Particle Segmentation

(Using Segment Anything Model)

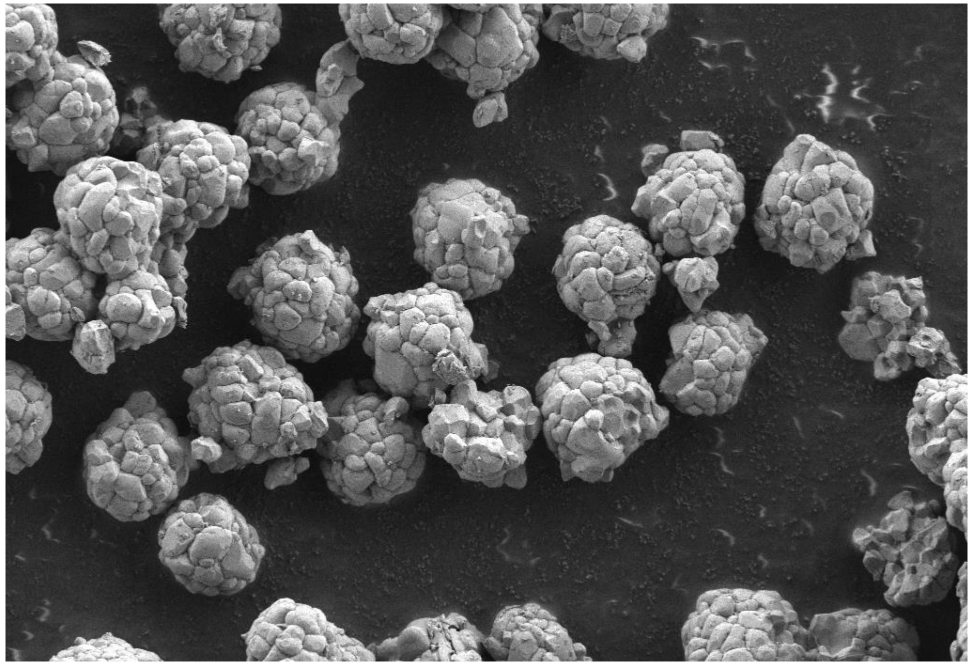
Computer Vision & Pattern Recognition Lab

kit 금오공과대학교
Kumoh National Institute of Technology

Aim

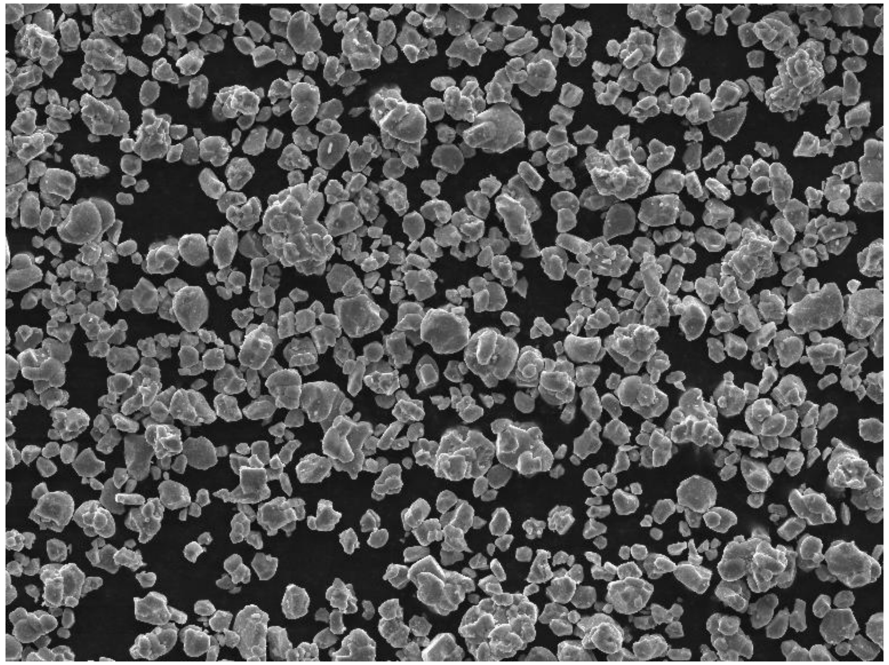
The aim of the project is to improve the segmentation by adapting the Segment anything model using LoRA.

1차시



1-SCN83L001C_1_2KV X 2K

2차시



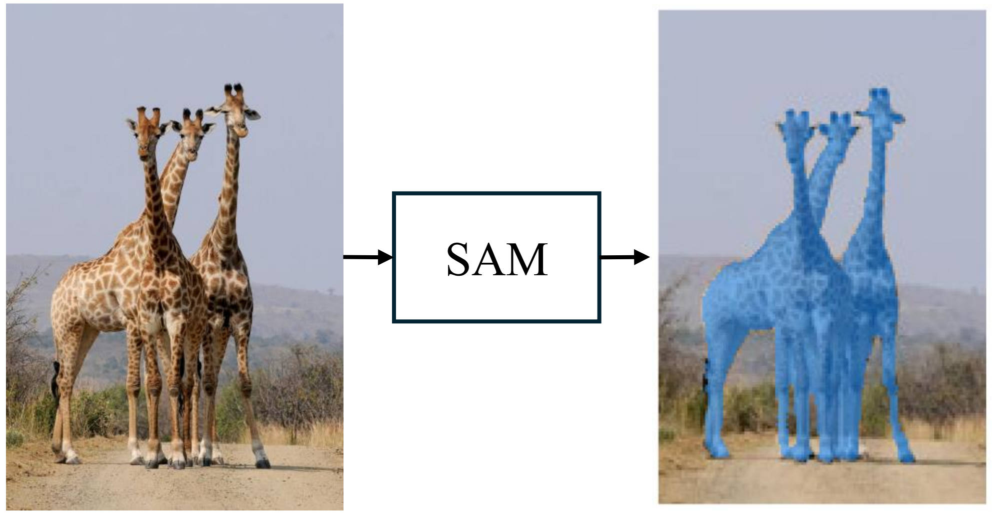
SIPL3-28AJ-1_2K_1_

Segment Anything Model

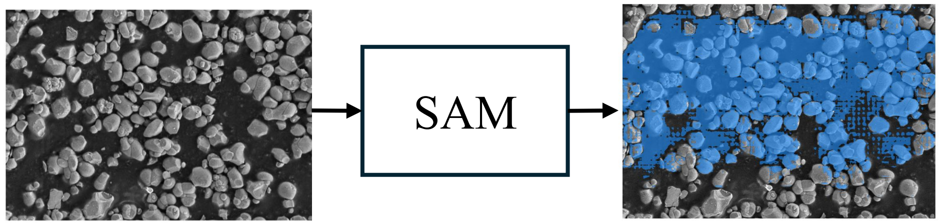
The Segment Anything Model (SAM) is a promptable image-segmentation foundation model introduced in April 2023[1].

It is trained on the SA-1B dataset—11 million images paired with 1.1 billion segmentation masks. As a result, it can generate rich, generalizable image representations which helps in segmenting objects in images. SAM can accurately segment almost any clearly describable object.

Describable Object



Non-describable Object



Both Segmentation results are produced using SAM predictor with base model

However, SAM struggles with non-describable object images.

To help SAM perform better we need to adapt it.

[1]Segment Anything

Adapting Segment Anything Model

Available methods for adapting SAM

1. Full Fine-tuning

   Initialize with pre-trained weights and all model parameters are updated while training.

   Has the highest accuracy but required a lot of training resources[1].

2. Feature Extractor and Custom Head

   The backbone is kept frozen and only the lightweight custom head is trained.

   Less training resources but adaptation is limited and underperform when domain shift is large[2].

3. Adaptors

   Insert small "bottleneck" layers (adapters) between transformer blocks. It freezes original weights and trains only adapter parameters.

   Adds just 5-10 % extra parameters per task and near full-fine-tune performance[2,3].

