurs when the optic axis lies parallel to the microscope stage and decreases to zero when the optic axis is vertical [1].

Biaxial minerals exhibit more complex bireflectance behavior. Maximum bireflectance occurs when the optic plane is parallel to the stage, yielding minimum and maximum reflectance values $R_1$ and $R_2$ related to the principal refractive indices $n_\alpha$ and $n_\gamma$, respectively, such that the maximum bireflectance is given by $R_2 - R_1$. Intermediate orientations produce intermediate reflectance values, including one associated with $n_\beta$, although this has limited practical significance [1]. Because reflectance varies strongly with crystallographic orientation and composition and commonly overlaps among different minerals, bireflectance is generally assessed qualitatively as absent, weak, moderate, or strong based on the degree of brightness change observed during stage rotation [1].

Care must be taken to distinguish true bireflectance from apparent brightness variations caused by sample tilt. Tilt-induced brightness changes recur every 180° of stage rotation, whereas true bireflectance produces variations every 90°. Accurate observations therefore require well-leveled, properly polished samples; tilted specimens should be repolished or adjusted prior to optical interpretation [1].

### 4.2.3 Color and Pleochroism

Most opaque minerals have colors in reflected light that range from nearly pure white to various shades of gray because some wavelengths are reflected more than others. Isotropic minerals are optically isotropic; therefore, reflection does not vary with orientation or with rotation of the stage [1]. If the mineral is anisotropic (tetragonal, hexagonal, orthorhombic, monoclinic, or triclinic), the light is split into two plane-polarized components for most mineral orientations. Each reflected component has its own pattern of reflection for different wavelengths of light, and the mineral will change from one reflection color to the other as the stage is rotated. The strength of pleochroism varies among anisotropic minerals. For some, such as arsenopyrite, it is faint, but for others, such as covellite, it is striking. The degree to which reflection pleochroism is displayed depends on mineral orientation and wavelength. In the work below, we use monochrome images; however, these pleochroism effects can also be visible as changes in pixel intensity when the mineral orientation is rotated [1].

## 4.3 Texture Analysis of Reflected-light Microscopy Images

### 4.3.1 Classical Approaches

**4.3.1.1 Manual and Semi-manual Workflows:** Historically, grain segmentation has relied heavily on manual or semi-manual interpretation by trained experts. Analysts delineate grain boundaries visually, either directly on micrographs (e.g., on printed images) or within interactive software. While expert judgment can accommodate complex visual cues and contextual information, manual workflows are inherently time-consuming,

operator-dependent, and difficult to reproduce; inter- and intra-operator variability and drift in criteria are well-documented concerns in quantitative microscopy [29].

Semi-manual approaches, combining automated preprocessing (filtering, thresholding, watershed) with human correction (brush/paint, split/merge, and boundary editing), can reduce workload but retain dependence on operator expertise and consistent protocols. Widely used platforms such as ImageJ [30], Fiji [31], Trainable Weka Segmentation [32], and ilastik [33] exemplify interactive, user-in-the-loop segmentation that improves practicality yet still requires careful calibration, documentation of parameter choices, and repeatability checks [29].

**4.3.1.2 Automated Workflows: Thresholding, Edge Detection, and Watershed-based Segmentation:** Thresholding-based methods are among the earliest and most widely used automated segmentation techniques. By partitioning pixel intensities into foreground and background (or multiple classes), these methods exploit contrast differences between grains or phases [34].

Edge-detection algorithms identify putative grain boundaries by locating regions of high intensity gradient. Common choices include gradient-magnitude operators [35], the Marr–Hildreth [36], and the Canny detector [37]. Watershed segmentation partitions an image into catchment basins on a topographic surface defined by intensity, gradient magnitude, or distance transforms [38].

Although these approaches can be computationally efficient, they are sensitive to non-uniform shading and illumination, noise, and highly textured images. Careful parameterization and calibration therefore remain essential to avoid over- or under-segmentation [39].

**4.3.1.3 Automated Workflows: Stereological Approaches and Intercept Methods:** Stereological techniques estimate grain size and phase statistics from systematic sampling on test lines or points, thereby avoiding full image segmentation while still requiring local boundary or phase identification at test probes [40]. Classic procedures include the lineal-intercept (Heyn) method for mean grain size [28], the planimetric/Jeffries count for grains per unit area [41], and point counting for phase (area/volume) fractions via the Delesse principle [42]. These methods have a long history in metallography and petrography, and we will specifically use the intercept method as a reference later in this work (see Figure 2).

However, stereological estimates rely on appropriate, design-based sampling to remain unbiased. The classic descriptors emanating from these methods, such as porosity, mean lineal intercepts $\bar{L}(\alpha)$, rose plots/ellipse fits for anisotropy (major/minor intercepts $L_a, L_b$, see Figure 3), and intercept–length distributions, do not directly recover detailed grain-shape descriptors, topology, or spatial connectivity, and can be less informative for anisotropic microstructures without specialized sampling strategies [43].

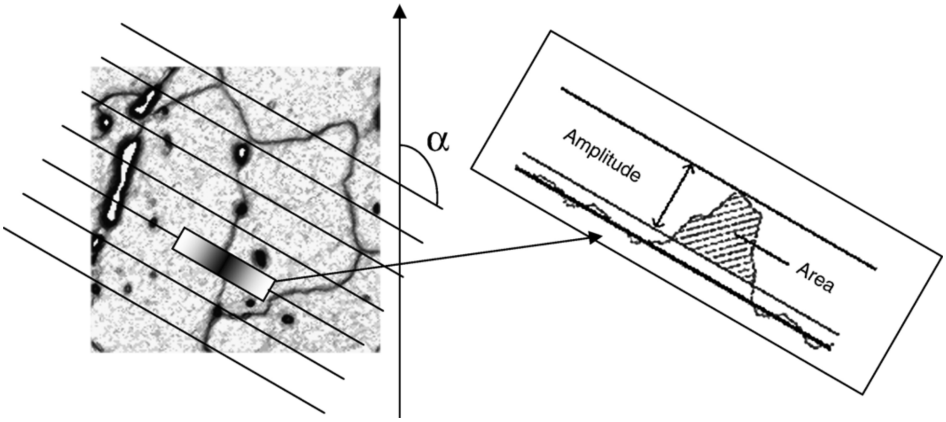



Figure 2: Gray-level intercept (edge) detection via line scans oriented at angle $\alpha$. Intensity is sampled along each scan line and a transition is flagged where the gradient peak passes a dual criterion: (i) peak amplitude above threshold and (ii) sufficient integrated peak area, suppressing noise and narrow spurious responses. The inset illustrates the 1-D profile with amplitude and area measures. The background is an inverted multiradial gradient image of hematite grains; parallel lines depict the scan geometry. Figure adapted from [2].

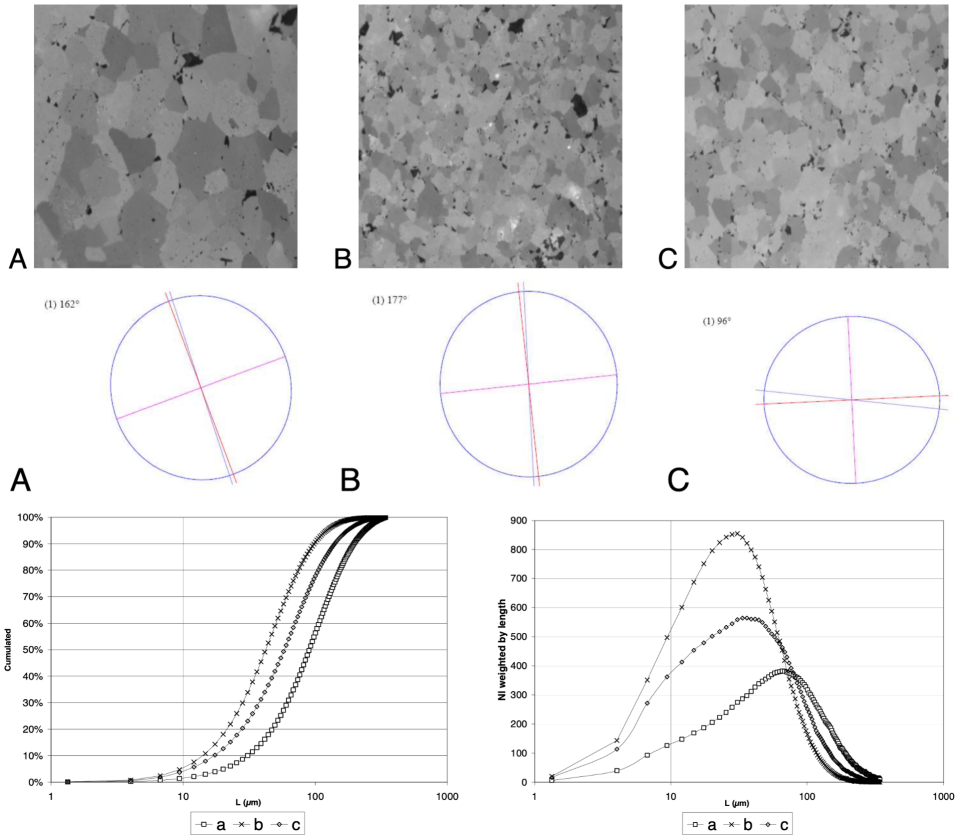



Figure 3: Stereological descriptors from lineal–intercept analysis for three specimens (A–C). Top: reflected–light micrographs. Middle: rose diagrams of mean lineal intercept $\bar{L}(\alpha)$; principal axes define the major/minor intercepts $(L_a, L_b)$ as an anisotropy summary, with A ($L_a = 77.96\,\mu$m, $L_b = 75.69\,\mu$m); B ($L_a = 37.89\,\mu$m, $L_b = 36.29\,\mu$m); C ($L_a = 63.65\,\mu$m, $L_b = 61.22\,\mu$m). Bottom left: cumulative distributions of intercept length $L$. Bottom right: intercept frequency weighted by length, highlighting characteristic scales. Figure adapted from [3].

**4.3.1.4 Automated Workflows: Variational, Graph-cut, and Spectral Formulations:** Other segmentation approaches formulate the task as an optimization that balances "fit to the data" with "avoid ragged, noisy boundaries." Variational models encourage uniform or smooth regions [44, 45]. Graph-cut methods cast the problem as finding a minimum cut in a graph over pixels, which can yield globally optimal solutions under common smoothness priors [46–49]. Spectral and graph-based approaches group pixels by connectivity using eigenvectors or fast region merging [50, 51].

These methods often require little or no training data, but they can be sensitive to parameter tuning and initialization, and they transfer poorly across modalities. A full review is beyond our scope.

Thus, the classical approaches discussed above have critical disadvantages, such as poor generalization and the need for extensive supervision to avoid over- or under-segmentation. This has motivated data-driven approaches that learn features and decision rules from annotated data to improve generalization in low-contrast, noisy, or topologically complex settings [52].

### 4.3.2 Data-driven Approaches in Segmentation

Machine learning offers a paradigm shift by learning segmentation rules directly from data rather than relying on hand-crafted heuristics [53].

**4.3.2.1 Fully Convolutional Networks:** Fully Convolutional Networks (FCNs) rose to prominence with the U-Net architecture [52] and numerous subsequent improvements [54–57], and they have been adopted for automated RLM segmentation [58–60]. DeepLab models [55, 56], used later in this work, are effective for mineral-grain segmentation because they capture a wide range of grain sizes and textures while producing sharp boundaries (see Figure 5). In petrographic thin-section studies they have proven competitive or state of the art [61–64], and they have also been applied to RLM [4, 65].

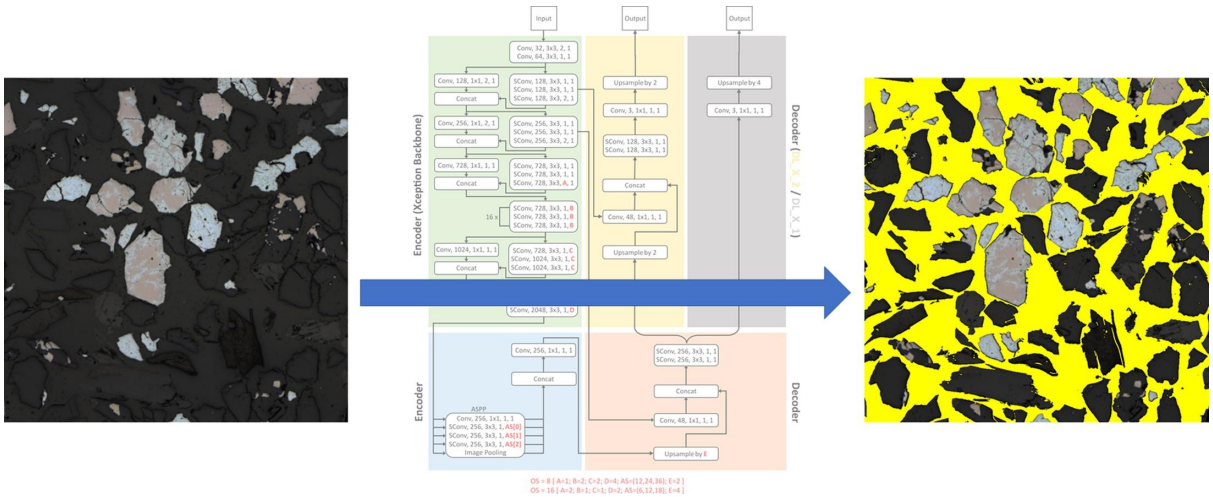



Figure 4: Automated segmentation of reflected-light ore micrographs using DeepLabV3+. Left: input micrograph. Center: encoder–decoder for pixel-wise semantic segmentation. Right: predicted mask separating opaque and non-opaque minerals from epoxy resin (yellow). Labels were obtained via correlative microscopy; the method reports >90% overall accuracy. Figure adapted from [4].

Instance segmentation, e.g., Mask R-CNN [66], produces a separate mask per grain, enabling size and shape measurements. Such methods have been adapted for materials science [67], mineralogy [68–70], and RLM [71,72]. Panoptic segmentation further unifies instance and semantic segmentation by also labeling background regions [73]. Building on these advances [74], panoptic segmentation has very recently been applied to microscopy [75] and to RLM in metallurgy [76], but, to our knowledge, not yet to mineralogy.

**4.3.2.2 Transformers:** Foundation models are large pretrained models that learn general features from broad data and can be adapted to many tasks [77]. In vision, many state-of-the-art foundation models (e.g., Mask2Former [78] and the Segment Anything Model (SAM) [5]) are transformer-based because they:

- **(i)** allow any part of an image to use information from any other part (long-range interactions),
- **(ii)** execute efficiently in parallel on modern hardware, and
- **(iii)** are straightforward to condition with prompts or tokens for new tasks [79].

These vision transformers capture global context, emphasize the most relevant patterns, and can be steered with prompts, making them a natural backbone for flexible segmentation systems.

**4.3.2.3 Segment Anything Model:** SAM is a promptable, class-agnostic segmentation foundation model that pairs a high-capacity Vision Transformer (ViT) image encoder with a lightweight, promptable mask decoder for fast interactive or fully automatic mask generation across diverse domains [5]. The ViT encoder first computes a global image embedding via self-attention; the mask decoder then (i) applies self-attention to the prompt tokens and (ii) uses cross-attention to query the image embedding with those tokens, so clicks/boxes directly steer which regions are segmented. A lightweight head outputs multiple mask proposals together with a learned IoU-quality score to rank alternatives [5]. SAM is trained on the SA-1B corpus (over a billion masks on tens of millions of images), enabling strong zero-shot generalization without task- or category-specific fine-tuning [5]. Architecturally, SAM comprises three components [5] (see Figure 5):

- **(i)** a ViT encoder that produces a dense, high-resolution latent embedding of the image,
- **(ii)** a prompt encoder that converts user inputs (points, boxes, rough masks) into compact tokens, and
- **(iii)** a mask decoder (a small Transformer with MLP heads; an MLP is a feed-forward network of fully connected layers with nonlinear activations) that fuses image embeddings with prompt tokens to predict multiple candidate masks and corresponding IoU-quality scores in a single forward pass.

This design amortizes the heavy image encoding (done once per image) and keeps the decoder as the fast interaction path (millisecond-scale updates). The ViT encoder comes in three versions: ViT-H (heavy; 636M parameters), ViT-L (large; 308M), and ViT-B (base; 93M). ViT-H improves substantially over ViT-B, but offers only marginal gains over ViT-L. When speed or computational cost matters, ViT-B can be a better choice;

however, when the best zero-shot segmentation quality is required, ViT-H is recommended [5].

Functionally, SAM supports three workflows:

- **(i)** interactive segmentation with iterative points/boxes,
- **(ii)** automatic mask generation that proposes object-like regions without prompts, and
- **(iii)** box/point-guided segmentation for integration with detectors or user clicks [5].

Compared to traditional pipelines (e.g., detection + mask heads) [55, 66, 74], SAM's promptable formulation decouples category recognition from region delineation, aiding transfer. Its Transformer design inherits global-context modeling from ViT/Transformers [79], while the quality head ranks plausible masks, which is helpful for ambiguous boundaries [5].

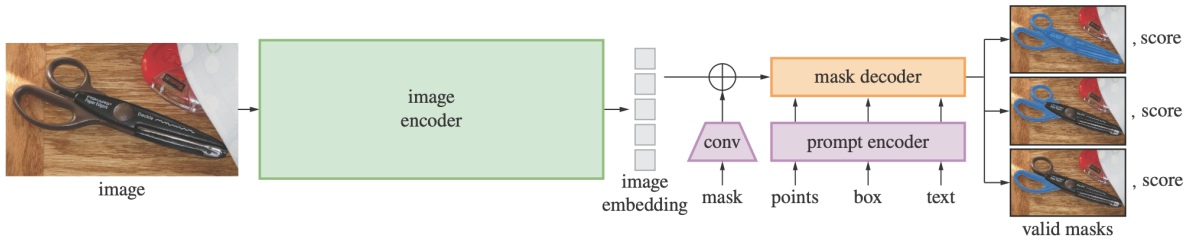



Figure 5: Overview of the SAM architecture. The model consists of a ViT-based image encoder that extracts dense image embeddings, a prompt encoder that maps user or algorithmic prompts into a shared embedding space, and a lightweight mask decoder that generates segmentation masks. Figure adapted from [5].

SAM shows strong zero-shot robustness across varied natural images and acquisition conditions, often producing high-quality masks from a few prompts [5]. Its scale underpins broad transfer without task-specific fine-tuning, enabling rapid adaptation to new domains and datasets [5]. Operationally, once the image is encoded, the lightweight decoder enables near-instant updates, supporting efficient human-in-the-loop labeling and exploratory analysis [5]. The model also returns multiple candidate masks per prompt with predicted IoU scores [5].

Consequently, SAM is being adopted across medicine [80–82], materials science [83], remote sensing [84, 85], and other areas [86, 87]. In mineralogy, work in transmitted-light microscopy is emerging [88, 89], but adoption otherwise remains limited at present.

Despite its strengths, SAM was not trained on scientific microscopy or materials-imaging corpora; its SA-1B pretraining focuses on natural-image distributions [5]. Indeed, RLM images often display weak or spatially variable contrast, with grain boundaries that are diffuse, discontinuous, or masked by polishing artifacts [1]. Intra-grain anisotropy, oxidation films, and surface damage can rival inter-grain differences, causing over-segmentation, while look-alike grains of the same phase drive under-segmentation [9]. Finally, differences in illumination, polarization, camera response, and operator choices across datasets make standardization and calibration essential [2]. Consequently, SAM can suffer from domain shift, causing models to generalize poorly across domains [90],

leading to over- or under-segmentation of grains, merging adjacent grains with weak boundaries, or hallucinating object-like regions from texture [5].

These limitations underscore the need for systematic evaluation and, where necessary, domain adaptation before deploying SAM in grain segmentation workflows.

**4.3.2.4 Adapting Foundation Models to Scientific Domains:** Fine-tuning adapts a pretrained model to a new task by updating parameters with task-specific data, but full end-to-end tuning is compute-heavy and can overfit in low-data regimes [77]. In practice, parameter-efficient strategies balance adaptability and stability:

- **Linear probing / head-only tuning:** freeze the encoder and train only lightweight task heads [91].
- **Partial unfreezing with discriminative learning rates:** unfreeze upper transformer blocks (or the decoder) and use larger learning rates for new layers, smaller for pretrained ones [92].
- **Adapter-based Parameter Efficient Tuning (PEFT):** insert small bottleneck adapters in each block and train only the adapters, keeping the backbone frozen [93].
- **Low-Rank Adapters (LoRA) and related methods:** replace full weight updates with trainable low-rank matrices on attention or MLP projections [94].
- **Prompt-based tuning:** learn a small set of tokens that condition the transformer (visual prompt tuning) while freezing the backbone [95].
- **Regularization and data efficiency:** strong augmentations, label smoothing, stochastic depth, and semi/self-training with pseudo-labels to curb overfitting and exploit unlabeled data [96].

In this work, we adopt PEFT and LoRA. First, we insert an input adapter before the first layer of the SAM image encoder to enable SAM to accept image stacks of shape $B \times W \times N$, where $N$ can be any number of polarizer orientations [93, 97]. This adapter can be any trainable neural network that outputs a tensor of shape $B \times W \times 3$, which is compatible with the remainder of the pipeline, since SAM normally takes images of shape $B \times W \times 3$.

Second, we apply LoRA only to the mask decoder of SAM [94]. Within each decoder block, we insert LoRA adapters on the self-attention and cross-attention projections (namely the $Q$, $K$, $V$, and output linear layers). The image encoder and prompt encoder remain frozen.

**4.3.2.5 Low-rank Adaptation for Reflected-light Microscopy:** LoRA enables efficient fine-tuning with a tiny fraction of parameters [94]. Concretely, for a pretrained weight $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, LoRA parameterizes the update as $\Delta W = BA$, with $A \in \mathbb{R}^{r \times d_{\text{in}}}$, $B \in \mathbb{R}^{d_{\text{out}} \times r}$, and $r \ll \min(d_{\text{in}}, d_{\text{out}})$; the effective weight is $W' = W + \frac{\alpha}{r} BA$. This preserves the pretrained representation while focusing learning capacity on a low-dimensional subspace [94], updating only $\ll 1\%$ of parameters, enabling rapid iteration, reduced overfitting in low-data regimes, and easy model/weight sharing. In RLM, LoRA can help mitigate problems related to domain shift [90] and data scarcity [98], while keeping

training cost relatively low. Practical settings often use $r \in [4, 16]$, $\alpha \in [8, 32]$, moderate weight decay, and early stopping, paired with domain-aware preprocessing to reduce overfitting [94]. Unless otherwise stated, we set $r = 8$, $\alpha = 8$, and scale by $\alpha/r$. This keeps the number of trainable parameters small while adapting the parts of the network that directly perform region delineation.

## 4.4 Metrics for Model Evaluation

Let $G \subset \Omega$ be the set of foreground pixels in the ground-truth mask and $P \subset \Omega$ the set of foreground pixels in the prediction. Define true positives as $\mathrm{TP} = |P \cap G|$, false positives as $\mathrm{FP} = |P \setminus G|$, false negatives as $\mathrm{FN} = |G \setminus P|$, and true negatives as $\mathrm{TN} = |\Omega \setminus (P \cup G)|$.

To assess segmentation quality and practical usability across datasets and scenarios, we report a concise set of metrics: Intersection-over-Union (IoU), Dice coefficient, Precision, Recall, and wall-clock CPU Time per image:

- **Intersection-over-Union (IoU):** IoU measures the overlap between the predicted and reference masks, normalized by their union. It is widely adopted in vision benchmarks because it penalizes both over- and under-segmentation in a single score [99].

$$\mathrm{IoU} = \frac{|P \cap G|}{|P \cup G|} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}.$$

- **Dice coefficient:** Dice is the harmonic mean of Precision and Recall and is closely related to IoU but somewhat more tolerant to slight boundary shifts [100].

$$\mathrm{Dice} = \frac{2\,|P \cap G|}{|P| + |G|} = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} = \frac{2\,\mathrm{IoU}}{1 + \mathrm{IoU}}.$$

- **Precision and Recall:** Precision quantifies the fraction of predicted ore pixels that are correct; Recall quantifies the fraction of true ore pixels that are recovered [101].

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad \mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}.$$

- **CPU Time (s):** Per-image CPU inference time captures computational cost on a standardized setup [102]. CPU Time is directly relevant for throughput in practical lab workflows and batch analyses. However, this metric is hardware- and implementation-dependent [103], and it does not reflect possible GPU acceleration or multi-threading gains [104]. In this work, CPU Time is therefore primarily reported as a comparative indicator across methods evaluated under identical conditions.

