

Supporting Information for

Generative learning of morphological and contrast heterogeneities for self-supervised electron micrograph segmentation

Wenhai Yuan^{1,2}, Bingqing Yao¹, Shengdong Tan¹, Fengqi You^{2*}, Qian He^{1,3*}

¹ Department of Material Science and Engineering, National University of Singapore, 9 Engineering Drive 1, EA #03-09, 117575, Singapore

² Systems Engineering, Cornell University, Ithaca, New York 14853, USA

³ Centre for Hydrogen Innovations, National University of Singapore, E8, 1 Engineering Drive 3, 117580, Singapore

This Supplementary Information file contains:

1. Supplementary Notes 1-4
2. Supplementary Figures 1-88
3. Supplementary Algorithms 1-7
4. Supplementary Tables 1-4

Supplementary Movies are available as separate files and include the following:

1. Supplementary Movie 1: This movie illustrates the generation and iterative refinement of synthetic HAADF-STEM images for PtSn-NPs@Al₂O₃, PtSn-Cluster@@Al₂O₃, Pt-SACs@C, Au@ZSM-5, Pt-SACs@NC, and Pd-NPs@C as training epochs advance. It captures the evolution of the generator's learning process, progressively assimilating real image features. Toward the end, a quick-flash sequence visually compares the final synthetic images with their corresponding experimental counterparts
2. Supplementary Movie 2: This movie captures an in-situ HAADF-STEM session during heating the Au-NPs@C sample. It demonstrates the real-time image acquisition process on the microscope system (Gatan PC, left panel) and the subsequent rapid analysis performed by a locally GPU-accelerated program (GPU PC, right panel). Within seconds, the system generates up-to-date analysis results, including segmentation visualization, particle count, mean size, and a size distribution plot.

*Corresponding Author. E-mail: heqian@nus.edu.sg; fengqi.you@cornell.edu;

Table of Contents

Supplementary Notes.....	7
Supplementary Note 1. Applying Segment Anything Model’s automatic mask generator (SAM-AMG).....	7
Supplementary Note 2. Preprocessing the SAM-AMG masks.....	9
Supplementary Note 3. Zero-shot performance of SAM-AMG masking.....	10
Supplementary Note 4. Extraction of morphology priors and generation of random masks.....	11
Supplementary Figures.....	12
Supplementary Fig. 1. The effect of parameter <i>points_per_side</i> (pps) on SAM-AMG.....	12
Supplementary Fig. 2 The effect of parameter <i>pred_iou_thresh</i> (pit) on SAM-AMG.....	13
Supplementary Fig. 3. The effect of parameter <i>stability_score_thresh</i> (sst) on SAM-AMG.....	14
Supplementary Fig. 4. The effect of <i>min_mask_region_area</i> on SAM-AMG.....	15
Supplementary Fig. 5. The effect of <i>max_mask_region_area</i> on SAM-AMG.....	16
Supplementary Fig. 6. The effect and time cost of model size on SAM-AMG.....	17
Supplementary Fig. 7 Examples of processing the SAM-masks to filter background and support regions.....	18
Supplementary Fig. 8. Examples of processing the SAM-masks to filter incomplete regions on image edge.....	19
Supplementary Fig. 9. Visualization of mask difference between SAM and manual-labeled ground truth.....	20
Supplementary Fig. 10. Examples of false-negatives (red) using SAM masking.....	21
Supplementary Fig. 11. Examples of false-positives (blue) using SAM masking.....	22
Supplementary Fig. 12. Examples of 10 reorganized masks based on SAM-mask.....	23
Supplementary Fig. 13. Histograms and KDE-plots of nanoparticles statistics from 330 masks generated from morphology priors (17420 particles).....	24
Supplementary Fig. 14. Histograms and KDE-plots of nanoparticles statistics from original 33 masks by manual labeling (2860 particles).....	25
Supplementary Fig. 15. Histograms and KDE-plots of nanoparticles statistics from 33 original masks generated by SAM (1901 particles).....	26
Supplementary Fig. 16. KDE-plots between manually labeled masks (red, 2,860 objects) and 10 randomly sampled subsets (green, 2,860 objects) from the full set of generated masks (33 images with 17,420 objects).....	27
Supplementary Fig. 17. KDE-plots of the nearest neighboring distances (NND) for each nanoparticle of the generated masks (17,420 objects from 330 images), manual masks (2,860 objects from 33 images), and SAM masks (1,901 objects from 330 images).....	28

Supplementary Fig. 18. Pair distribution function G(r) (PDF) averaged on each nanoparticle of the generated masks (17,420 objects from 330 images), manual masks (2,860 objects from 33 images), and SAM masks (1,901 objects from 330 images).	29
Supplementary Fig. 19. Histograms and KDE-plots of nanoparticles statistics from 330 totally random generated masks (23317 particles).....	30
Supplementary Fig. 20. Loss and validation metrics of pix2pix trained for 1000 epochs on SAM coarse masks.	31
Supplementary Fig. 21. 1 st round pix2pix training log on SAM coarse masks.....	32
Supplementary Fig. 22. 2 nd round pix2pix training log on SAM coarse masks.....	33
Supplementary Fig. 23. 3 rd round pix2pix training log on SAM coarse masks.....	34
Supplementary Fig. 24. 4 th round pix2pix training log on SAM coarse masks.	35
Supplementary Fig. 25. 5 th round pix2pix training log on SAM coarse masks.	36
Supplementary Fig. 26. Comparison between the experimental images (gray) and the best LPIPS generation from SAM masks (magma).....	37
Supplementary Fig. 27. Synthetic images of the best LPIPS model from random masks...	38
Supplementary Fig. 28. Comparison between the experimental images (gray) and the best SSIM generation from SAM masks (magma).....	39
Supplementary Fig. 29. Synthetic images of the best SSIM model from random masks....	40
Supplementary Fig. 30. Loss and validation metrics of pix2pix trained for 1000 epochs on manual-labeled masks.	41
Supplementary Fig. 31. 1 st round pix2pix training log on manual-labeled masks.....	42
Supplementary Fig. 32. 2 nd round pix2pix training log on manual-labeled masks.....	43
Supplementary Fig. 33. 3 rd round pix2pix training log on manual-labeled masks.....	44
Supplementary Fig. 34. 4 th round pix2pix training log on manual-labeled masks.	45
Supplementary Fig. 35. 5 th round pix2pix training log on manual-labeled masks.	46
Supplementary Fig. 36. Comparison between the synthetic images generated from the best SSIM model trained by SAM-masking (left) and human-masking (right).	47
Supplementary Fig. 37. Comparison between the synthetic images generated from the best LPIPS model trained by SAM-masking (left) and human-masking (right).	48
Supplementary Fig. 38. Representative examples of 18 augmentation strategies.	49
Supplementary Fig. 39. UMAP visualization of raw synthetic data and original data.....	50
Supplementary Fig. 40. UMAP visualization of adapted synthetic data and original data.	51
Supplementary Fig. 41. PCA visualization of raw synthetic data and original data.....	52
Supplementary Fig. 42. PCA visualization of adapted synthetic data and original data.	53
Supplementary Fig. 43. 3D t-SNE visualization of original synthetic data and original data by CBAM-UNet++.....	54
Supplementary Fig. 44. 3D t-SNE visualization of 1000 adapted synthetic data and original data by CBAM-UNet++.....	55

Supplementary Fig. 45. 3D t-SNE visualization of original synthetic data and original data by flatted vectoring.....	56
Supplementary Fig. 46. 3D t-SNE visualization of 1000 adapted synthetic data and original data by flatted vectoring.....	57
Supplementary Fig. 47. Loss and validation metrics of CBAM-UNet++ trained for on 33 manual-labeled images (error band from 5 times runs ensuring reproducibility).....	58
Supplementary Fig. 48. Loss and validation metrics of UNet++ trained for on 33 manual-labeled images.....	59
Supplementary Fig. 49. Loss and validation metrics of SE-UNet++ trained for on 33 manual-labeled images.....	60
Supplementary Fig. 50. Loss and validation metrics of ECA-UNet++ trained for on 33 manual-labeled images.....	61
Supplementary Fig. 51. Loss and validation metrics of CBAM-UNet++ trained for on 5940 synthetic HAADF-STEM images.	62
Supplementary Fig. 52. Grad-CAM of EMcopilot on 33 experimental images.	63
Supplementary Fig. 53. Grad-CAM of manual labeling trained-model on 33 experimental images.....	64
Supplementary Fig. 54. Prediction different between EMcopilot and manual labeling method on 33 experimental images.....	65
Supplementary Fig. 55. Loss and validation metrics of CBAM-UNet++ trained for 5940 synthetic images hybrid with 5 manual labeled images (randomly selected).	66
Supplementary Fig. 56. Segmentation metrics comparison of CBAM-UNet++ trained by ML, EMcopilot and five-shot learning.....	67
Supplementary Fig. 57. Loss and validation metrics of pix2pix trained for 1000 epochs of PtSn Clusters@Al ₂ O ₃	68
Supplementary Fig. 58. 1 st round pix2pix training log of PtSn Clusters@Al ₂ O ₃	69
Supplementary Fig. 59. 2 nd round pix2pix training log of PtSn Clusters@Al ₂ O ₃	70
Supplementary Fig. 60. 3 rd round pix2pix training log of PtSn Clusters@Al ₂ O ₃	71
Supplementary Fig. 61. 4 th round pix2pix training log of PtSn Clusters@Al ₂ O ₃	72
Supplementary Fig. 62. 5 th round pix2pix training log of PtSn Clusters@Al ₂ O ₃	73
Supplementary Fig. 63. Loss and validation metrics of pix2pix trained for 1000 epochs of SACs.....	74
Supplementary Fig. 64. 1 st round pix2pix training log of SACs.....	75
Supplementary Fig. 65. 2 nd round pix2pix training log of SACs.....	76
Supplementary Fig. 66. 3 rd round pix2pix training log of SACs.....	77
Supplementary Fig. 67. 4 th round pix2pix training log of SACs.	78
Supplementary Fig. 68. 5 th round pix2pix training log of SACs.	79

Supplementary Fig. 69. Experimental HAADF-STEM image (gray) and EMcopilot synthetic image (magma) of Pt single-atom catalysts (SACs) supported by nitrogen-doped amorphous carbon (NC) substrate (Pt-SACs@NC), scale bar 2 nm. The image is obtained from the public dataset associated with the work ⁹ by Mitchell's group	80
Supplementary Fig. 70. Experimental HAADF-STEM image (gray) and EMcopilot synthetic image (magma) of Pd-SACs@NC, scale bar 2 nm. The image is obtained from the public dataset associated with the work ⁹ by Mitchell's group.	81
Supplementary Fig. 71. Experimental HAADF-STEM image (gray) and EMcopilot synthetic image (magma) of Ni-SACs@NC, scale bar 2 nm. The image is obtained from the public dataset associated with the work ⁹ by Mitchell's group.	82
Supplementary Fig. 72. Experimental HAADF-STEM image (gray) and EMcopilot synthetic image (magma) of Ru-SACs@NC, scale bar 2 nm. The image is obtained from the public dataset associated with the work ⁹ by Mitchell's group.	83
Supplementary Fig. 73. Experimental BF-TEM image (gray) and EMcopilot synthetic image (magma) of Pd nanoparticles supported on carbon (Pd@C), scale bar 50 nm. The image is obtained from the public dataset associated with the work ¹⁰ by Kirkland's group.	
.....	84
Supplementary Fig. 74. Loss and validation metrics of pix2pix trained for 1000 epochs of Au@ZSM-5.....	85
Supplementary Fig. 75. 1 st round pix2pix training log of Au@ZSM-5.....	86
Supplementary Fig. 76. 2 nd round pix2pix training log of Au@ZSM-5.....	87
Supplementary Fig. 77. 3 rd round pix2pix training log of Au@ZSM-5.....	88
Supplementary Fig. 78. 4 th round pix2pix training log of Au@ZSM-5.....	89
Supplementary Fig. 79. 5 th round pix2pix training log of Au@ZSM-5.....	90
Supplementary Fig. 80. Loss and validation metrics of pix2pix trained for 1000 epochs of Pd@C.	91
Supplementary Fig. 81. 1 st round pix2pix training log of Pd@C.	92
Supplementary Fig. 82. 2 nd round pix2pix training log of Pd@C.....	93
Supplementary Fig. 83. 3 nd round pix2pix training log of Pd@C.....	94
Supplementary Fig. 84. 4 th round pix2pix training log of Pd@C.	95
Supplementary Fig. 85. 5 th round pix2pix training log of Pd@C.	96
Supplementary Fig. 86. Visualization of identifying nanoparticles dynamics and analyzing PSD during in situ experiments (Au@C).....	97
Supplementary Fig. 87. Evolution of particle counts and size during in situ experiments..	98
Supplementary Fig. 88. Schematic illustration of the PC connections of ARM200F system for the in-situ experiments.....	99
Supplementary Algorithms	100
Supplementary Algorithm 1. Processing binary masks obtained from SAM-AMG.....	100

Supplementary Algorithm 2. Extracting morphology priors from processed SAM masks.	101
Supplementary Algorithm 3. Generating reorganized masks based on morphology priors.	102
Supplementary Algorithm 4. Generating totally random masks without morphology priors.	103
Supplementary Algorithm 5. Training and validating the generator.	104
Supplementary Algorithm 6. Augmenting the raw generated images by adding noise and contrast variation.	105
Supplementary Algorithm 7. Training and validating the segmentor.	106
Supplementary Tables	107
Supplementary Table 1. Summary of reported methods and comparison with this work.	107
Supplementary Table 2. Summary of datasets, validation objectives, evaluation metrics, and result references.	109
Supplementary Table 3. Settings of ablation studies.	113
Supplementary Table 4. Statistics of training and inference time.	114
Reference	115

Supplementary Notes

Supplementary Note 1. Applying Segment Anything Model’s automatic mask generator (SAM-AMG).

Segment Anything Model (SAM)¹ is a state-of-the-art segmentation model based on a Transformer² architecture, designed to achieve high-performance zero-shot segmentation without requiring task-specific fine-tuning. Due to its adaptability across diverse imaging domains, SAM has been widely applied in fields such as biomedical imaging^{3,4}, remote sensing⁵, and materials science⁶⁻⁸. In this study, we employed SAM for the initial segmentation of HAADF-STEM images containing supported nanoparticles, generating high-quality masks that serve as a foundation for subsequent data generation. We utilized point-based prompting, where a grid of sampled points is provided as input to guide SAM in detecting and segmenting target regions. To optimize SAM’s performance on HAADF-STEM images, we systematically explored several key parameters of the mask generator, among which *points_per_side*, *pred_iou_thresh*, and *stability_score_thresh* exhibited the most significant impact.

Our trials revealed that *points_per_side* (pps) determines the density of prompt points across the image. Lower values result in incomplete segmentation due to missed targets, as shown in **Supplementary Fig. 1**, whereas higher values improve mask coverage but significantly increase computational cost. As a trade-off between accuracy and efficiency, a higher setting (such as 32) was found to be preferable. The *pred_iou_thresh* (pit) parameter, which filters candidate masks based on predicted Intersection-over-Union (IoU) scores, affects the model’s ability to detect low-contrast nanoparticles. A threshold that is too high may cause weakly contrasting particles to be ignored, leading to incomplete segmentation, while a lower value increases sensitivity to these structures. We found that setting pit to 0.5 allows the model to recognize most nanoparticles without introducing excessive background noise, as demonstrated in **Supplementary Fig. 2**. The *stability_score_thresh* (sst) parameter, which determines the robustness of the generated masks, plays a crucial role in filtering out unstable segmentations. Lower values result in excessive segmentation of background regions, particularly the supporting substrate, which is undesirable for accurate particle identification. Maintaining a high threshold, such as 0.9, effectively prevents spurious mask generation in the background and ensures the integrity of individual nanoparticles, as shown in **Supplementary Fig. 3**.

Additionally, *points_per_batch* must be adjusted according to GPU/CPU memory limitations, as excessively high values can lead to out-of-memory errors when processing high-resolution images. These parameter optimizations are crucial for enhancing the applicability of SAM in electron microscopy segmentation tasks, ensuring the production of high-quality masks that support subsequent image synthesis and model training.

In addition to tuning the internal parameters of SAM, applying area-based filtering provides an effective way to refine segmentation masks by removing background regions and improving overall accuracy. While SAM inherently includes a *min_mask_region_area* parameter to filter out extremely small regions, we introduced an additional *max_mask_region_area* parameter to impose an upper limit on mask size, allowing for bidirectional control. Our experiments showed that the minimum area threshold had a relatively minor impact on segmentation results, as illustrated in **Supplementary Fig. 4**. However, as demonstrated in **Supplementary Fig. 5**, the *max_mask_region_area* parameter significantly influenced the outcome. When *max_mask_region_area* was set too high, large background regions were misclassified as objects, leading to excessive segmentation. By moderately reducing this value, we were able to effectively filter out unwanted background while retaining nanoparticle masks. However, it is crucial to note that in cases where the sample contains exceptionally large nanoparticles, an excessively low *max_mask_region_area* value may inadvertently remove these valid structures. Therefore, the *max_mask_region_area* parameter should be adjusted adaptively based on the characteristics of individual images to achieve an optimal balance between maximizing nanoparticle recognition and minimizing background interference.

Furthermore, we compared the performance of the three available SAM model variants—huge (ViT-H), large (ViT-L), and base (ViT-B)—to evaluate their efficiency in HAADF-STEM image segmentation (**Supplementary Fig. 6**). While the segmentation accuracy remained relatively stable across different models, the inference time varied significantly. Using CPU-based execution as a benchmark, the processing times for ViT-H, ViT-L, and ViT-B were 163s, 128s, and 99s, respectively. Consequently, for applications where efficiency is a priority, such as on-the-fly processing or real-time analysis, the ViT-B variant offers a more practical choice due to its significantly reduced computational burden while maintaining reasonable segmentation performance.

Supplementary Note 2. Preprocessing the SAM-AMG masks.

The raw segmentation masks generated by SAM-AMG often contain artifacts that can negatively impact subsequent processing and model training. These artifacts primarily fall into two categories: background/support and edge artifacts from incomplete particle segmentation. To ensure the quality and reliability of the segmentation masks, we apply a preprocessing pipeline that filters out invalid regions based on pixel intensity and spatial location constraints, as described in **Supplementary Algorithm 1**, which outlines the complete algorithm used for refining SAM-generated binary masks.

The first issue arises from the inclusion of low-intensity regions in the SAM-generated masks, which may correspond to background noise or the support structure of the sample rather than actual nanoparticles. Since supported nanoparticles in HAADF-STEM imaging exhibit significantly higher intensity than the surrounding support, we apply an intensity-based filtering step to remove low-intensity regions. For each segmented mask, we compute the mean pixel intensity within each detected region. Regions with an intensity below a predefined threshold (set to 60 in this study) are considered part of the support or noise and are removed. This filtering significantly enhances the mask quality by ensuring only high-contrast nanoparticle regions are retained, as illustrated in **Supplementary Fig. 7**.

The second issue relates to edge artifacts caused by incomplete segmentation at image boundaries. SAM occasionally detects particles that are partially cropped at the edges of the image, leading to fragmented or incomplete masks. These artifacts can introduce bias in model training, as the presence of truncated particles does not accurately represent the actual physical morphology of nanoparticles. To mitigate this issue, we apply an edge-filtering step that removes segmented regions near the image boundary. Specifically, we label all connected components in the processed mask and discard any regions that intersect a predefined margin (5 pixels) from the image boundary. This approach ensures that only fully contained particles are preserved while filtering out incomplete regions, as demonstrated in **Supplementary Fig. 8**.

Supplementary Note 3. Zero-shot performance of SAM-AMG masking.

Building on the preprocessed SAM-AMG masks in **Supplementary Note 2**, we qualitatively evaluate their segmentation accuracy. As shown in **Supplementary Fig. 9**, a direct comparison with manually labeled ground truth reveals substantial discrepancies, with both false negatives (FNs) and false positives (FPs) occurring frequently.

SAM often fails to detect nanoparticles with low contrast or irregular morphology, leading to a high FN rate. These missed particles are highlighted in **Supplementary Fig. 10**, where manually labeled regions absent in SAM segmentation are marked in red. The tendency of under-segmentation suggests that the model struggles to adapt to the contrast variations and structural complexity inherent in HAADF-STEM imaging.

Conversely, SAM also introduces numerous FPs, mistakenly segmenting background noise or support structures as nanoparticles. As shown in **Supplementary Fig. 11**, these incorrect segmentations are highlighted in blue, indicating that the model lacks domain-specific constraints, leading to excessive background detection.

Supplementary Note 4. Extraction of morphology priors and generation of random masks.