   Low Rank Adaptor (LoRA)

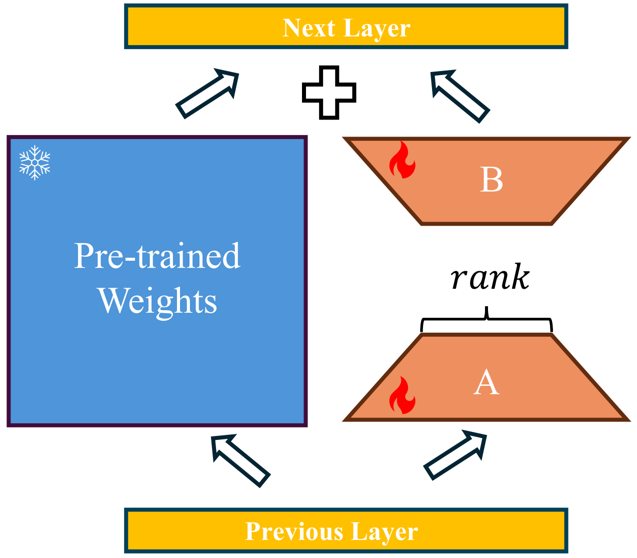
   Keeps base weights frozen and only adds a few parameters to update the weights in the attention layers.

   Matches full fine-tuning with no added inference latency[4].

[1] A survey on Transfer Learning
[2] Parameter-Efficient Transfer Learning for NLP
[3] The Power of Scale for Parameter-Efficient Prompt Tuning
[4] LoRA: Low-Rank Adaptation of Large Language Models

Adapting Segment Anything Model

Low Rank Adaptor (LoRA)



LoRA freezes the pre-trained model weights and injects trainable rank decomposition matrices (A and B) into each layer.

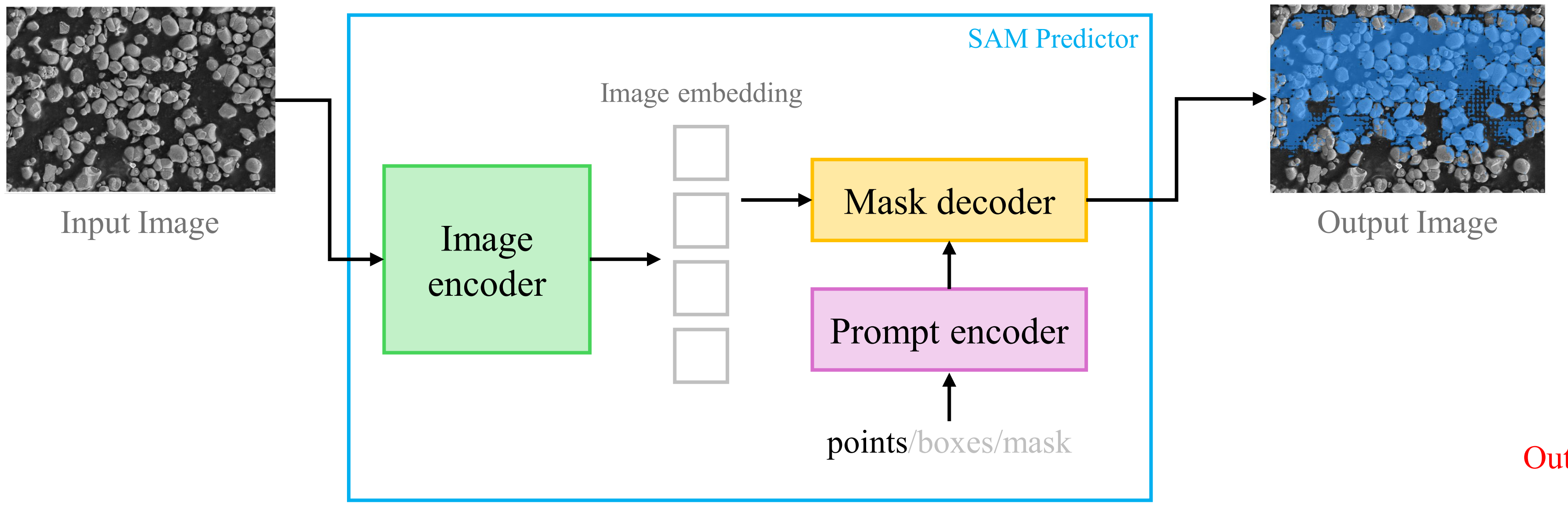
While training, the A and B matrices are updated simultaneously and is then added to the pre-trained weights.

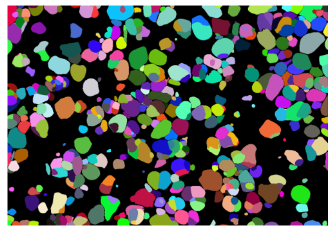
LoRA performs on par with traditional fine-tuning with no additional inference latency[1].

[1]LoRA: Low-Rank Adaptation of Large Language Models

Adapting Segment Anything Model

Need for SAM adaptation?



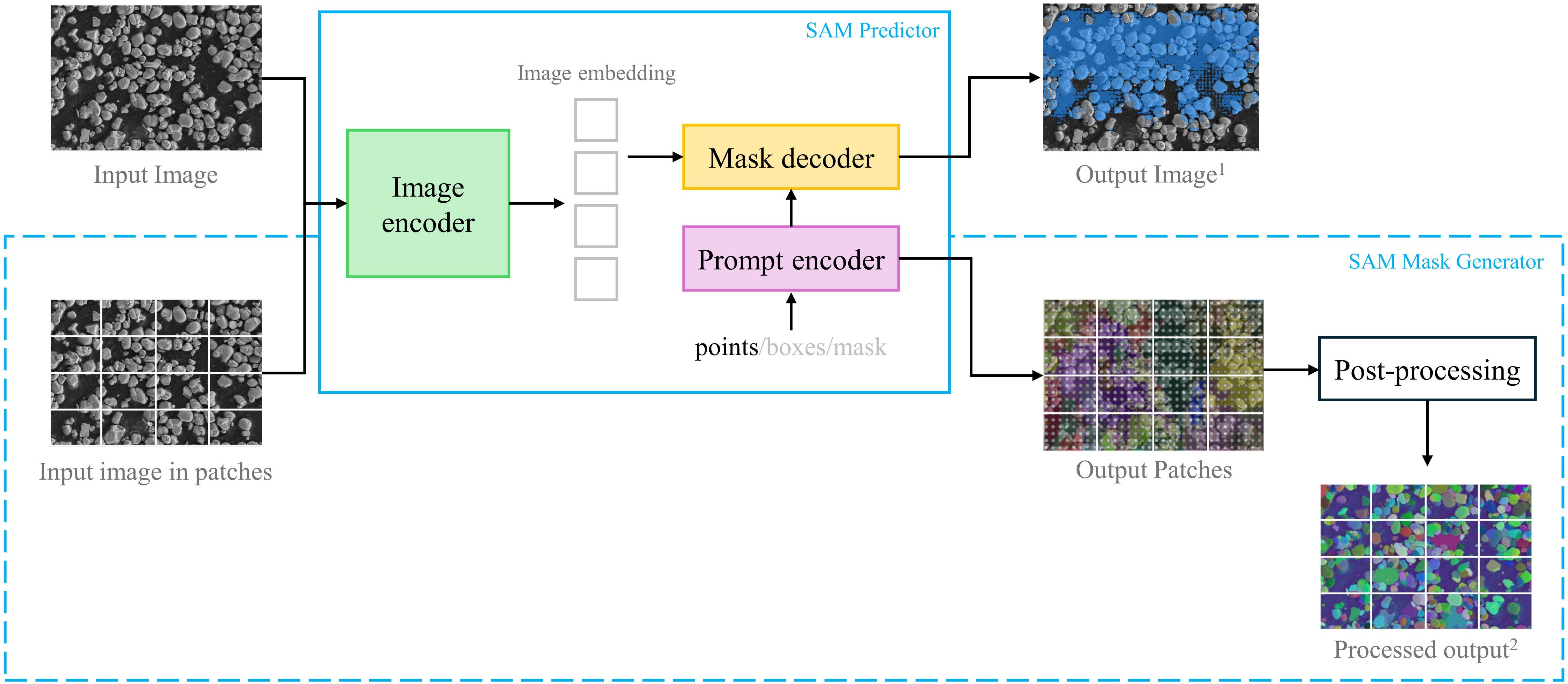

Output of SAM Mask generator

Adapting Segment Anything Model

Need for SAM adaptation?