## 4.5 Objectives and Structure of the Thesis

Automated grain segmentation remains challenging because methods must balance accuracy, robustness, and scalability across diverse materials and imaging conditions [2]. Classical techniques lack adaptability, whereas supervised deep learning often depends on large, domain-specific annotations that are costly and brittle under domain shift [90]. These constraints motivate exploring alternative paradigms that reduce the annotation burden while preserving generalization, leading to the following research objectives:

- **Primary objective:** Evaluate the applicability of SAM to grain segmentation in RLM.
- **Secondary objective:** Assess whether PEFT/LoRA can mitigate domain shift and improve transfer.

In the remainder of this thesis, we first benchmark SAM against DeepLabV3+ on two open-source RLM ore datasets. We then evaluate its performance on a dolomite RLM dataset imaged at six polarization angles and attempt to improve it via fine-tuning. Next, we evaluate and fine-tune SAM on three RLM datasets containing glass and silica grains from solar panels. We provide a reproducible Colab notebook and a private repository for code use and inspection. Finally, we discuss results and limitations and present our conclusions.

In the remainder of this thesis, for brevity, we denote SAM with a ViT-H encoder as **SAM-H** and SAM with a ViT-B encoder as **SAM-B**. The PEFT/LoRA–fine-tuned SAM with a ViT-B encoder is written as SAM-$\mathrm{B}_{\mathrm{PEFT/LoRA}}$.

# 5 Benchmarking SAM on Two Open-source Datasets

## 5.1 Introduction

In this chapter, we benchmark SAM-H and SAM-B against DeepLabV3+ on two open-source RLM ore datasets: an iron-ore set (FeM, [6]) and a copper-ore set (Cu, [7]), both available on Zenodo.

Our objectives are twofold. First, we measure SAM's zero-shot transfer to RLM ore images. Second, we compare SAM with a supervised DeepLabV3+ model [55,56] trained in-domain on the provided binary grain masks (ore vs. resin).

All methods are evaluated on identical train/validation/test splits using the same pixel-wise metrics (IoU, Dice, precision, recall) and wall-clock CPU time. We report in-distribution accuracy and cross-dataset generalization (FeM→Cu and Cu→FeM) to probe robustness to changes in mineralogy, grain morphology, and contrast.

## 5.2 Materials and Methods

### 5.2.1 The Datasets

Two open-source datasets are used in this benchmark:

- **(i)** the FeM dataset (iron ore, see Figure 6), and
- **(ii)** the Cu dataset (copper ore, see Figure 7).

Both datasets consist of paired RLM RGB images and corresponding binary grain masks that label ore regions versus embedding resin [6,7]. The binary grain masks were generated using a correlative microscopy workflow in which optical images were registered to SEM acquisitions, and SEM-derived information was thresholded to produce objective binary labels.

- **FeM (iron ore):** FeM contains 81 correlated image pairs. Each pair comprises a 24-bit RGB image of an itabiritic iron-ore concentrate from the Quadrilátero Ferrífero (Brazil), mainly hematite and quartz, with minor magnetite and goethite, and an 8-bit binary grain mask in which pixels are labeled `ore`=0 and `resin`=255. Samples were size-classified and density-concentrated ($-149{+}105\,\mu$m, density $> 3.2$), cold-mounted in epoxy, then ground and polished. Imaging used a reflected-light microscope with a $\times 10$ (NA 0.20) objective; fields were registered to SEM, yielding images of $999 \times 756$ pixels at $1.05\,\mu$m/pixel [6].
- **Cu (copper ore):** Cu contains 121 correlated image pairs. Each pair includes a 24-bit RGB image of a copper ore from Yauri (Cusco, Peru) with complex mineralogy (sulfides, oxides, silicates, native copper) and an 8-bit binary grain mask with labels `ore`=0 and `resin`=255. Samples were size-classified ($+74{-}100\,\mu$m), cold-mounted in epoxy, then ground and polished. Imaging used a $\times 20$ (NA 0.40) reflected-light objective; fields were registered to SEM, producing images of $1017 \times 753$ pixels at $0.53\,\mu$m/pixel [7].

These datasets display pronounced differences in grain size, contrast, and phase complexity, making them well suited for evaluating robustness and cross-domain generalization.

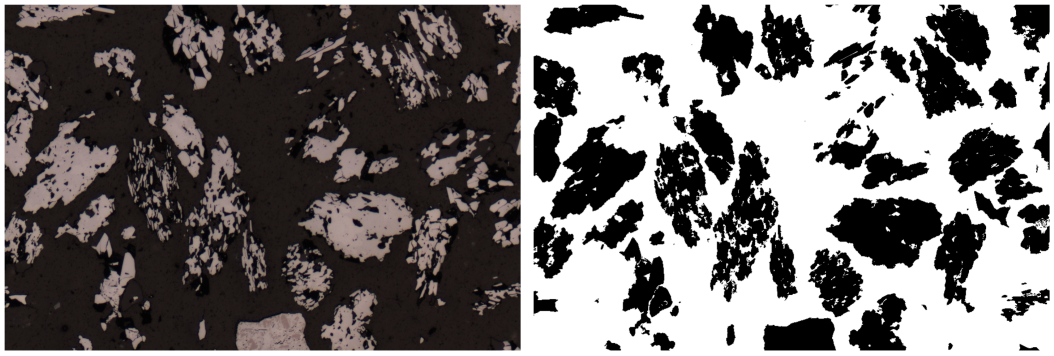


Figure 6: Example from the FeM dataset [6]. Left: RLM RGB image of an iron-ore polished section. Right: binary grain mask (ore vs. resin) obtained from SEM-derived reference segmentation after correlative registration.

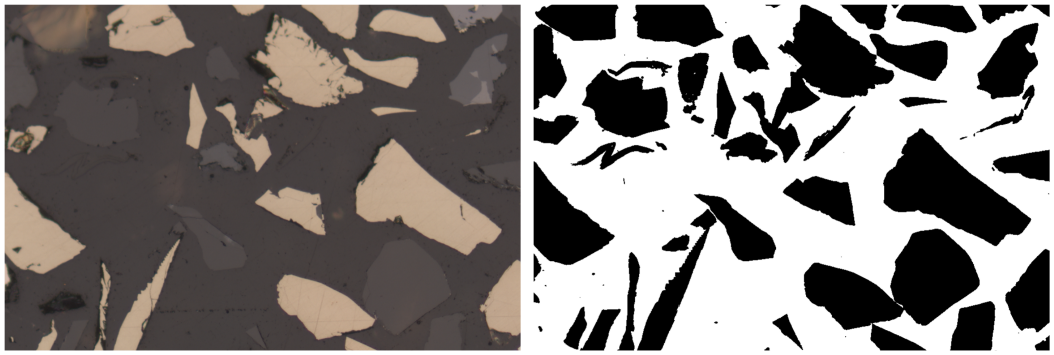


Figure 7: Example from the Cu dataset [7]. Left: RLM RGB image of a copper-ore polished section. Right: binary grain mask (ore vs. resin) obtained from SEM-derived reference segmentation after correlative registration.

### 5.2.2 DeepLabV3+ Training

As a supervised reference, DeepLabV3+ [55, 56] was trained independently on each dataset using the provided binary grain masks. We trained the network for 10 epochs with pixel-wise cross-entropy loss, appropriate for semantic segmentation where each pixel belongs to a single class [105], using an 80/10/10% train/validation/test split.

Training convergence is summarized by the training and validation loss curves shown in Figures 8 and 9.

### 5.2.3 Benchmarking SAM against DeepLabV3+

After training the DeepLabV3+ network, we compare its segmentation performance with SAM-B and SAM-H using qualitative inspection and quantitative evaluation. Qualitatively, we inspect characteristic failure modes, boundary leakage, fragmentation, and omission of fine-scale features. Quantitatively, we report pixel-wise metrics (IoU, Dice, precision, recall) and CPU time.

All evaluations are performed on held-out test images from each dataset, using the same 80/10/10% train/validation/test split used to train DeepLabV3+. Metrics are computed against the known ground-truth binary grain masks.

As an additional test of cross-domain generalization, DeepLabV3+ trained on the Cu dataset is evaluated on FeM, and vice versa; the same cross-domain protocol is applied to the SAM models. The same metrics are reported in each case.

### 5.2.4 Experimental Design

First, we train a DeepLabV3+ model separately on each of the two datasets, yielding two domain-specific DeepLab models. Each dataset is split into training, validation, and test partitions.

Next, we perform full-field inference on the Cu and FeM test sets using SAM-B, SAM-H, and both DeepLab models. Each DeepLab model is evaluated both in-distribution (its own domain) and cross-dataset (the other domain). In total, we evaluate four models on two test sets, producing eight sets of predicted masks.

To evaluate the results, we compute quantitative metrics (pixel-wise IoU, Dice, precision, recall) and wall-clock CPU time. We also provide qualitative assessments: for each image, we generate a $2 \times 2$ overview panel highlighting FP and FN for each method.

These tests allow us to report in-distribution accuracy and cross-dataset generalization.

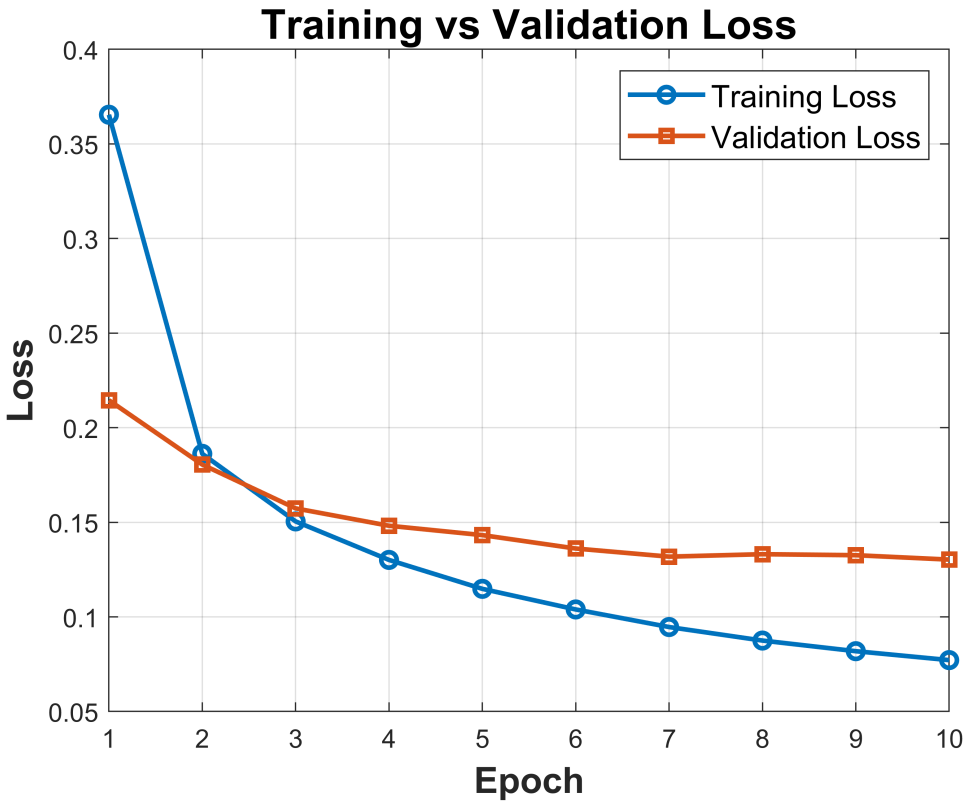



Figure 8: DeepLabV3+ training dynamics on the Cu dataset. Training loss (blue) and validation loss (red) as a function of epoch.

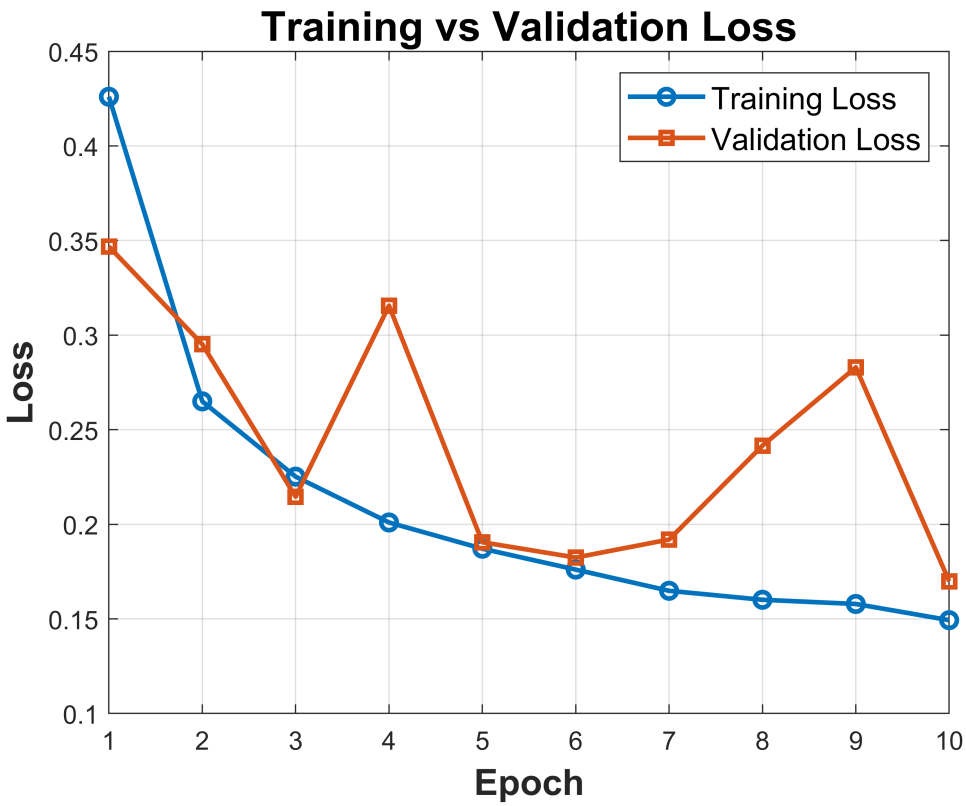



Figure 9: DeepLabV3+ training dynamics on the FeM dataset. Training loss (blue) and validation loss (red) as a function of epoch.

## 5.3 Results

### 5.3.1 Qualitative Comparison

Results were compared qualitatively. For each image in the dataset, we generated a $2 \times 2$ overview panel highlighting FP and FN for each method. Examples are shown in Figures 10 and 11.

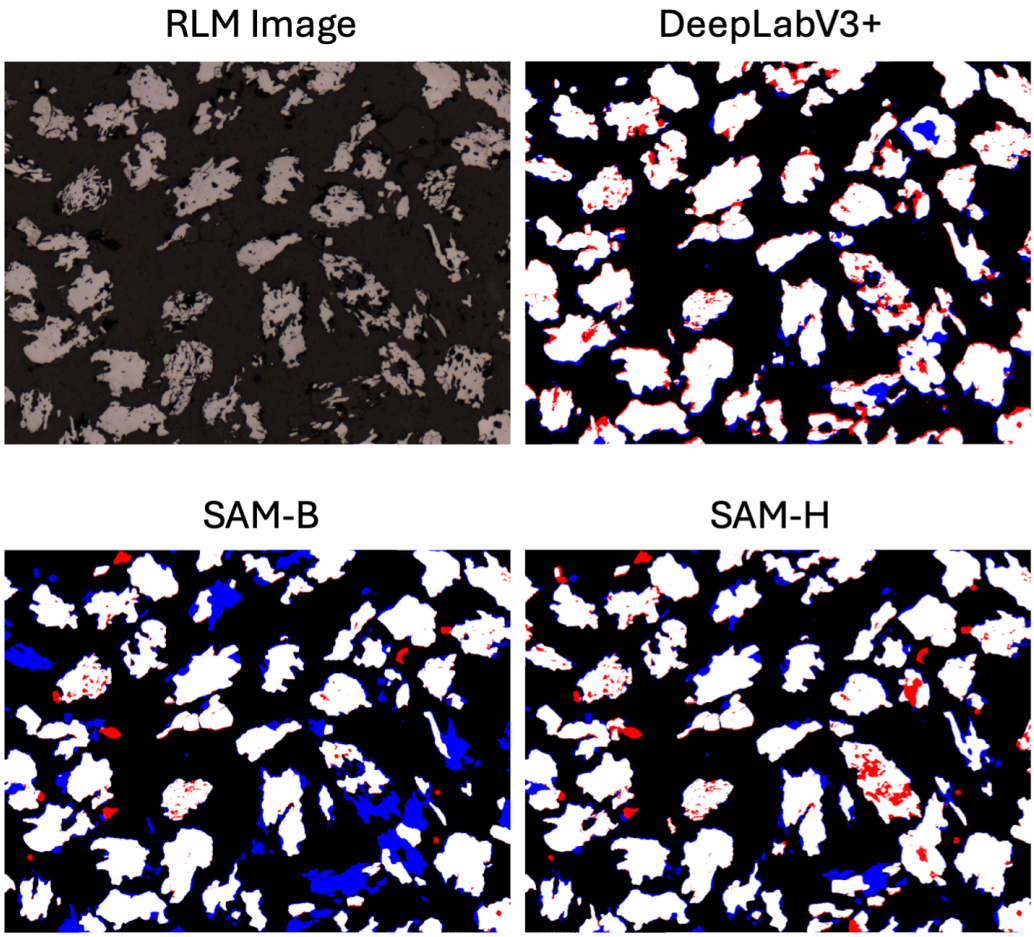



Figure 10: Qualitative segmentation comparison on the Cu dataset. Shown are the original RLM image, the DeepLabV3+ prediction, SAM-B and SAM-H predictions, and the SEM-derived binary grain mask. The Cu dataset exhibits strong textural heterogeneity, which challenges boundary delineation for all methods. Red = FP, blue = FN.

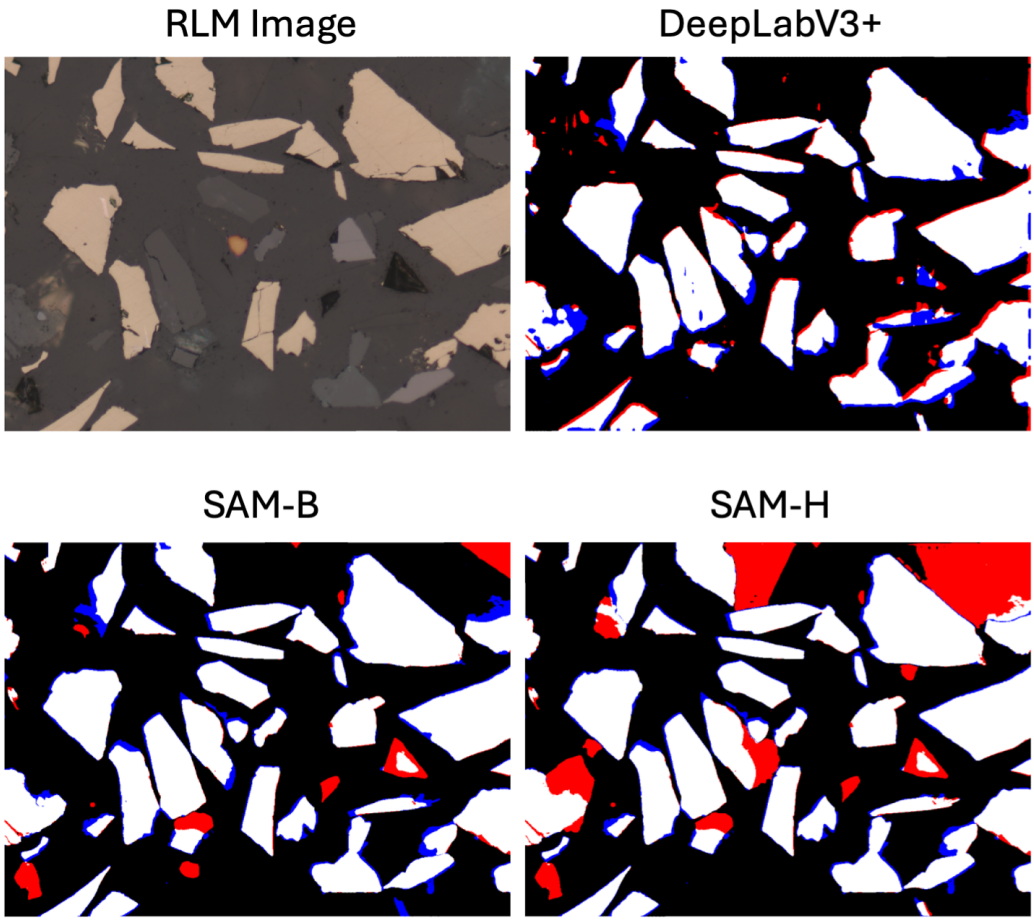



Figure 11: Qualitative segmentation comparison on the FeM dataset. Shown are the original RLM image, the DeepLabV3+ prediction, SAM-B and SAM-H predictions, and the SEM-derived binary grain mask. Compared to the Cu dataset, the FeM images show larger variation in reflectivity across grains. Red = FP, blue = FN.

### 5.3.2 Quantitative Comparison

Quantitative performance metrics for the Cu and FeM datasets are summarized in Tables 1 and 2. Reported values correspond to per-image statistics (median and standard deviation) computed over the respective test sets.

Table 1: Summary statistics for shape, accuracy, and CPU runtime metrics for SAM-B, SAM-H, and DeepLabV3+ in the Cu dataset.

| | SAM-B | | SAM-H | | DeepLabV3+ | |
|---|---|---|---|---|---|---|
| Metric | Median | Std | Median | Std | Median | Std |
| IoU | 0.798 | 0.034 | 0.766 | 0.049 | 0.796 | 0.024 |
| Dice | 0.888 | 0.021 | 0.867 | 0.032 | 0.886 | 0.015 |
| Precision | 0.930 | 0.036 | 0.831 | 0.059 | 0.917 | 0.018 |
| Recall | 0.864 | 0.023 | 0.900 | 0.012 | 0.868 | 0.019 |
| CPU Time (s) | 3.106 | 0.084 | 4.087 | 0.071 | 0.105 | 0.007 |

Table 2: Summary statistics for shape, accuracy, and CPU runtime metrics for SAM-B, SAM-H, and DeepLabV3+ in the FeM dataset.

| | SAM-B | | SAM-H | | DeepLabV3+ | |
|---|---|---|---|---|---|---|
| Metric | Median | Std | Median | Std | Median | Std |
| IoU | 0.705 | 0.035 | 0.830 | 0.013 | 0.840 | 0.031 |
| Dice | 0.827 | 0.024 | 0.907 | 0.008 | 0.913 | 0.019 |
| Precision | 0.943 | 0.013 | 0.927 | 0.012 | 0.908 | 0.044 |
| Recall | 0.737 | 0.040 | 0.890 | 0.013 | 0.915 | 0.014 |
| CPU Time (s) | 3.062 | 0.543 | 4.054 | 0.062 | 0.105 | 0.026 |

### 5.3.3 Cross-dataset Generalization

To assess robustness to domain shift, we trained on one dataset and evaluated on the other without retraining. These results were evaluated qualitatively (see Figures 12 and 13) and quantitatively (see Tables 3 and 4).