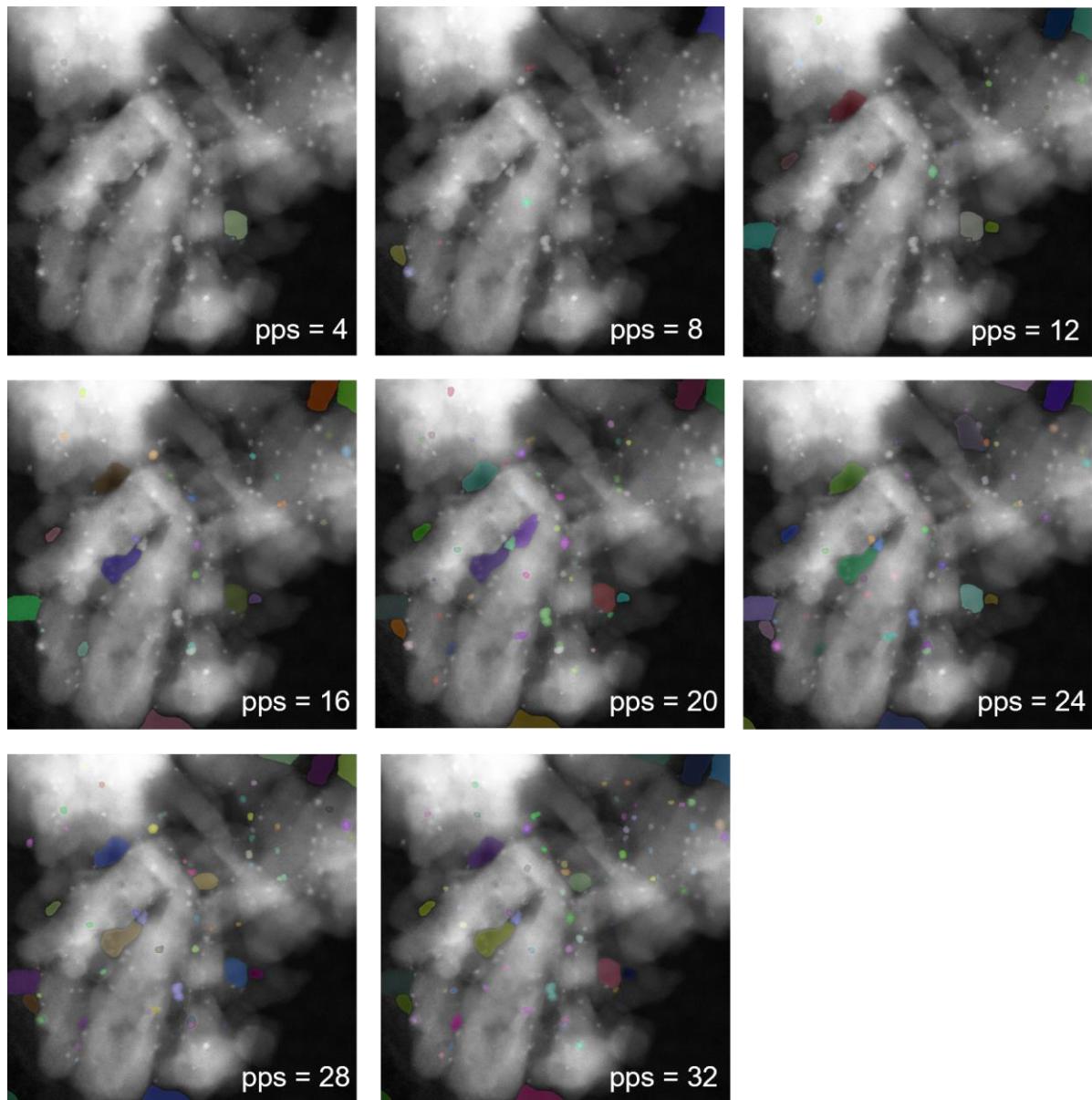
To enhance the representativeness of synthetic masks, we extracted morphology priors from processed SAM masks and leveraged them to generate new masks. The process begins with analyzing the geometric properties of individual particles within the SAM masks. This extraction procedure, illustrated in **Supplementary Algorithm 2**, follows a systematic workflow to measure key morphological features such as area, perimeter, eccentricity, aspect ratio, circularity, and solidity. The resulting statistical distributions for these six descriptors, derived from 33 processed SAM masks containing 1901 particles, are presented in **Supplementary Fig. 15**.

Based on the extracted priors, we implemented a morphology-based mask generation strategy (**Supplementary Algorithm 3**) that rearranges individual particle instances while preserving their morphological integrity. The workflow involves applying affine transformations, including random translation and rotation, to redistribute particles within new masks while maintaining their original statistical distributions (**Supplementary Fig. 12**). Using this approach, each SAM mask was expanded into ten new masks, yielding a total of 330 masks (**Supplementary Fig. 13**).

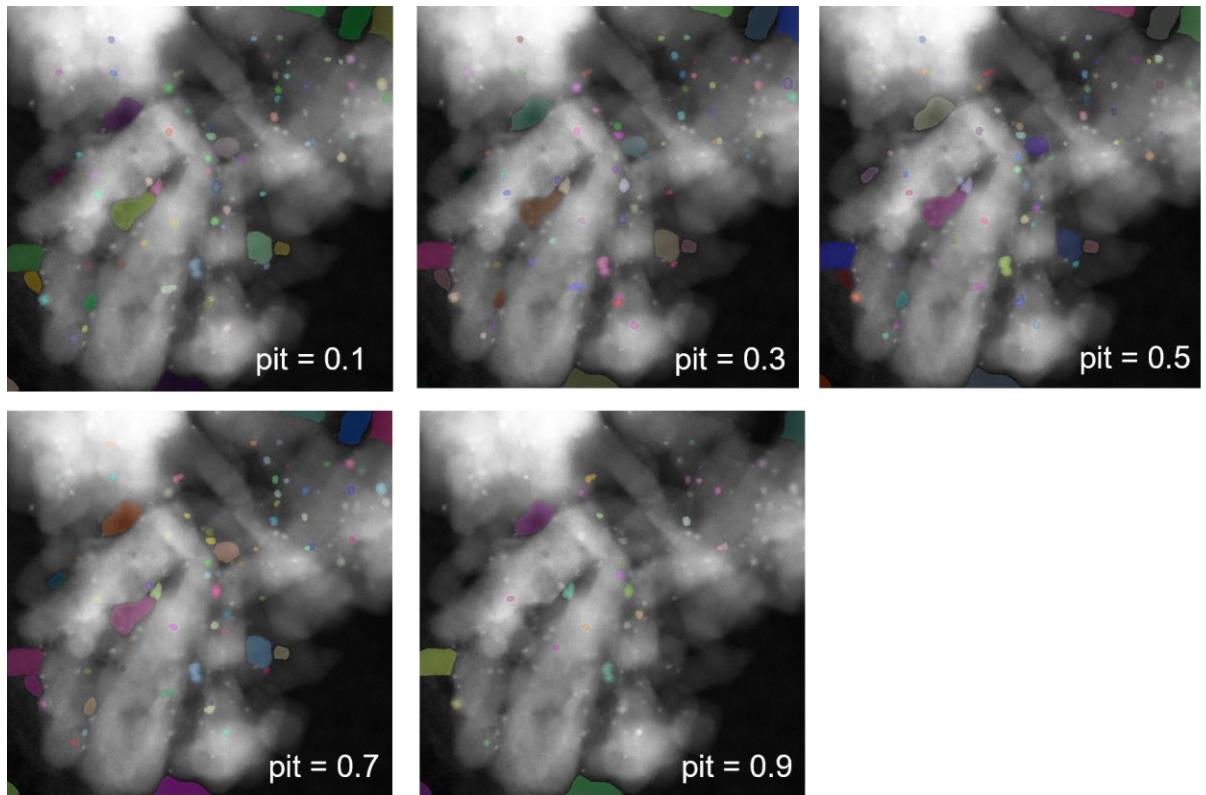
A comparative analysis of the statistical distributions of morphology descriptors provides evidence of the validity of this approach. As shown in **Supplementary Fig. 14**, the particle statistics of the original manually labeled masks (2860 particles) closely match those obtained from SAM-generated masks (**Supplementary Fig. 15**). The morphological properties of the newly generated masks remain highly consistent with the original distributions, as seen in **Supplementary Fig. 13**, confirming the reliability of this morphology-driven augmentation method.

To highlight the importance of morphology priors in mask generation, we further compare our method with an alternative approach based on fully random mask generation. Unlike our morphology-informed strategy, the random method disregards geometric constraints and instead places particles in a purely stochastic manner, as detailed in the random mask generation process (**Supplementary Algorithm 4**). The resulting masks exhibit severe deviations from the expected morphology distributions, as evident in **Supplementary Fig. 19**, where the statistical distributions are markedly different from those observed in SAM and manual masks. This contrast underscores the necessity of integrating morphology priors to ensure the generation of realistic synthetic masks.

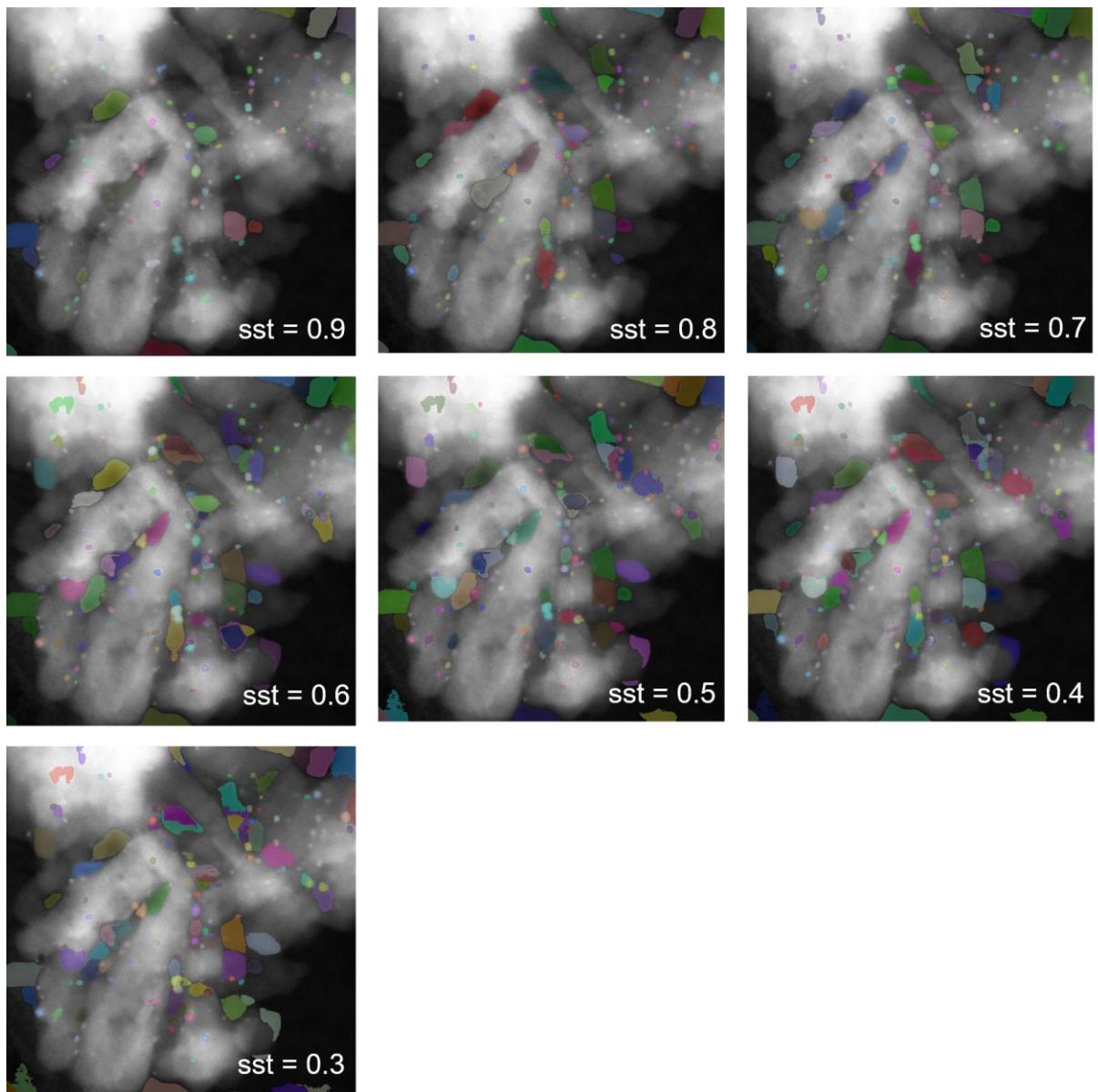
Supplementary Figures



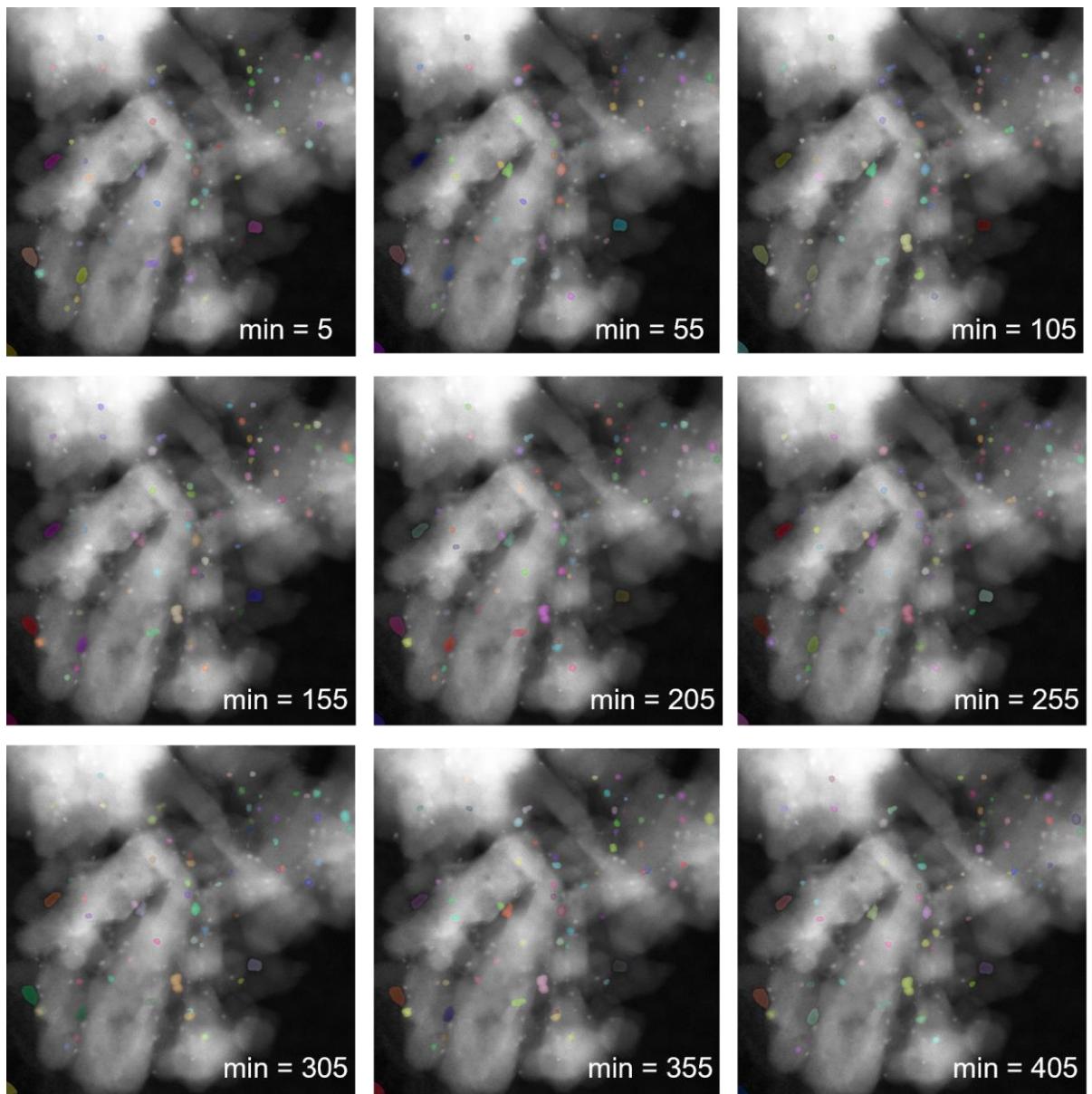
Supplementary Fig. 1. The effect of parameter `points_per_side` (pps) on SAM-AMG.



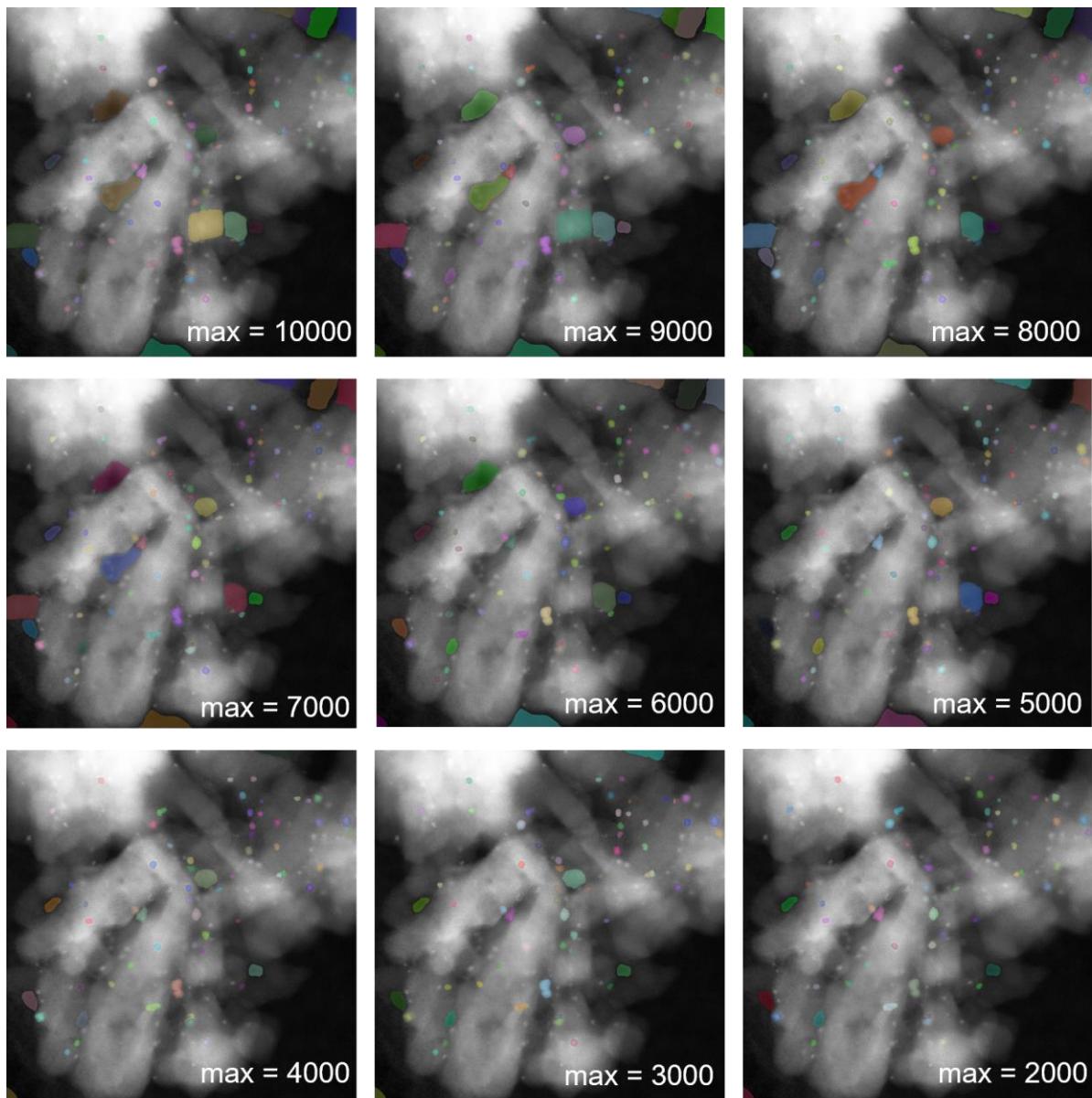
Supplementary Fig. 2 The effect of parameter *pred_iou_thresh* (pit) on SAM-AMG.