[1]SAM (default) predictor does single-object segmentation so there is only one color

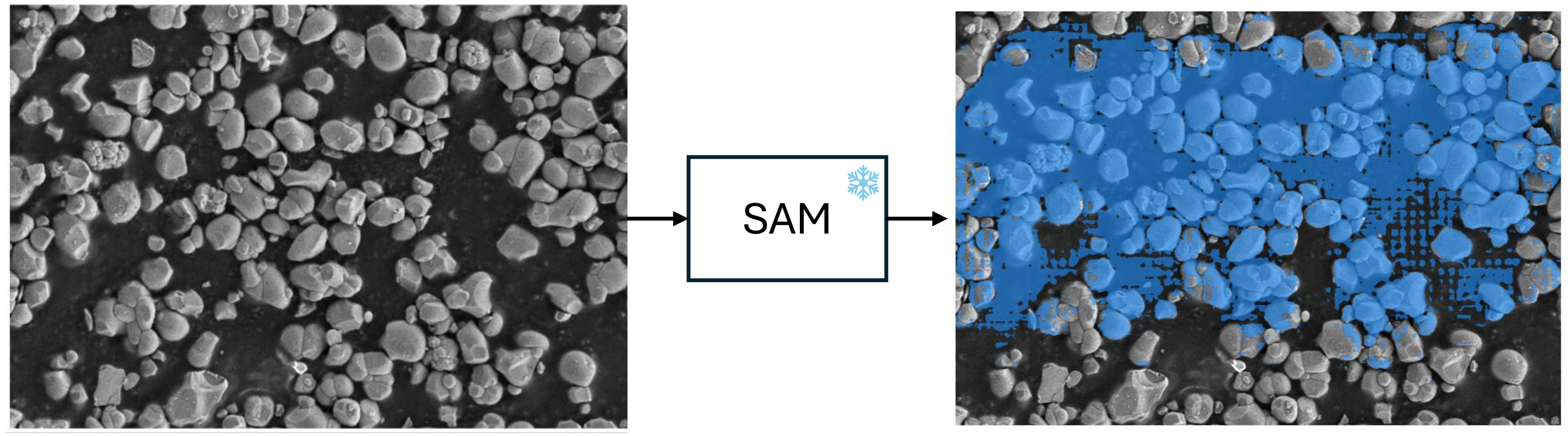
[2]Mask generator randomly adds color to each individual detection during post-processing



Adapting Segment Anything Model

Need for SAM adaptation?

Why this matters?



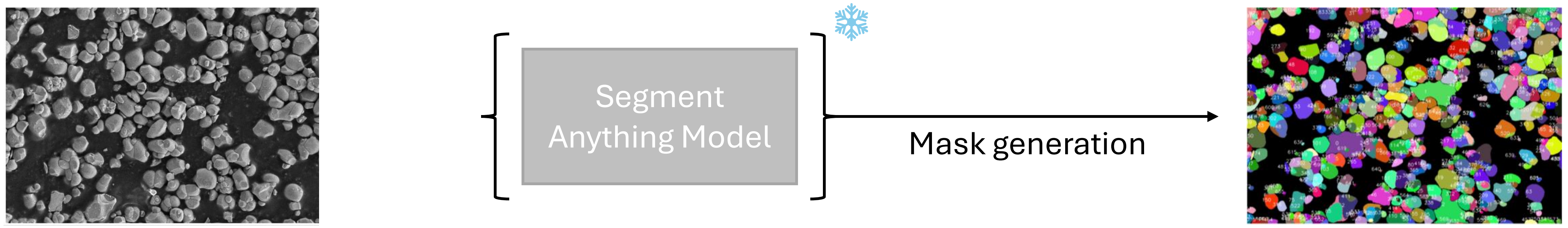
SAM predictor result gives us insight on what SAM actually understands about the image[1].

If we improve SAM predictor result, then the Mask Generator output will also be improved

[1]MaskSAM: Towards Auto-prompt SAM with Mask Classification for Volumetric Medical Image Segmentation