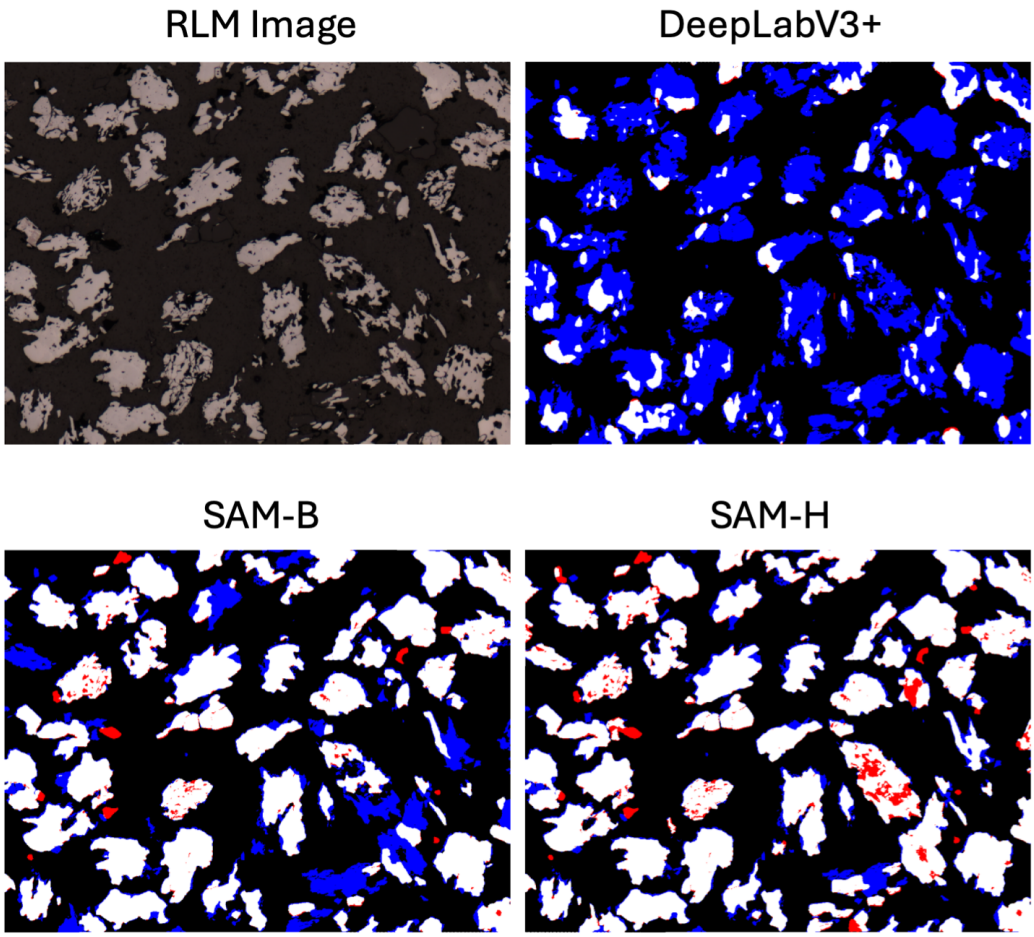



Figure 12: Cross-dataset evaluation: models trained on the Cu dataset and evaluated on the FeM dataset. Shown are the original RLM image, the DeepLabV3+ prediction, SAM-B and SAM-H predictions, and the SEM-derived binary grain mask. Red = FP, blue = FN.

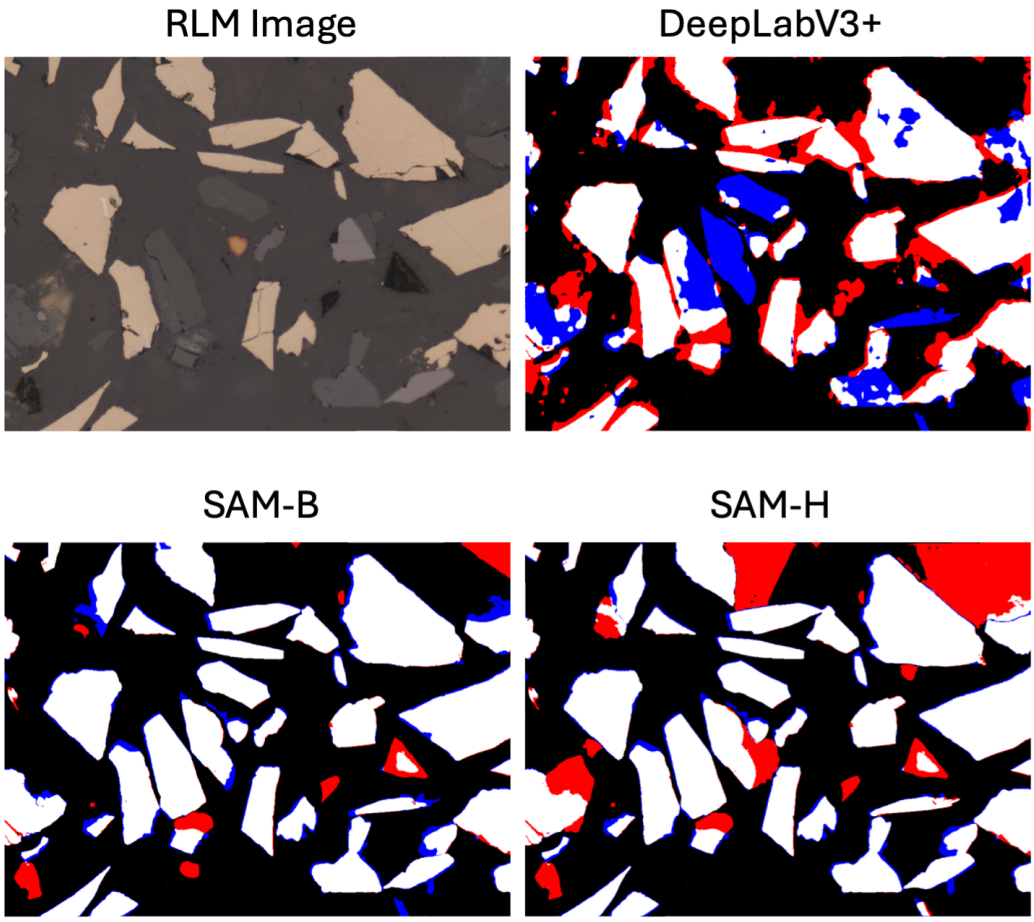



Figure 13: Cross-dataset evaluation: models trained on the FeM dataset and evaluated on the Cu dataset. Shown are the original RLM image, the DeepLabV3+ prediction, SAM-B and SAM-H predictions, and the SEM-derived binary grain mask. Red = FP, blue = FN.

Table 3: Summary statistics for shape, accuracy, and CPU runtime metrics for SAM-B, SAM-H, and DeepLabV3+, the latter trained on the Cu dataset, applied to the FeM dataset.

| | SAM-B | | SAM-H | | DeepLabV3+ | |
|---|---|---|---|---|---|---|
| Metric | Median | Std | Median | Std | Median | Std |
| IoU | 0.705 | 0.035 | 0.830 | 0.013 | 0.200 | 0.025 |
| Dice | 0.827 | 0.024 | 0.907 | 0.008 | 0.333 | 0.035 |
| Precision | 0.943 | 0.013 | 0.927 | 0.012 | 0.989 | 0.004 |
| Recall | 0.737 | 0.040 | 0.890 | 0.013 | 0.200 | 0.025 |
| CPU Time (s) | 3.069 | 0.112 | 4.077 | 0.036 | 0.106 | 0.009 |

Table 4: Summary statistics for shape, accuracy, and CPU runtime metrics for SAM-B, SAM-H, and DeepLabV3+, the latter trained on the FeM dataset, applied to the Cu dataset.

| | SAM-B | | SAM-H | | DeepLabV3+ | |
|---|---|---|---|---|---|---|
| Metric | Median | Std | Median | Std | Median | Std |
| IoU | 0.798 | 0.034 | 0.766 | 0.049 | 0.578 | 0.051 |
| Dice | 0.888 | 0.021 | 0.867 | 0.032 | 0.732 | 0.041 |
| Precision | 0.930 | 0.036 | 0.831 | 0.059 | 0.707 | 0.044 |
| Recall | 0.864 | 0.023 | 0.900 | 0.012 | 0.773 | 0.057 |
| CPU Time (s) | 3.125 | 0.084 | 4.129 | 0.058 | 0.105 | 0.008 |

## 5.4 Discussion and Conclusion

Across both ore datasets, SAM and DeepLabV3+ perform strongly. SAM generally matches DeepLabV3+ in both qualitative (Figures 10, 11) and quantitative evaluations (Tables 1, 2), with SAM-H outperforming SAM-B while incurring approximately 30% more CPU time.

However, DeepLabV3+ is far more CPU-efficient, approximately 30–40× faster, making it the safer choice when high-quality, target-domain labels are available (Tables 1, 2).

Under domain shift, the contrast reverses. Qualitatively (Figures 12, 13), DeepLabV3+ exhibits poor transfer; quantitatively (Tables 3, 4), the supervised model shows large drops in IoU, Dice, and recall.

In short, SAM can match DeepLabV3+ at a substantially higher compute cost, yet it tends to generalize much better across datasets, offering a compelling zero-/low-shot baseline to jump-start analysis and accelerate dataset creation.

# 6 Grain detection in Dolomite Reflected-light Microscopy Images

## 6.1 Introduction

In this chapter, we present a detailed case study on grain detection in dolomite RLM images. The dataset is particularly challenging: it contains closely packed grains with heterogeneous sizes and reflectivities, multiple mineral phases, and imperfect or spurious boundaries.

The image stack was acquired at six polarization angles, providing an additional dimension of information, and is accompanied by a grain-boundary map computed via edge detection.

While this grain-boundary map is not a true ground-truth segmentation, it enables comparison of classical stereological and intercept-based descriptors with the corresponding descriptors derived from the SAM pipeline.

In addition, the multi-angle polarization stack can inform preprocessing steps that improve SAM-based segmentation.

Our objectives are threefold. First, we evaluate preprocessing strategies that exploit the polarization stack to improve SAM segmentation quality. Second, we compare classical descriptors derived from SAM segmentations with those computed from the grain-boundary map. Third, we investigate fine-tuning SAM-B with PEFT/LoRA adaptation [94] to assess whether lightweight, domain-specific adaptation mitigates domain shift and data scarcity.

## 6.2 Materials and Methods

### 6.2.1 The Dataset

The dataset comprises RLM images of a dolomite section acquired at six rotation angles (0°, 30°, 60°, 90°, 120°, 150°) (see Figures 14–16).

It consists of 10 stacks of 6 RLM monochrome 8-bit images and corresponding 8-bit grain-boundary maps, with `pore`=100, `resin/matrix`=50, and `grain`=0. The grain-boundary maps were generated using an edge-detection algorithm. The resolution is $1280 \times 1024$ pixels, which exceeds what can be fed into SAM on our current workstation. Therefore, the dataset was cut into 12 tiles (Figure 16) of $300 \times 300$ pixels, yielding a total workable dataset of 120 stacks.

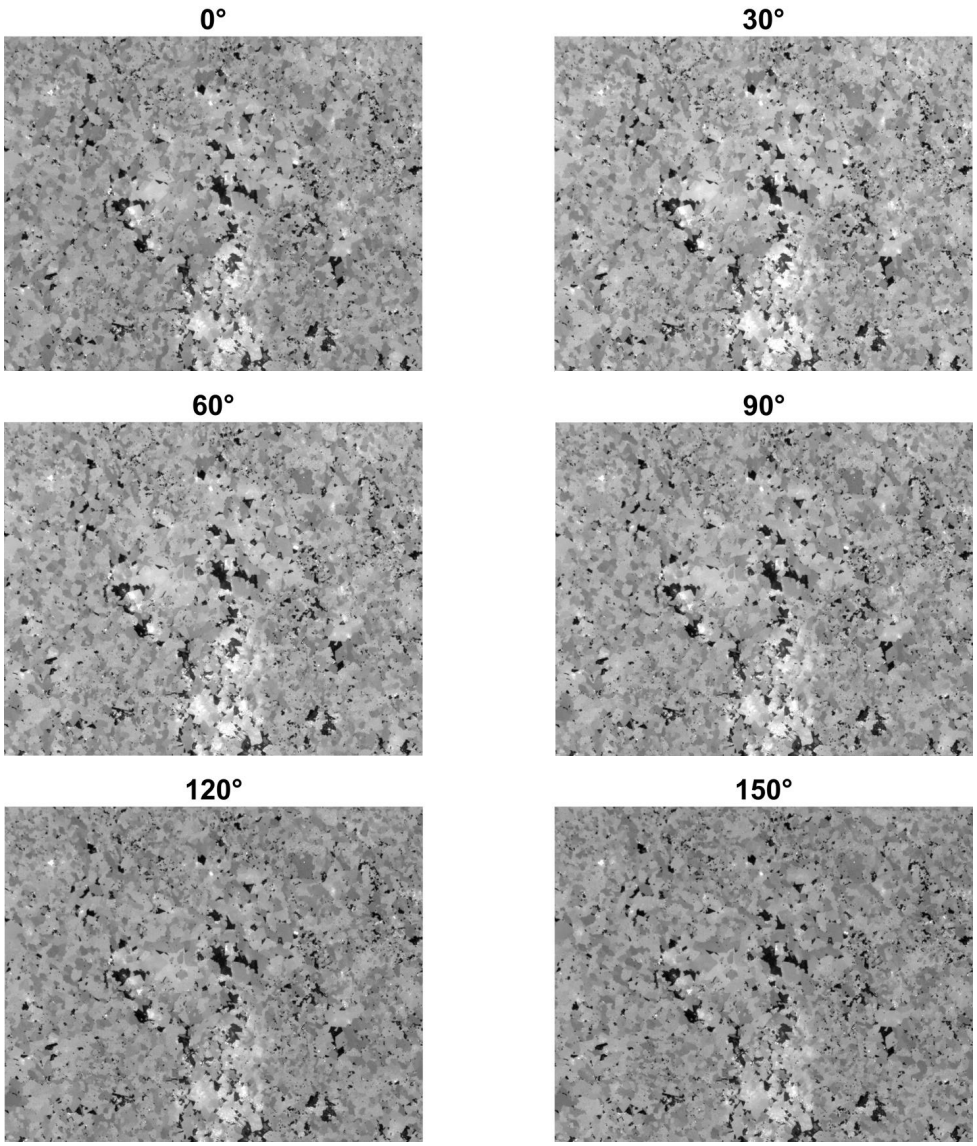



Figure 14: Original RLM tiles acquired at multiple rotation angles (0°, 30°, 60°, 90°, 120°, 150°). Variations in brightness and texture across angles provide complementary information for grain-boundary detection.

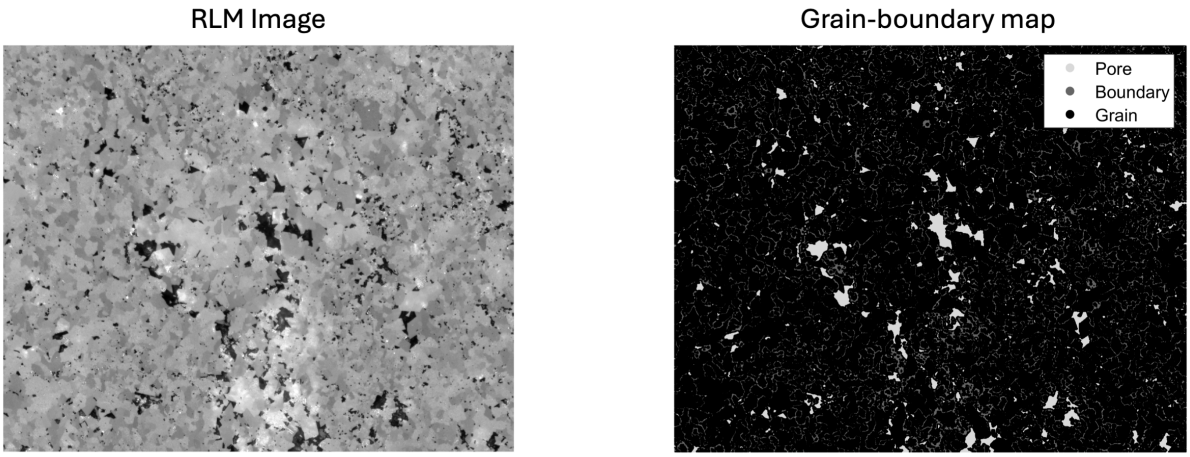



Figure 15: Original grayscale RLM image (left) and 8-bit grain-boundary map (right), with pores (light gray), resin/matrix (dark gray), and grains (black). The grain-boundary map was generated using an edge-detection algorithm. This image serves as a reference representation of grain topology for subsequent quantitative analyses.

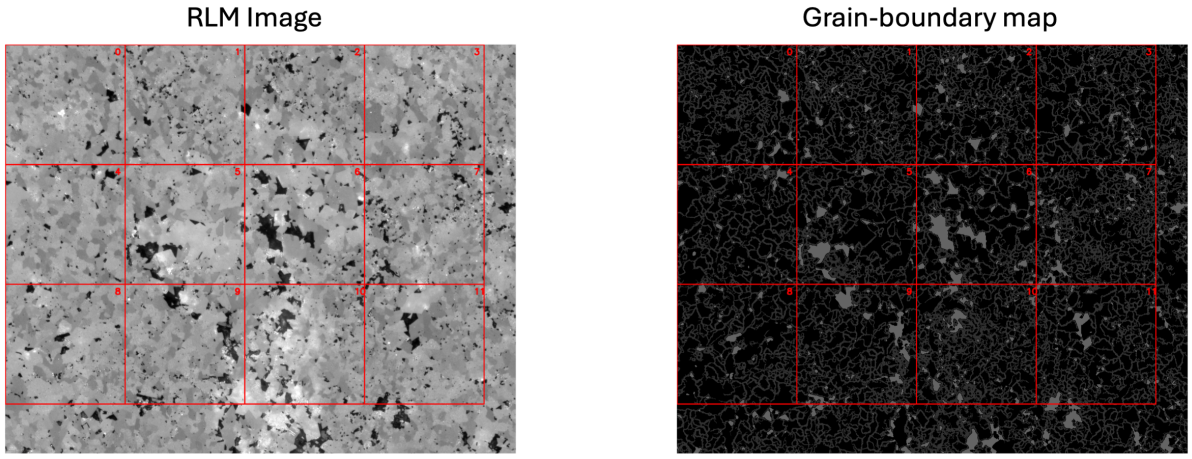



Figure 16: Subdivision of the full field into individual tiles used for localized analysis. Tiling allows spatial variability in grain properties to be quantified across the sample while satisfying the RAM and CPU/GPU constraints. Original grayscale RLM image (left) and grain-boundary map (right), with pores (light gray), resin/matrix (dark gray), and grains (black).

### 6.2.2 Experimental Design

First, we test several preprocessing methods on the stack of polarized images and evaluate their performance with SAM-B and SAM-H. This evaluation is primarily qualitative.

Then, from the grain-boundary map, we compute a set of classical stereological and intercept-based descriptors. We also calculate the same descriptors from the binary grain masks obtained by the SAM models.

An additional goal of this work is to assess whether PEFT/LoRA fine-tuning can improve SAM-B to approach the performance of SAM-H. Training requires a binary grain mask as a reference; therefore, we use binary grain masks predicted by SAM-H (with PCA-based preprocessing) as pseudo–ground truth for training.

After training, we evaluate the performance of SAM-B and SAM-B$_{\text{PEFT/LoRA}}$ against SAM-H, considering both single-mask and full-field performance.

### 6.2.3 Preprocessing Methods for SAM Input Data

We evaluated several preprocessing strategies to assess the impact on segmentation quality and downstream quantitative measurements: (i) monochrome images, where the per-pixel median intensity is replicated across the three RGB channels; (ii) bireflectance images, where the bireflectance map is replicated across the three RGB channels; and (iii) PCA images, where the first three principal components constitute the three channels (see Figure 17).

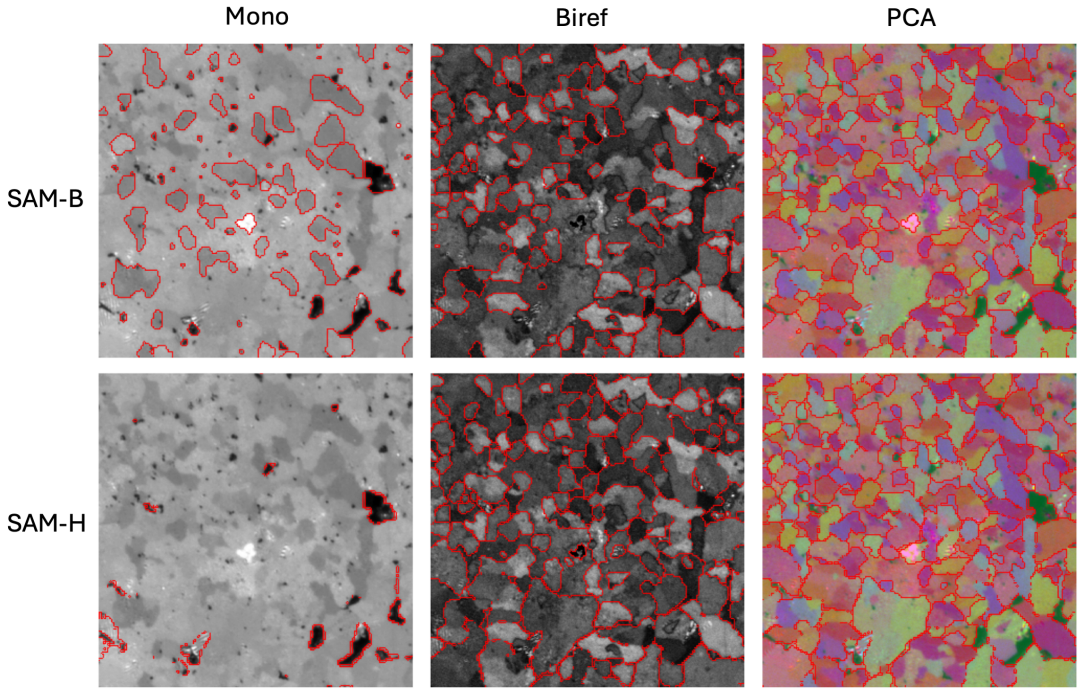



Figure 17: Comparison of preprocessing strategies for SAM segmentation. Rows correspond to SAM-B and SAM-H, while columns show (i) monochrome images; (ii) bireflectance images; and (iii) PCA images.

### 6.2.4 Data Analysis with Classical Methods

Classical stereological and intercept-based descriptors were extracted from the grain-boundary map and from binary grain masks predicted by the SAM models.

From all binary grain masks, we computed the mean lineal intercepts $\bar{L}(\alpha)$; rose diagrams and ellipse fits for anisotropy (major/minor intercepts, $L_a$, $L_b$); and intercept-length distributions [28, 40].

Bireflectance, defined as the maximal intensity difference across all recorded rotation angles, $\sigma$, and corrected for the major/minor intercepts ($L_a$, $L_b$) [1], was also computed from all binary grain masks.

Porosity was estimated only from the grain-boundary map by counting pixels labeled as pores [42].

### 6.2.5 Fine-tuning SAM-B

PCA preprocessing yields descriptors that closely match those obtained from the grain-boundary map, while SAM-B underperforms. The lesser performance of SAM-B is also visible in, e.g., Figure 17.

As a case study and proof of principle for PEFT/LoRA-based fine-tuning, and in the absence of true binary grain masks, we fine-tune the adapter/LoRA attached to SAM-B to emulate SAM-H performance, using SAM-H binary grain masks as teacher targets in the training loop. From the SAM-H analysis, we extracted 17 279 single masks for training.

Concretely, we insert an input adapter before the first layer of the image encoder to accept polarization stacks of shape $B \times H \times W \times 6$, and apply a LoRA module in the mask decoder while keeping the backbone frozen [94].

Training proceeds for 10 epochs with a pixel-wise cross-entropy loss [105] (see Figure 18). Training, validation, and testing were done using an 80/10/10% train/validation/test split for the single masks. Note that when training with single masks, every single mask from the binary grain masks counts as a data entry, amounting to 17 279 masks. For full-field inference, we always use the entire dataset of tiles (120 data points).

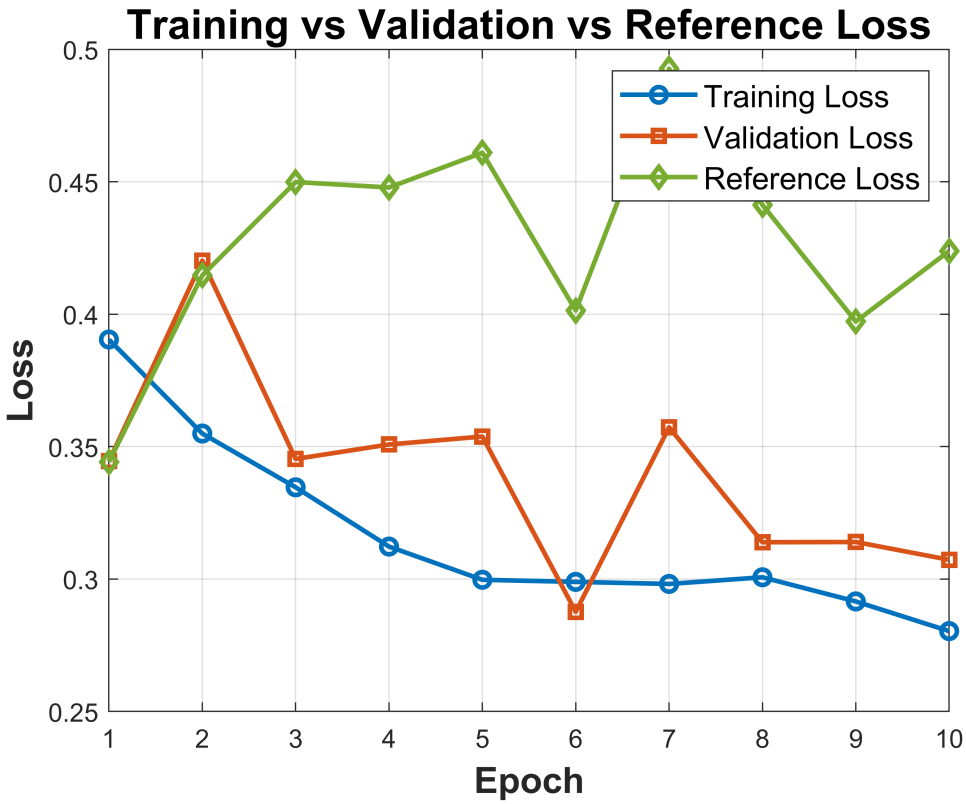



Figure 18: Training (blue) and validation (red) loss of SAM-$B_{PEFT/LoRA}$ as a function of epoch. The gap between curves reflects the degree of generalization. The reference loss (green) is also indicated, which is the loss of the native SAM-B.