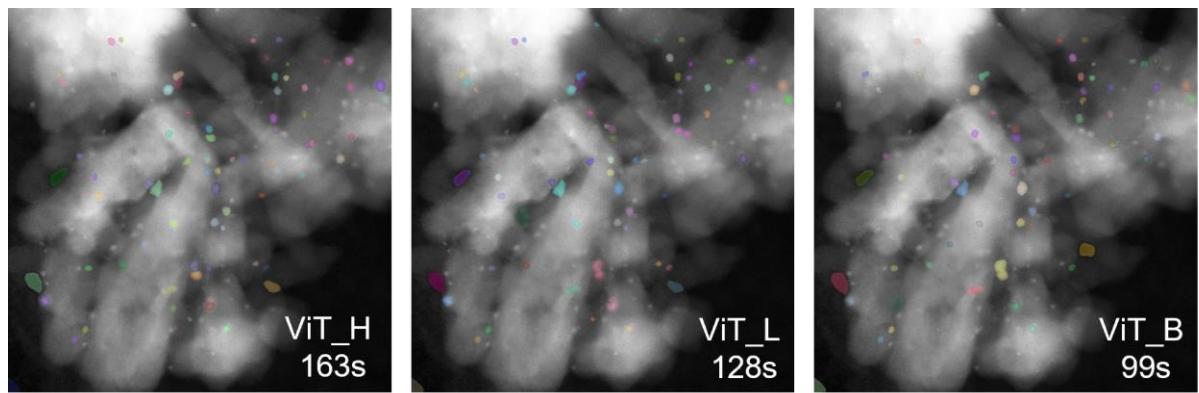
Supplementary Fig. 3. The effect of parameter *stability_score_thresh* (sst) on SAM-AMG.



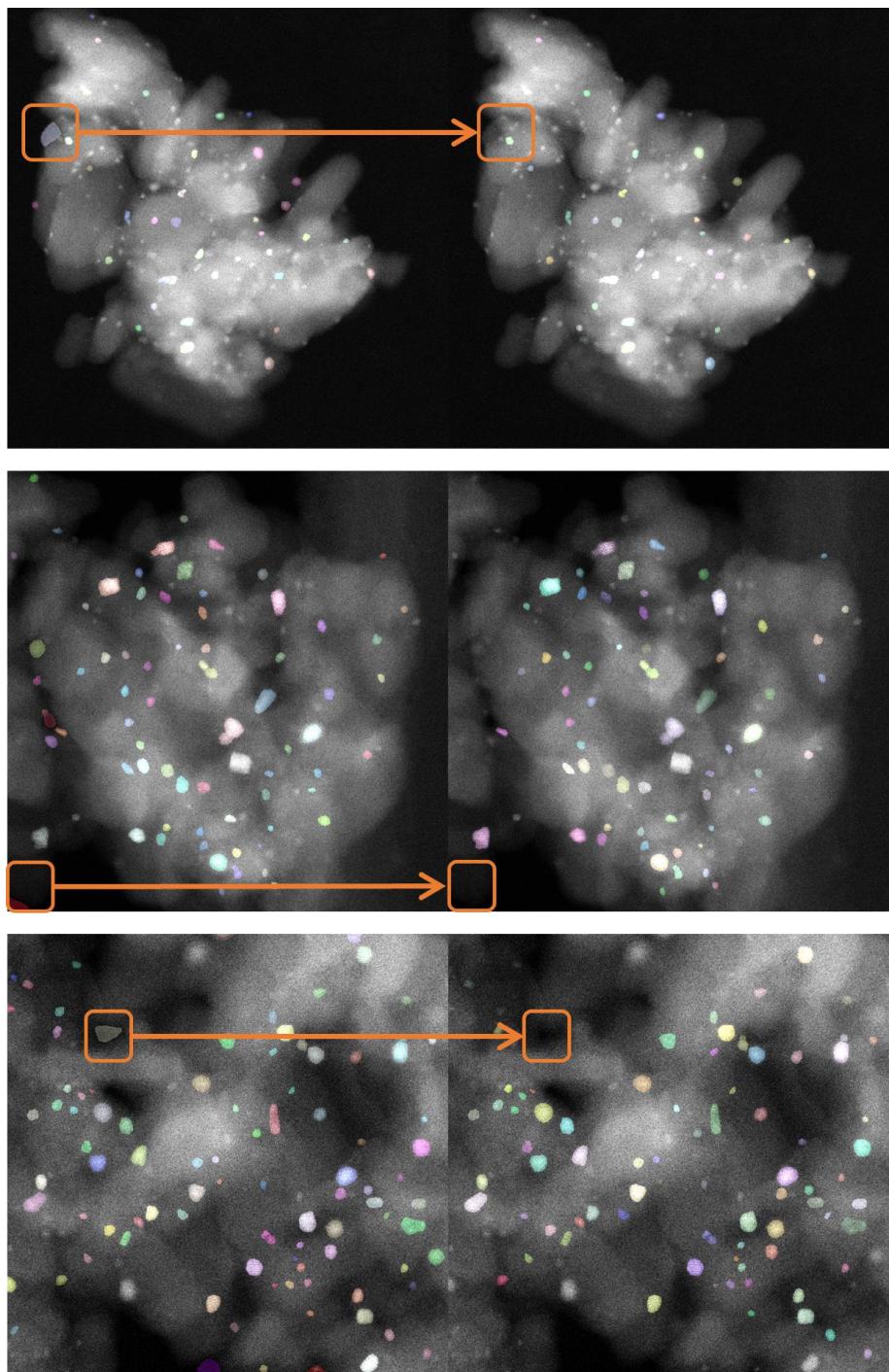
Supplementary Fig. 4. The effect of *min_mask_region_area* on SAM-AMG.



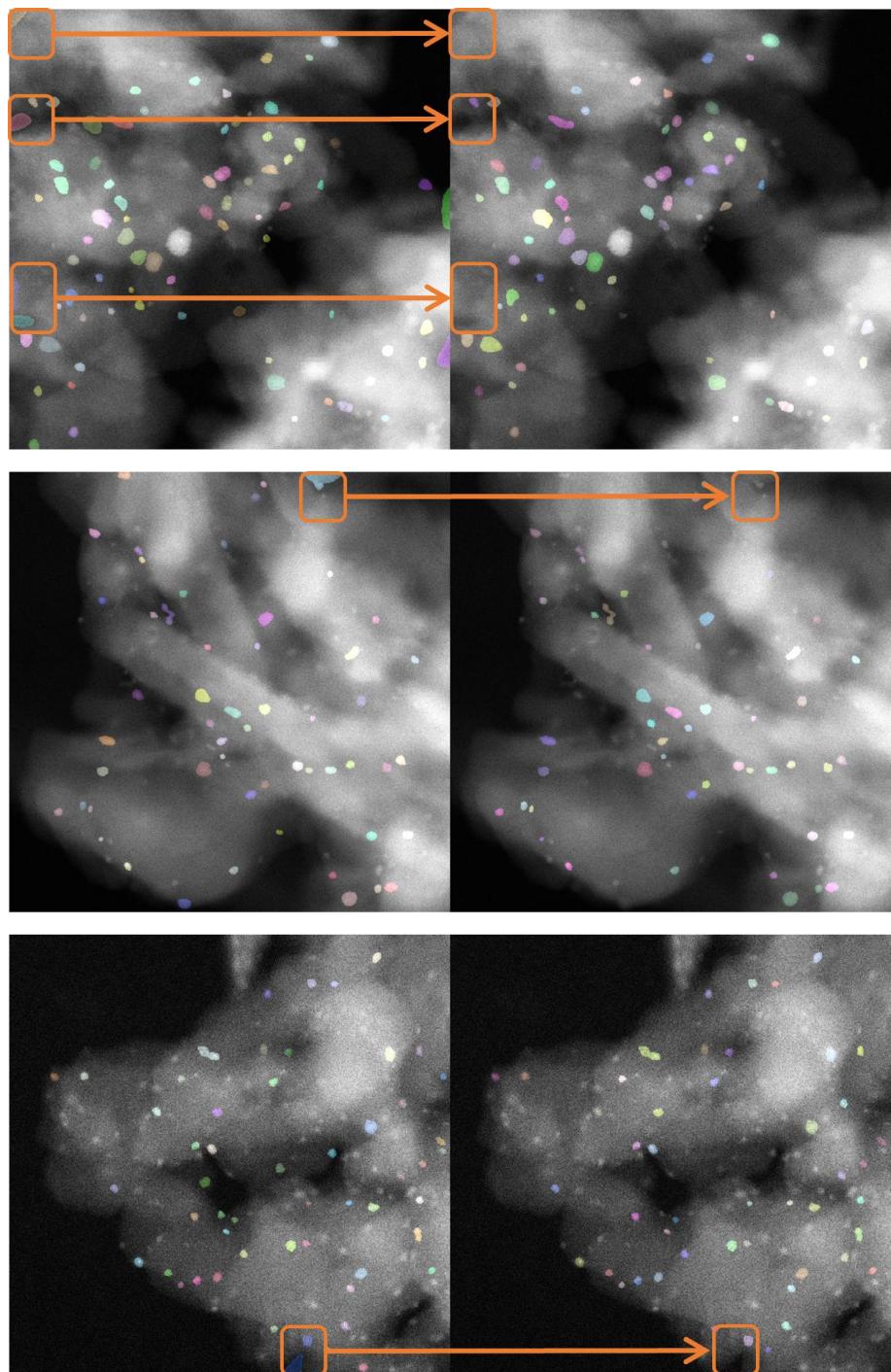
Supplementary Fig. 5. The effect of *max_mask_region_area* on SAM-AMG.



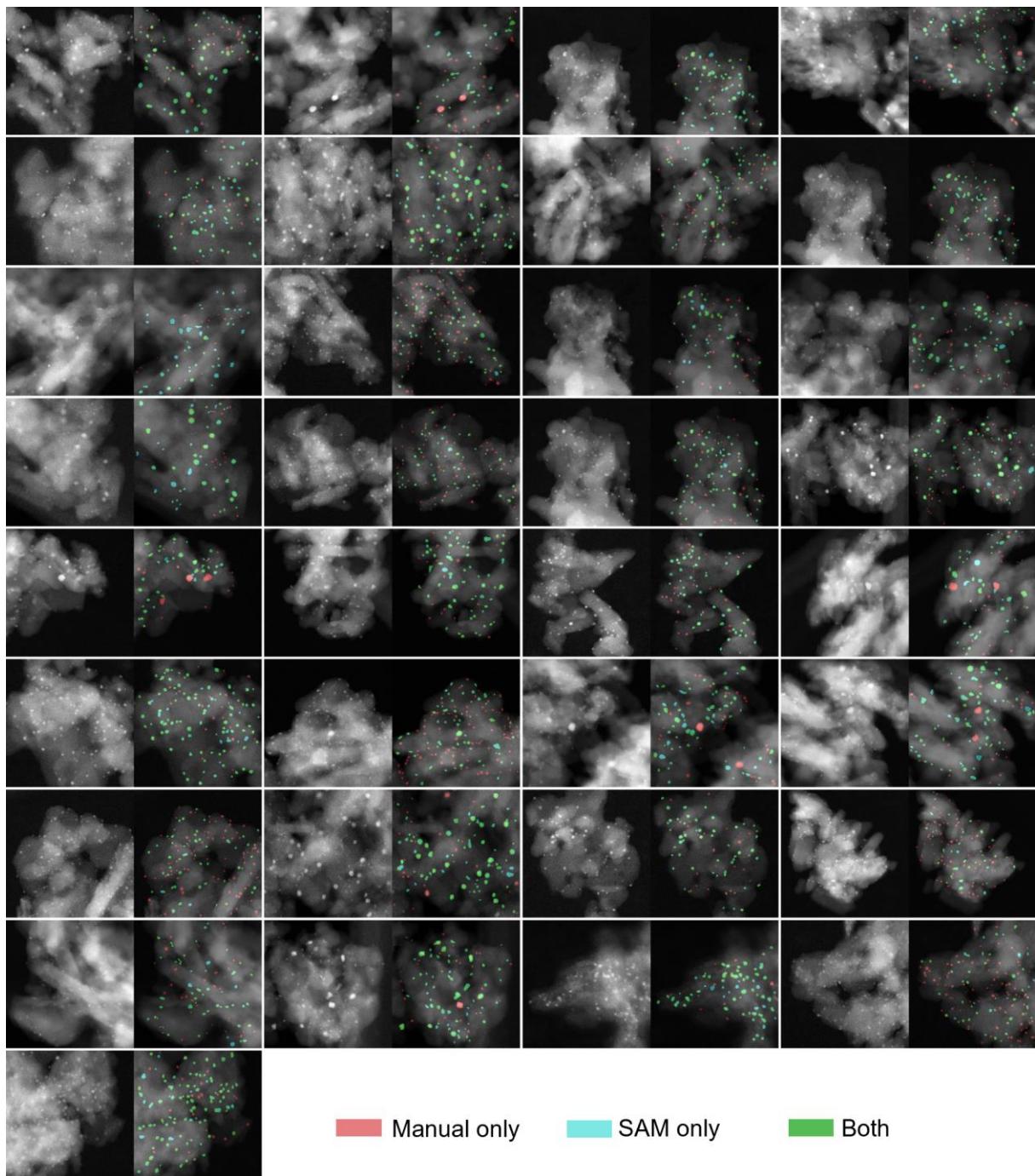
Supplementary Fig. 6. The effect and time cost of model size on SAM-AMG



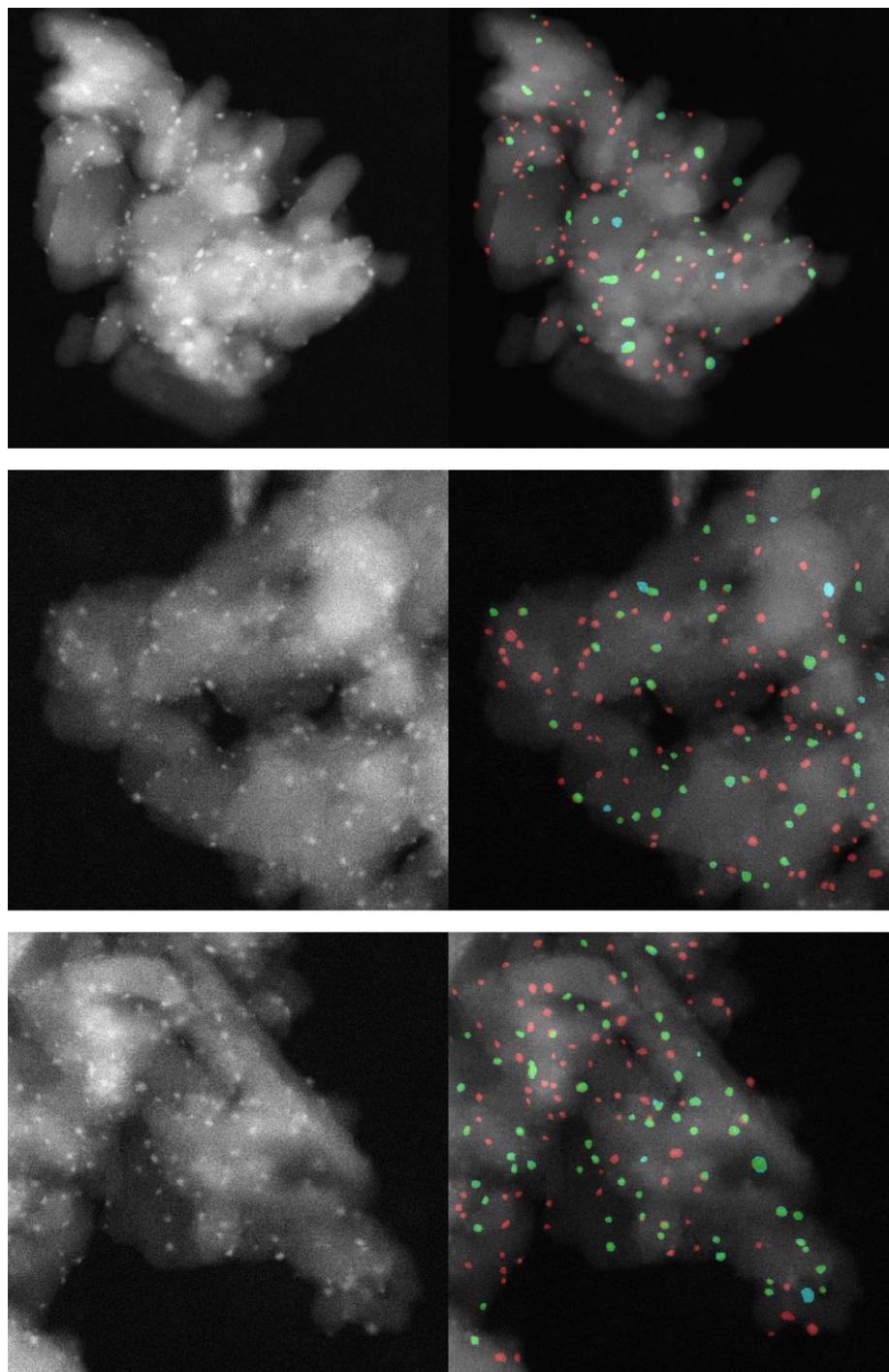
Supplementary Fig. 7 Examples of processing the SAM-masks to filter background and support regions.



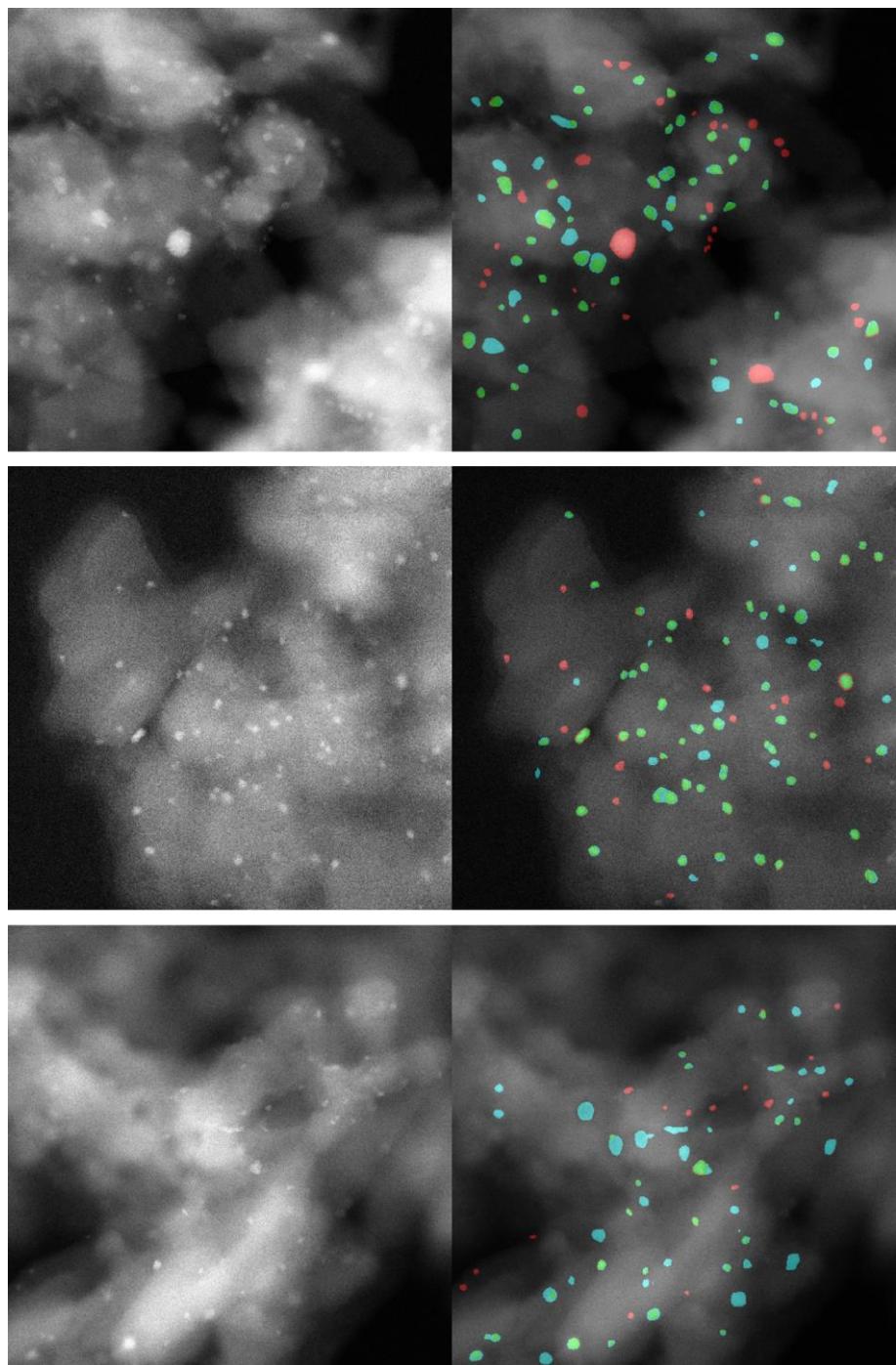
Supplementary Fig. 8. Examples of processing the SAM-masks to filter incomplete regions on image edge.