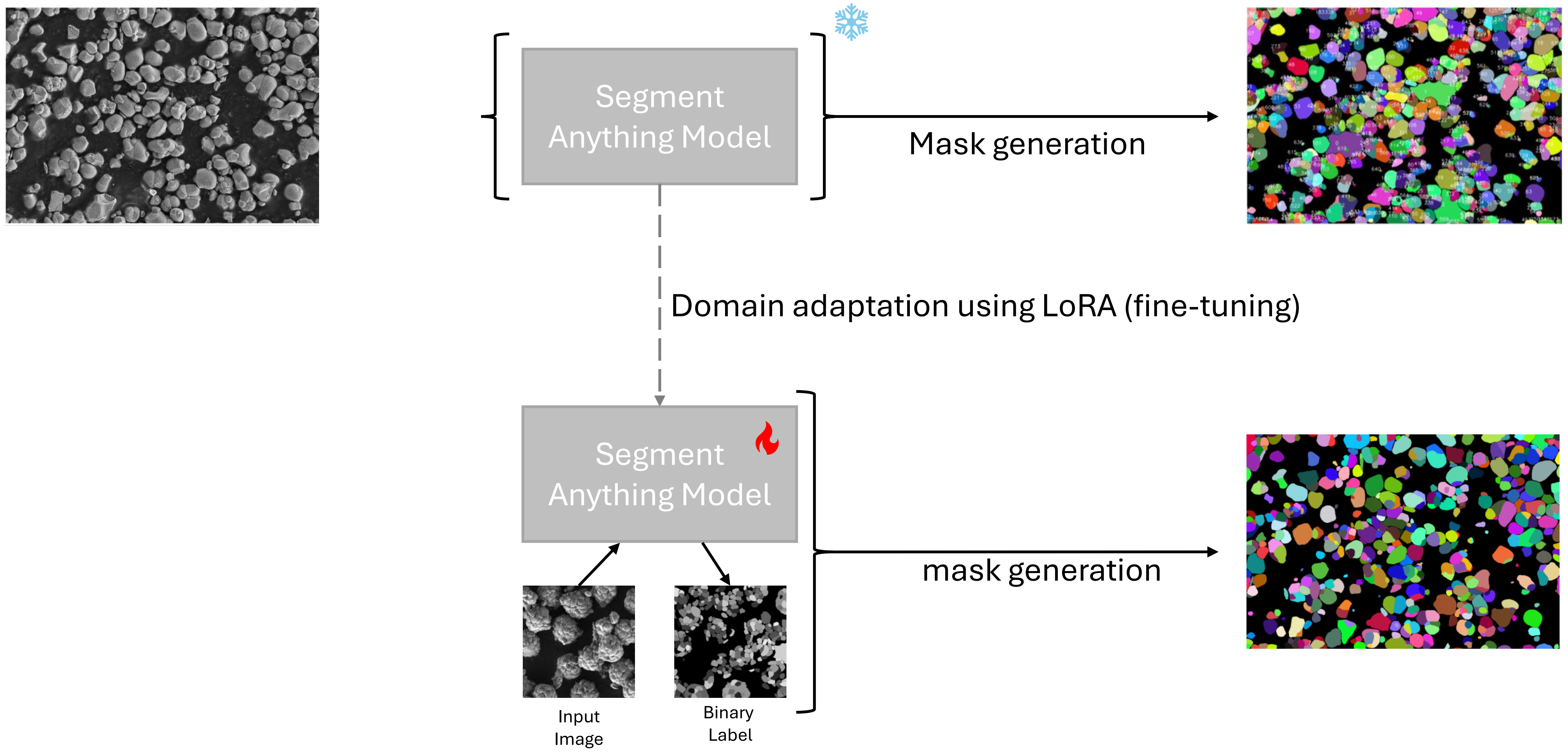
SAM Adaptation using LoRA

Approach



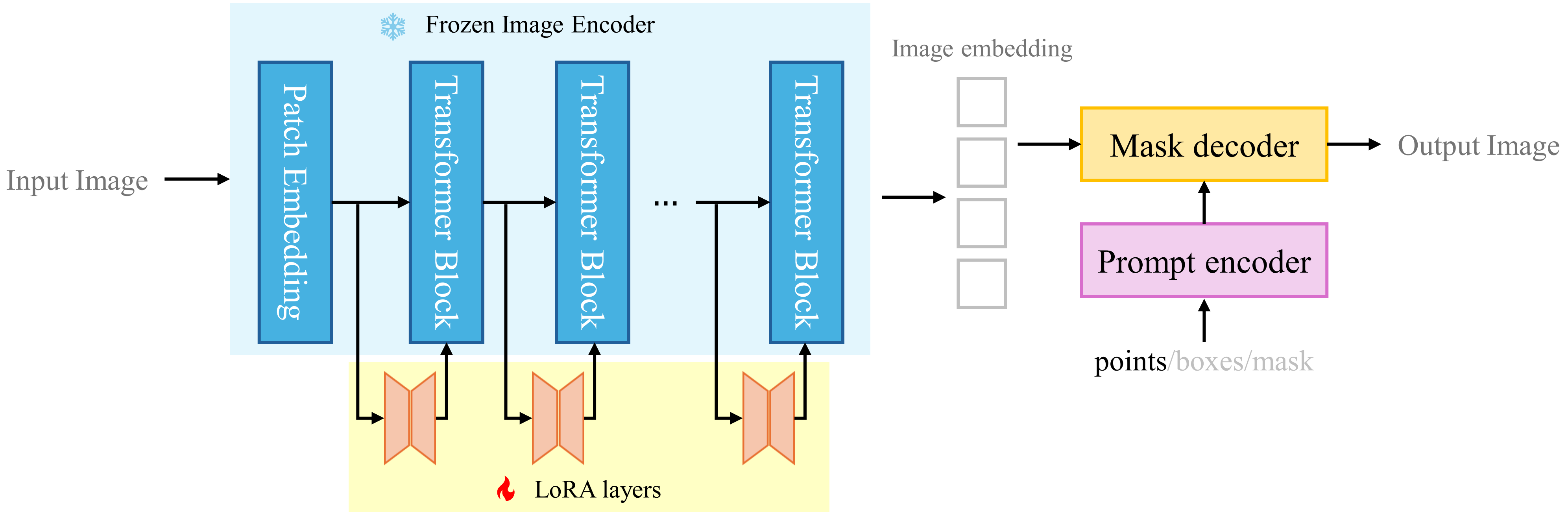
SAM Adaptation using LoRA

Approach



SAM Adaptation using LoRA

Approach



LoRA layers are added only to the encoder because we need to learn representations that is different from its trained data[2]. As encoder is responsible for capturing visual information, most of the adaptation researches implement LoRA in the encoder[3].

[2]Customize Segment Anything Model for Multi-Modal Semantic Segmentation with Mixture of LoRA Experts
[3]AM-SAM: Automated Prompting and Mask Calibration for Segment Anything Model

Model Training

Dataset

Total Images: 243

Image shapes: [(1024x768) - 83 images
(768, 666)] - 160 images

Dataset Split Ratio: [7:2:1]

Augmentations performed: [Sliding window,
Random crop (0-90),
Random flip (horizontal, vertical),
Random contrast,
Random gamma]

Total number of augmented images: 5820

Final training Image shape: 512x512

Model Training

Training Parameters

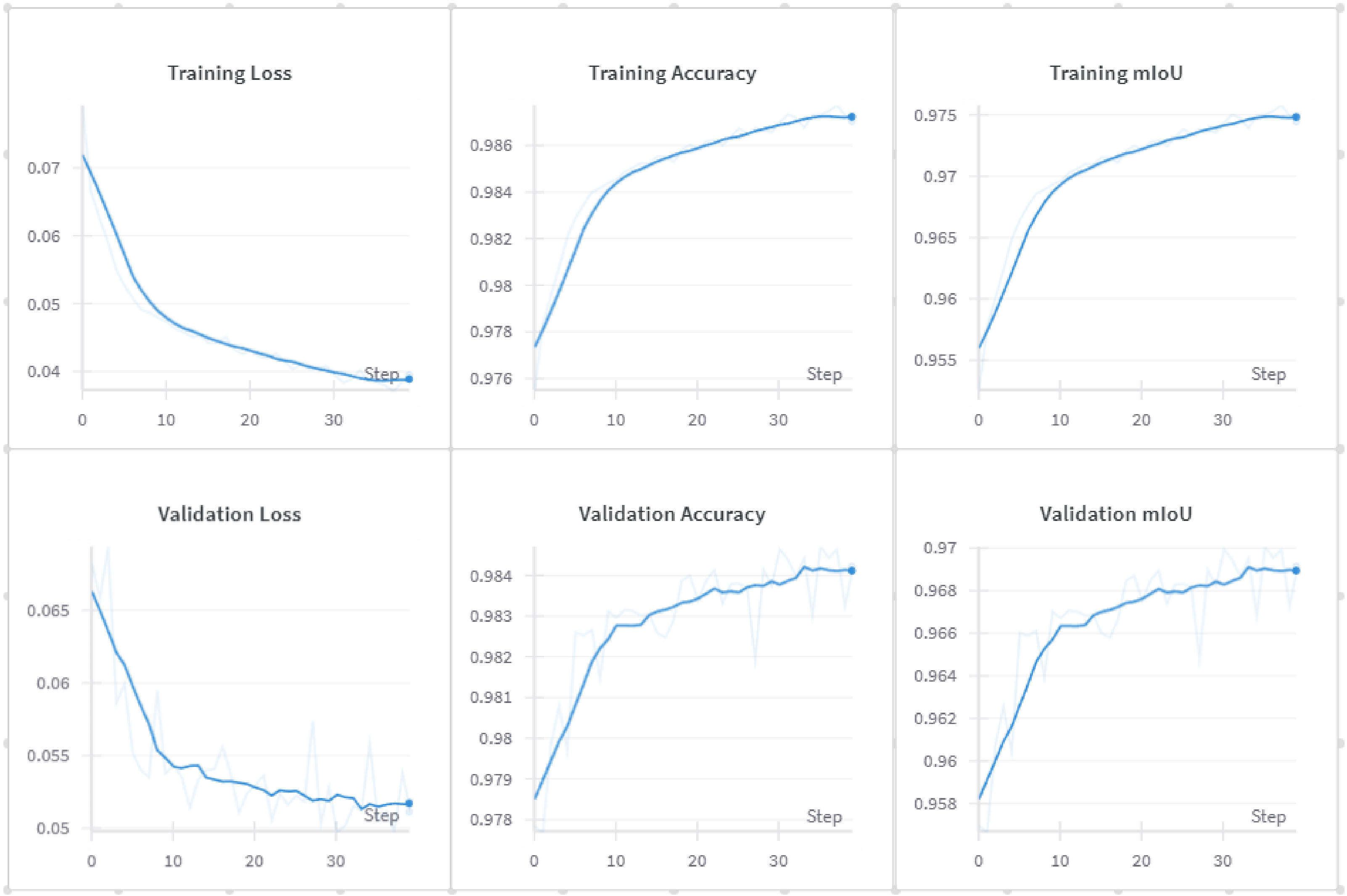
LoRA Rank:	512
SAM Model:	Base Model with ~91M parameters

Epochs:	40
Batch Size:	04
Learning Rate:	0.0001
Loss Function:	Dice Cross-entropy
Optimizer:	Adam, with weight decay – 0.01

Used Weights and Biases for logging experiment statistics.

Model Training Results

Training Stats



Results → Test Set Dice – 96.14 (avg)

Qualitative Results

Mask Generator Hyper parameters

Points per side = 100,	Samples a regular grid of points over the image
Pred iou threshold = 0.90,	Only select objects that surpass prediction iou threshold of 90%.
Stability score threshold = 0.92,	SAM slightly tweaks the mask boundary a few times to see if it stays roughly the same. If the mask changes too much (stability under 0.92), it’s considered unreliable and discarded.
Crop n layers = 1,	
Crop n points downscale factor = 2,	On smaller patches SAM uses fewer points, here 2 means, 100 /2 = 50 points per patch
Min mask region area = 100	After masks are proposed, any tiny blob smaller than 100 pixels is thrown away.