## 6.3 Results

### 6.3.1 Classical Descriptors from the Grain-boundary Maps

**6.3.1.1 Porosity:** We calculated porosity maps from the entire dataset, as summarized in Table 5. For an example, see Figure 19.

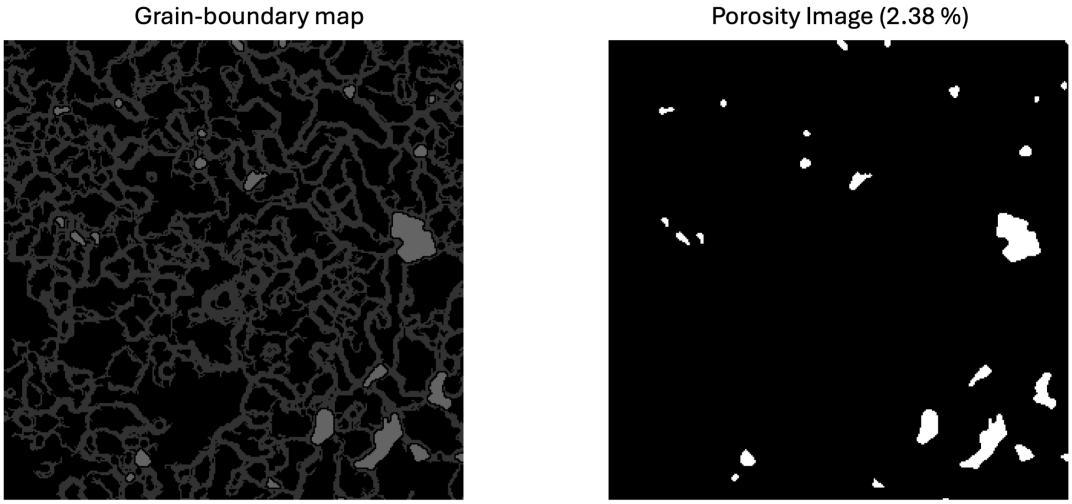



Figure 19: Porosity map (right), with pores (white), derived from the grain-boundary map (left), with pores (light gray), resin/matrix (dark gray), and grains (black).

**6.3.1.2 Intercept Distributions:** Mean lineal intercepts $\bar{L}(\alpha)$, rose plots/ellipse fits for anisotropy (major/minor intercepts $L_a, L_b$) (see Figure 20), and intercept–length distributions were analyzed (see Figure 21). Results are reported in Table 5.

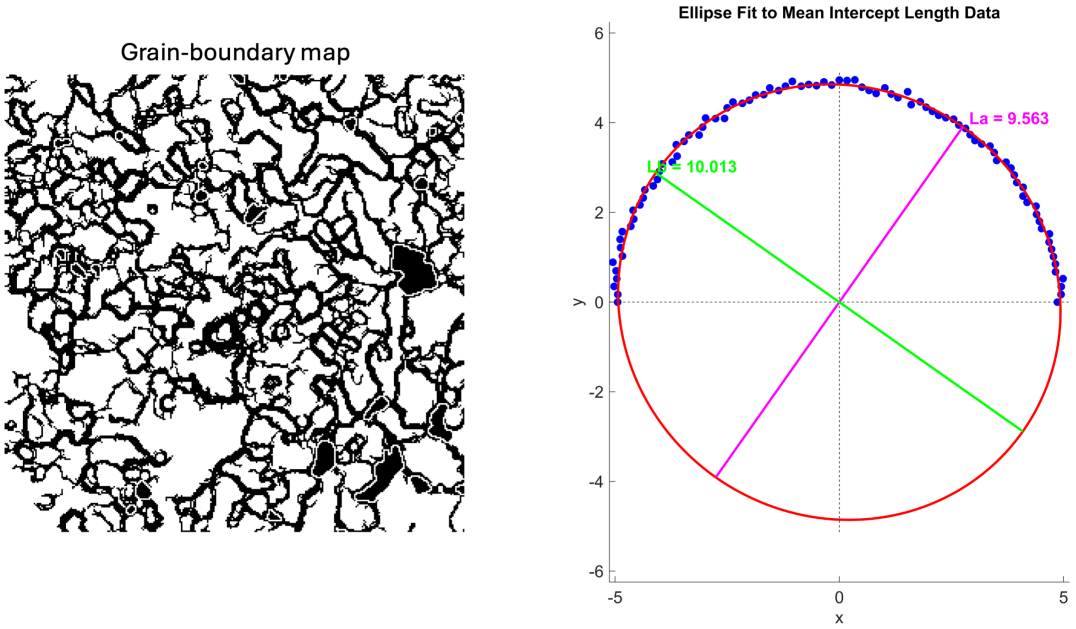



Figure 20: Grain-boundary map used for intercept analysis (left). Rose plot with major/minor intercepts $L_a, L_b$ (right).

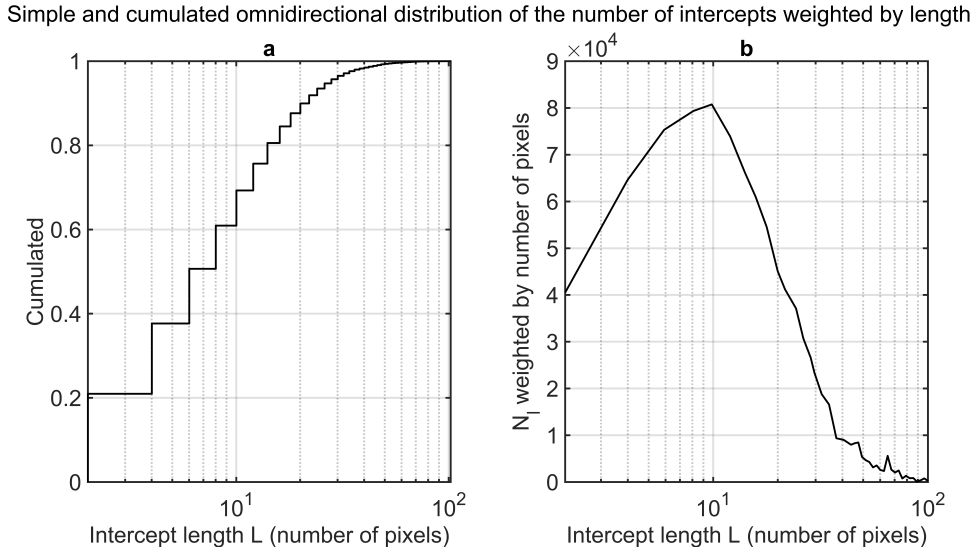



Figure 21: Simple and cumulative omnidirectional intercept-length distributions as calculated from the grain-boundary maps. These distributions provide estimates of characteristic grain dimensions.

**6.3.1.3 Bireflectance:** We calculated bireflectance values from the entire dataset, as summarized in Table 5. For an example, see Figure 22.

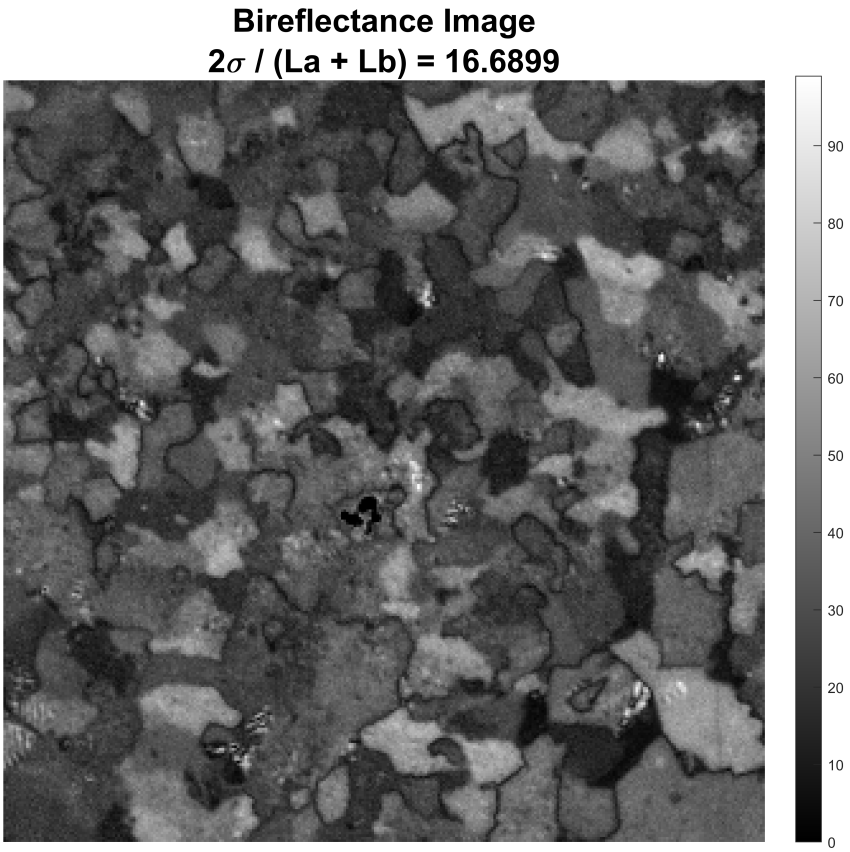



Figure 22: Bireflectance image derived from the intensity difference $\sigma$ across all recorded rotation angles, with correction for the major/minor intercepts $L_a, L_b$.

Table 5: Summary statistics of classical descriptors as extracted from all tiles using the grain-boundary map.

| Parameter | Min | Max | Median | Mean | Std |
|---|---|---|---|---|---|
| Porosity (pct) | 0.7 | 11.4 | 3.48 | 4.13 | 2.1 |
| La (px) | 7.02 | 12.48 | 9.2 | 9.36 | 1.18 |
| Lb (px) | 7.08 | 12.85 | 9.55 | 9.73 | 1.27 |
| Bireflectance (-) | 11.18 | 29.61 | 19 | 19.57 | 3.84 |

### 6.3.2 Classical Descriptors from the SAM Models

**6.3.2.1 Intercept Distributions:** Starting from the full-field binary grain masks (as opposed to the grain-boundary maps from the previous step; see Figure 23), mean lineal intercepts $\bar{L}(\alpha)$, rose plots/ellipse fits for anisotropy (major/minor intercepts $L_a, L_b$) (see Figure 24), and intercept–length distributions were analyzed (see Figure 25). For an overview of all methods, see Figure 26.

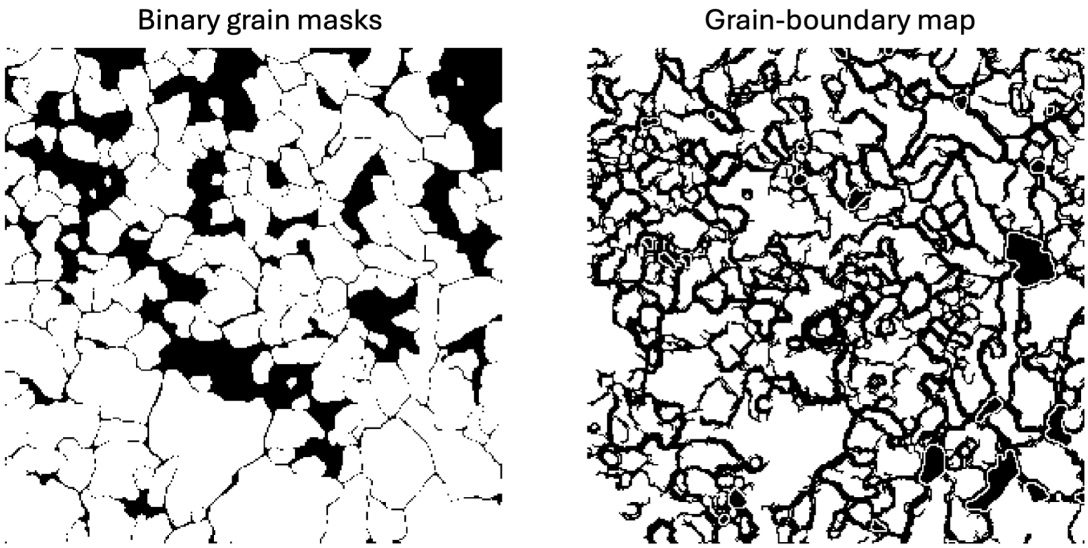



Figure 23: Comparison between the binary grain mask derived from PCA-preprocessed SAM-H segmentation (left) and the classical grain-boundary map constructed with edge detection (right).

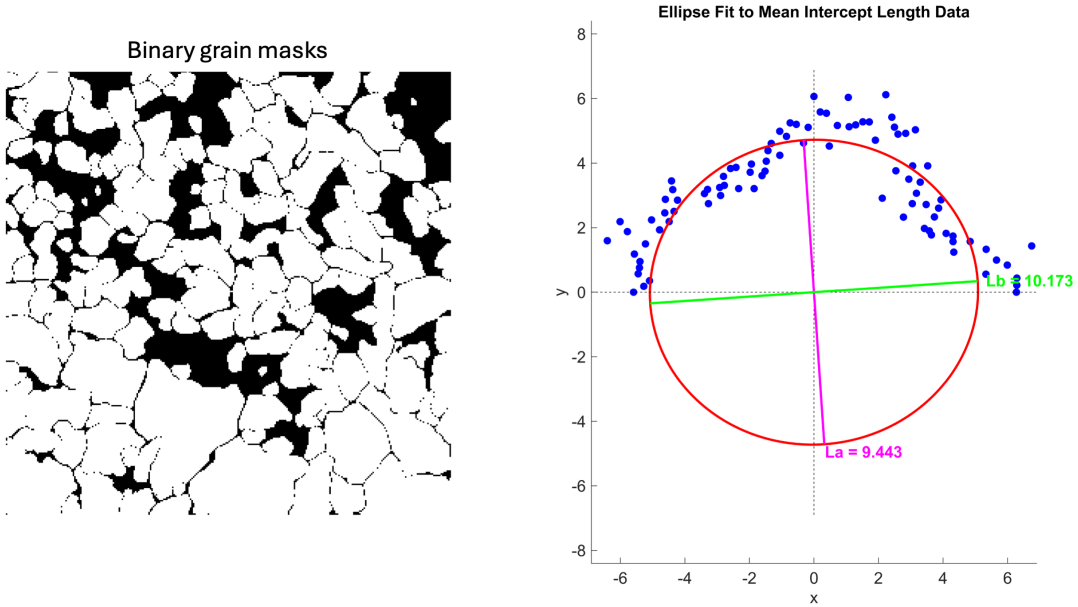



Figure 24: Binary grain mask used for intercept analysis based on PCA-preprocessed SAM-H segmentation (left); rose plot with major/minor intercepts $L_a, L_b$ (right).

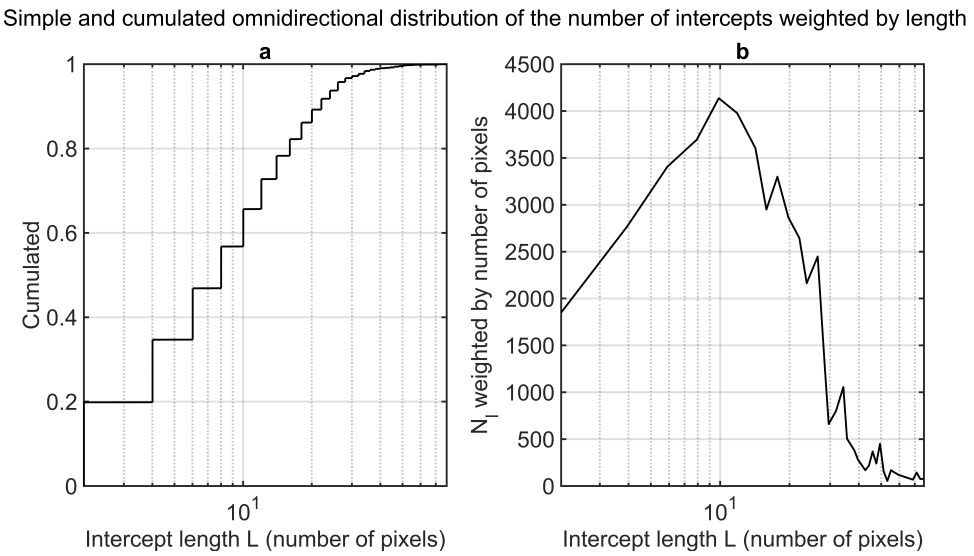



Figure 25: Simple and cumulative omnidirectional intercept-length distributions as calculated from the PCA-preprocessed SAM-H binary grain mask. These distributions provide estimates of characteristic grain dimensions.

**6.3.2.2 Bireflectance:** We calculated bireflectance values for the entire dataset, as summarized in Table 6 (for SAM-H and PCA only). For an overview of all methods, see Figure 26.

Table 6: Summary statistics of parameters extracted from all tiles using SAM-H and PCA preprocessing.

| Parameter | Min | Max | Median | Mean | Std |
|---|---|---|---|---|---|
| La (px) | 6.77 | 28.53 | 12.92 | 13.41 | 3.39 |
| Lb (px) | 8.72 | 30.83 | 14 | 14.77 | 3.9 |
| Bireflectance (-) | 5.46 | 26.92 | 13.33 | 13.8 | 3.79 |

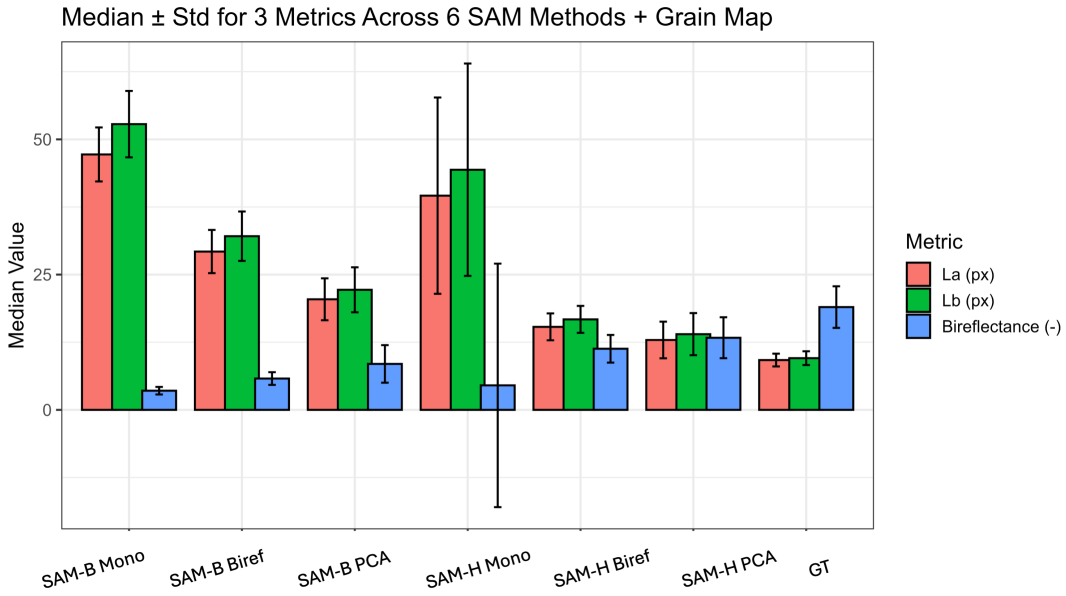



Figure 26: Comparison of the median values of all quantitative descriptors derived from SAM (across backbones and preprocessing methods) and from the grain-boundary map (GT). Error bars indicate inter-tile variability.

### 6.3.3 Full-field and Single-mask Segmentation

Here, we report the segmentation performance of native SAM-B compared to SAM-$B_{PEFT/LoRA}$, using quantitative pixel-wise metrics (IoU, Dice, precision, and recall). The results of the fine-tuning on single masks are illustrated in Figure 27 and summarized in Table 7.

Full-field inference is illustrated in Figure 28 and reported in Table 8. The metrics are computed using binary grain masks predicted by SAM-H and PCA preprocessing as the pseudo–ground truth.

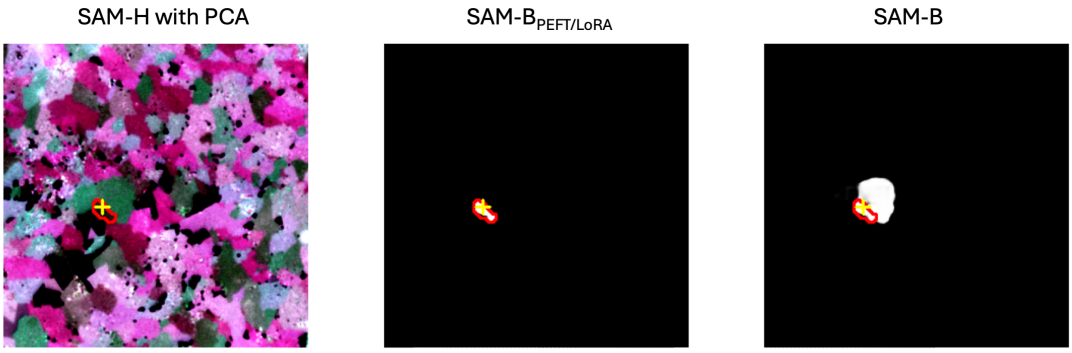



Figure 27: Mask-level improvement achieved through SAM fine-tuning. Individual mask segmentation of SAM-H and PCA preprocessing (left), SAM-B$_{\text{PEFT/LoRA}}$ (middle), and native SAM-B (right).

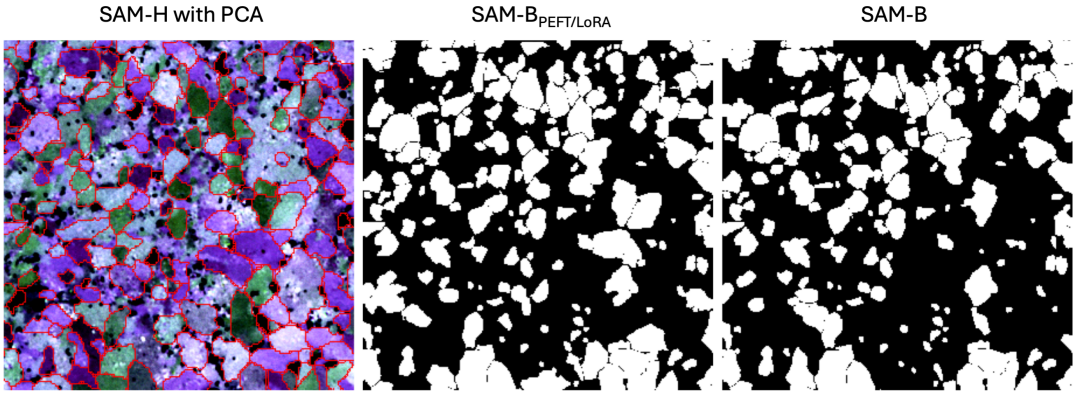



Figure 28: Full-field segmentation improvement achieved through SAM fine-tuning. Full-field binary grain masks as predicted by SAM-H and PCA preprocessing (left), SAM-B$_{\text{PEFT/LoRA}}$ (middle), and native SAM-B (right).