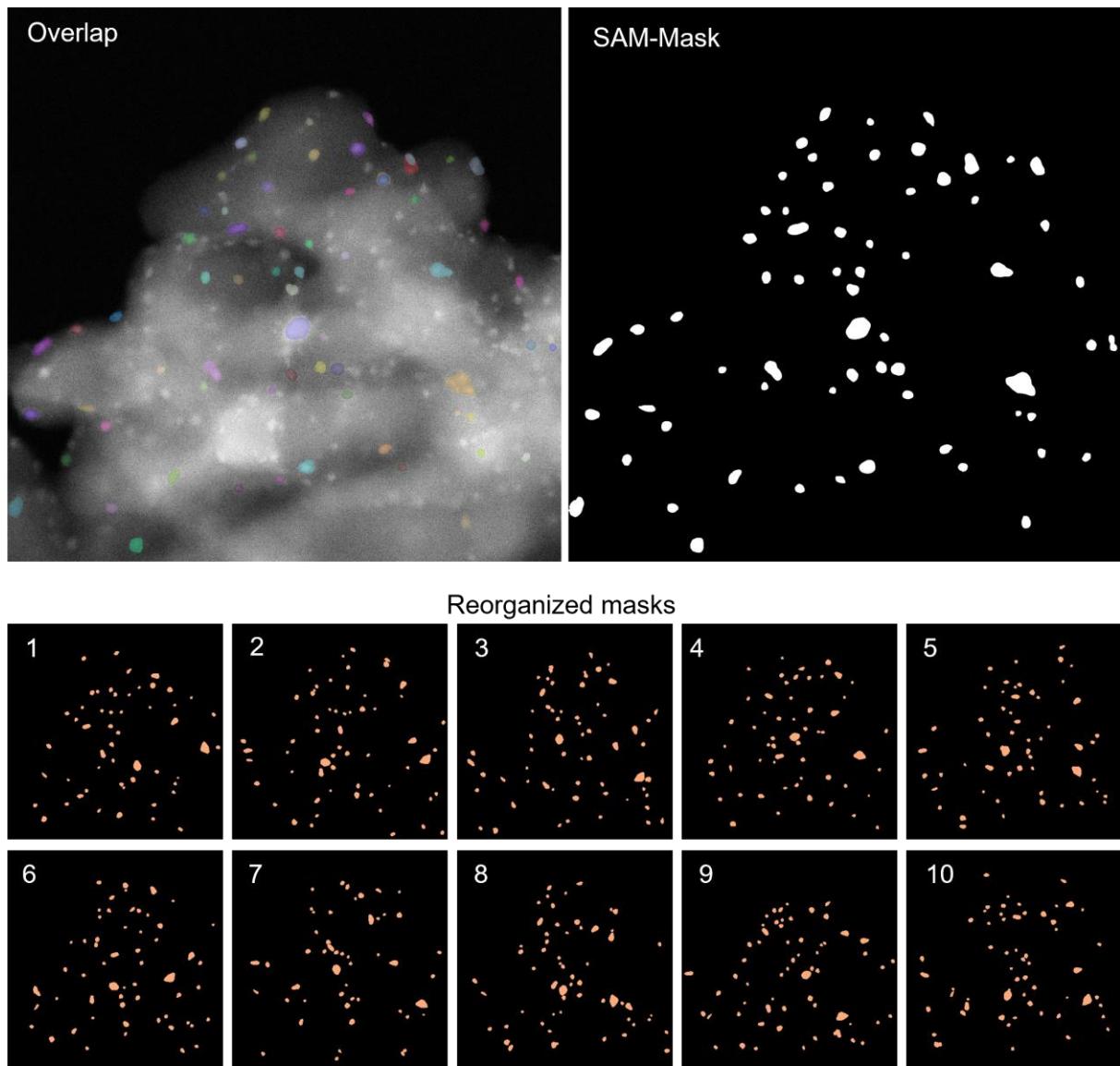
Supplementary Fig. 9. Visualization of mask difference between SAM and manual-labeled ground truth.



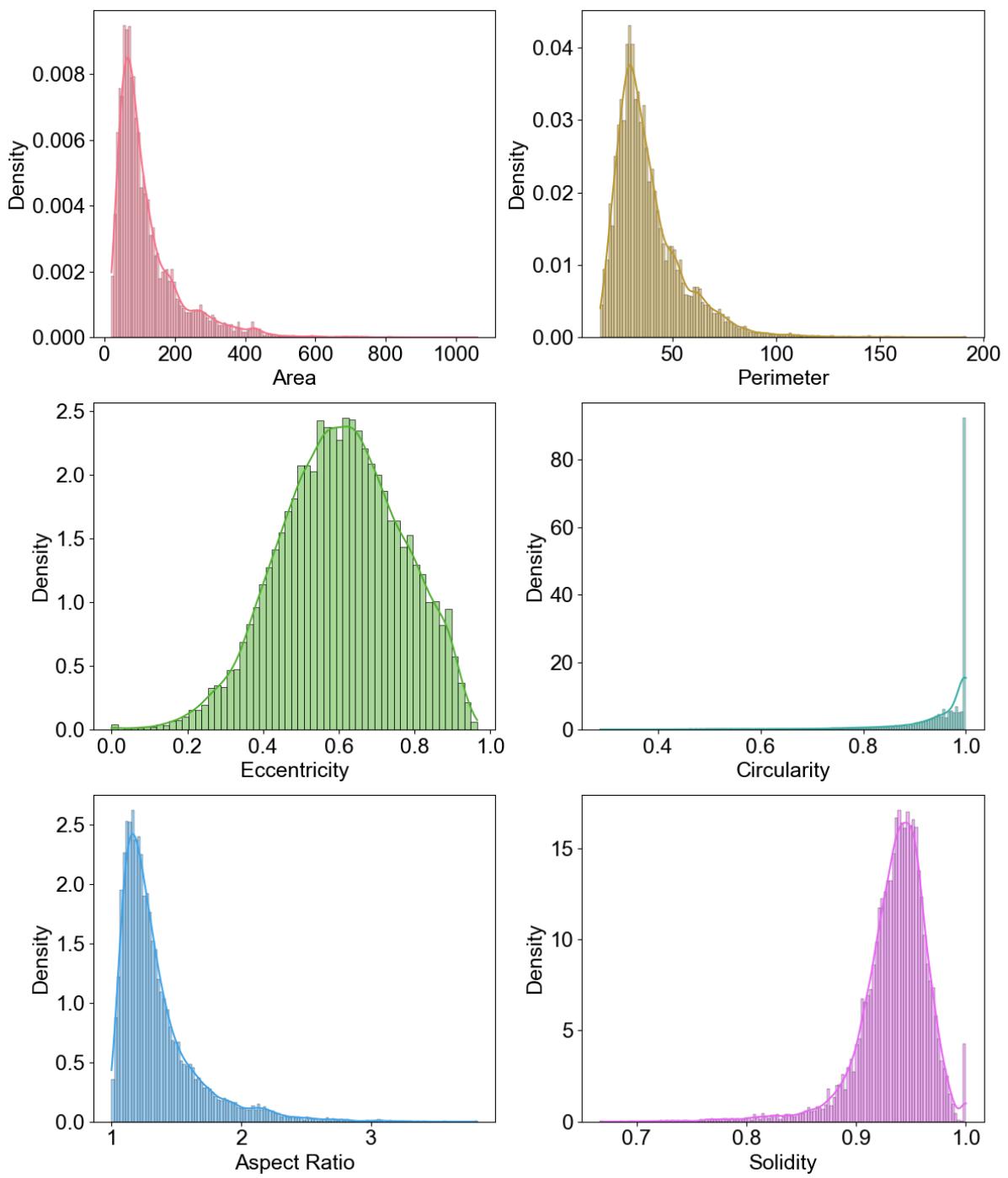
Supplementary Fig. 10. Examples of false-negatives (red) using SAM masking.



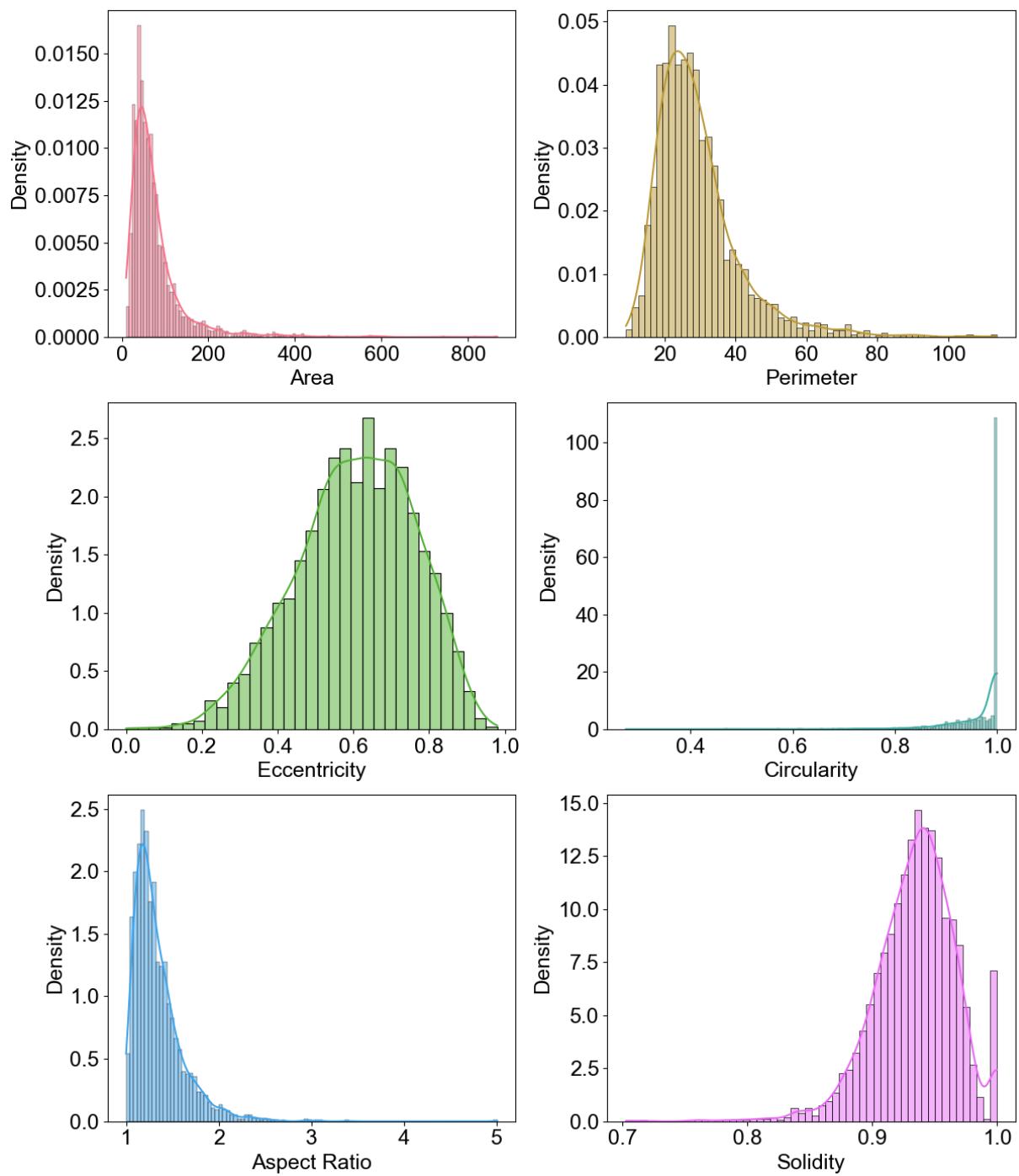
Supplementary Fig. 11. Examples of false-positives (blue) using SAM masking.



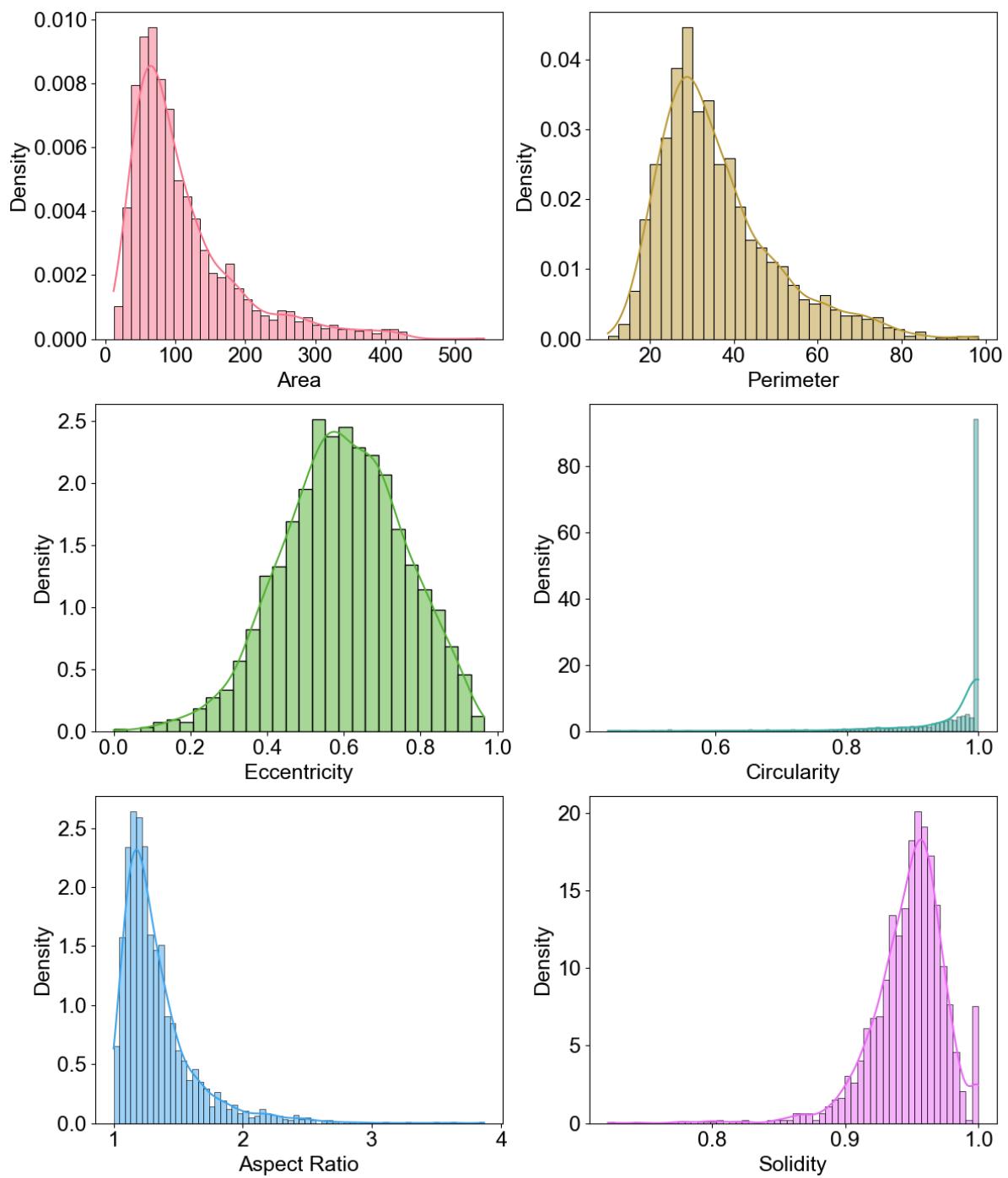
Supplementary Fig. 12. Examples of 10 reorganized masks based on SAM-mask.



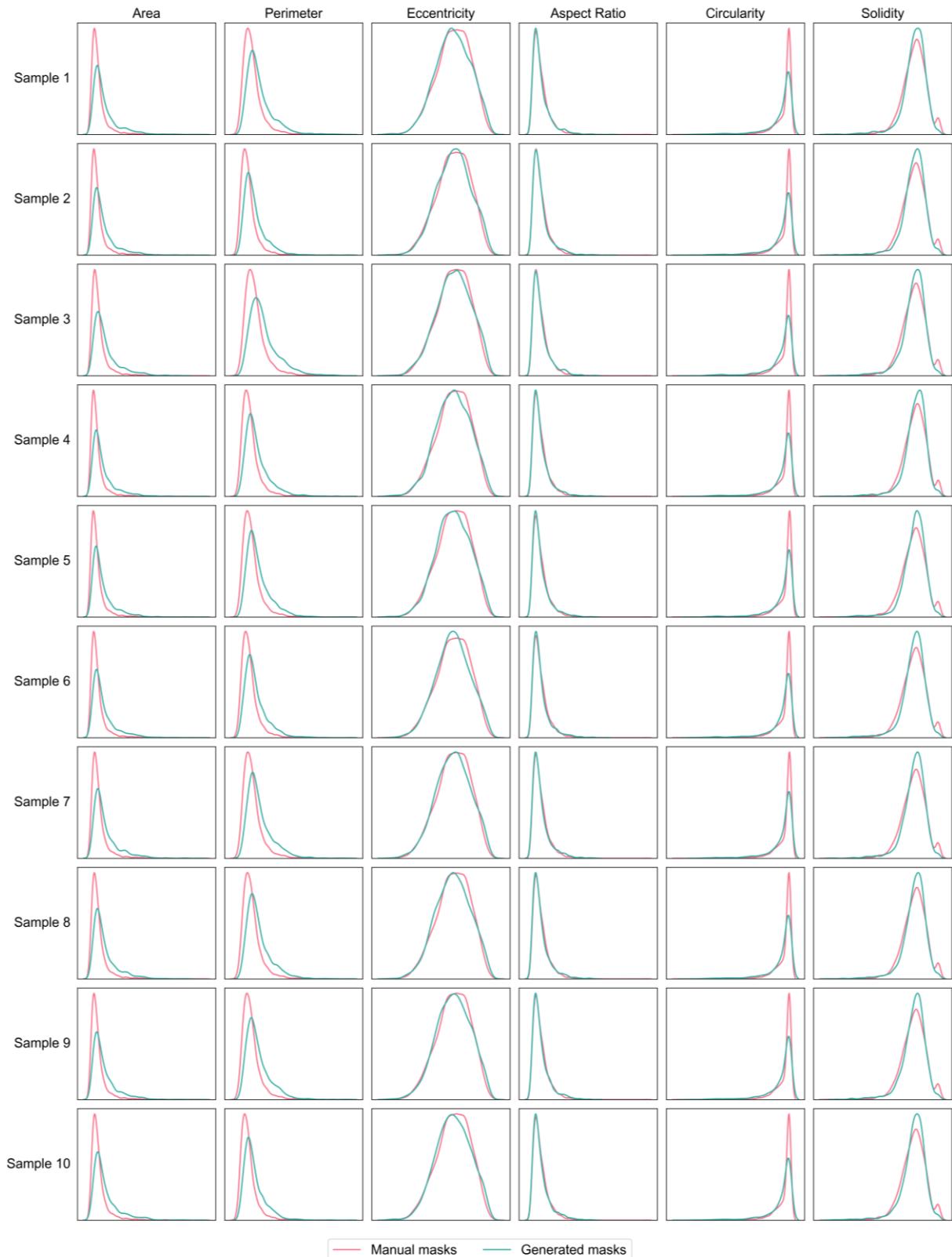
Supplementary Fig. 13. Histograms and KDE-plots of nanoparticles statistics from 330 masks generated from morphology priors (17420 particles).



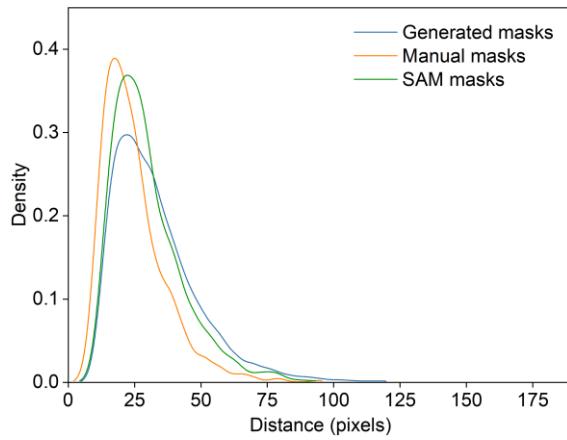
Supplementary Fig. 14. Histograms and KDE-plots of nanoparticles statistics from original 33 masks by manual labeling (2860 particles).



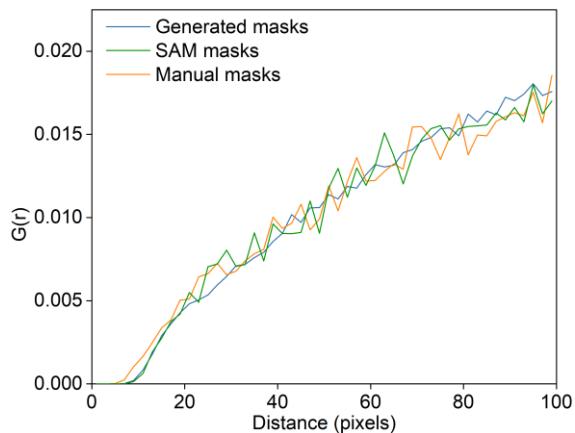
Supplementary Fig. 15. Histograms and KDE-plots of nanoparticles statistics from 33 original masks generated by SAM (1901 particles).