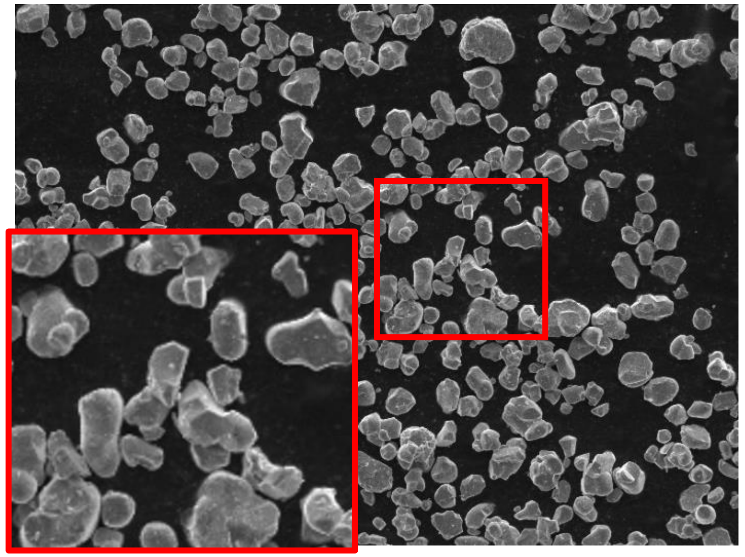
These values are set for all of the experiments.