Table 7: Summary statistics for SAM-$B_{PEFT/LoRA}$ and SAM-B (Single Mask)

| | SAM-$B_{PEFT/LoRA}$ | | SAM-B | |
|---|---|---|---|---|
| Metric | Median | Std | Median | Std |
| IOU | 0.730 | 0.234 | 0.632 | 0.280 |
| DICE | 0.844 | 0.204 | 0.774 | 0.267 |
| PREC | 0.902 | 0.232 | 0.775 | 0.316 |
| REC | 0.907 | 0.200 | 0.899 | 0.201 |

Table 8: Summary statistics for SAM-$B_{PEFT/LoRA}$ and SAM-B (Full Field)

| | SAM-$B_{PEFT/LoRA}$ | | SAM-B | |
|---|---|---|---|---|
| Metric | Median | Std | Median | Std |
| IOU | 0.570 | 0.097 | 0.530 | 0.112 |
| DICE | 0.726 | 0.087 | 0.693 | 0.109 |
| PREC | 0.841 | 0.054 | 0.859 | 0.062 |
| REC | 0.632 | 0.100 | 0.576 | 0.116 |

## 6.4 Discussion and Conclusion

From a qualitative evaluation (see Figure 17), SAM segments individual grains reasonably well, likely rivaling human performance when appropriate preprocessing is used. It is therefore unsurprising that the descriptors calculated from binary grain masks predicted by SAM-H and PCA preprocessing are consistent with those obtained from the grain-boundary map (see Figure 26).

However, SAM's best configuration appears to slightly overestimate particle size, likely due to aggressive downsampling and built-in mask thresholds (including hard cutoffs on mask area). By contrast, the grain-boundary maps contain considerable noise and spurious boundaries, which propagate into stereological measurements. Moreover, grain-boundary maps are a different data modality than the binary grain masks predicted by SAM and therefore present inherently different inputs to classical algorithms (see Figure 23).

SAM-B underperforms, both qualitatively (see Figure 17) and in reproducing classical descriptors (see Figure 26). As a case study and proof of principle for PEFT/LoRA-based fine-tuning, and in the absence of true binary grain masks, we fine-tuned SAM-B to emulate SAM-H, using SAM-H binary grain masks as teacher targets. The results indicate improvements at both the prompted single-mask level (see Figure 27, Table 7) and the full-field level (see Figure 28, Table 8). This provides an important proof of concept for mitigating domain shift and data scarcity and can also be extended to fine-tuning in combination with SAM-H, provided that ground-truth masks and sufficient computing power are available. In addition, the PEFT setup allows loading, training, and inference on stacks of $N$ polarized images with shape $B \times H \times W \times N$, giving the optimizer access to all available information in the stack.

In conclusion, for RLM segmentation in mineralogy, appropriate model choice is critical, as illustrated by the performance gap between SAM encoders. In addition, mineralogically informed preprocessing is advisable, since results improve markedly with steps

such as bireflectance or PCA before input to SAM. PEFT/LoRA can further enhance performance. These findings support a workflow in which foundation models are first guided by preprocessing and then selectively adapted, rather than fully retrained.

The findings in this chapter should nevertheless be interpreted with care, especially regarding PEFT/LoRA adaptation. A true ground truth was not available for this dataset, so one model was effectively trained on targets produced by another model. Moreover, the gains are incremental. This likely reflects the need for more aggressive adaptation, for example, introducing adapters or LoRA modules in the encoder and MLP heads, and the inherent limitations of the ViT-B encoder.

# 7 Grain Detection in Solar Panel Reflected-light Microscopy Images

## 7.1 Introduction

This chapter extends the application of SAM-based grain segmentation to three datasets of solar panels that differ in the proportions and size distributions of silicon and glass grains. The datasets face the particular challenge of glass grain reflectivities being close to that of the resin background, but binary grain masks are provided by an automated mineralogy BSE system.

The objective is to assess whether SAM, using different encoders that can be fine-tuned with a limited number of annotated examples, can reliably perform automatic grain detection across the datasets and support the extraction of grain-scale descriptors commonly used in quantitative analysis.

Classical image-analysis methods based on ground-truth masks serve as the reference for comparing grain-scale descriptors, and quantitative performance metrics (IoU, Dice, precision, and recall) are computed with respect to the ground truth.

## 7.2 Materials and Methods

### 7.2.1 The Dataset

Three independent datasets of solar-panel sections are considered in this study. Each dataset consists of 8-bit RGB RLM images which were paired, cut and resampled with a custom-made algorithm to match automatically annotated 8-bit RGB ground-truth annotations, with `resin` $= [0, 0, 0]$; the color encodes the mineral type (see Figure 29). For this pipeline, we convert the masks into binary grain masks in which `resin`=0 and `grain`=1. The datasets differ in grain morphology, size distribution, and spatial density. The resolution is $1024 \times 768$ pixels, which can be fed into SAM on our current workstation, yielding a total workable dataset of 131 images per dataset.

| | |
|---|---|
| Si | 0;128;255 |
| Glass (Verre) | 128;64;64 |
| Al | 0;0;64 |
| AlSiCaO | 0;0;255 |
| AlSi(O) | 0;255;255 |
| Silver | 0;255;0 |
| AgPb | 176;216;255 |
| Al(O) | 255;0;0 |
| SiO | 174;174;0 |
| Barite | 255;0;255 |
| Fe | 128;128;255 |
| Pyrite | 255;255;0 |
| Oxyde de fer | 64;128;128 |
| CuZn | 0;166;0 |
| Cu | 255;128;64 |
| AgAlO | 128;128;0 |
| FeCrO | 0;170;170 |
| FeMnO | 202;255;255 |
| TiCaSiO | 0;64;128 |
| FeTiO | 70;0;140 |
| TiO | 128;0;255 |
| Sn (Pb) | 128;0;255 |
| PbO | 64;0;0 |
| FeCaAlO | 128;128;192 |
| FeCaO | 128;128;192 |
| FTi | 255;255;206 |
| NaCaSiO(Al) - Plagioclase? | 255;0;128 |
| CaO | 128;64;64 |

Figure 29: Mineral-class color map used for constructing the binary grain masks. Each row lists the class name and its RGB value $[R, G, B]$ as exported by the automated mineralogy BSE system.

Representative examples from the three datasets are shown in Figures 30–32, illustrating the diversity of grain shapes and spatial organization encountered in the solar-panel sections.

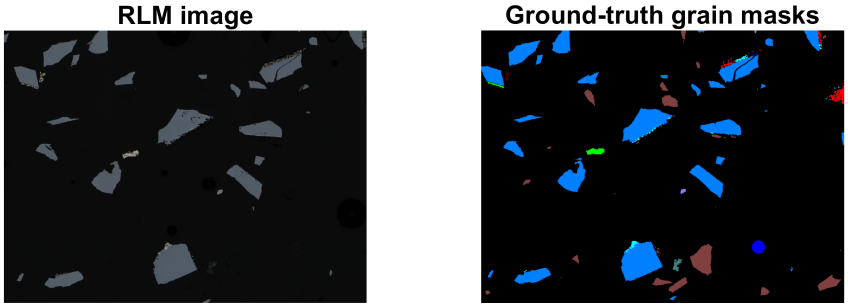



Figure 30: Optical reflected microscopy (left) and ground-truth annotations (right) of dataset 1.

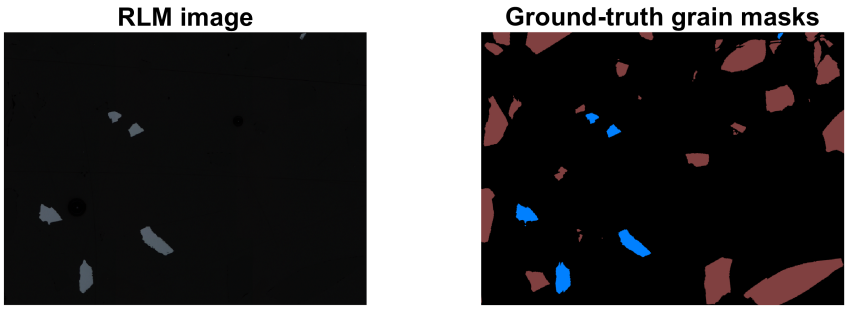



Figure 31: Optical reflected microscopy (left) and ground-truth annotations (right) of dataset 2.

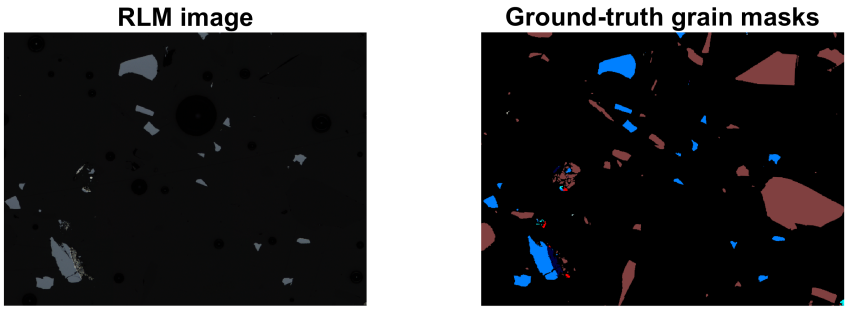



Figure 32: Optical reflected microscopy (left) and ground-truth annotations (right) of dataset 3.

### 7.2.2 Fine-tuning SAM-B for Automatic Grain Detection

In the previous chapter, we explored PEFT/LoRA fine-tuning of SAM-B because no training masks were available. SAM-H already produced strong results; unlike SAM-B, its outputs were good enough to serve as targets for supervising SAM-$B_{PEFT/LoRA}$ training. In the present study, however, we have true binary grain masks and observe full-field performance of SAM-B that almost matches that of SAM-H (see Figure 33 for an example, and Tables 12–14). For fine-tuning, we nevertheless resort to SAM-B, which is more computationally efficient.

SAM was fine-tuned independently on each dataset using the same PEFT/LoRA strategy as in the previous chapter. However, since the dataset is not a stack of polarized imaged, there is no adapter before the model; we feed the RGB images directly into SAM.

Training and validation loss curves for the three datasets are shown in Figures 34–36. In all cases, convergence is achieved within a limited number of epochs, indicating effective adaptation to the target domain.

Since binary grain masks are available in this dataset, we fine-tune SAM-B directly on these masks, using pixel-wise cross-entropy loss [105]. Training, validation, and testing were performed using an 80/10/10% train/validation/test split. Note that training was done with single masks: each individual mask from the ground-truth binary grain masks counts as a data entry, amounting to 1 416, 1 165, and 1 603 masks for datasets 1, 2, and 3, respectively. For full-field inference, we always use the entire dataset of tiles (131 data points per dataset).

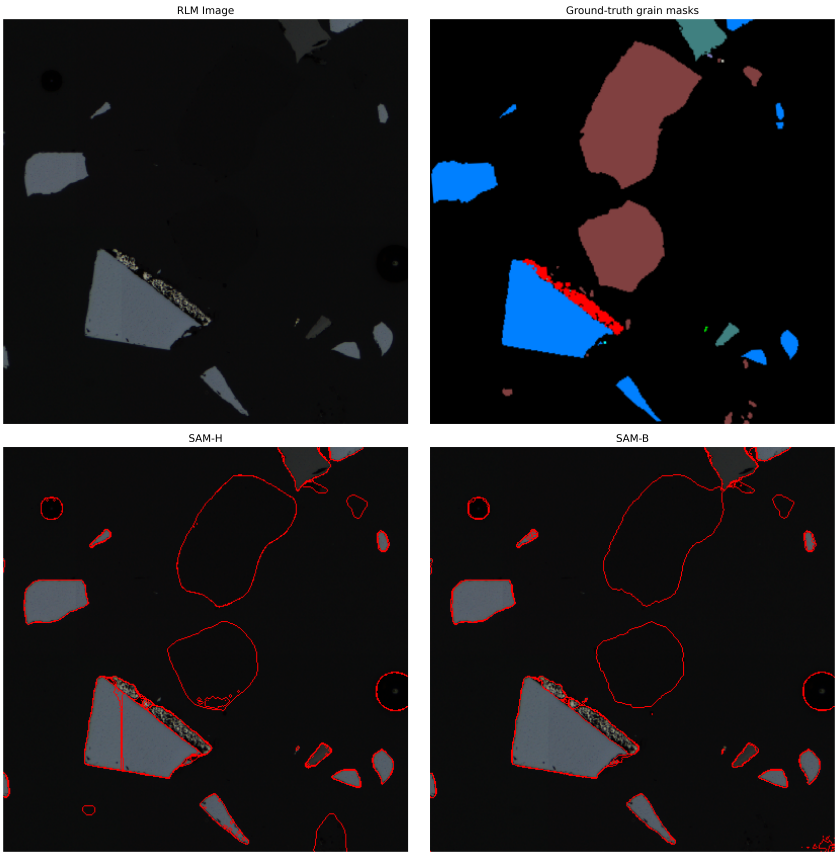



Figure 33: Original RLM image (upper left), ground-truth grain masks (upper right), and the segmentation performance of native SAM-H (lower left) and SAM-B (lower right). Predicted grain contours are shown in red and overlaid on the original RLM image.

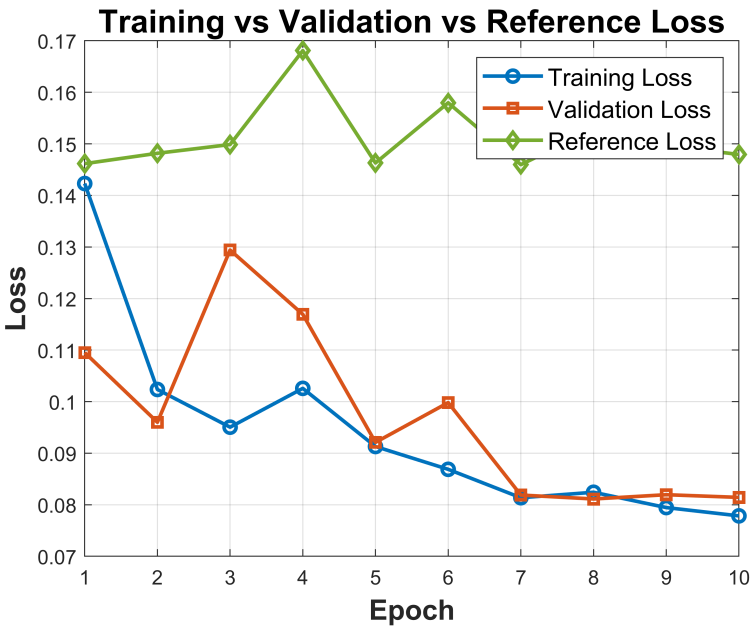



Figure 34: Training (blue) and validation (red) loss of SAM-B$_{\text{PEFT/LoRA}}$ as a function of epoch. The reference loss (green) is also indicated, which is the loss of native SAM-B. Results from dataset 1.

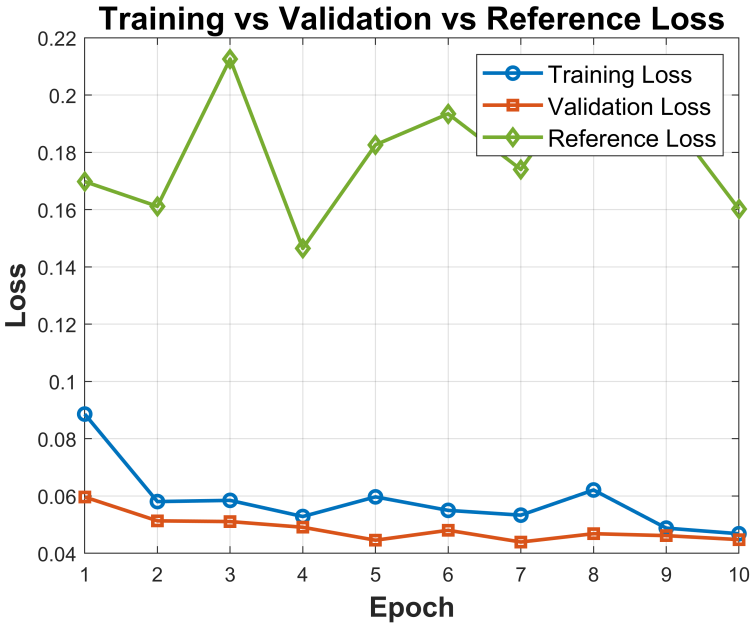



Figure 35: Training (blue) and validation (red) loss of SAM-B$_{\text{PEFT/LoRA}}$ as a function of epoch. The reference loss (green) is also indicated, which is the loss of native SAM-B. Results from dataset 2.

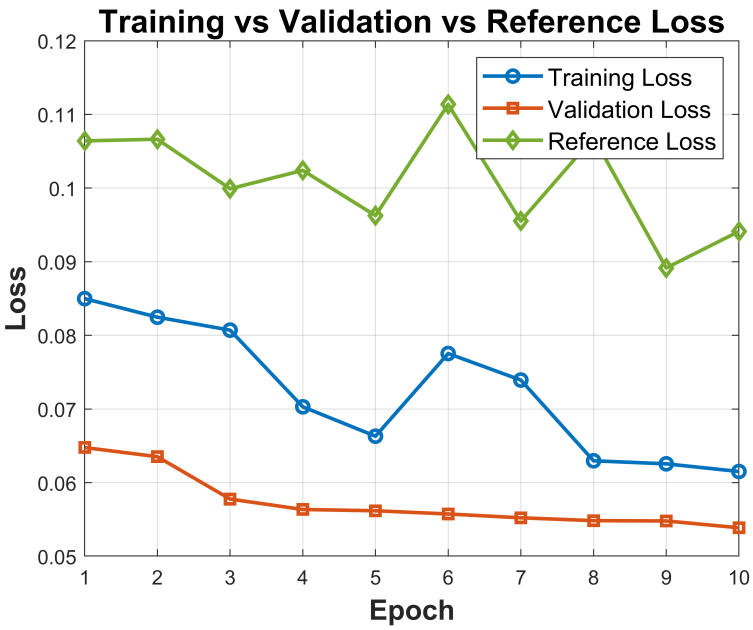



Figure 36: Training (blue) and validation (red) loss of SAM-$B_{PEFT/LoRA}$ as a function of epoch. The reference loss (green) is also indicated, which is the loss of native SAM-B. Results from dataset 3.

### 7.2.3 Experimental Design

Once PEFT/LoRA adaptation is complete, we work with three SAM variants: SAM-B, SAM-$B_{PEFT/LoRA}$, and SAM-H.

Using the binary grain masks, we evaluate full-field inference of these three models over the entire dataset by computing IoU, Dice, precision, and recall between the predicted binary grain masks and the ground truth.

Second, we apply classical lineal-intercept methods to compare the major and minor intercepts, $L_a$ and $L_b$, as well as the intercept-length distributions, derived from the predicted and ground-truth binary grain masks.

For consistency with the rest of this work, and because it is part of the analysis pipeline, we report rose diagrams of line intercepts and the major/minor intercepts ($L_a$, $L_b$). However, because the samples in these datasets were ground and polished, these orientation measures have limited meaning from a mineralogy or materials-science point of view. All measurements were computed in pixel units because the microscope calibration (pixels-per-$\mu$m) was not available.

## 7.3 Results

### 7.3.1 Full-field and Single-mask Segmentation

The results of SAM-$B_{PEFT/LoRA}$ and native SAM-B on prompted single-mask segmentation are illustrated in Figure 37 and reported in Tables 9–11.

Binary grain masks predicted by full-field inference of SAM-H, SAM-$B_{PEFT/LoRA}$, and native SAM-B are illustrated in Figure 38 and reported in Tables 12–14.


Figure 37: Mask-level improvement achieved through SAM fine-tuning. Ground-truth binary grain mask and prompted single mask indicated in red (left), SAM-$B_{PEFT/LoRA}$ (middle), and native SAM-B (right).


Figure 38: Full-field segmentation improvement achieved through SAM fine-tuning. Ground-truth binary grain masks (left), SAM-$B_{PEFT/LoRA}$ (middle), and native SAM-B (right).

Table 9: Summary statistics for SAM-$B_{PEFT/LoRA}$and SAM-B (Single Mask) on dataset 1.

| | SAM-$B_{PEFT/LoRA}$ | | SAM-B | |
|---|---|---|---|---|
| Metric | Median | Std | Median | Std |
| IOU | 0.922 | 0.120 | 0.878 | 0.093 |
| DICE | 0.960 | 0.085 | 0.935 | 0.057 |
| PREC | 0.940 | 0.046 | 0.881 | 0.091 |
| REC | 0.996 | 0.126 | 1.000 | 0.034 |

Table 10: Summary statistics for SAM-$B_{PEFT/LoRA}$ and SAM-B (Single Mask) on dataset 2.

| | SAM-$B_{PEFT/LoRA}$ | | SAM-B | |
|---|---|---|---|---|
| Metric | Median | Std | Median | Std |
| IOU | 0.975 | 0.038 | 0.968 | 0.063 |
| DICE | 0.987 | 0.020 | 0.984 | 0.036 |
| PREC | 0.994 | 0.038 | 0.988 | 0.065 |
| REC | 0.990 | 0.027 | 0.995 | 0.022 |

Table 11: Summary statistics for SAM-$B_{PEFT/LoRA}$ and SAM-B (Single Mask) on dataset 3.

| | SAM-$B_{PEFT/LoRA}$ | | SAM-B | |
|---|---|---|---|---|
| Metric | Median | Std | Median | Std |
| IOU | 0.969 | 0.079 | 0.949 | 0.120 |
| DICE | 0.984 | 0.049 | 0.974 | 0.078 |
| PREC | 0.980 | 0.083 | 0.950 | 0.120 |
| REC | 0.990 | 0.007 | 0.999 | 0.003 |

Table 12: Summary statistics for SAM-H, SAM-$B_{PEFT/LoRA}$ and SAM-B (Full Field) on dataset 1.

| | SAM-H | | SAM-$B_{PEFT/LoRA}$ | | SAM-B | |
|---|---|---|---|---|---|---|
| Metric | Median | Std | Median | Std | Median | Std |
| IOU | 0.899 | 0.057 | 0.863 | 0.069 | 0.856 | 0.070 |
| DICE | 0.930 | 0.043 | 0.927 | 0.042 | 0.922 | 0.043 |
| PREC | 0.921 | 0.053 | 0.889 | 0.072 | 0.864 | 0.070 |
| REC | 0.982 | 0.012 | 0.975 | 0.015 | 0.986 | 0.016 |

Table 13: Summary statistics for SAM-H, SAM-$B_{PEFT/LoRA}$ and SAM-B (Full Field) on dataset 2.

| | SAM-H | | SAM-$B_{PEFT/LoRA}$ | | SAM-B | |
|---|---|---|---|---|---|---|
| Metric | Median | Std | Median | Std | Median | Std |
| IOU | 0.925 | 0.132 | 0.826 | 0.146 | 0.942 | 0.037 |
| DICE | 0.912 | 0.101 | 0.905 | 0.097 | 0.970 | 0.020 |
| PREC | 0.953 | 0.142 | 0.839 | 0.150 | 0.953 | 0.035 |
| REC | 0.985 | 0.002 | 0.982 | 0.005 | 0.987 | 0.005 |

Table 14: Summary statistics for SAM-H, SAM-$B_{PEFT/LoRA}$ and SAM-B (Full Field) on dataset 3.

| | SAM-H | | SAM-$B_{PEFT/LoRA}$ | | SAM-B | |
|---|---|---|---|---|---|---|
| Metric | Median | Std | Median | Std | Median | Std |
| IOU | 0.902 | 0.042 | 0.891 | 0.054 | 0.886 | 0.055 |
| DICE | 0.957 | 0.025 | 0.942 | 0.032 | 0.940 | 0.033 |
| PREC | 0.936 | 0.062 | 0.929 | 0.058 | 0.919 | 0.058 |
| REC | 0.975 | 0.005 | 0.962 | 0.009 | 0.966 | 0.007 |

### 7.3.2 Classical Lineal-intercept Descriptors

**7.3.2.1 Ground-truth Binary Grain Masks:** Classical lineal-intercept descriptors were extracted directly from the ground-truth binary grain masks. Mean lineal intercepts $\bar{L}(\alpha)$, rose plots/ellipse fits for anisotropy (major/minor intercepts $L_a, L_b$) (see Figures 39–41), and intercept–length distributions were calculated (see Figures 42–44).