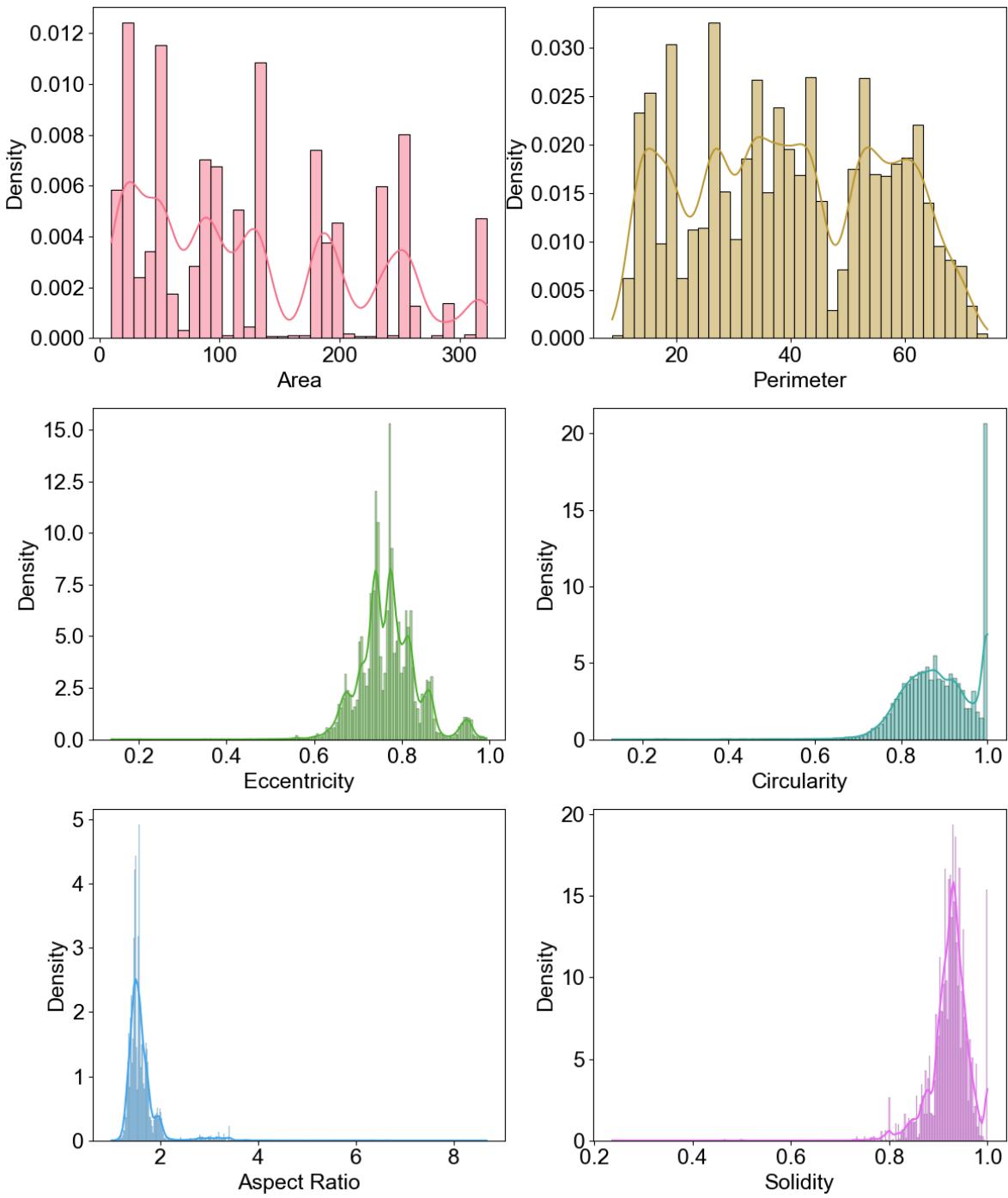
Supplementary Fig. 16. KDE-plots between manually labeled masks (red, 2,860 objects) and 10 randomly sampled subsets (green, 2,860 objects) from the full set of generated masks (33 images with 17,420 objects).



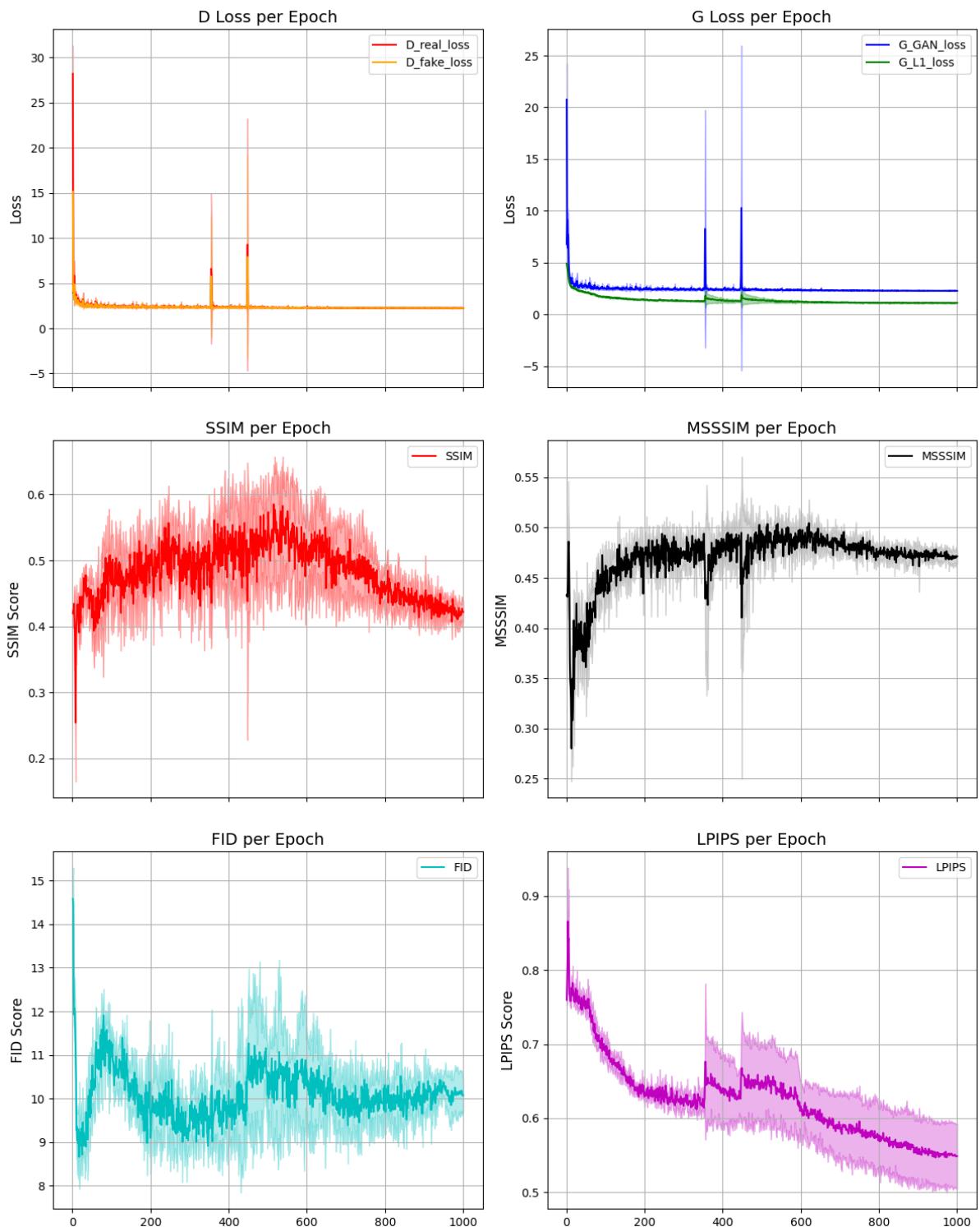
Supplementary Fig. 17. KDE-plots of the nearest neighboring distances (NND) for each nanoparticle of the generated masks (17,420 objects from 330 images), manual masks (2,860 objects from 33 images), and SAM masks (1,901 objects from 330 images).



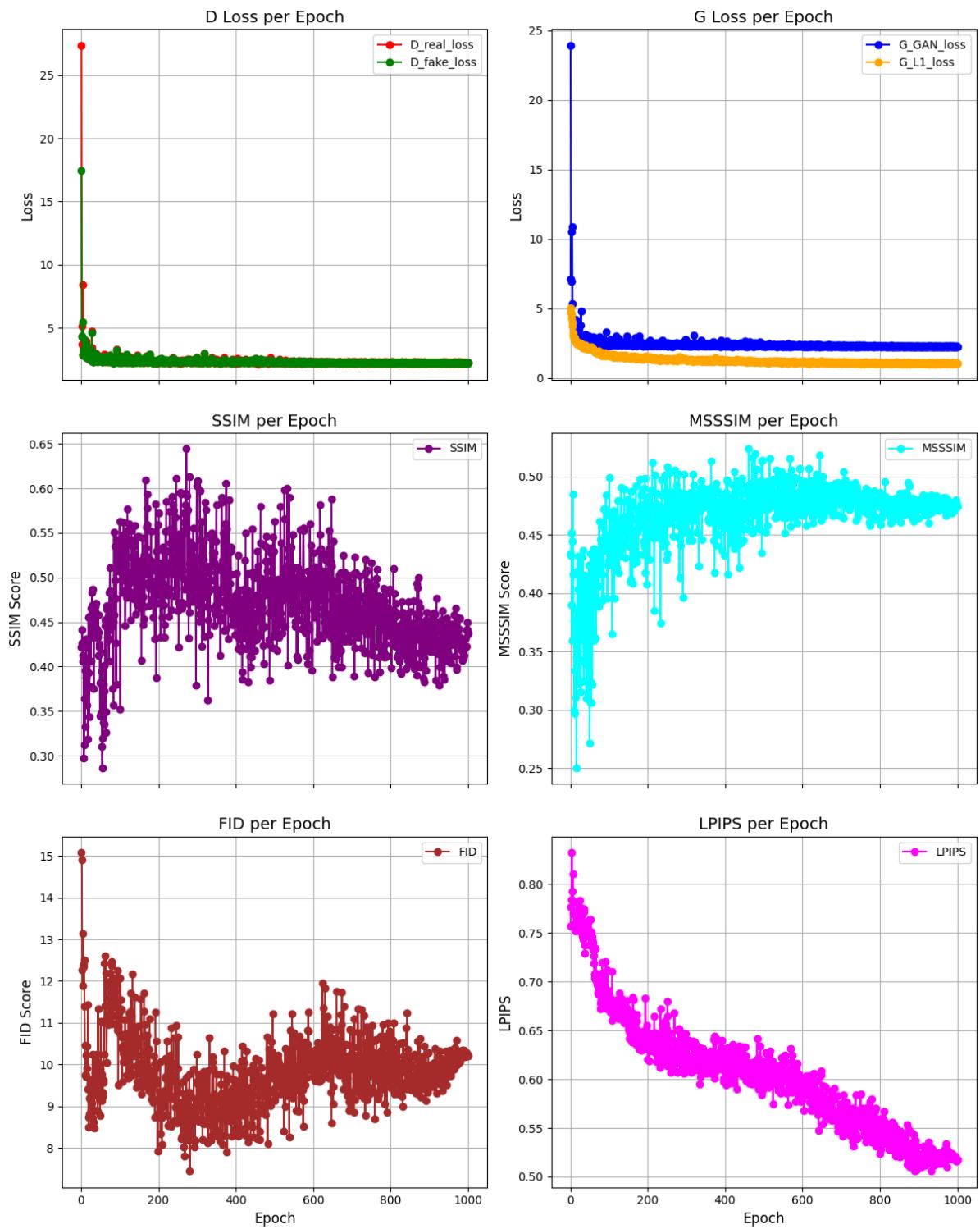
Supplementary Fig. 18. Pair distribution function $G(r)$ (PDF) averaged on each nanoparticle of the generated masks (17,420 objects from 330 images), manual masks (2,860 objects from 33 images), and SAM masks (1,901 objects from 330 images).



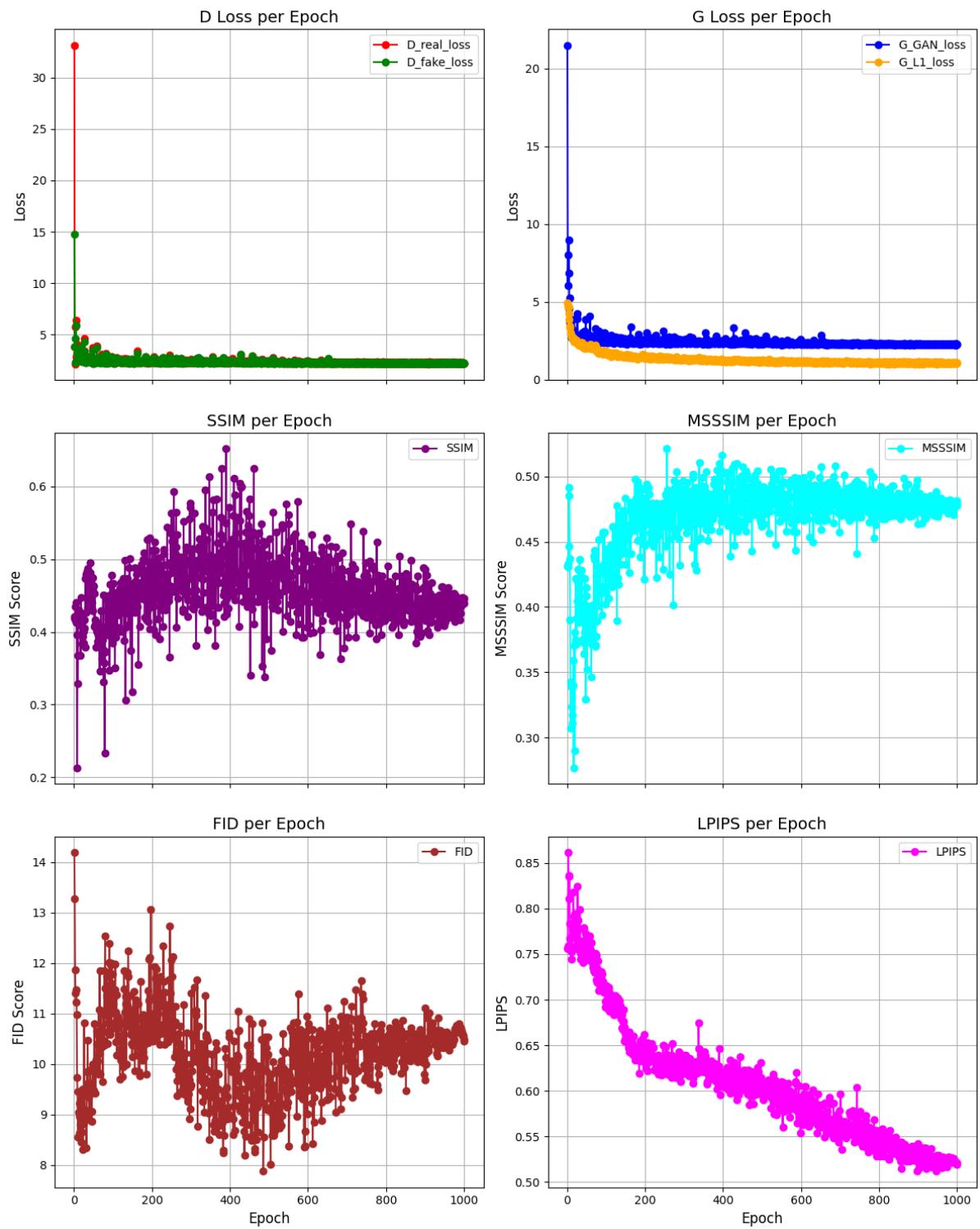
Supplementary Fig. 19. Histograms and KDE-plots of nanoparticles statistics from 330 totally random generated masks (23317 particles).



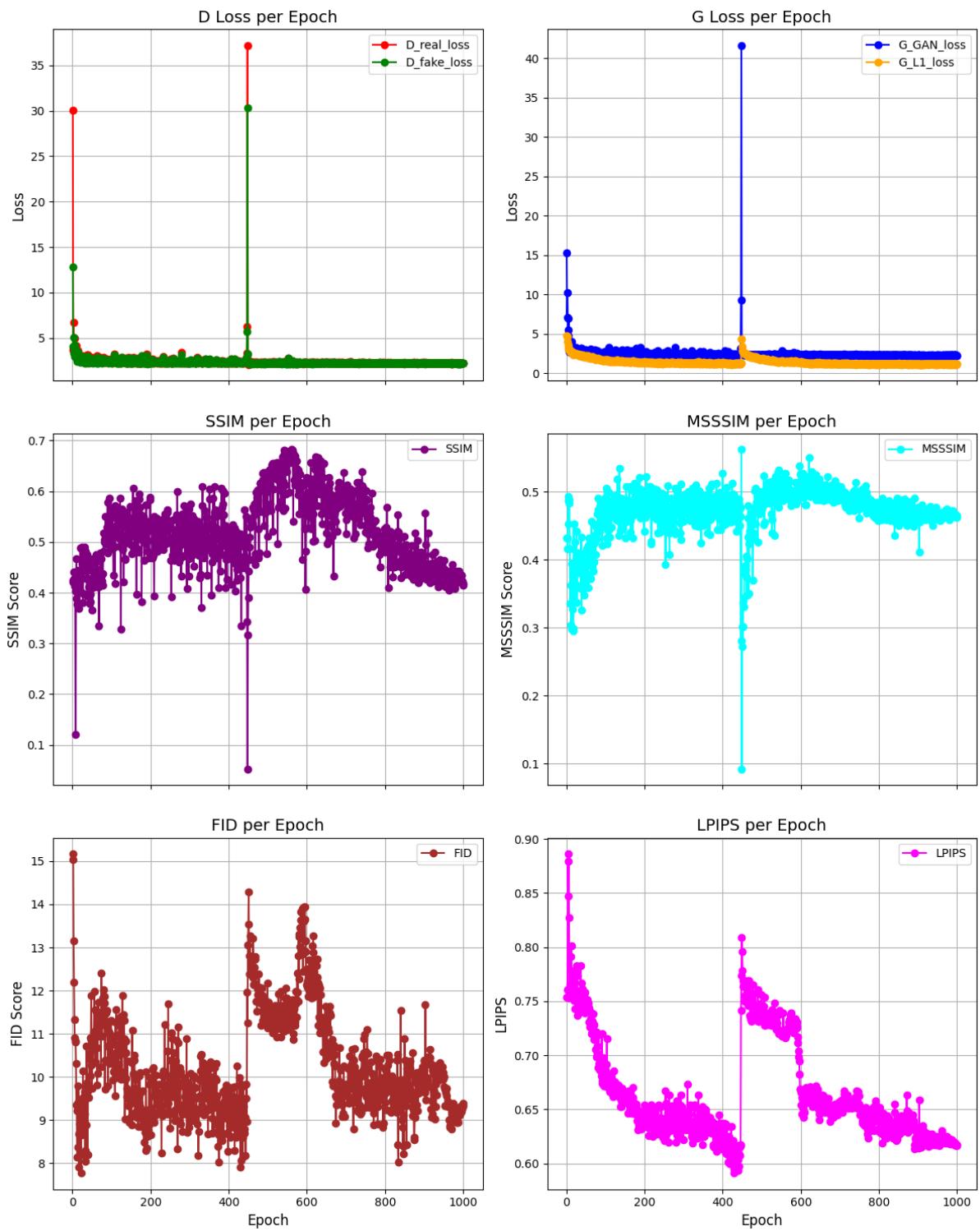
Supplementary Fig. 20. Loss and validation metrics of pix2pix trained for 1000 epochs on SAM coarse masks.