SAM Fine-Tuning

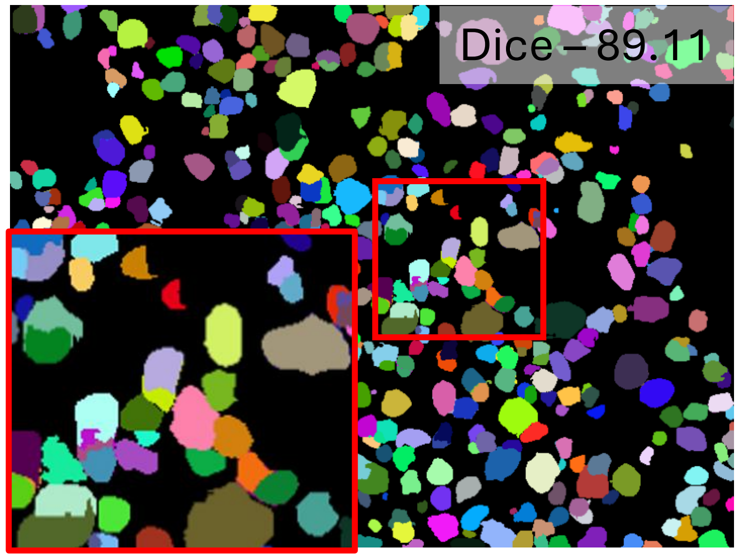
Qualitative Results

Test Case 1

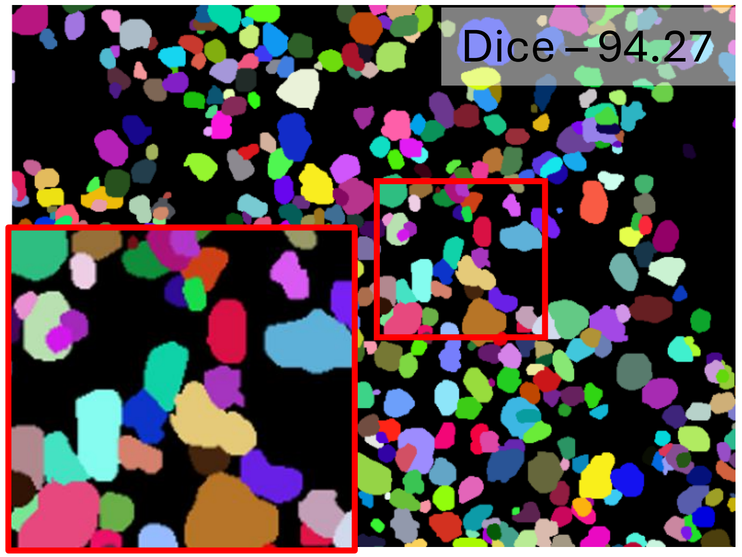
Input Image



❄️ SAM-MG



🔥 SAM-MG

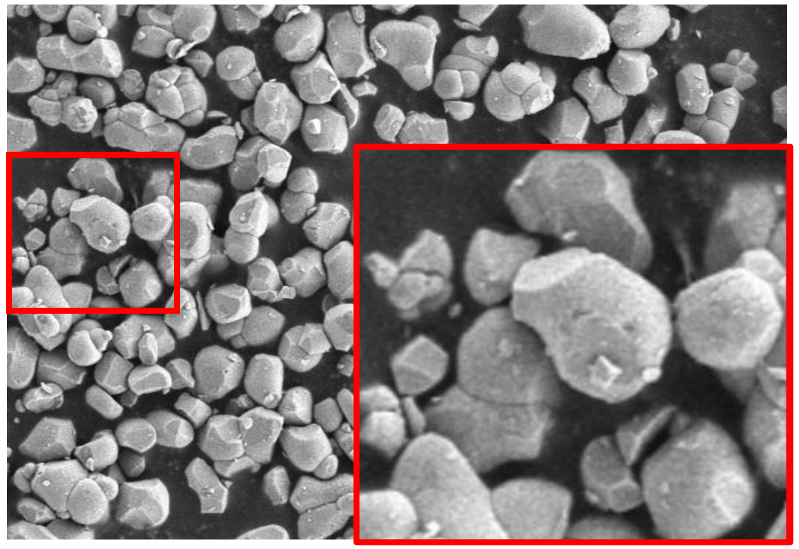


SAM Fine-Tuning

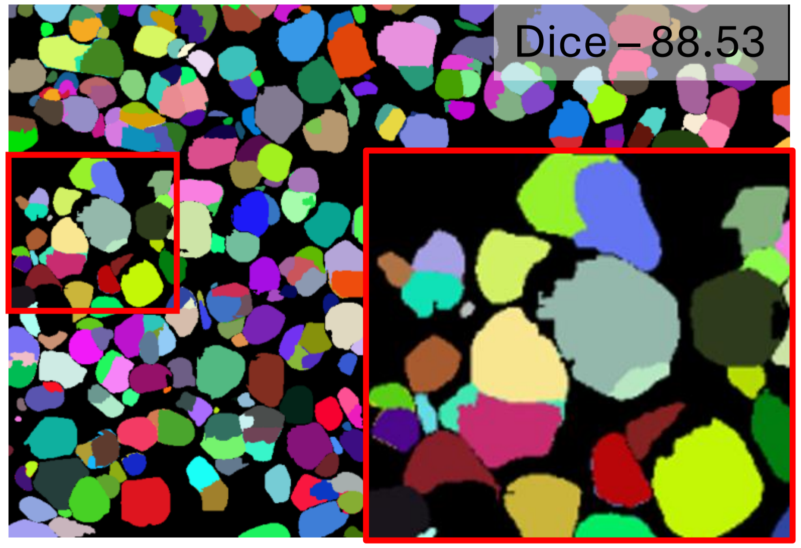
Qualitative Results

Test Case 2

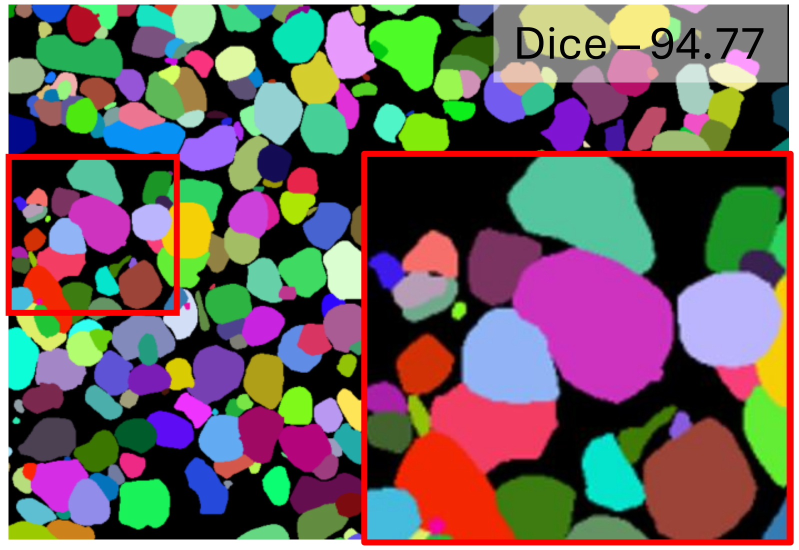
Input Image



❄ SAM-MG



🔥 SAM-MG

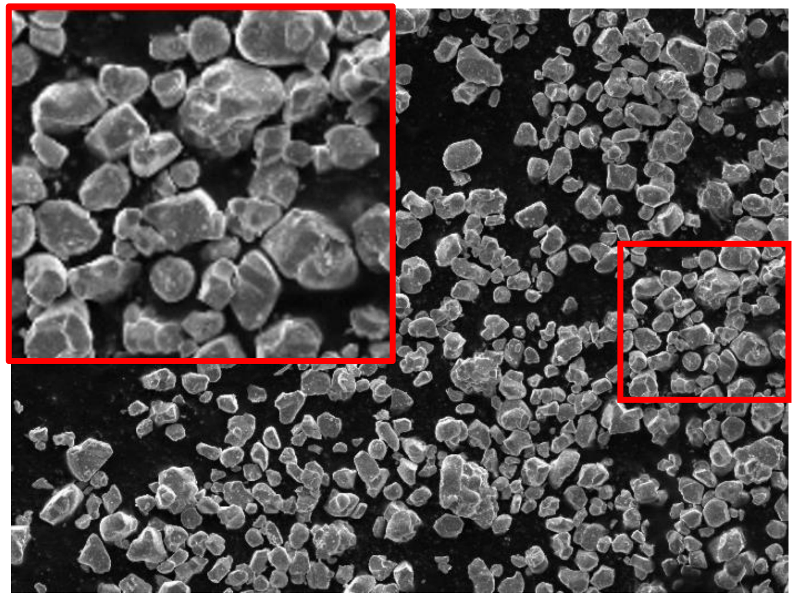


SAM Fine-Tuning

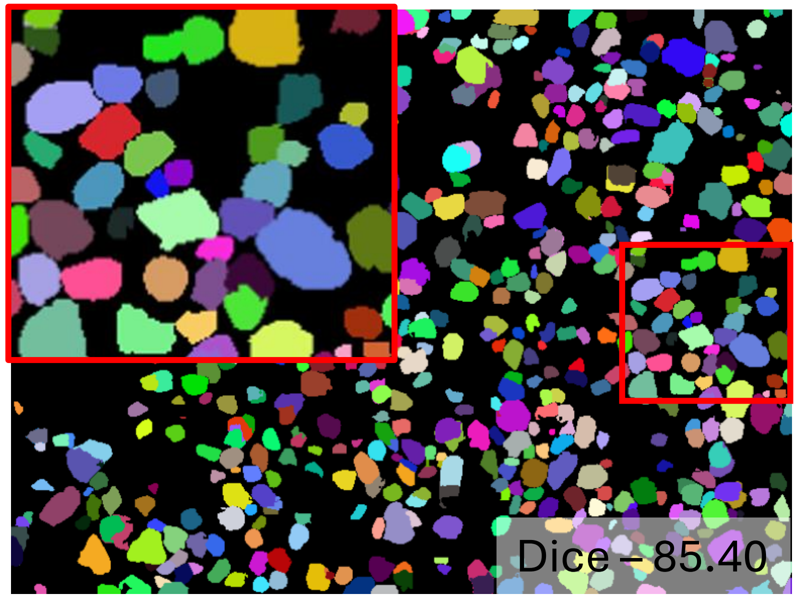
Qualitative Results

Test Case 3 – Lowest Dice Score

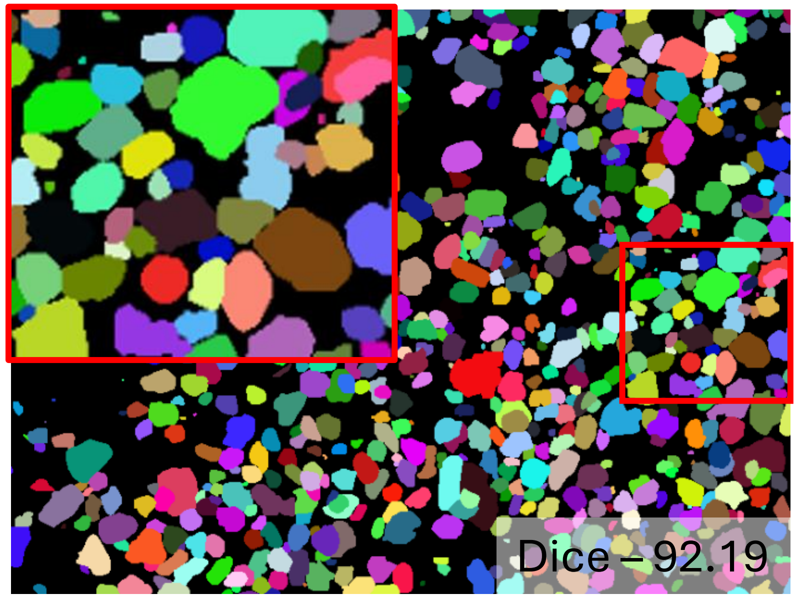
Input Image



❄️ SAM-MG



🔥 SAM-MG

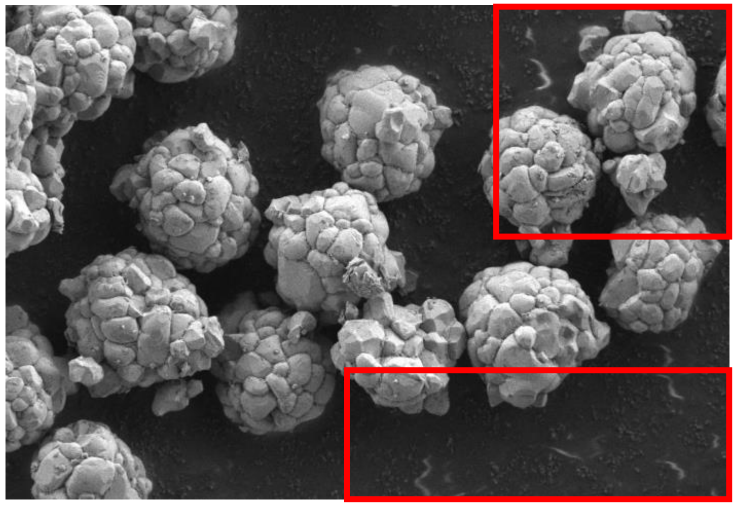


SAM Fine-Tuning

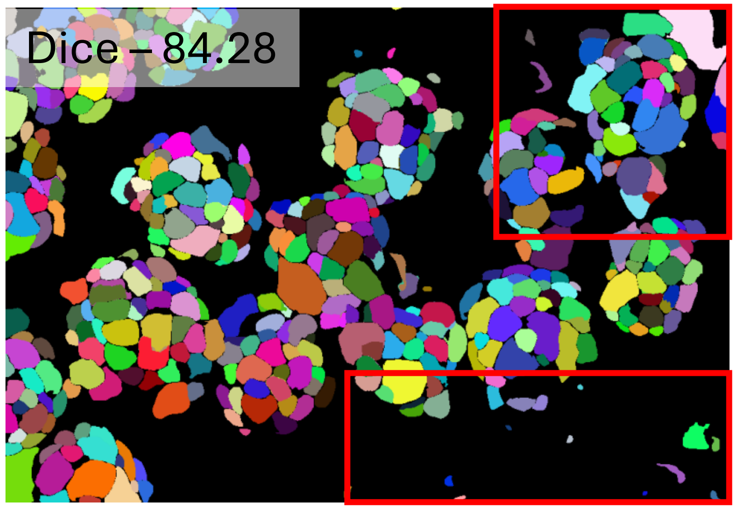