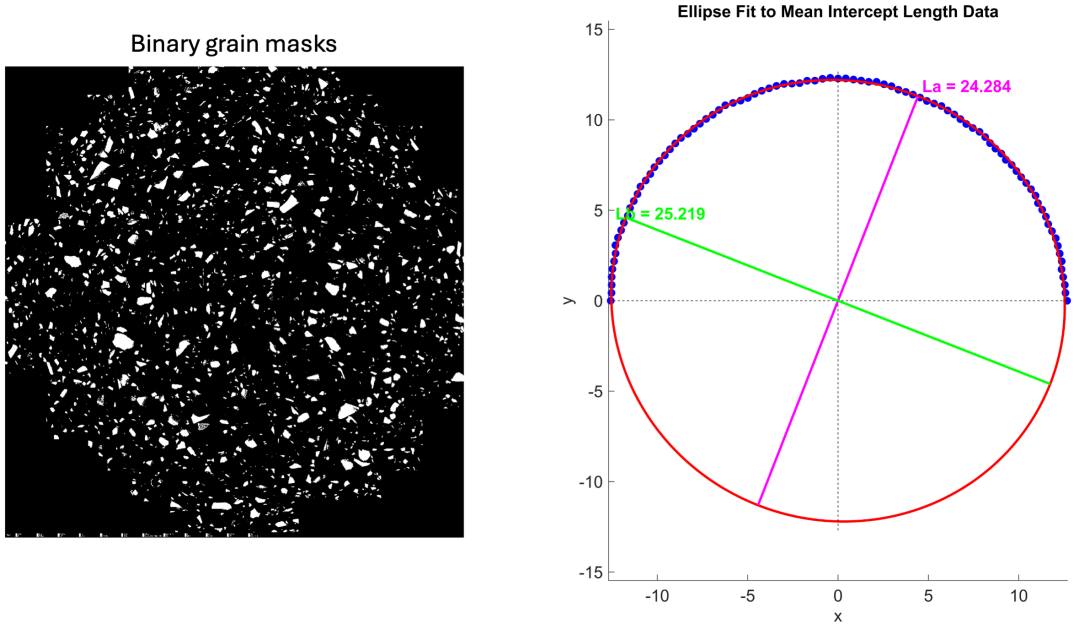



Figure 39: Binary grain masks (left); rose plot with major/minor intercepts $L_a, L_b$ (right). Obtained from the ground-truth binary grain masks in dataset 1.

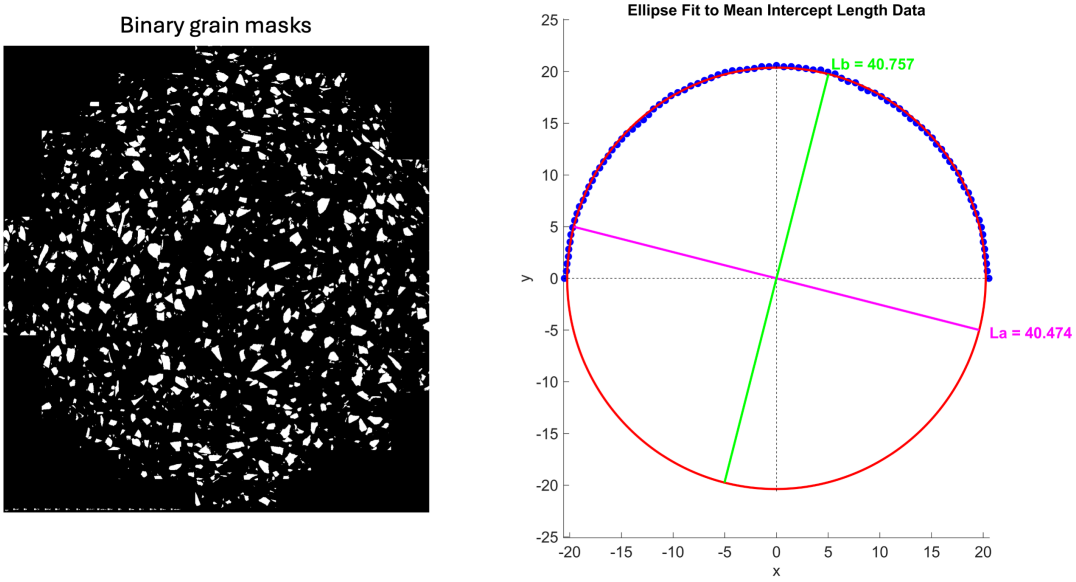



Figure 40: Binary grain masks (left); rose plot with major/minor intercepts $L_a, L_b$ (right). Obtained from the ground-truth binary grain masks in dataset 2.

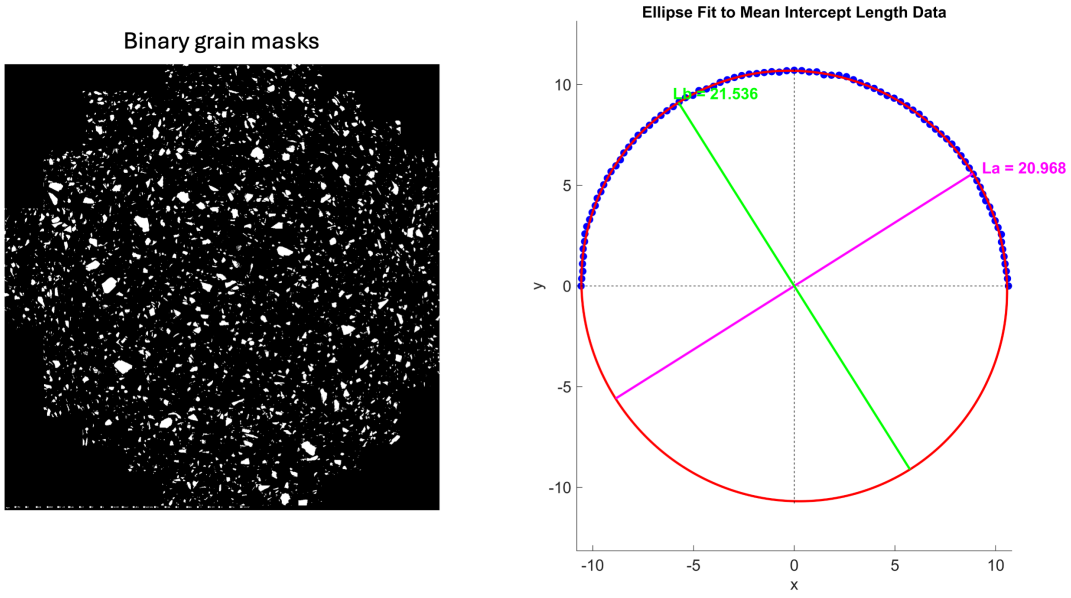



Figure 41: Binary grain masks (left); rose plot with major/minor intercepts $L_a, L_b$ (right). Obtained from the ground-truth binary grain masks in dataset 3.

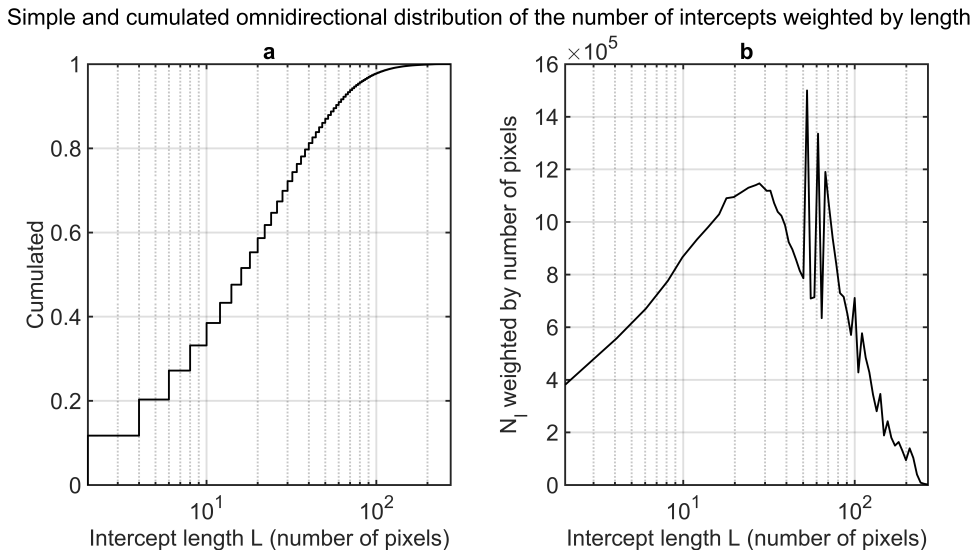



Figure 42: Simple and cumulative omnidirectional intercept-length distributions calculated from the binary grain masks. Obtained from the ground-truth binary grain masks in dataset 1.

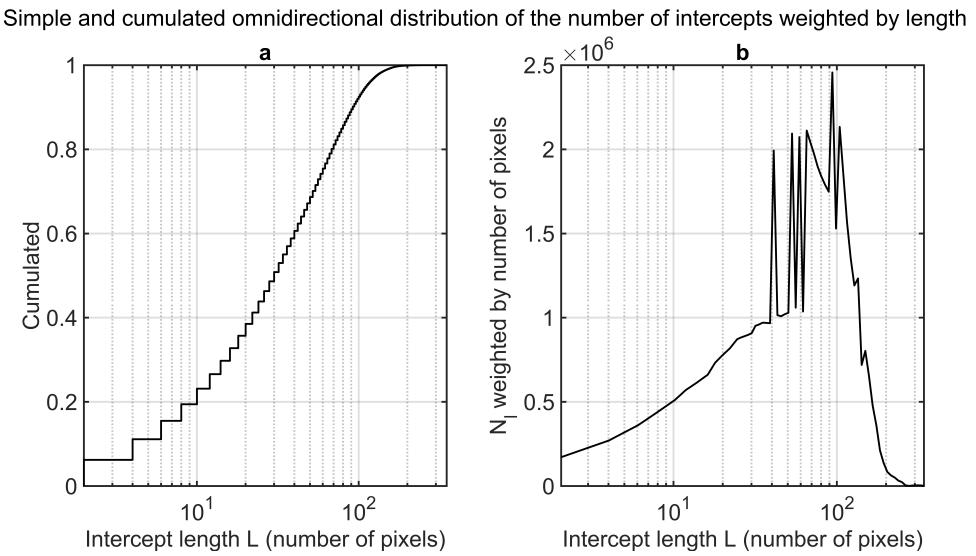



Figure 43: Simple and cumulative omnidirectional intercept-length distributions calculated from the binary grain masks. Obtained from the ground-truth binary grain masks in dataset 2.

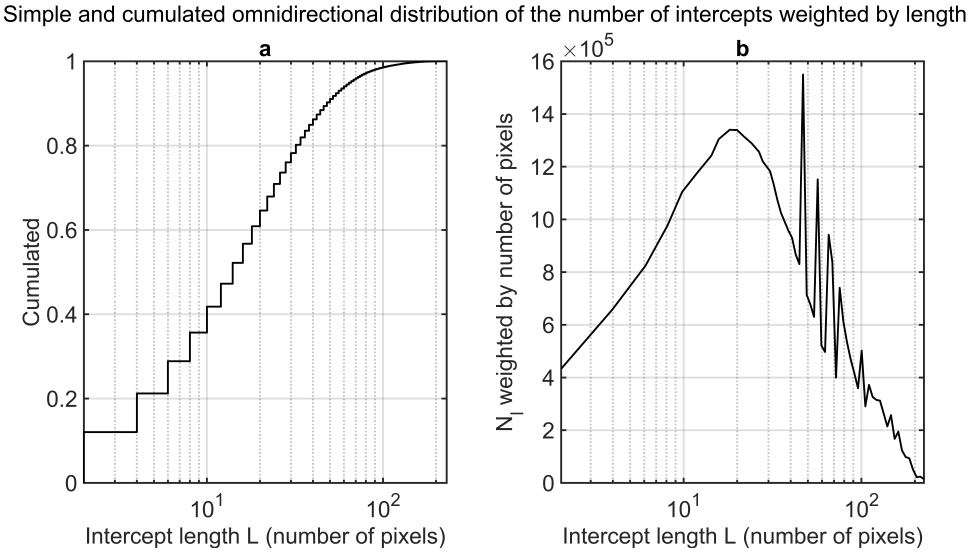



Figure 44: Simple and cumulative omnidirectional intercept-length distributions calculated from the binary grain masks. Obtained from the ground-truth binary grain masks in dataset 3.

**7.3.2.2 SAM Models:** The same analyses were repeated on binary grain masks predicted by the SAM models. Mean lineal intercepts $\bar{L}(\alpha)$, rose plots/ellipse fits for anisotropy (major/minor intercepts $L_a, L_b$) (see Figures 45–47), and intercept–length distributions were calculated (see Figures 48–50).

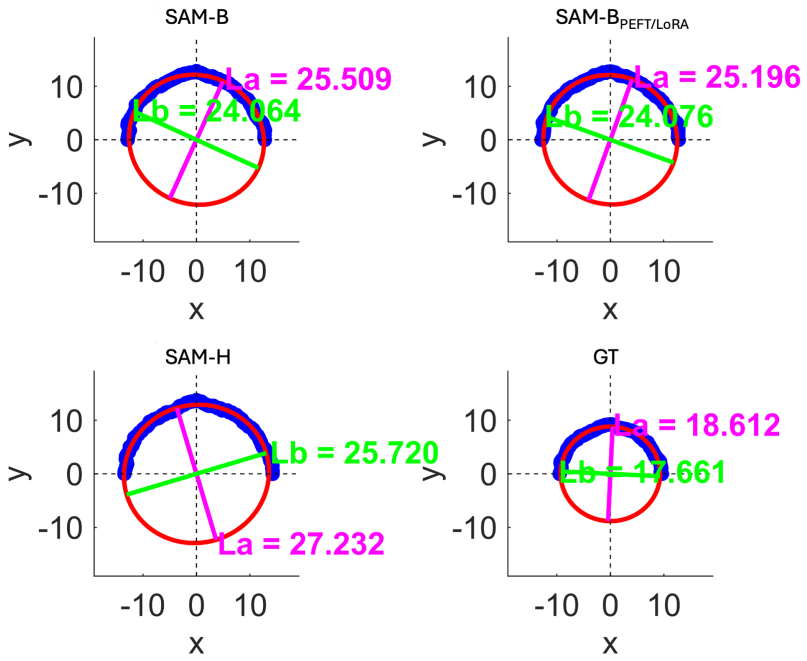



Figure 45: Rose plots with major/minor intercepts $L_a, L_b$. Obtained from native SAM-B (upper left), SAM-B$_{\text{PEFT/LoRA}}$ (upper right), native SAM-H (lower left), and the ground-truth binary grain masks (lower right). Obtained from dataset 1.

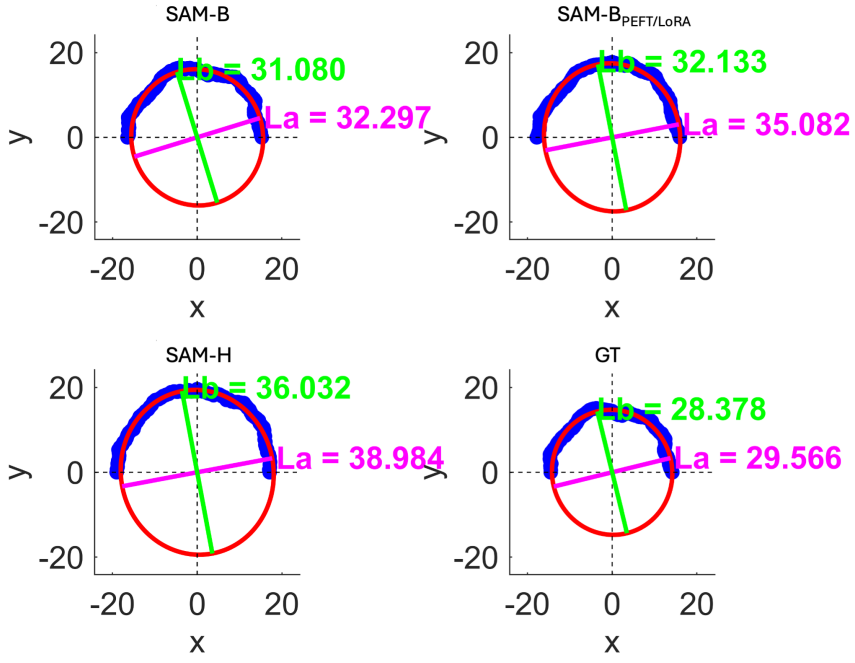



Figure 46: Rose plots with major/minor intercepts $L_a, L_b$. Obtained from native SAM-B (upper left), SAM-B$_{\text{PEFT/LoRA}}$ (upper right), native SAM-H (lower left), and the ground-truth binary grain masks (lower right). Obtained from dataset 2.

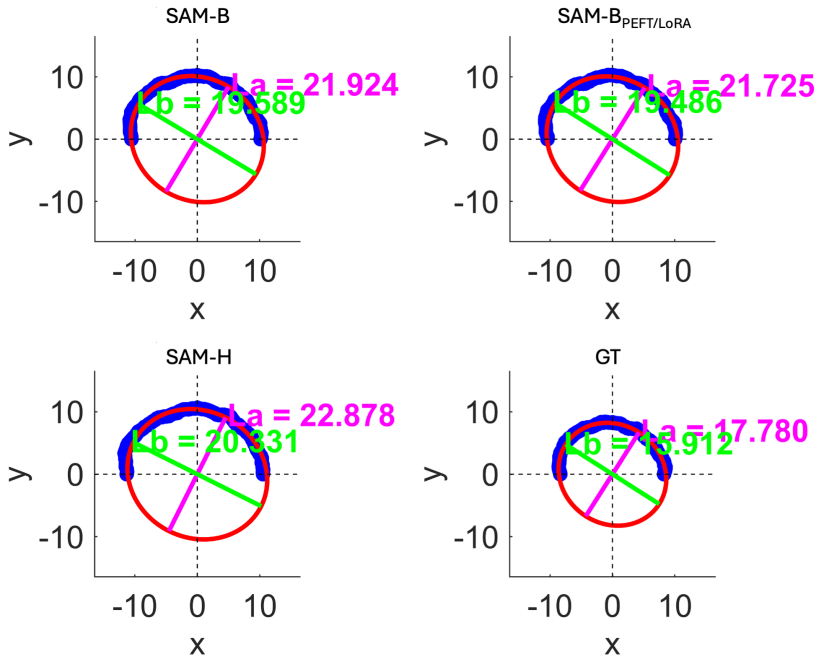



Figure 47: Rose plots with major/minor intercepts $L_a, L_b$. Obtained from native SAM-B (upper left), SAM-B$_{\text{PEFT/LoRA}}$ (upper right), native SAM-H (lower left), and the ground-truth binary grain masks (lower right). Obtained from dataset 3.

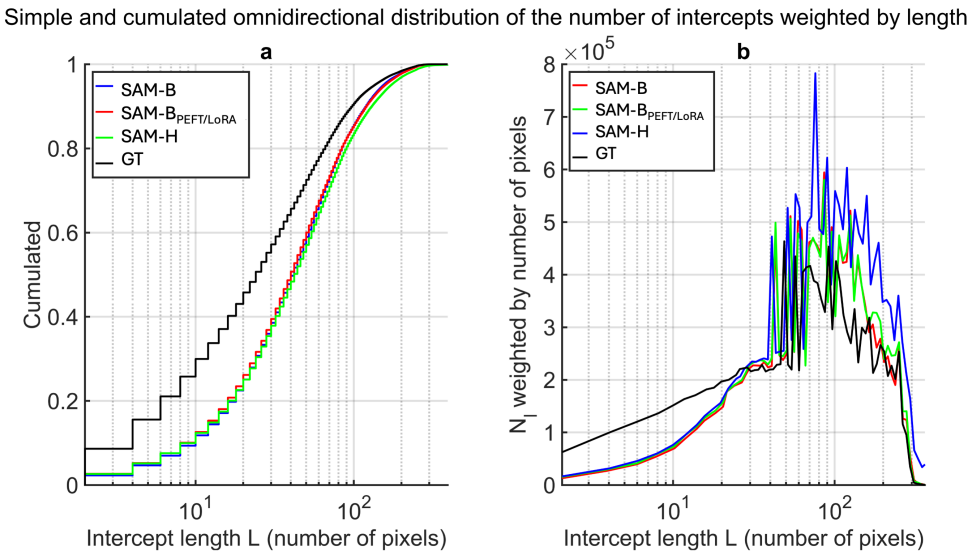



Figure 48: Simple and cumulative omnidirectional intercept-length distributions. Blue = native SAM-B, Red = SAM-B$_{\text{PEFT/LoRA}}$, Green = native SAM-H, and Black = ground-truth binary grain masks. Obtained from dataset 1.

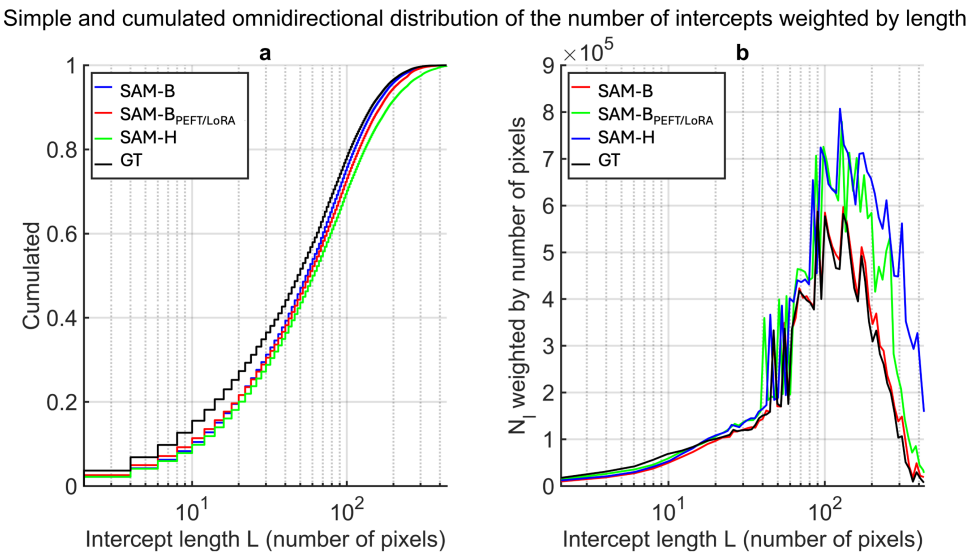



Figure 49: Simple and cumulative omnidirectional intercept-length distributions. Blue = native SAM-B, Red = SAM-B$_{\mathrm{PEFT/LoRA}}$, Green = native SAM-H, and Black = ground-truth binary grain masks. Obtained from dataset 2.

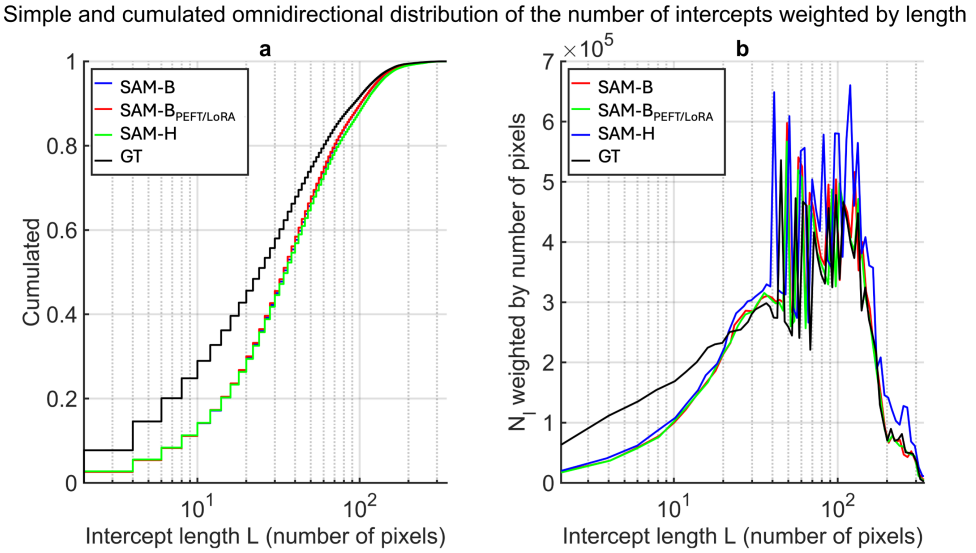



Figure 50: Simple and cumulative omnidirectional intercept-length distributions. Blue = native SAM-B, Red = SAM-B$_{\mathrm{PEFT/LoRA}}$, Green = native SAM-H, and Black = ground-truth binary grain masks. Obtained from dataset 3.

### 7.3.3 Major/minor Intercepts

A quantitative comparison of the major and minor intercepts ($L_a$, $L_b$) across the three datasets, computed from binary grain masks produced by the different SAM models and from the ground-truth binary grain masks, is presented in Figures 51–53. Data are reported as medians with standard deviations, calculated across tiles.