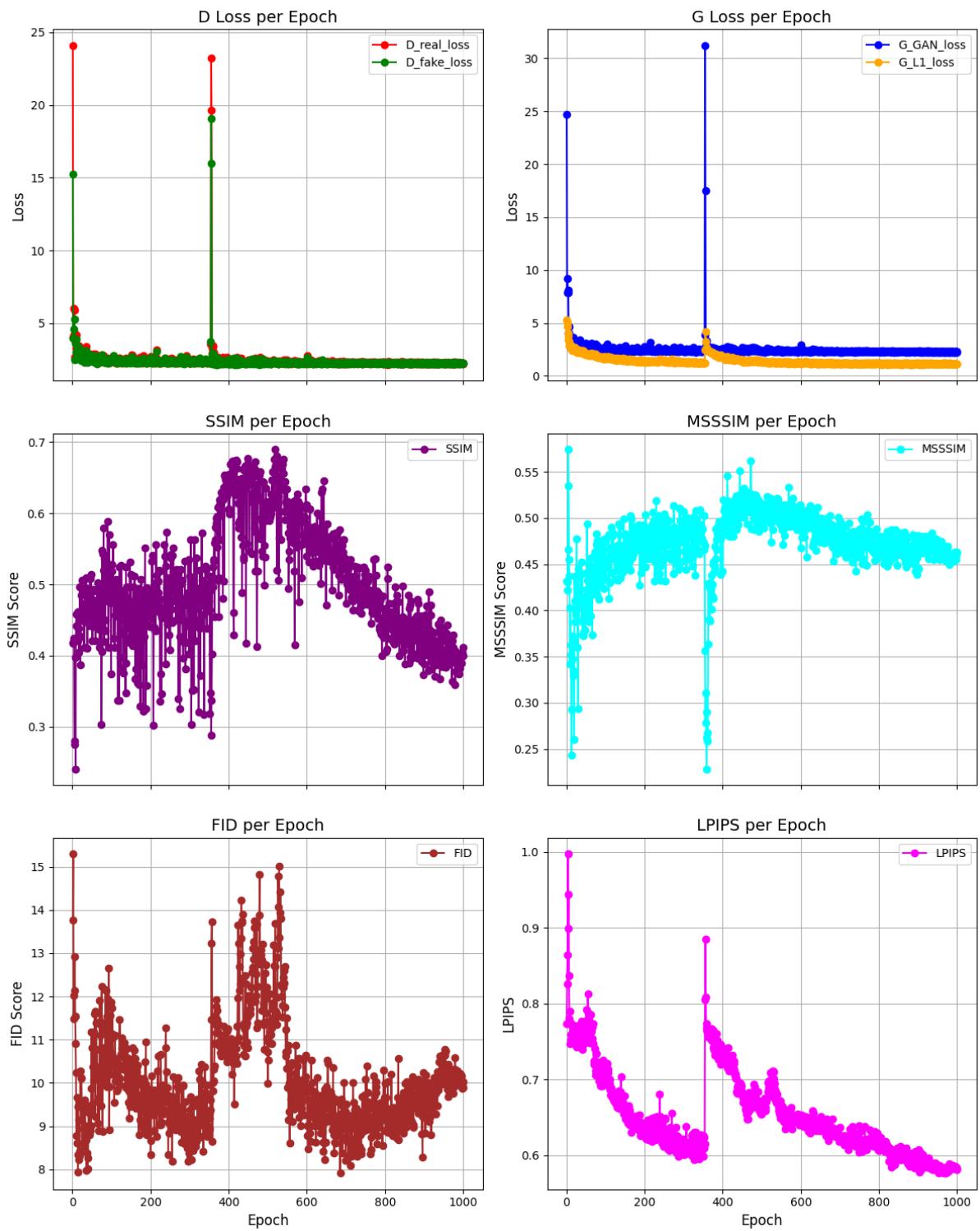
Supplementary Fig. 21. 1st round pix2pix training log on SAM coarse masks.



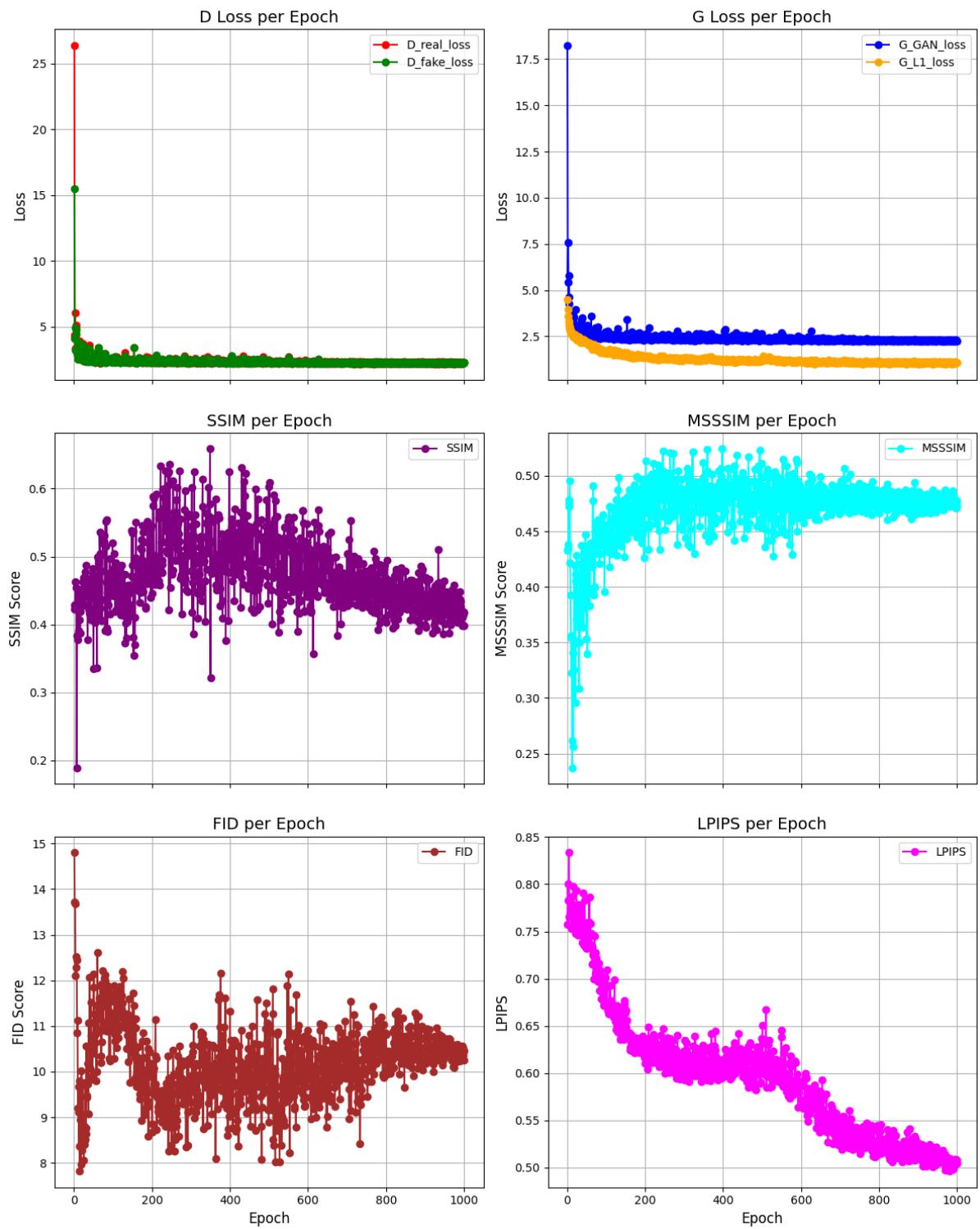
Supplementary Fig. 22. 2nd round pix2pix training log on SAM coarse masks.



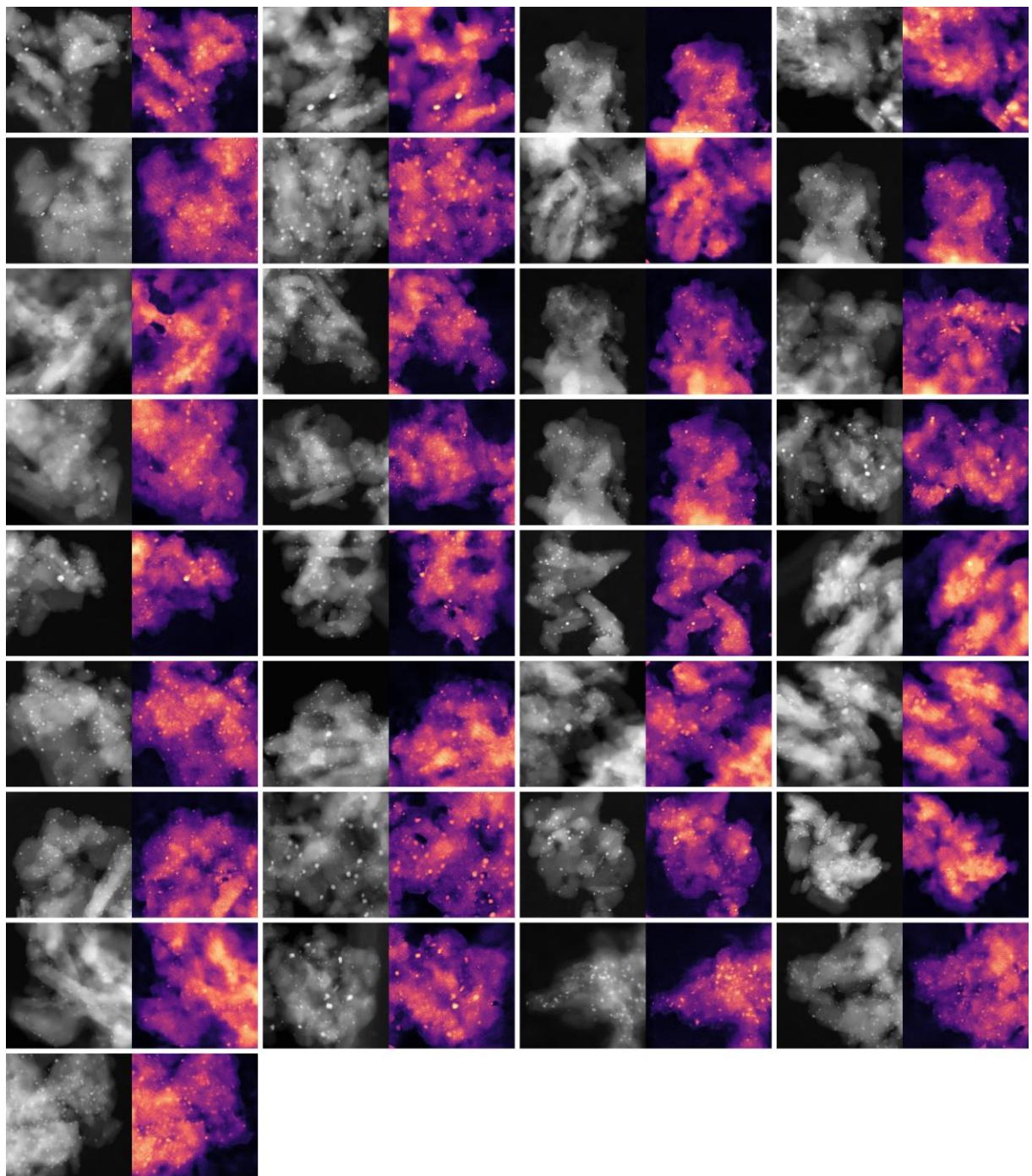
Supplementary Fig. 23. 3nd round pix2pix training log on SAM coarse masks.



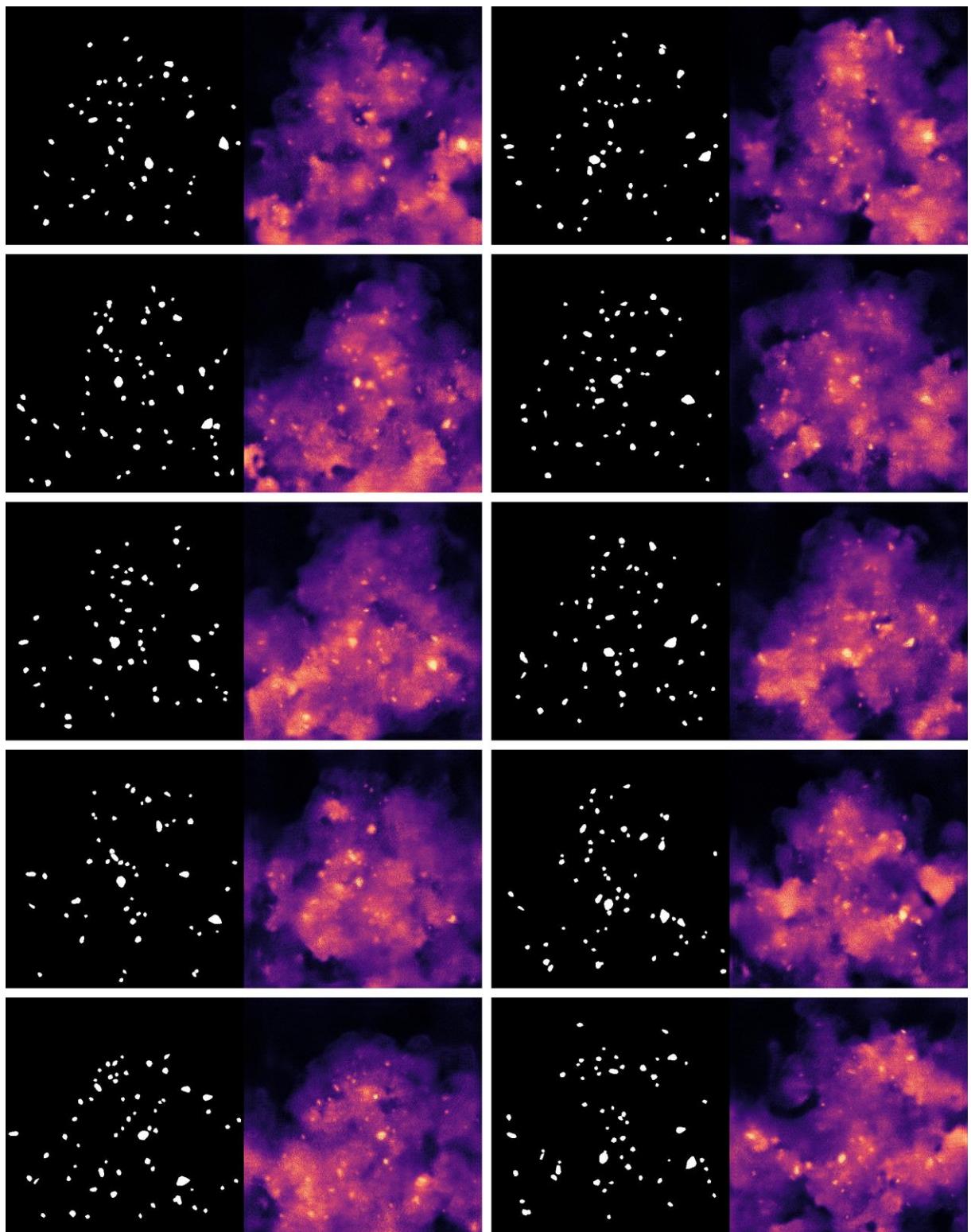
Supplementary Fig. 24. 4th round pix2pix training log on SAM coarse masks.



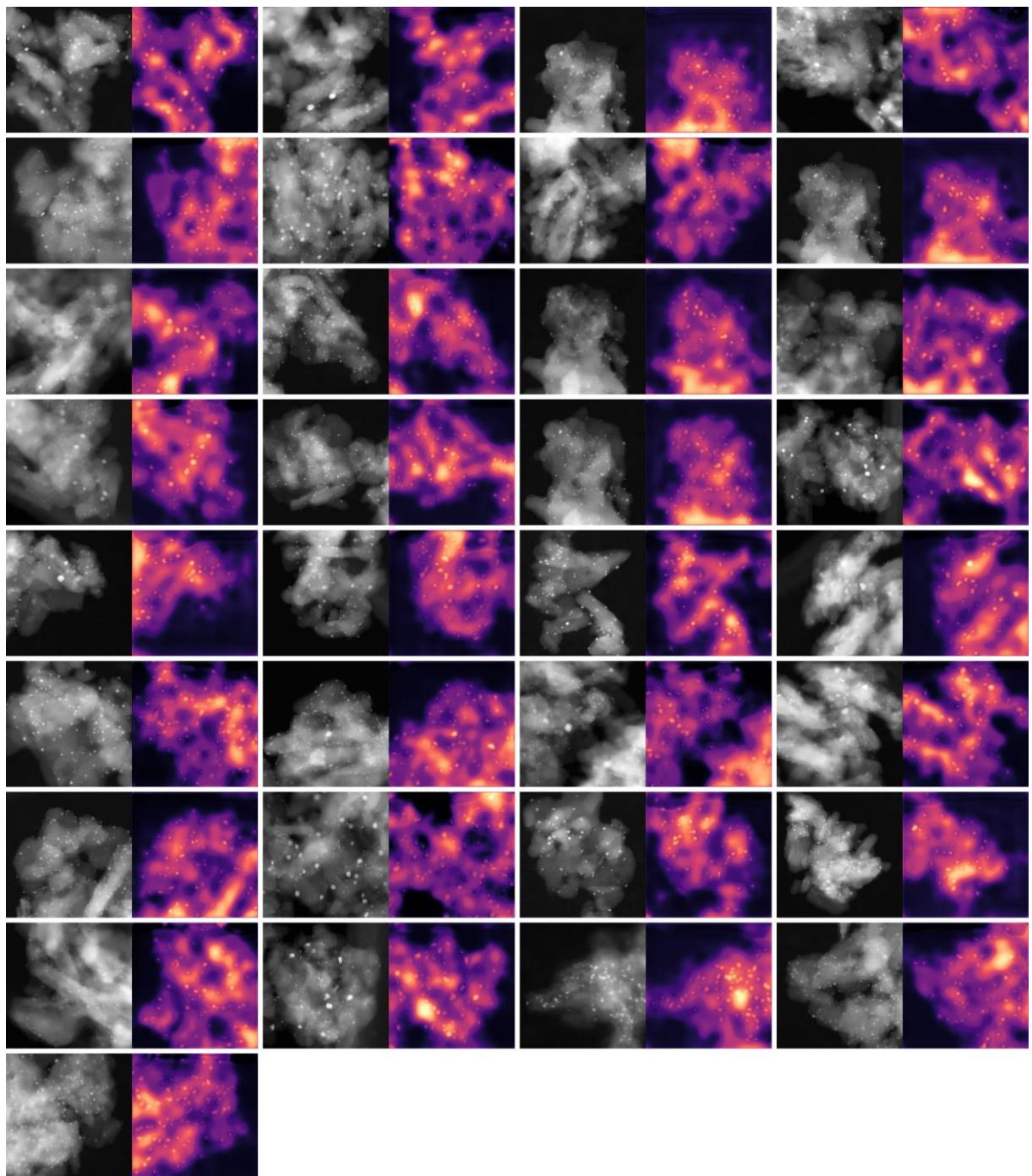
Supplementary Fig. 25. 5th round pix2pix training log on SAM coarse masks.



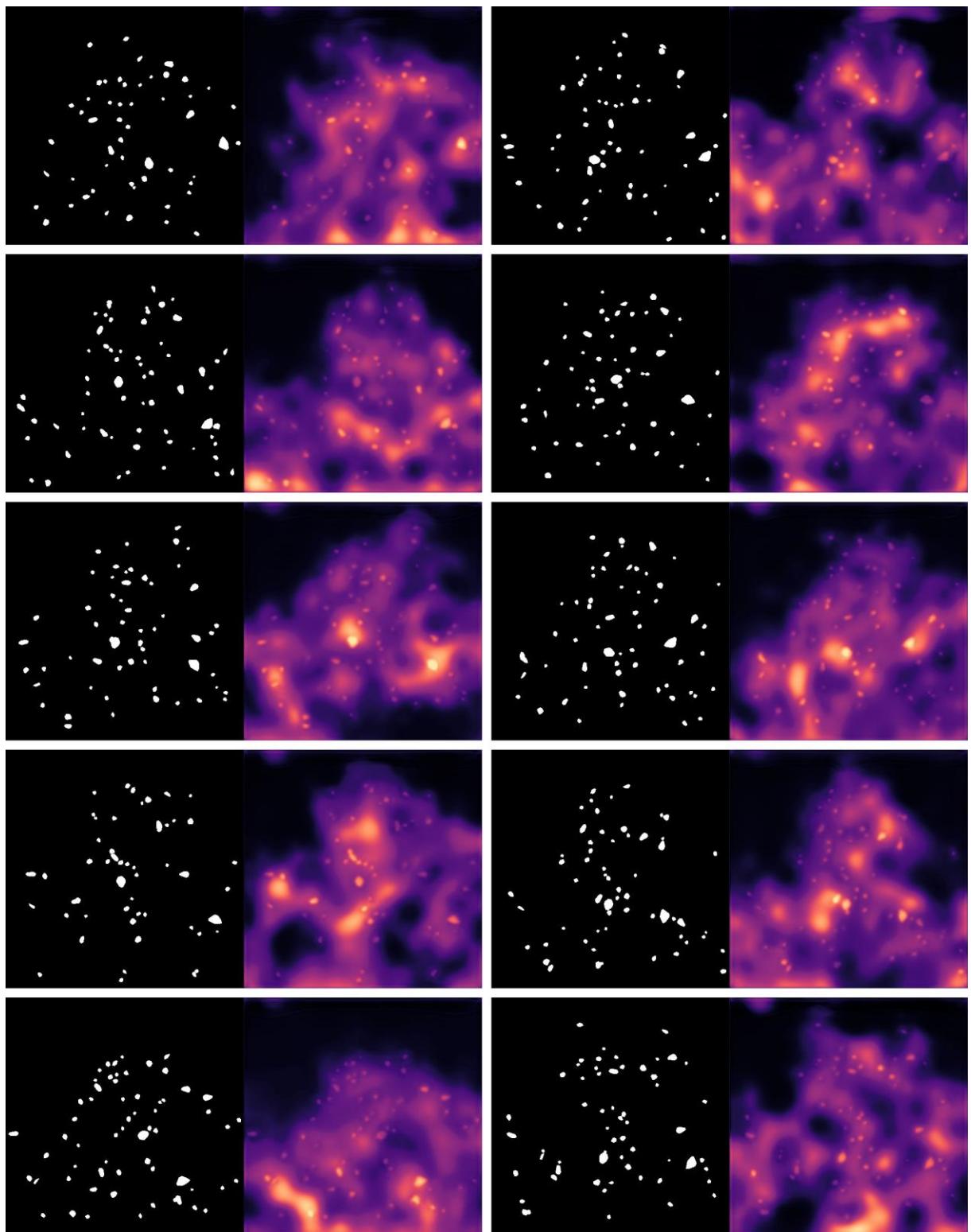
Supplementary Fig. 26. Comparison between the experimental images (gray) and the best LPIPS generation from SAM masks (magma).