Qualitative Results

Test Case 4 – Highest Dice Score

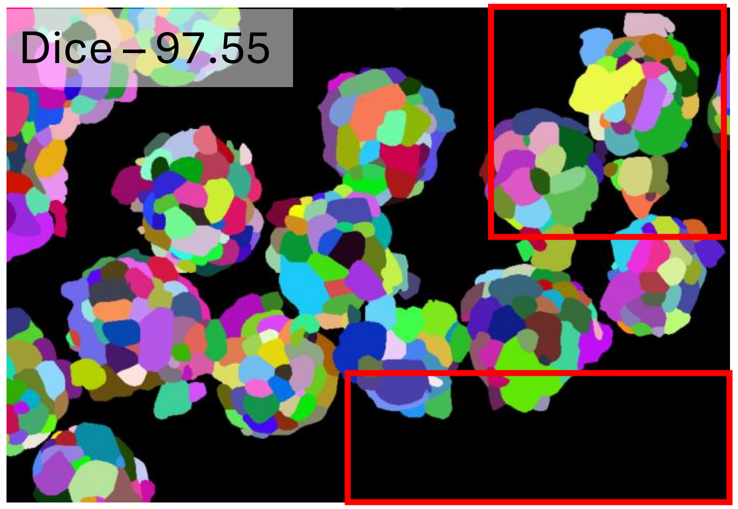
Input Image



❄️ SAM-MG

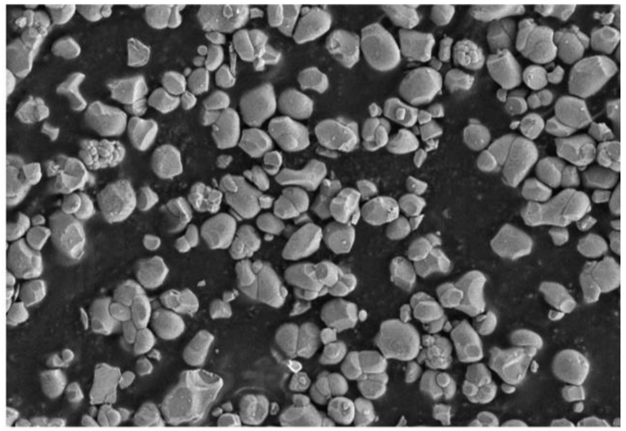


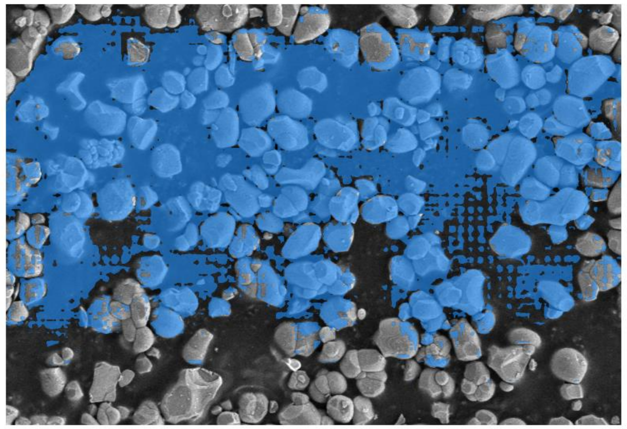
🔥 SAM-MG

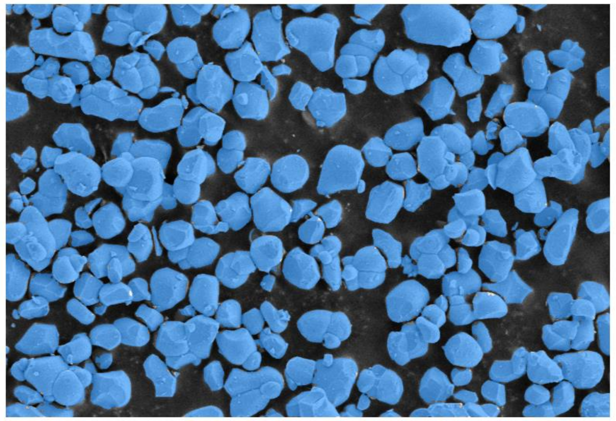


SAM Fine-Tuning

Qualitative Results



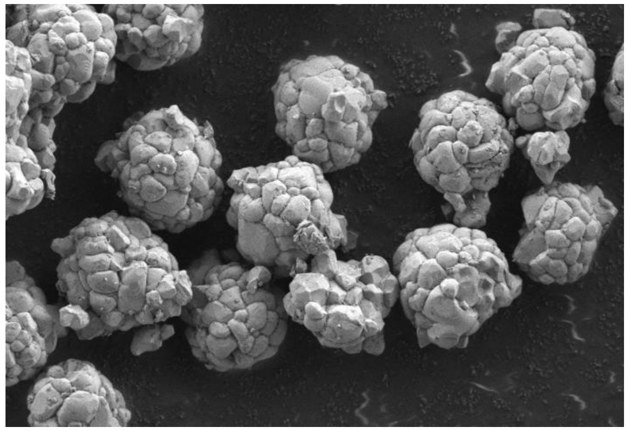


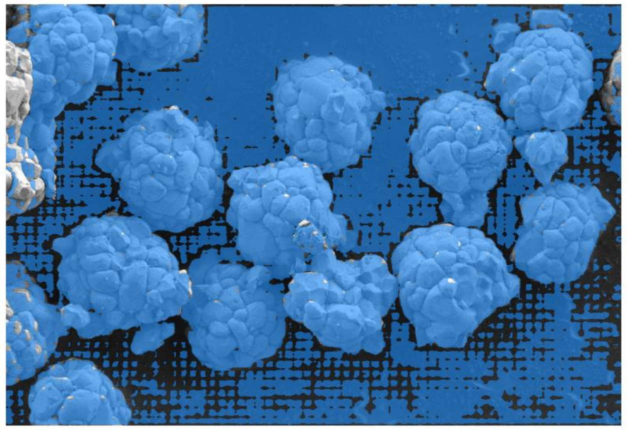


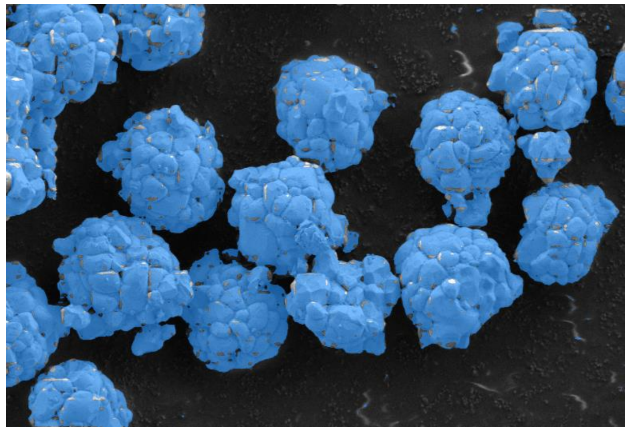
Input Images

SAM Predictor outputs

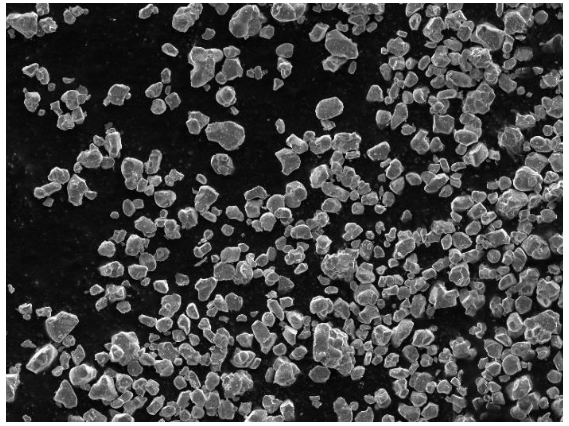
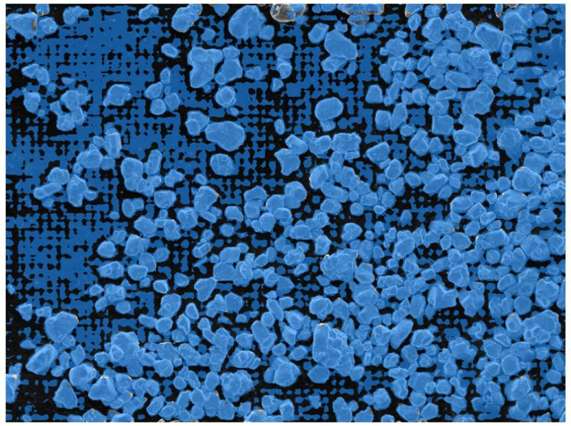
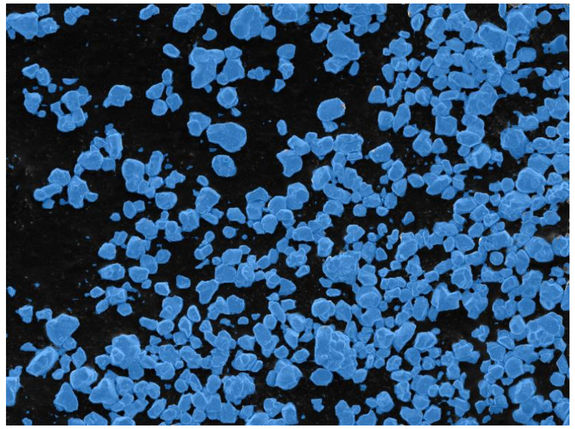
LoRA-SAM Predictor outputs







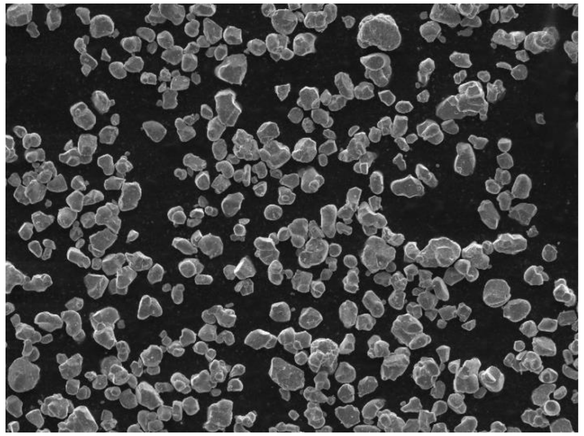
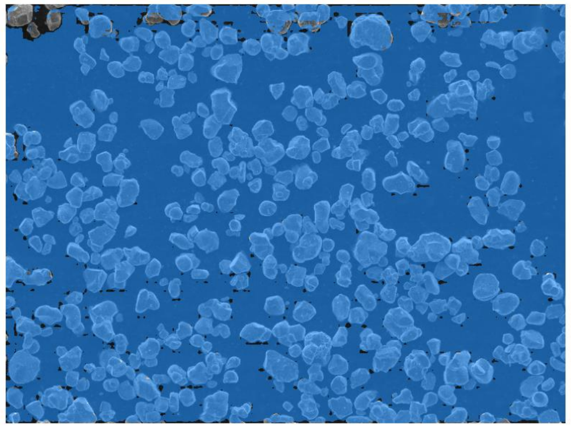
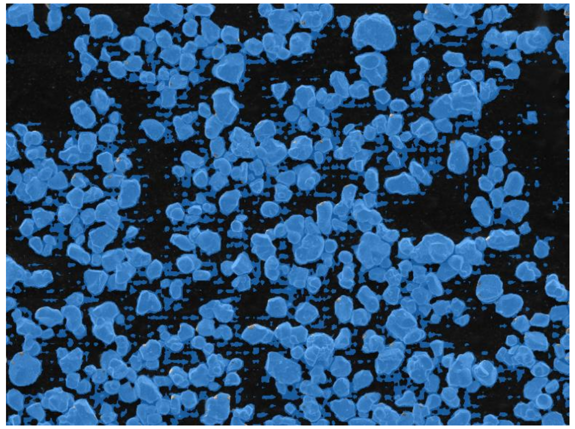