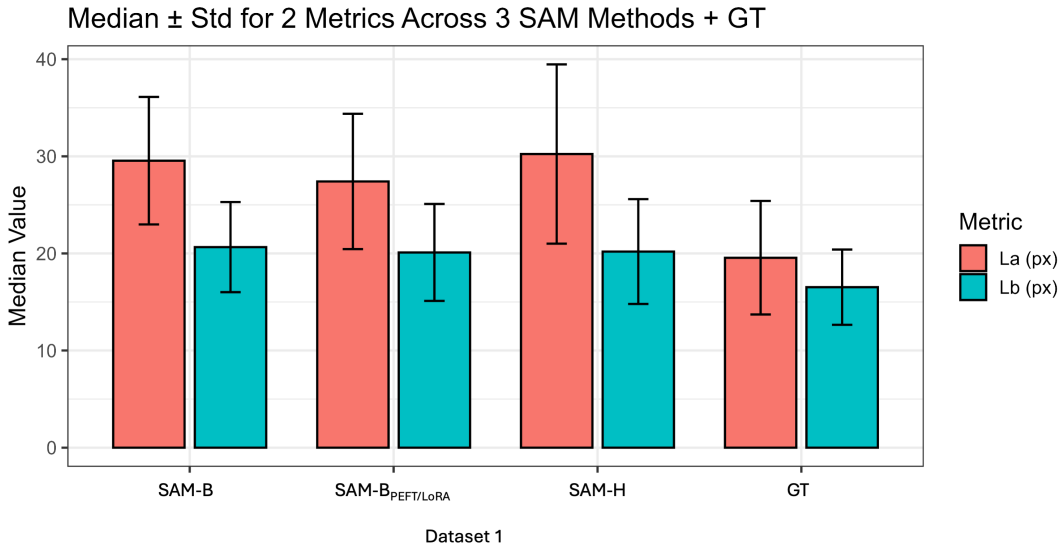



Figure 51: Comparison of major/minor intercepts $L_a, L_b$ obtained using different SAM methods and the ground-truth binary grain masks in dataset 1.

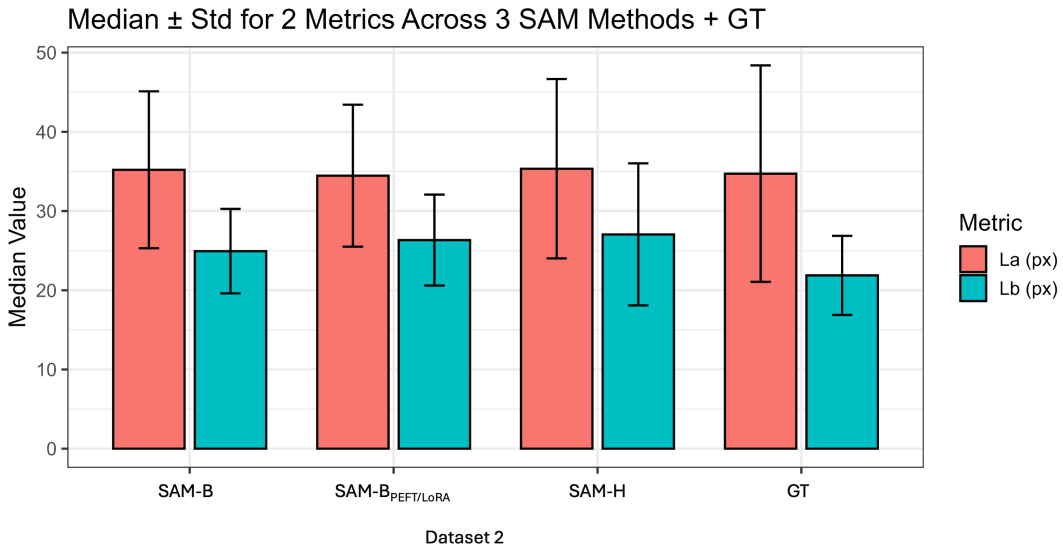



Figure 52: Comparison of major/minor intercepts $L_a, L_b$ obtained using different SAM methods and the ground-truth binary grain masks in dataset 2.

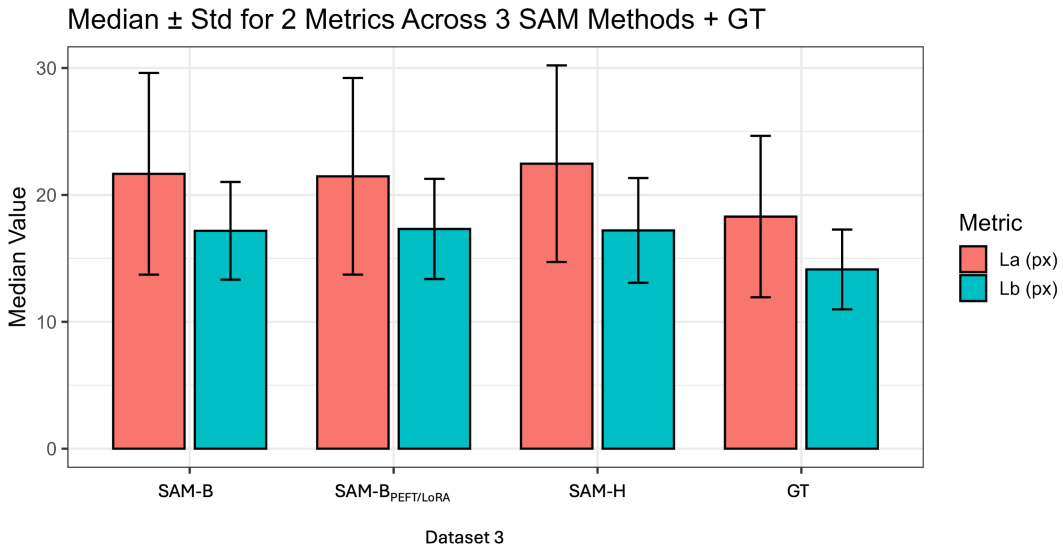



Figure 53: Comparison of major/minor intercepts $L_a, L_b$ obtained using different SAM methods and the ground-truth binary grain masks in dataset 3.

## 7.4 Discussion and Conclusion

This chapter demonstrates that SAM models can effectively perform automatic grain detection across all three solar-panel datasets, as evidenced by the metrics reported in Tables 12–14.

Once again, performance can be improved using PEFT/LoRA adaptation, as illustrated in Figure 37 and reported in Tables 9–11. However, despite the availability of ground-truth binary grain masks in this dataset, enabling supervised training, this is not an ideal use case for such training, since native SAM already performs very well and only incremental gains are achievable.

When comparing the full-field binary grain masks predicted by the models with the ground-truth binary grain masks, the resulting grain-scale descriptors diverge to some extent (Figures 51–53).

Specifically, SAM tends to overestimate particle dimensions (Figures 45–47) and systematically underestimate the fraction of smaller particles (Figures 48–50). As noted in the previous chapter, this likely has two causes:

- **(i)** the encoder downsamples the image to a $64 \times 64$ grid [5], which attenuates fine particles (see also Figure 38), and
- **(ii)** default model settings filter out individual masks based on descriptors such as size, omitting them from the final full-field inference [5].

These issues can be mitigated, but doing so lies beyond the scope of this work.

Finally, due to time constraints, we did not investigate whether segmentation metrics differ between glass and crystalline silica particles, nor whether PEFT/LoRA adjustments could help discriminate the more challenging low-contrast particles. As a follow-on, it would be interesting to add a semantic classification head to the model to identify the mineral phase within each particle fraction, as demonstrated in [5].

Nevertheless, this chapter again highlights the potential of SAM as a robust and adaptable tool for quantitative microstructural analysis in mineralogical and materials-science imaging contexts.

# 8 Code and Repository

This thesis is accompanied by a public code repository at https://github.com/adriaancampo/LoRA_Training that contains the complete implementation of the proposed training and evaluation pipeline. The repository is designed to be modular, extensible, and reproducible, allowing both experimentation and reuse for related segmentation tasks.

## 8.1 Code Architecture and Design Concept

The core concept of the codebase is the adaptation of a large pre-trained vision foundation model to a domain-specific segmentation task using parameter-efficient fine-tuning. Rather than modifying the original model weights directly, LoRA modules are injected into selected linear layers of the network. This design minimizes the number of trainable parameters and reduces computational and memory requirements. All components are orchestrated by a single entry-point script, which enables reproducible experiments through command-line arguments.

## 8.2 Repository Structure

The repository is organized as follows:

- `main/master.py`: Main entry point for training and evaluation. This script parses a comprehensive set of command-line arguments that fully specify the experimental configuration:
  - `-data_dir`: Path to the directory containing the input images and their corresponding ground-truth annotations.
  - `-lora_r`: Rank of the LoRA matrices, which controls the number of additional trainable parameters introduced into the model.
  - `-lora_alpha`: Scaling factor applied to the LoRA updates, determining their relative contribution during training.
  - `-early_layers`: Index of the earliest model layer into which LoRA adapters are injected.
  - `-late_layers`: Index of the latest model layer adapted with LoRA, enabling control over the depth of model specialization.
  - `-epochs`: Number of training epochs performed during fine-tuning.
  - `-preprocessing`: Input preprocessing strategy used to convert multi-channel image stacks into three-channel representations. The available options are:
    - `cnn`: Processes the input stack using a trainable CNN. When selected, the CNN is jointly trained and saved alongside the LoRA checkpoints.
    - `pca`: Constructs an RGB image from the pixel-wise principal components of the input stack.
    - `proj`: Applies a simple weighted, pixel-wise linear projection of the input channels.

- ⋆ `biref`: Generates a bireflectance image by computing per-pixel channel range statistics.
- ⋆ `mono`: Produces a monochrome image by aggregating channel intensities into a single grayscale representation.

Based on these parameters, the script initializes the base model and LoRA adapter modules, loads and preprocesses the dataset, and orchestrates the training and validation loops.

- ◦ `main/src/my_project/sam_wrapper.py`: Encapsulates SAM and provides a simplified interface for forward passes, loss computation, and inference.
- ◦ `main/src/my_project/adapter_setup.py`: Defines the insertion and configuration of LoRA modules, including rank, scaling factors, and the selection of adapted layers.
- ◦ `main/src/my_project/data_loader.py`: Implements dataset loading, preprocessing integration, batching, and train–validation splitting.
- ◦ `main/src/my_project/preprocess_module.py`: Contains the preprocessing routines for polarized light microscopy images.
- ◦ `main/src/my_project/utils.py`: Provides auxiliary functionality, including logging utilities, evaluation metrics, and visualization helpers, among other scripts and functions.
- ◦ `main/data/`: Directory containing example datasets and placeholder structures for custom input data.

This modular layout allows individual components to be replaced or extended without modifying the rest of the pipeline.

## 8.3 Usage and Reproducibility

For convenience, the complete setup and execution of the codebase in Google Colab is provided as a single executable cell. The user only needs to copy and paste the code block into a Colab notebook cell and execute it, at https://colab.research.google.com/. The code block below is self-contained and can be executed without modification.

```
1
2  import os, subprocess, textwrap, shutil
3
4  # ------------------------------------------------
5  # Private Key
6  # ------------------------------------------------
7  # This private SSH key is written into the Colab runtime (~/.ssh/id_ed25519)
8  # so that GitHub can authenticate the clone operation via SSH.
9  # NOTE: Colab runtimes are ephemeral; the key is stored only for the lifetime
10 # of the session. Anyone with access to this notebook output could misuse it.
11 PRIVATE_KEY = """-----BEGIN OPENSSH PRIVATE KEY-----
12 b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
13 QyNTUxOQAAACCfYWWn4rUenOHaQGkFDpgWcJhLDX8frUOjdouTjLOXYAAAAJgV3s2ZFd7N
14 mQAAAAtzc2gtZWQyNTUxOQAAACCfYWWn4rUenOHaQGkFDpgWcJhLDX8frUOjdouTjLOXYA
15 AAAEC+IZBXu1nCNTWWEmNMuHEBsPR9bgRHLcfSKEnywPVq/J9hZafitR6c4dpAaQUOmBZw
16 mEsNfx+tQ6N2i5OMs5dgAAAAEGNvbGFiLWRlcGxveS1rZXkBAgMEBQ==
17 -----END OPENSSH PRIVATE KEY-----
18 """
19
20 # SSH URL of the repository to clone.
21 # (Using SSH enables cloning private repositories if the key has access.)
22 REPO_SSH = "git@github.com:adriaancampo/LoRA_Training.git"
23
24 # Target directory name inside the Colab filesystem.
25 DEST_DIR = "LoRA_Training"
26 # ------------------------------------------------
27
28 # ------------------------------------------------
29 # 1) Create ~/.ssh with correct permissions
30 # ------------------------------------------------
31 home = os.path.expanduser("~")
32 ssh_dir = os.path.join(home, ".ssh")
33 os.makedirs(ssh_dir, exist_ok=True)
34
35 # SSH requires restrictive permissions on ~/.ssh
36 os.chmod(ssh_dir, 0o700)
37
38 # ------------------------------------------------
39 # 2) Write the private key to ~/.ssh/id_ed25519
40 # ------------------------------------------------
41 key_path = os.path.join(ssh_dir, "id_ed25519")
42 with open(key_path, "w") as f:
43     # Ensure a trailing newline at the end of the key file
44     f.write(PRIVATE_KEY.strip() + "\n")
45
46 # SSH refuses to use a key if file permissions are too open
47 os.chmod(key_path, 0o600)
48
49 # ------------------------------------------------
50 # 3) Add GitHub host key to known_hosts (avoid interactive prompt)
```

```
# ------------------------------------------------
# Colab sessions start clean; we populate known_hosts so that SSH can verify
# the host key without asking "Are you sure you want to continue connecting?"
known_hosts = os.path.join(ssh_dir, "known_hosts")
with open(known_hosts, "a") as f:
    subprocess.run(
        # Use ssh.github.com on port 443 (often works behind restricted networks)
        ["ssh-keyscan", "-p", "443", "ssh.github.com"],
        stdout=f,
        stderr=subprocess.DEVNULL,
        text=True,
    )
os.chmod(known_hosts, 0o644)

# ------------------------------------------------
# 4) Configure SSH to use port 443 and the written key
# ------------------------------------------------
# GitHub SSH usually uses port 22, but some networks block it.
# This config forces "github.com" connections to route via ssh.github.com:443.
config_path = os.path.join(ssh_dir, "config")
with open(config_path, "w") as f:
    f.write(textwrap.dedent("""\
        Host github.com
          HostName ssh.github.com
          Port 443
          User git
          IdentityFile ~/.ssh/id_ed25519
          IdentitiesOnly yes
          StrictHostKeyChecking yes

        Host ssh.github.com
          Port 443
          User git
          IdentityFile ~/.ssh/id_ed25519
          IdentitiesOnly yes
          StrictHostKeyChecking yes
    """))
os.chmod(config_path, 0o600)

# ------------------------------------------------
# 5) Test SSH authentication
# ------------------------------------------------
# GitHub may return exit code 1 even when authentication succeeds.
print("--- SSH test ---")
subprocess.run(["ssh", "-T", "git@github.com"], text=True)

# ------------------------------------------------
# 6) Clone repository (fresh clone)
# ------------------------------------------------
# Remove any previous clone to guarantee a clean state for reproducibility.
```

```
if os.path.exists(DEST_DIR):
    print(f"Deleting existing '{DEST_DIR}' ...")
    shutil.rmtree(DEST_DIR)

print(f"Cloning into '{DEST_DIR}' ...")
#subprocess.run(["git", "clone", REPO_SSH, DEST_DIR], check=True)
subprocess.run(
    ["git", "clone", "--branch", "my-edits", "--single-branch", REPO_SSH,
    DEST_DIR],
    check=True
)


# ------------------------------------------------
# 7) Install repository in editable mode
# ------------------------------------------------
# -e installs the package so local changes are immediately reflected.
!pip install -e LoRA_Training

# ------------------------------------------------
# 8) Install SAM + Download checkpoints
# ------------------------------------------------
!pip install git+https://github.com/facebookresearch/segment-anything.git
!mkdir -p /content/checkpoints
!wget -O /content/checkpoints/sam_vit_b_01ec64.pth \
  https://dl.fbaipublicfiles.com/segment_anything/sam_vit_b_01ec64.pth

# ------------------------------------------------
# 9) Run the training script with the chosen configuration
# ------------------------------------------------
# Arguments:
# --data_dir: dataset directory in the Colab filesystem
# --lora_r: LoRA rank
# --lora_alpha: LoRA scaling factor
# --early_layers / --late_layers: adapter injection layer range
# --epochs: training epochs
# --preprocess: preprocessing mode (biref, pca, cnn, proj, mono)
!python LoRA_Training/master.py \
  --data_dir "/content/LoRA_Training/data/" \
  --lora_r 8 \
  --lora_alpha 8 \
  --early_layers 0 \
  --late_layers 6 \
  --epochs 10 \
  --preprocess "biref"
```

The script first configures SSH authentication within the Colab runtime, enabling secure access to the GitHub repository. It then clones the repository, installs the project as an editable Python package, and executes the main training script with a predefined set of command-line arguments. During execution, intermediate results, logs, and generated figures are written to the Colab runtime file system. All output files, including visualizations, can be accessed via the `Files` panel on the left-hand side of the Colab interface (see Figure 54). The code block produces a set of single-mask (see Figure 55) and full-field (see Figure 56) inference images and creates a CSV file summarizing the evaluation metrics. In addition, the training progress is tracked (see Figure 57). Interested readers with a sufficiently powerful computer can also pull the repository using the SSH key provided in this example.

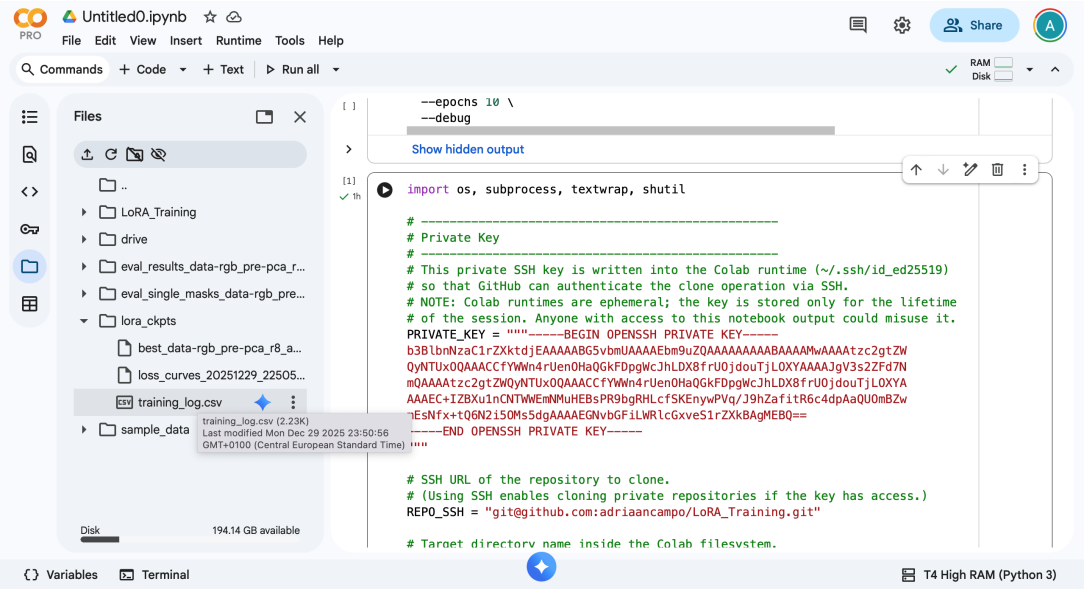



Figure 54: The code can be run in Google Colab, and the results of the training and inference can be found in the `Files` panel on the left-hand.

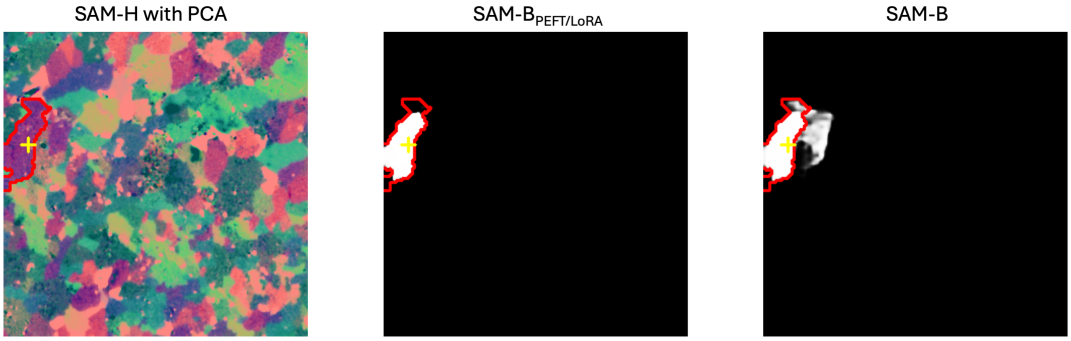



Figure 55: Mask-level improvement achieved through SAM-B$_{\text{PEFT/LoRA}}$ fine-tuning. Individual mask segmentation of SAM-H and PCA preprocessing (left), SAM-B$_{\text{PEFT/LoRA}}$ (middle), and native SAM-B (right).

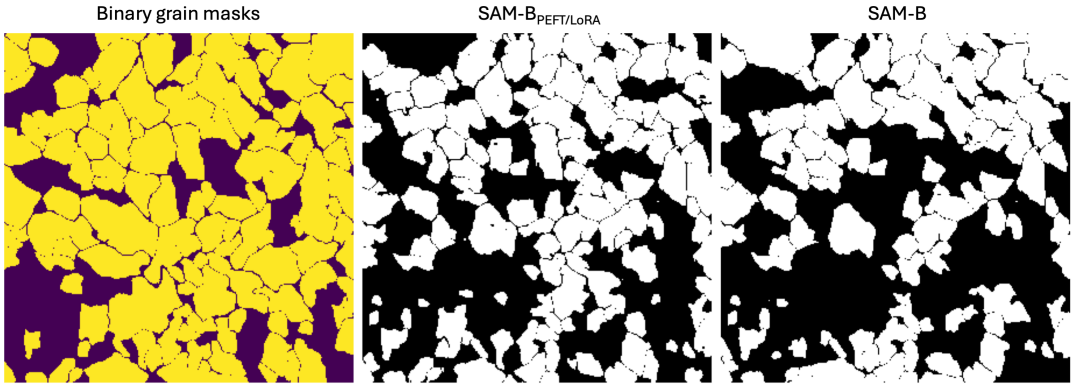



Figure 56: Full-field segmentation improvement achieved through SAM-B$_{\text{PEFT/LoRA}}$ fine-tuning. Binary grain masks as predicted by full field inference of SAM-H and PCA preprocessing (left), SAM-B$_{\text{PEFT/LoRA}}$ (middle), and native SAM-B (right).

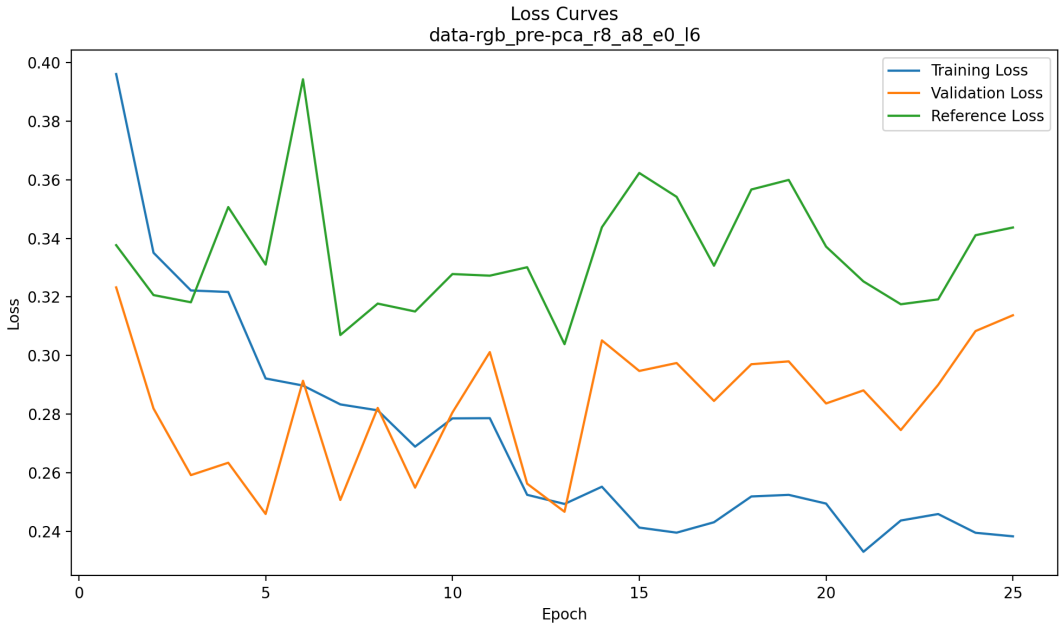



Figure 57: Training (blue) and validation (red) loss of SAM-B$_{\text{PEFT/LoRA}}$ as a function of epoch. The reference loss (green) is also indicated, which is the loss of native SAM-B.