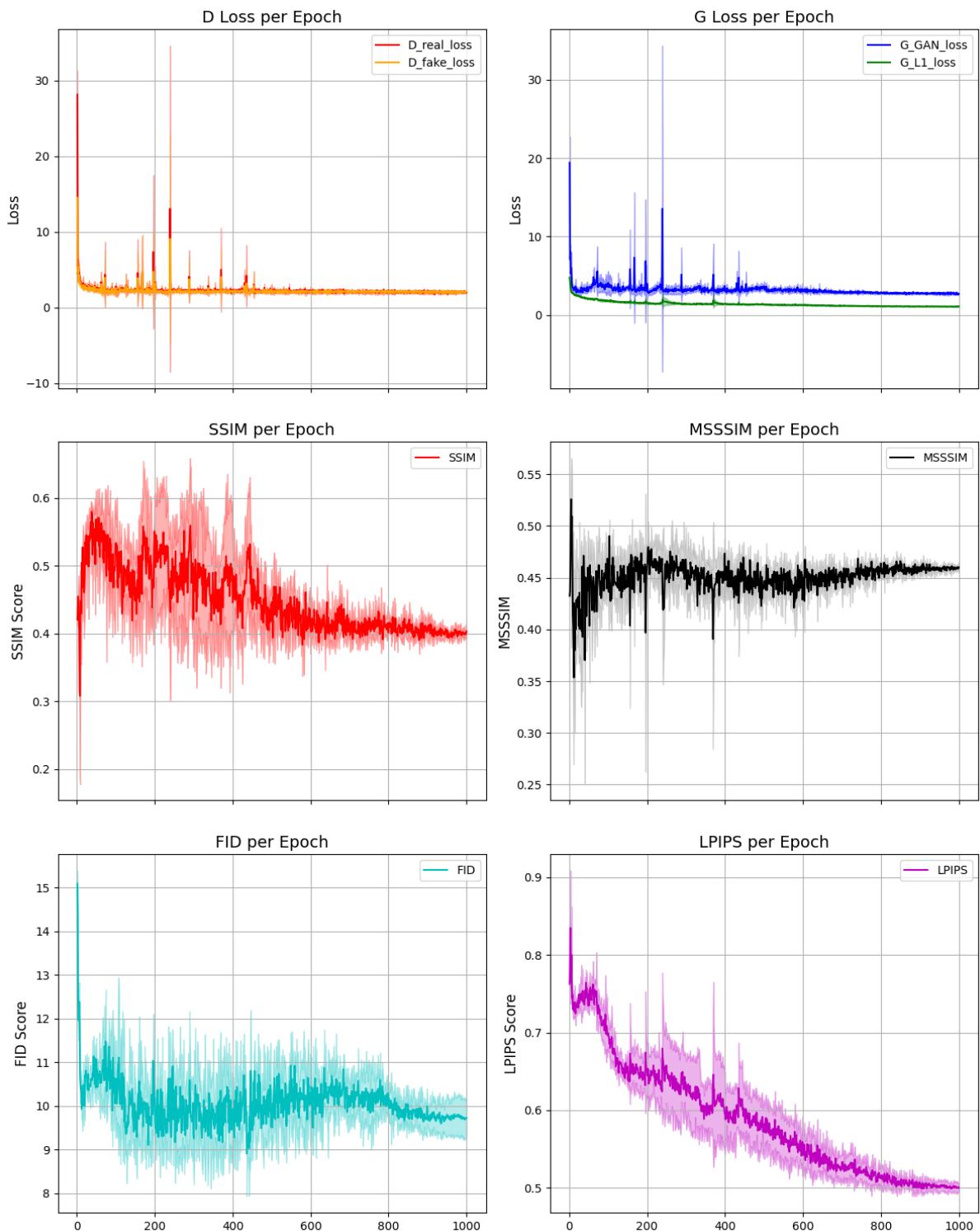
Supplementary Fig. 27. Synthetic images of the best LPIPS model from random masks.



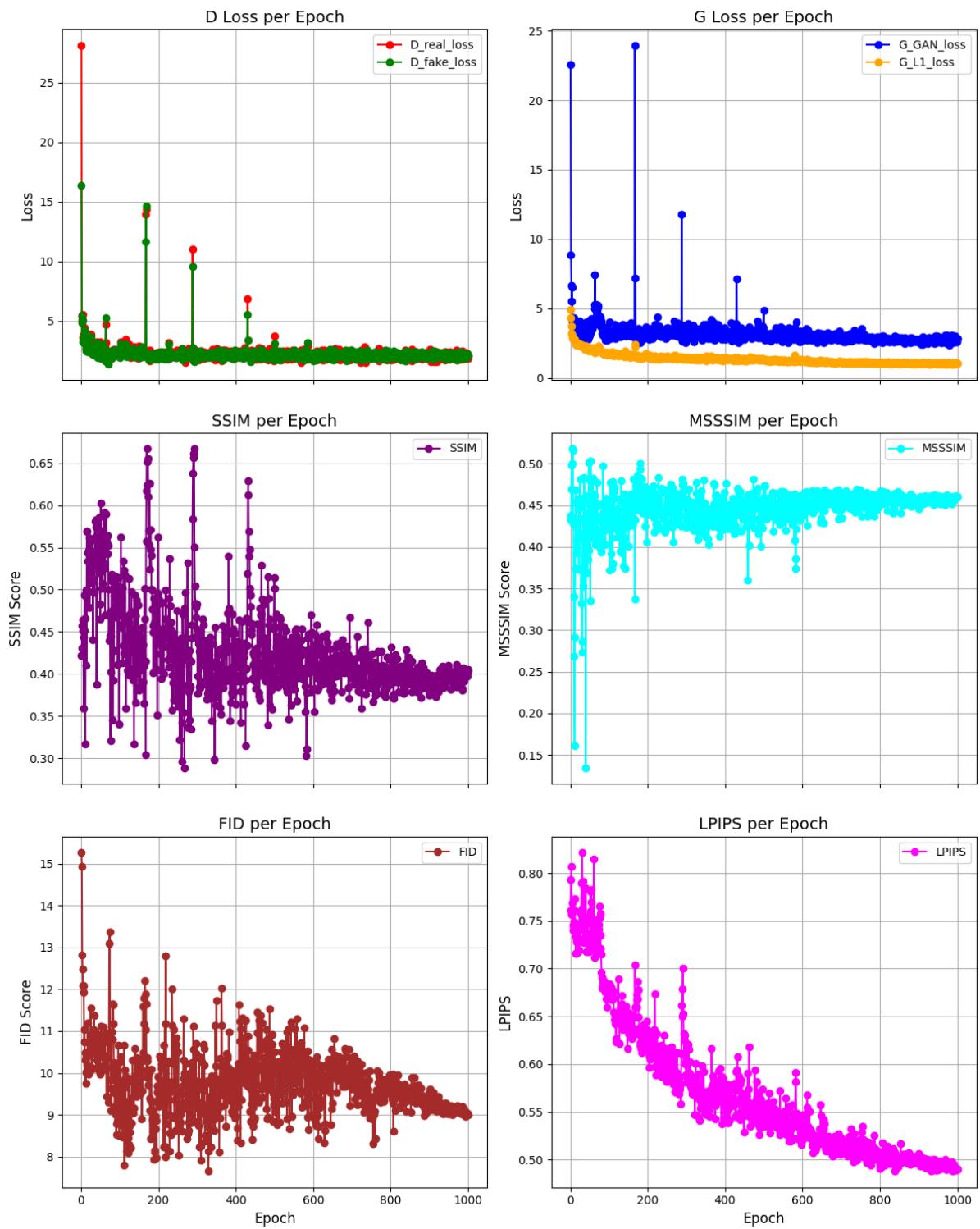
Supplementary Fig. 28. Comparison between the experimental images (gray) and the best SSIM generation from SAM masks (magma).



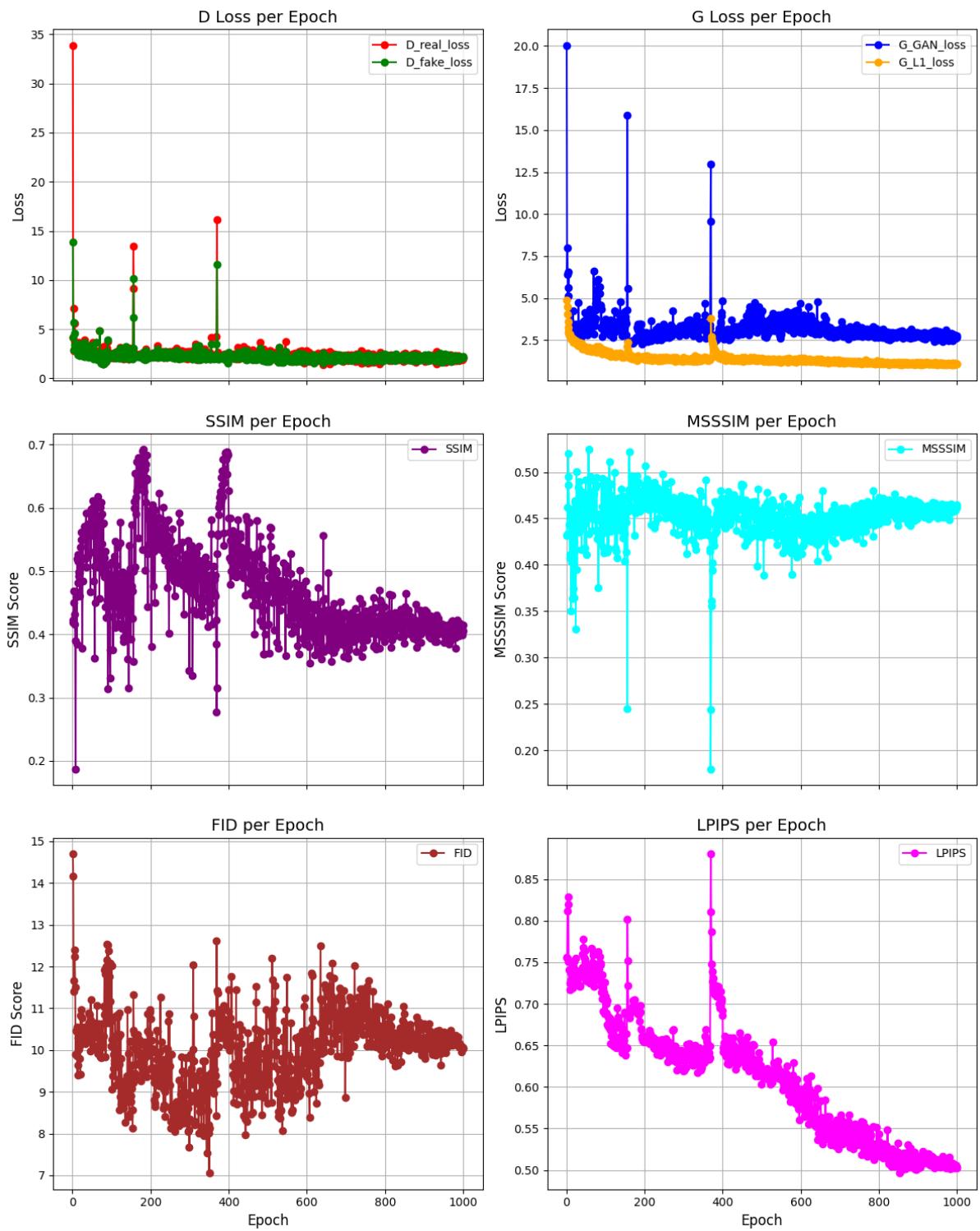
Supplementary Fig. 29. Synthetic images of the best SSIM model from random masks.



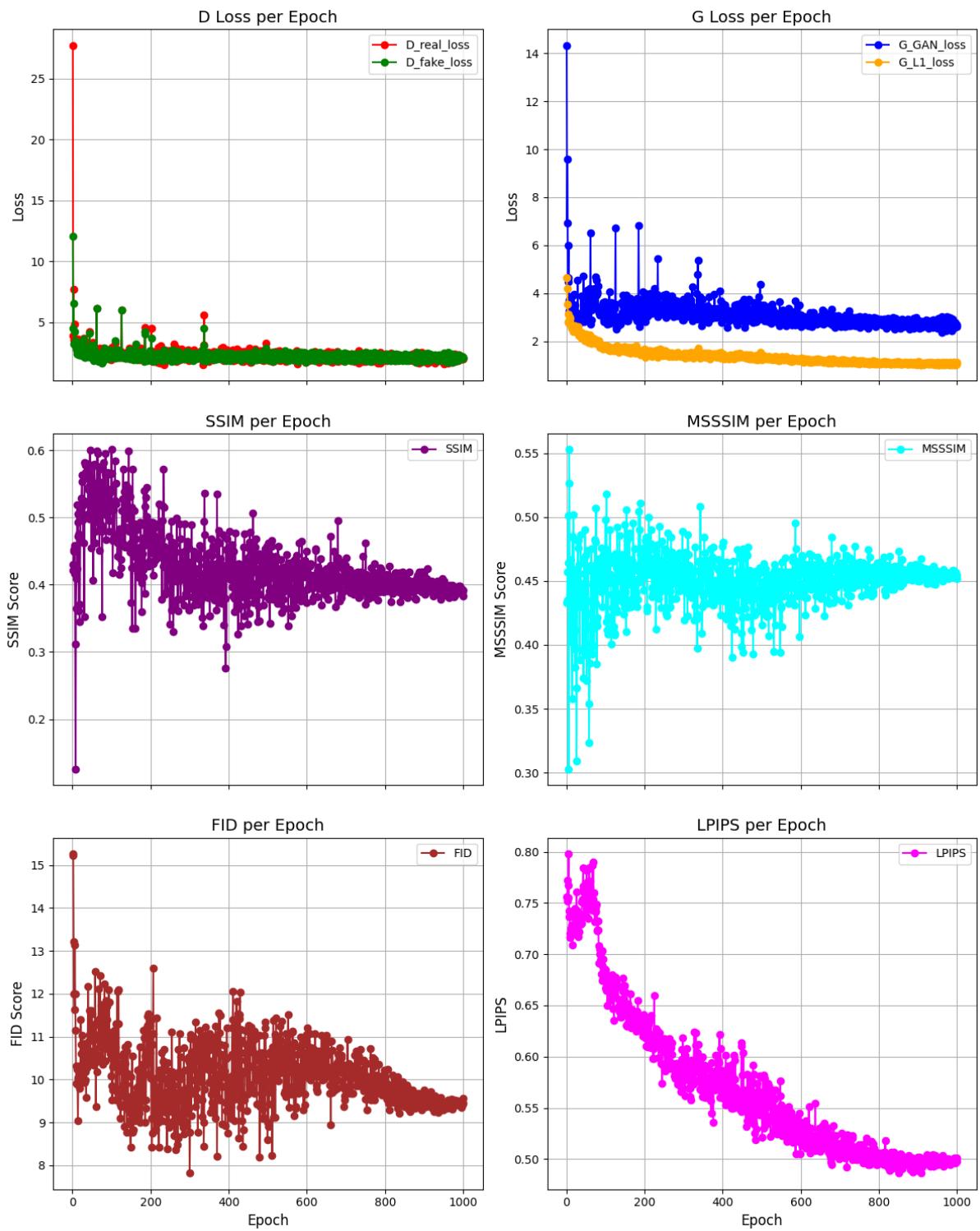
Supplementary Fig. 30. Loss and validation metrics of pix2pix trained for 1000 epochs on manual-labeled masks.



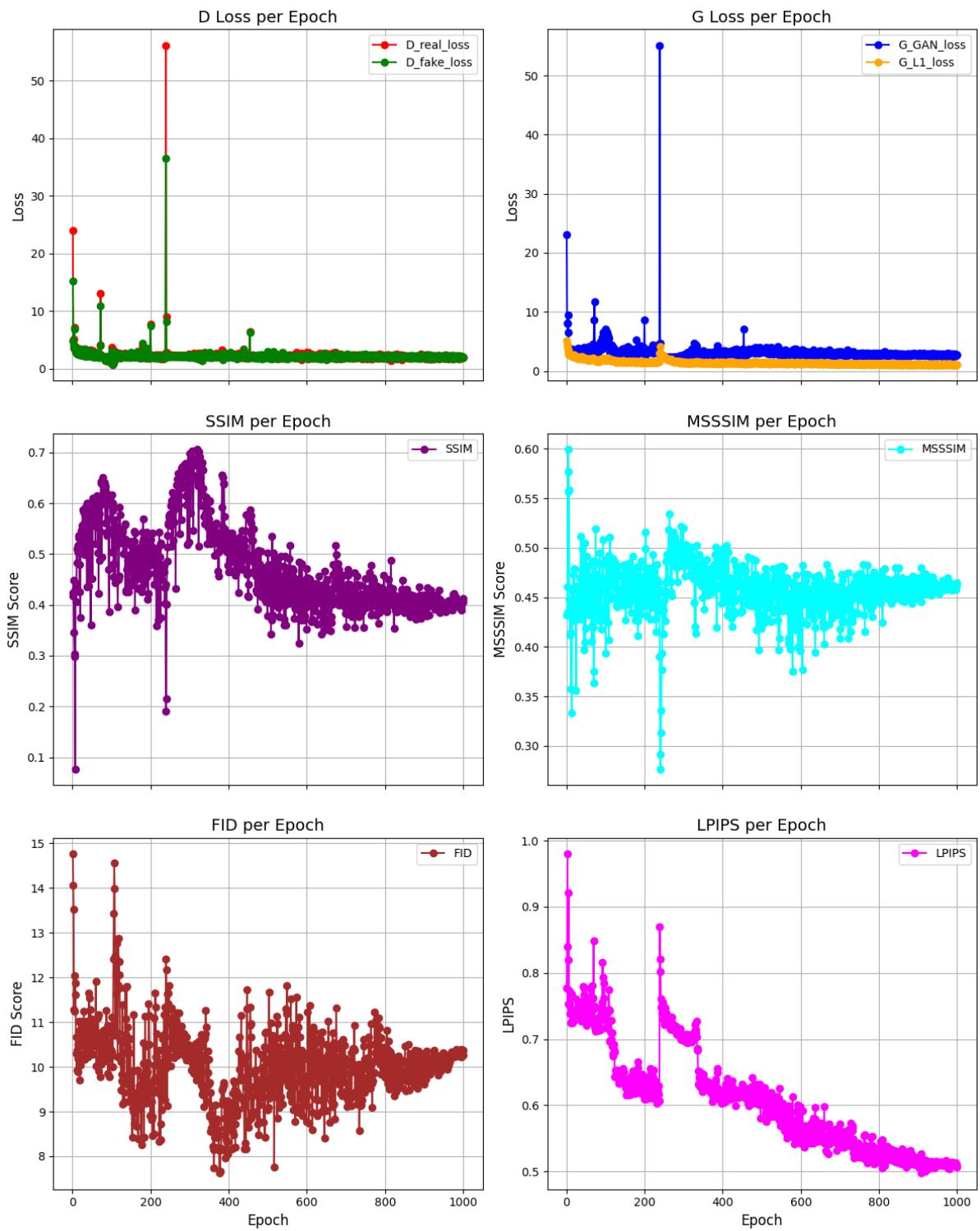
Supplementary Fig. 31. 1st round pix2pix training log on manual-labeled masks.