Qualitative Results

Input Images

SAM Predictor outputs

LoRA-SAM Predictor outputs

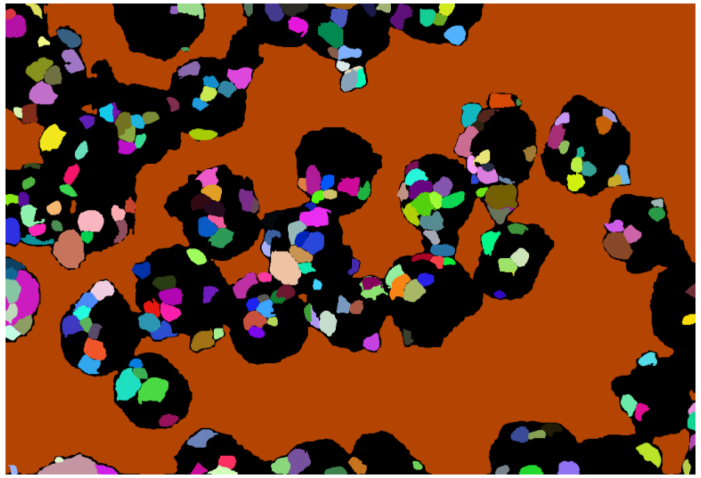
Ablation Studies

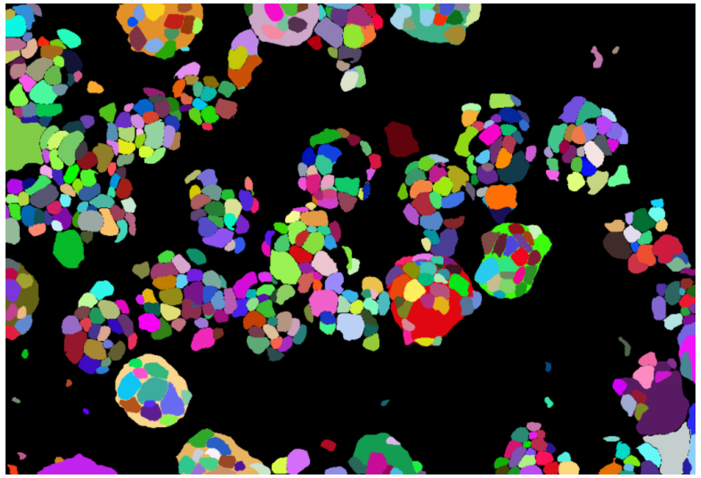
LoRA Rank Experiments

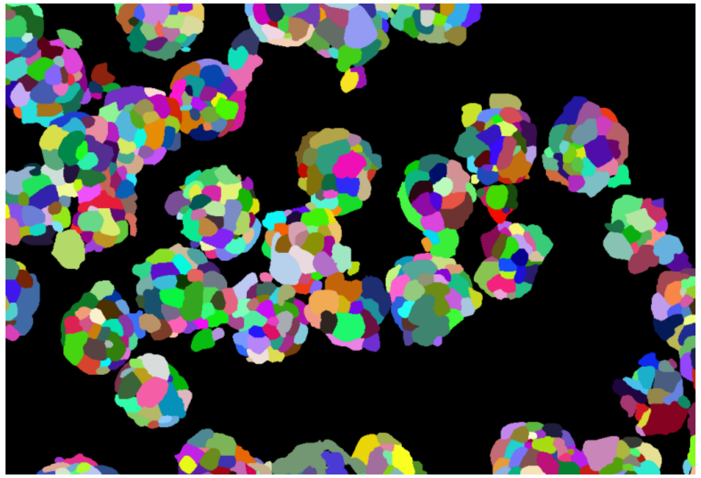
SAM Base model parameters = ~91M

Rank	Trainable Parameters	Dice Score
128	2.8M	82.39
256	5.73M	89.56
512	11.44M	96.14
1024	22.9M	96.22

Rank 1024 has little performance increase but requires more training resources


Rank 128


Rank 256


Rank 512

Ablation Studies

Learning Rate Experiments

Learning Rate	Dice Score
0.01	89.50
0.001	94.29
0.0001	**96.14**
0.00001	93.75

Batch Size Experiments

Batch Size	Dice Score
2	90.61
4	92.77
8	**96.14**
16	94.55
32	95.48