# 9 Discussion and Conclusion

The results in this thesis show that foundation models can be effectively repurposed for quantitative RLM, consistent with reports that large-scale pre-trained models learn transferable representations that extend beyond their original training domains, even under substantial domain shift [77, 106, 107].

However, our benchmarks show that zero-shot SAM does not uniformly outperform task-specific supervised networks. When abundant, domain-specific annotations are available, fully supervised models retain a clear advantage, consistent with prior findings in scientific image segmentation [80]. By contrast, the strength of Vision Transformer–based models lies in their robustness and flexibility, whereas classical deep networks are prone to severe generalization breakdowns under distribution shift [108].

The cross-domain experiments show that differences in reflectance and grain morphology can readily corrupt the outputs of fully supervised, task-specific CNN segmenters such as DeepLabV3+. This pattern mirrors results in microscopy and medical imaging, where dataset bias and acquisition variability (scanner/vendor, staining/protocols) remain dominant obstacles to generalization [109–111]. By contrast, Vision Transformer–based models exhibit stronger robustness and flexibility [112, 113], but at a substantial computational cost, a trade-off that has already prompted the development of lighter SAM variants [114, 115].

Despite SAM's observed and well-documented robustness [5], it is obvious that mineralogically informed preprocessing can significantly improve the alignment between SAM's learned representations and the target domain. These results support the broader argument that domain knowledge remains indispensable, even in the era of foundation models [116]. Rather than replacing expert insight, foundation models benefit from it when embedded within carefully designed preprocessing pipelines [80].

Secondly, our PEFT/LoRA fine-tuning experiments further support this conclusion. By inserting low-rank adapters, we adapt SAM to internalize domain-specific boundary cues. This is broadly consistent with evidence that LoRA can achieve competitive performance gains [94]. That said, our fine-tuning stack is not final; in future iterations we expect additional gains from adapting the image encoder and the mask-decoder MLPs. To encourage collaboration and critical review, we provide a Colab-ready, reproducible codebase with clear repo structure, fixed seeds, and evaluation scripts.

Finally, the comparison between classical intercept-based measurements and SAM-derived descriptors underscores the need to cross-validate and rigorously characterize these methods. Architectural and systems choices in SAM, internal downsampling/upsampling, decoder configuration and thresholds, prompt handling, and practical constraints (large RAM footprint and slower CPU inference that often force tiled processing), can introduce systematic discrepancies relative to classical pipelines. These findings argue for expanded validation, including calibration to physical ground truth, sensitivity analyses to tiling/grid alignment, and ablations over decoder settings [80].

SAM applications in transmitted-light microscopy are only beginning to emerge [83, 88, 89], and, to our knowledge, this is the first study to adapt and fine-tune SAM for mineral identification in RLM. We anticipate that SAM and other foundation models will increasingly be integrated into RLM pipelines; however, mineralogically informed preprocessing and lightweight adaptation via PEFT/LoRA will be crucial to mitigate domain shift and data scarcity. Equally important is task-specific validation, before deployment.

# 10 Limitations

Despite the encouraging results, several limitations of this study must be acknowledged. First, SAM already performs surprisingly well on unseen mineralogical samples without any domain-specific training. This strong zero-shot behavior highlights SAM's generalization capacity, which is arguably its most valuable feature. However, this also implies that the training cases presented in this thesis delivered only incremental improvements. It remains to be shown to what extent we can improve SAM on more challenging RLM datasets.

Secondly, in this work we fine-tuned SAM using PEFT/LoRA while keeping the model backbone frozen. This reduces training memory and avoids backpropagation through the heavy vision transformer encoder. Nevertheless, in our current implementation the full forward pass, including the encoder, is still executed at every training iteration. As a result, the practical speed-up of decoder-only adaptation is smaller than what would be possible in principle. Decoder-focused adaptation becomes substantially more efficient only if encoder image embeddings are cached, after which only the prompt encoder and mask decoder need to be executed during optimization. Implementing such caching would require changes to the data and prompt pipeline and was outside the scope of this study.

Thirdly, while LoRA tuning yields measurable improvements, these gains are inherently limited when the image encoder remains untouched, because the feature representation cannot fully adapt to mineralogical textures, grain boundaries, and domain-specific appearance variations. Extending adapters into the encoder or MLP would likely improve specialization, but would require running and backpropagating through the entire model, dramatically increasing computational cost.

Additionally, fine-tuning SAM for a specific domain introduces a trade-off between specialization and generalization. Increasing task-specific performance risks reducing robustness to unseen samples, thereby weakening the key advantage of SAM as a foundation model. The extent to which we can improve SAM while preserving its generalizability across RLM domains remains to be explored.

Finally, SAM was trained and evaluated only with pixel-wise metrics. Such scores can look strong even when grain boundaries are misaligned, over-smoothed, or interrupted by tiny gaps/bridges, errors that distort grain counts and phase-contact statistics [117, 118]. Accordingly, we recommend reevaluating these methods with boundary- and topology-aware metrics that are tailored to the downstream processing streams.

# Bibliography


[1] W. D. Nesse, *Introduction to Optical Mineralogy*, 5th ed. New York: Oxford University Press, 2019.

[2] E. Pirard, S. Lebichot, and W. Krier, "Particle texture analysis using polarized light imaging and grey level intercepts," *International Journal of Mineral Processing*, vol. 84, no. 1-4, pp. 299–309, 2007.

[3] R. Jaimes Contreras, D. Pilawski, A. Califice, and E. Pirard, "Quantitative microtexture analysis of carbonate rocks using bireflectance imaging," in *Proceedings of IAMG 2010*, Budapest, Hungary, Aug. 2010, iAMG 2010 (29 August–2 September 2010).

[4] M. P. Filippo, O. d. F. M. Gomes, G. A. O. P. da Costa, and G. L. A. Mota, "Deep learning semantic segmentation of opaque and non-opaque minerals from epoxy resin in reflected light microscopy images," *Minerals Engineering*, vol. 170, p. 107007, 2021.

[5] A. Kirillov, E. Mintun, N. Ravi, and et al., "Segment anything," in *ICCV*, 2023.

[6] O. d. F. M. Gomes, S. Paciornik, M. P. Filippo, G. A. O. P. da Costa, and G. L. A. Mota, "Fem dataset – an iron ore labeled images dataset for segmentation training and testing," Jun. 2021.

[7] ——, "Cu dataset – a copper ore labeled images dataset for segmentation training and testing," Jun. 2021.

[8] B. Schulz, D. Sandmann, and S. Gilbricht, "Sem-based automated mineralogy and its application in geo- and material sciences," *Minerals*, vol. 10, no. 11, p. 1004, 2020.

[9] R. Fandrich, Y. Gu, D. Burrows, and K. Moeller, "Modern sem-based mineral liberation analysis," *International Journal of Mineral Processing*, vol. 84, no. 1-4, pp. 310–320, 2007.

[10] Y. Gu, "Automated scanning electron microscope based mineral liberation analysis: An introduction to jkmrc/fei mineral liberation analyser," *Journal of Minerals & Materials Characterization & Engineering*, vol. 2, no. 1, pp. 33–41, 2003.

[11] W. R. Goodall, P. J. Scales, and A. R. Butcher, "The use of qemscan and diagnostic leaching in the characterisation of visible gold in complex ores," *Minerals Engineering*, vol. 18, no. 8, pp. 877–886, 2005.

[12] T. Hrstka, P. Gottlieb, R. Skála, K. Breiter, and D. Motl, "Automated mineralogy and petrology—applications of tescan integrated mineral analyzer (tima)," *Journal of Geosciences*, vol. 63, no. 1, pp. 47–63, 2018.

[13] Carl Zeiss Microscopy GmbH, "Zeiss mineralogic mining — product info, release 4.2," Carl Zeiss Microscopy GmbH, Jena, Germany, Tech. Rep., 2021.

[14] R. D. Barker, S. L. L. Barker, S. Wilson, and E. D. Stock, "Quantitative mineral mapping of drill core surfaces i: A method for µxrf mineral calculation and mapping of hydrothermally altered, fine-grained sedimentary rocks from a carlin-type gold deposit," *Economic Geology*, vol. 116, no. 4, pp. 803–819, 2021.

[15] R. D. Barker, S. L. L. Barker, M. J. Cracknell, E. D. Stock, and G. Holmes, "Quantitative mineral mapping of drill core surfaces ii: Long-wave infrared mineral characterization using µxrf and machine learning," *Economic Geology*, vol. 116, no. 4, pp. 821–836, 2021.

[16] A. López-Benito, J. C. Catalina, D. Alarcón, Ú. Grunwald, P. Romero, and R. Castroviejo, "Automated ore microscopy based on multispectral measurements of specular reflectance i: A comparative study of some supervised classification techniques," *Minerals Engineering*, vol. 146, p. 106136, 2020.

[17] A. Chopard, P. Marion, J.-J. Royer, R. Taza, H. Bouzahzah, and M. Benzaazoua, "Automated sulfides quantification by multispectral optical microscopy," *Minerals Engineering*, vol. 131, pp. 38–50, 2019.

[18] J. C. Catalina, R. Castroviejo, Ú. Grunwald, and A. López-Benito, "Automated characterization of metal ores by multispectral reflected-light microscopy," in *Procemin–Geomet 2021*, Santiago, Chile, 2021.

[19] J. C. Á. Iglesias, O. d. F. M. Gomes, and S. Paciornik, "Automatic recognition of hematite grains under polarized reflected light microscopy through image analysis," *Minerals Engineering*, vol. 24, no. 12, pp. 1264–1270, 2011.

[20] D. Pirrie, A. R. Butcher, M. R. Power, P. Gottlieb, and G. L. Miller, "Rapid quantitative mineral and phase analysis using automated scanning electron microscopy (qemscan); potential applications in forensic geoscience," *Geological Society, London, Special Publications*, vol. 232, pp. 123–136, 2004.

[21] N. Keulen, S. N. Malkki, and S. Graham, "Automated quantitative mineralogy applied to metamorphic rocks," *Minerals*, vol. 10, no. 1, p. 47, 2020.

[22] P. Vallejos, J. Yianatos, and L. Vinnett, "A model structure for size-by-liberation recoveries in flotation," *Minerals*, vol. 11, no. 2, p. 194, 2021.

[23] J. Duan, D. Fornasiero, and J. Ralston, "Calculation of the flotation rate constant of chalcopyrite particles in an ore," *International Journal of Mineral Processing*, vol. 72, no. 1-4, pp. 227–237, 2003.

[24] L. Little, M. Becker, J. Wiese, and A. N. Mainza, "Auto-sem particle shape characterisation: Investigating fine grinding of ug2 ore," *Minerals Engineering*, vol. 82, pp. 92–100, 2015.

[25] P. T. L. Koh, F. P. Hao, L. K. Smith, T. T. Chau, and W. J. Bruckard, "The effect of particle shape and hydrophobicity in flotation," *International Journal of Mineral Processing*, vol. 93, no. 2, pp. 128–134, 2009.

[26] L. Pérez-Barnuevo, E. Pirard, and R. Castroviejo, "Automated characterisation of intergrowth textures in mineral particles—a case study," *Minerals Engineering*, vol. 52, pp. 136–142, 2013.

[27] P. I. Guntoro, Y. Ghorbani, M. Parian, A. R. Butcher, J. Kuva, and J. Rosenkranz, "Development and experimental validation of a texture-based 3d liberation model," *Minerals Engineering*, vol. 164, p. 106828, 2021.

[28] *Standard Test Methods for Determining Average Grain Size*, ASTM International Std. E0112-24, 2024. [Online]. Available: https://www.astm.org/e0112-24.html

[29] E. A. Holm, R. Cohn, N. Gao, A. R. Kitahara, T. P. Matson, B. Lei, and S. R. Yarasi, "Overview: Computer vision and machine learning for microstructural characterization and analysis," *Metallurgical and Materials Transactions A*, vol. 51, pp. 5985–5999, 2020.

[30] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, "Nih image to imagej: 25 years of image analysis," *Nature Methods*, vol. 9, no. 7, pp. 671–675, 2012.

[31] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, "Fiji: an open-source platform for biological-image analysis," *Nature Methods*, vol. 9, no. 7, pp. 676–682, 2012.

[32] I. Arganda-Carreras, V. Kaynig, C. Rueden, K. W. Eliceiri, J. Schindelin, A. Cardona, and H. S. Seung, "Trainable weka segmentation: a machine learning tool for microscopy pixel classification," *Bioinformatics*, vol. 33, no. 15, pp. 2424–2426, 2017.

[33] S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, F. Jug, P. Tomancak, and A. Kreshuk, "ilastik: interactive machine learning for (bio)image analysis," *Nature Methods*, vol. 16, no. 12, pp. 1226–1232, 2019.

[34] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[35] I. Sobel and G. Feldman, "A 3×3 isotropic gradient operator for image processing," Stanford Artificial Intelligence Laboratory (SAIL), Tech. Rep., 1968, stanford AI Project memo.

[36] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[37] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[38] S. Beucher, "The watershed transformation applied to image segmentation," *Scanning Microscopy*, vol. 6, no. Supplement, pp. 299–314, 1992.

[39] L. Vincent, "Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 176–201, 1993.

[40] L. M. Cruz-Orive, "Particle size distributions from sectional data: Stereology," *Journal of Microscopy*, vol. 107, no. 2, pp. 145–158, 1976.

[41] Z. Jeffries, "The determination of grain size in metals," *Transactions of the American Institute of Mining Engineers*, vol. 54, pp. 594–600, 1915.

[42] A. Delesse, "Procédé mécanique pour déterminer la composition des roches," *Annales des Mines*, vol. 13, pp. 379–388, 1848.

[43] L. M. Cruz-Orive, "Stereology: A historical survey," *Image Analysis & Stereology*, vol. 36, no. 3, pp. 153–177, 2017.

[44] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, 1989.

[45] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, 2001.

[46] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE TPAMI*, 2004.

[47] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images," in *ICCV*, 2001.

[48] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE TPAMI*, 2004.

[49] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," in *SIGGRAPH*, 2004.

[50] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE TPAMI*, 2000.

[51] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, 2004.

[52] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, ser. Lecture Notes in Computer Science, vol. 9351. Springer, 2015, pp. 234–241.

[53] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[54] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE TPAMI*, 2017.

[55] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE TPAMI*, 2018.

[56] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," arXiv:1706.05587, 2017.

[57] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017.

[58] J. C. 'Alvarez Iglesias, R. B. M. Santos, and S. Paciornik, "Deep learning discrimination of quartz and resin in optical microscopy images of minerals," *Minerals Engineering*, vol. 138, pp. 79–85, 2019.

[59] B. De Castro, M. Benzaazoua, S. Roychowdhury, A. Chopard, F. Quintal Lauzon, and B. Plante, "Novel technique for the preparation and analysis of powder-based polished sections by automated optical mineralogy: Part 2—use of deep learning approach for transparent mineral detection," *Minerals Engineering*, vol. 206, p. 108508, 2024.

[60] E. J. Y. Koh, E. Amini, G. J. McLachlan, and N. Beaton, "Utilising convolutional neural networks to perform fast automated modal mineralogy analysis for thin-section optical microscopy," *Minerals Engineering*, vol. 173, p. 107230, 2021.

[61] N. Saxena, R. J. Day-Stirrat, A. Hows, and R. Hofmann, "Application of deep learning for semantic segmentation of sandstone thin sections," *Computers Geosciences*, vol. 152, p. 104778, 2021.

[62] J. Zhong, Y. Meng, and Z. Liu, "Multichannel sandstone thin sections identification based on improved deeplab v3 plus neural network," *ACS Omega*, vol. 9, no. 26, pp. 28 611–28 625, 2024.

[63] J. Bardelli, F. Pape, and coauthors, "Analysis of clastic sedimentary rocks using image semantic segmentation: exploring a computational processing method in petrographic studies," in *IAPG Congreso de Exploraci'on y Desarrollo de Hidrocarburos (CONEXPLO), Sesiones Generales: Evaluaci'on de Formaciones*, 2021, compares U-Net and DeepLab V3+ for thin-section mineral segmentation.

[64] I. Aydın, T. K. Şener, A. D. Kılıç, and H. Derviş, "Improving rock type identification through advanced deep learning-based segmentation models: A comparative study," *Applied Sciences*, vol. 15, no. 3, p. 1630, 2025.

[65] V. Z. Martins and coauthors, "Deep learning domain adaptation applied for minerals segmentation in reflected light microscopy images," *Preprints*, 2024, preprint, version 1.

[66] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[67] K. Rathod, A. K. Choudhary, A. Jansche, G. Ketzer-Raichle, T. Bernthaler, and G. Schneider, "Gegra: Approaching a generic model for quantitative grain size analysis from materials microscopy data using deep learning," *Materials Characterization*, vol. 217, p. 114379, 2024.

[68] A. Bukharev, S. Budennyy, O. Lokhanova, B. Belozerov, and E. Zhukovskaya, "The task of instance segmentation of mineral grains in digital images of rock samples (thin sections)," in *2018 International Conference on Artificial Intelligence: Applications and Innovations (IC-AIAI)*. IEEE, 2018.

[69] H. Liu, Y.-L. Ren, X. Li, Y.-X. Hu, J.-P. Wu, B. Li, L. Luo, Z. Tao, X. Liu, J. Liang, Y.-Y. Zhang, X.-Y. An, and W.-K. Fang, "Rock thin-section analysis and identification based on artificial intelligent technique," *Petroleum Science*, vol. 19, no. 4, pp. 1605–1621, 2022.

[70] M. R. Iyas, N. I. Setiawan, and I. W. Warmada, "Mask r-cnn for rock-forming minerals identification on petrography: Case study at monterado, west kalimantan," in *E3S Web of Conferences*, vol. 200, 2020, p. 06007.

[71] R. B. M. Santos, K. S. Augusto, J. C. 'Alvarez Iglesias, S. Rodrigues, S. Paciornik, J. S. Esterle, and A. L. A. Domingues, "A deep learning system for collotelinite segmentation and coal reflectance determination," *International Journal of Coal Geology*, vol. 263, p. 104111, 2022.

[72] B. A. P. Ferreira, K. S. Augusto, J. C. 'Alvarez Iglesias, T. D. P. Caldas, R. B. M. Santos, and S. Paciornik, "Instance segmentation of quartz in iron ore optical microscopy images by deep learning," *Minerals Engineering*, vol. 211, p. 108681, 2024.

[73] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *CVPR*, 2019.

[74] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.

[75] L. Van Doorselaer and coauthors, "Panoptic segmentation for complete labeling of fruit microstructure in 3d micro-ct images with deep learning," *Plant Phenomics*, 2025.

[76] T. Girerd, G. Mart'inez-Arellano, A. T. Clare, and A. Speidel, "A facile methodology to identify microstructural grains on etched surfaces using panoptic segmentation," *Materials Characterization*, vol. 227, p. 115224, 2025.

[77] R. Bommasani, D. A. Hudson, E. Adeli, *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[78] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *CVPR*, 2022.

[79] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[80] J. Ma, B. Wang, Y. He, Y. Wang, Y. Lu, H. Zhang, Y. Zhang, Y. Wang, and X. Guo, "Segment anything in medical images," *arXiv preprint arXiv:2304.12306*, 2023.

[81] Y. Zhang, Z. Liu, Y. He *et al.*, "Samed: A medical image segmentation framework based on the segment anything model," *arXiv preprint arXiv:2304.14660*, 2023.

[82] R. Deng *et al.*, "Segment anything model (sam) for digital pathology: Assess zero-shot segmentation on whole slide imaging," *arXiv:2304.04155*, 2023.

[83] C. Li, X. Han, C. Yao, and X. Ban, "Matsam: Efficient materials microstructure extraction via visual large model," 2024.

[84] S. Ren *et al.*, "Segment anything, from space?" in *WACV*, 2024.

[85] D. Wang, J. Zhang, B. Du, M. Xu, L. Liu, D. Tao, and L. Zhang, "Samrs: Scaling-up remote sensing segmentation dataset with segment anything model," in *NeurIPS Datasets and Benchmarks*, 2023.

[86] W. Zhang *et al.*, "Adapting the segment anything model for plant identification via explainable contrastive learning," *Horticulturae*, vol. 10, no. 4, p. 398, 2024.

[87] K. D. Nguyen *et al.*, "A sam-based solution for hierarchical panoptic segmentation in agricultural scenes," 2023.

[88] A. Vellappally, S. Hou, and J. Emmings, "Automated grain segmentation and mineral classification in rock thin sections," in *EAGE Annual Conference & Exhibition*. EAGE, 2024, pp. 1–5.

[89] Y. Ren *et al.*, "Multi-channel attention transformer for rock thin-section image segmentation," *Journal of Rock Mechanics and Geotechnical Engineering*, 2024.

[90] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, Eds., *Dataset Shift in Machine Learning*, ser. Neural Information Processing. Cambridge, MA: MIT Press, 2009.

[91] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2661–2671.

[92] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[93] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[94] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[95] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *Computer Vision – ECCV 2022*, ser. Lecture Notes in Computer Science. Springer, 2022.

[96] K. Sohn, D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *arXiv preprint arXiv:2001.07685*, 2020.

[97] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo, "Adaptformer: Adapting vision transformers for scalable visual recognition," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[98] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf, "Transfer learning in natural language processing," *Proceedings of NAACL*, 2019.

[99] P. Jaccard, "The distribution of the flora in the alpine zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[100] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[101] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[102] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C.-J. Wu, B. Reagen, P. Micikevicius *et al.*, "MLPerf Inference Benchmark," in *Proceedings of the 47th Annual International Symposium on Computer Architecture (ISCA)*, 2020, pp. 446–459.

[103] J. Pineau, K. Sinha, V. Larivière, H. Larochelle *et al.*, "The Machine Learning Reproducibility Checklist (v2.0)," 2020, accessed: 2026-01-01.

[104] A. Canziani, A. Paszke, and E. Culurciello, "An Analysis of Deep Neural Network Models for Practical Applications," *arXiv preprint arXiv:1605.07678*, 2016.

[105] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[106] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," in *European Conference on Computer Vision (ECCV)*, 2020.

[107] S. Azizi, B. Mustafa, F. Ryan, Z. Beaver, J. Freyberg, J. Deaton, A. Loh, A. Karthikesalingam, S. Kornblith, T. Chen, V. Natarajan, M. Norouzi, D. J. Fleet, G. Hinton *et al.*, "Big self-supervised models advance medical image classification," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[108] J. Chen, J. Mei, X. Li, Y. Lu, Q. Yu, Q. Wei, X. Luo, Y. Xie, E. Adeli, Y. Wang, M. P. Lungren, S. Zhang, L. Xing, L. Lu, A. Yuille, and Y. Zhou, "Transunet: Rethinking the u-net architecture design for medical image segmentation through the lens of transformers," *Medical Image Analysis*, vol. 97, p. 103280, 2024.

[109] H. Guan and M. Liu, "Domain adaptation for medical image analysis: A survey," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 3, pp. 1173–1185, 2022.

[110] K. Stacke, G. Eilertsen, J. Unger, and C. Lundström, "Measuring domain shift for deep learning in histopathology," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 2, pp. 325–336, 2021.

[111] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study," *PLOS Medicine*, vol. 15, no. 11, p. e1002683, 2018.

[112] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, pp. 111–132, 2022.

[113] Y. Bai, J. Mei, A. Yuille, and C. Xie, "Are transformers more robust than cnns?" in *NeurIPS 2021*, 2021.

[114] X. Zhao, W. Li, X. Cheng *et al.*, "Fast segment anything," *arXiv preprint arXiv:2306.12156*, 2023.

[115] C. Zhang, S. Rahman, X. Wang *et al.*, "Towards lightweight sam for mobile applications," *arXiv preprint arXiv:2306.14289*, 2023.

[116] Y. Wang *et al.*, "An empirical study on the robustness of the segment anything model," *Pattern Recognition*, p. 110685, 2024.

[117] A. A. Taha and A. Hanbury, "Metrics for evaluating 3d medical image segmentation: Analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, no. 1, p. 29, 2015.

[118] A. Reinke, M. D. Tizabi, L. Maier-Hein *et al.*, "Understanding metric-related pitfalls in image analysis validation," *Nature Methods*, vol. 21, no. 2, pp. 182–194, 2024.