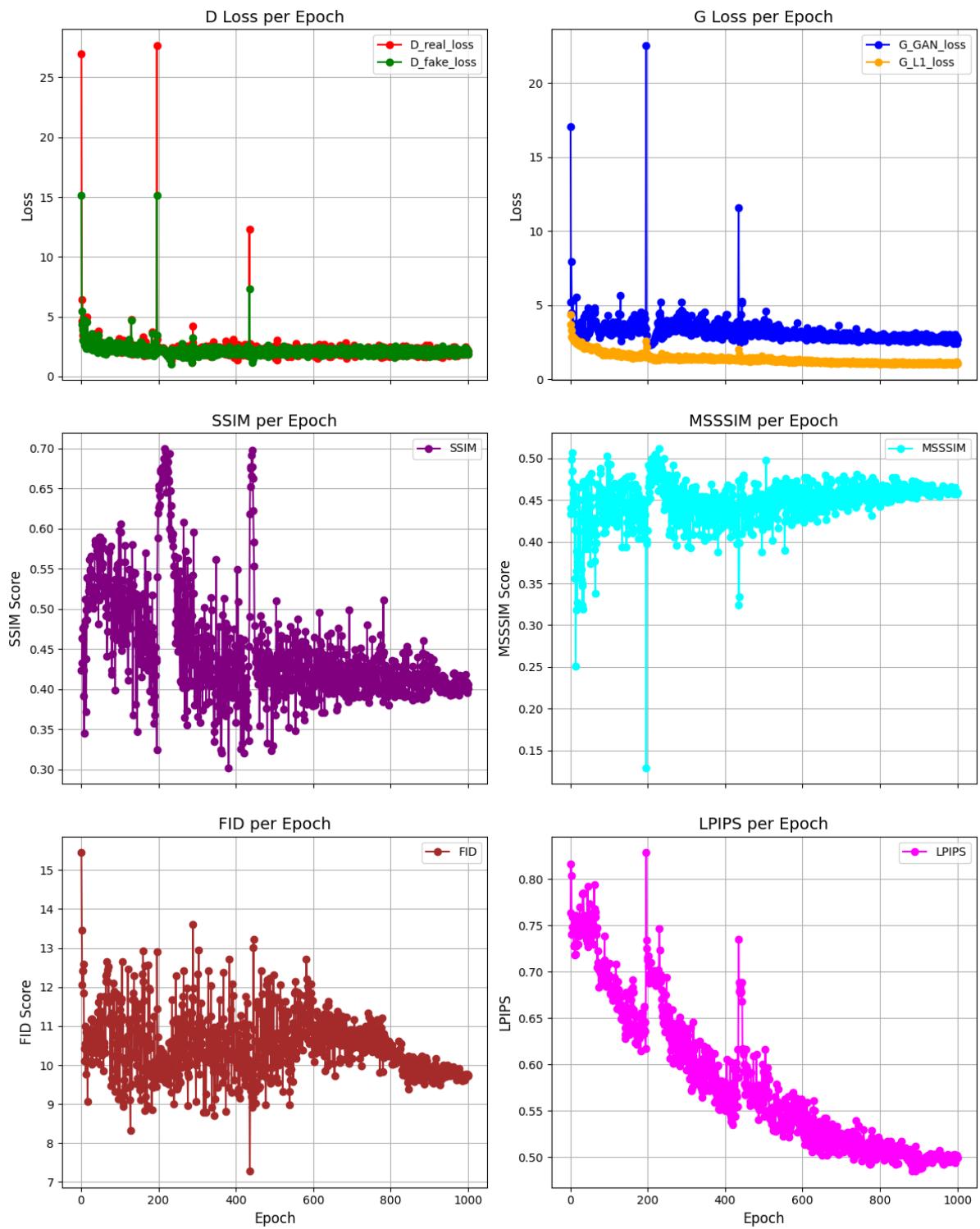
Supplementary Fig. 32. 2nd round pix2pix training log on manual-labeled masks.



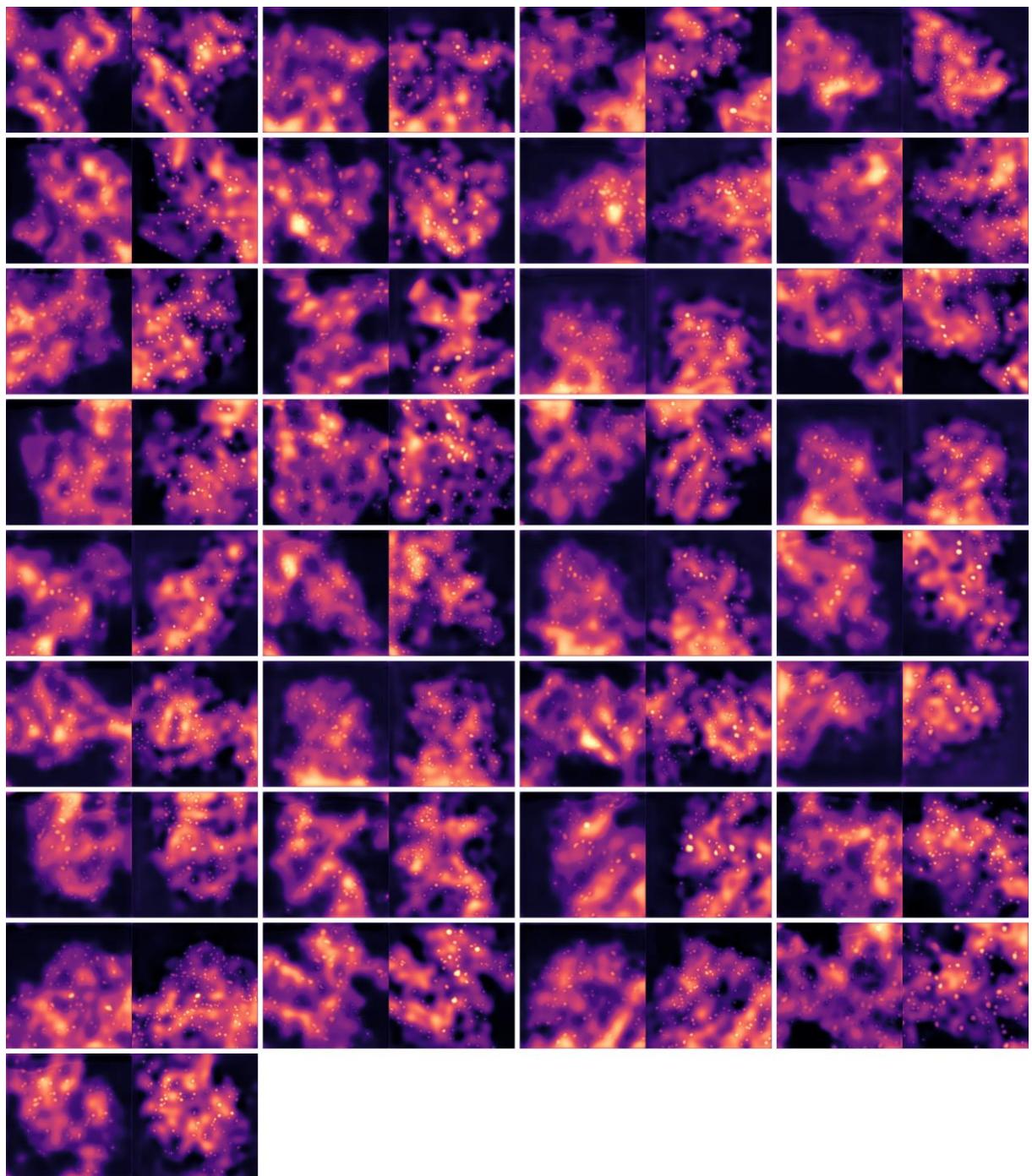
Supplementary Fig. 33. 3nd round pix2pix training log on manual-labeled masks.



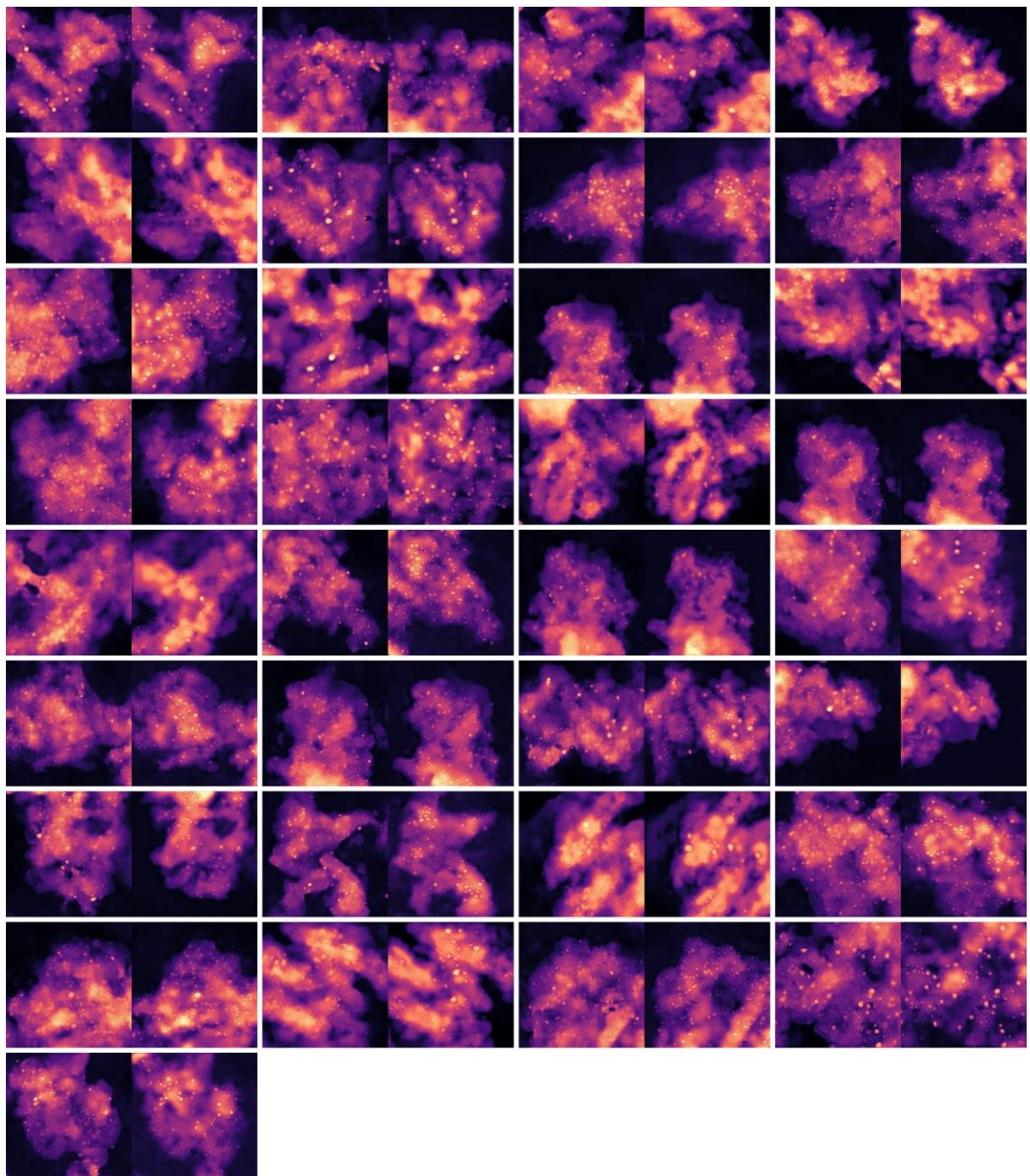
Supplementary Fig. 34. 4th round pix2pix training log on manual-labeled masks.



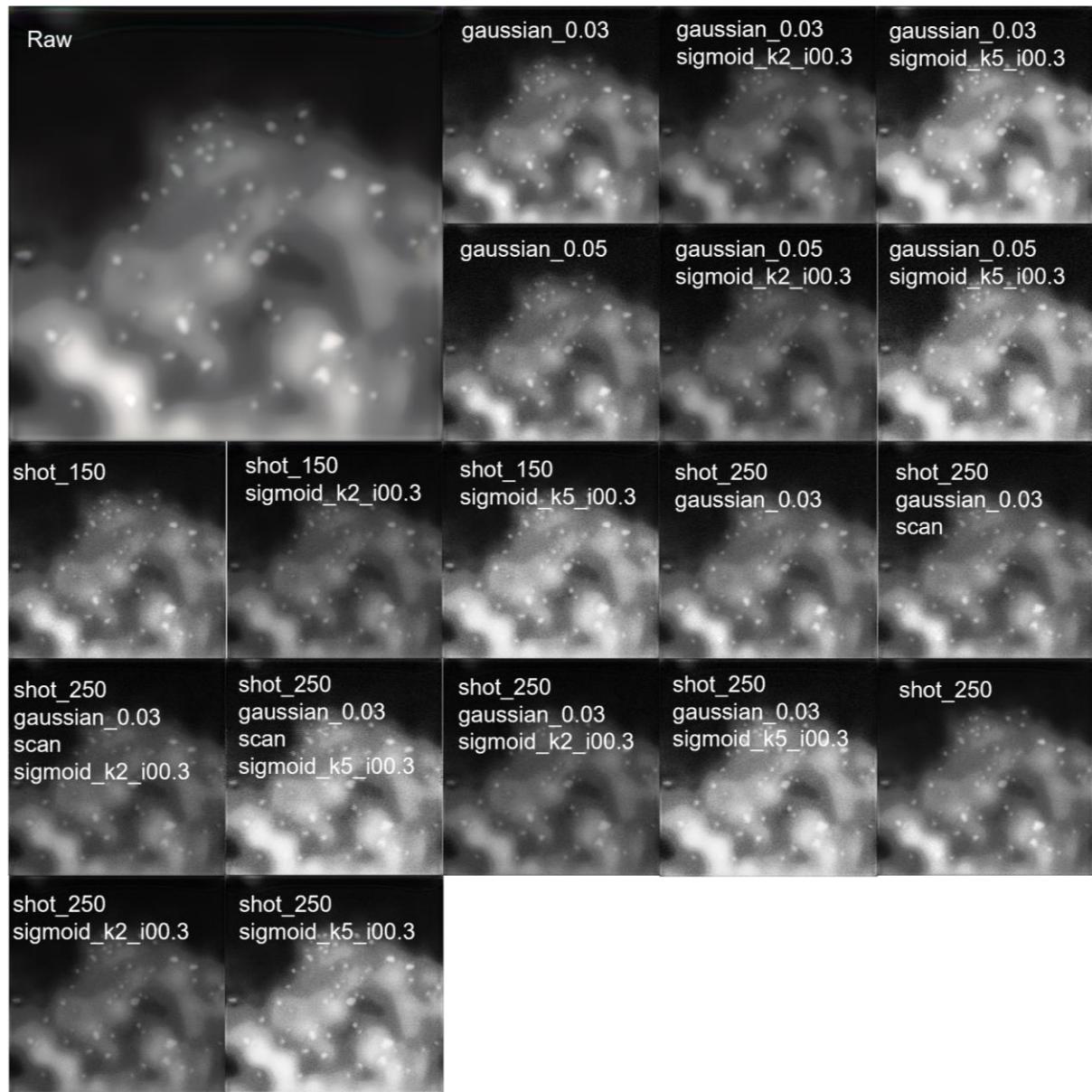
Supplementary Fig. 35. 5th round pix2pix training log on manual-labeled masks.



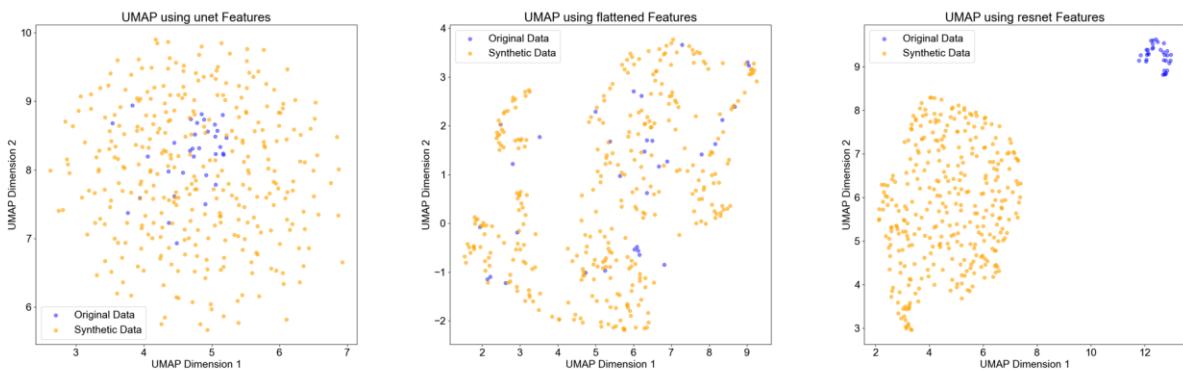
Supplementary Fig. 36. Comparison between the synthetic images generated from the best SSIM model trained by SAM-masking (left) and human-masking (right).



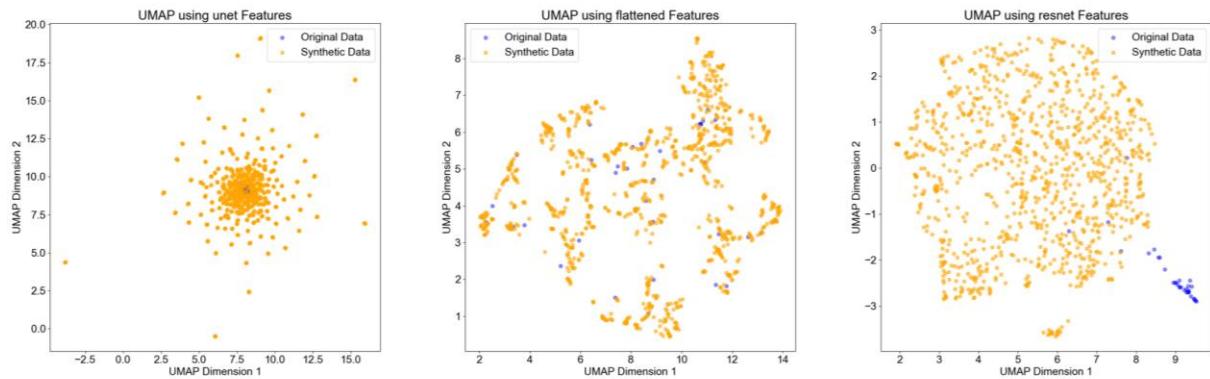
Supplementary Fig. 37. Comparison between the synthetic images generated from the best LPIPS model trained by SAM-masking (left) and human-masking (right).



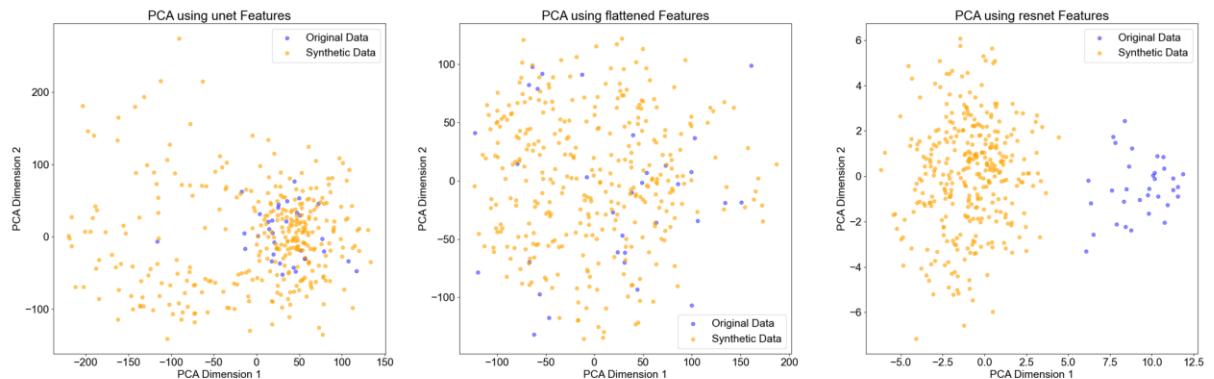
Supplementary Fig. 38. Representative examples of 18 augmentation strategies.



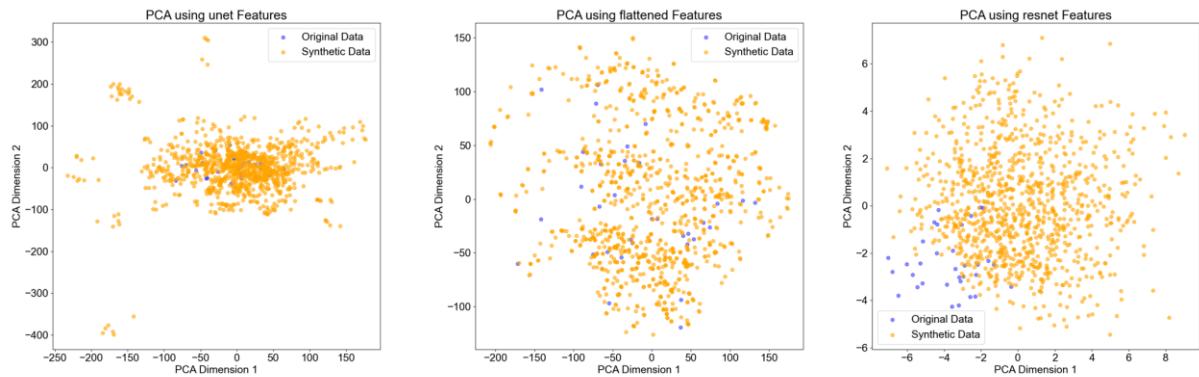
Supplementary Fig. 39. UMAP visualization of raw synthetic data and original data.



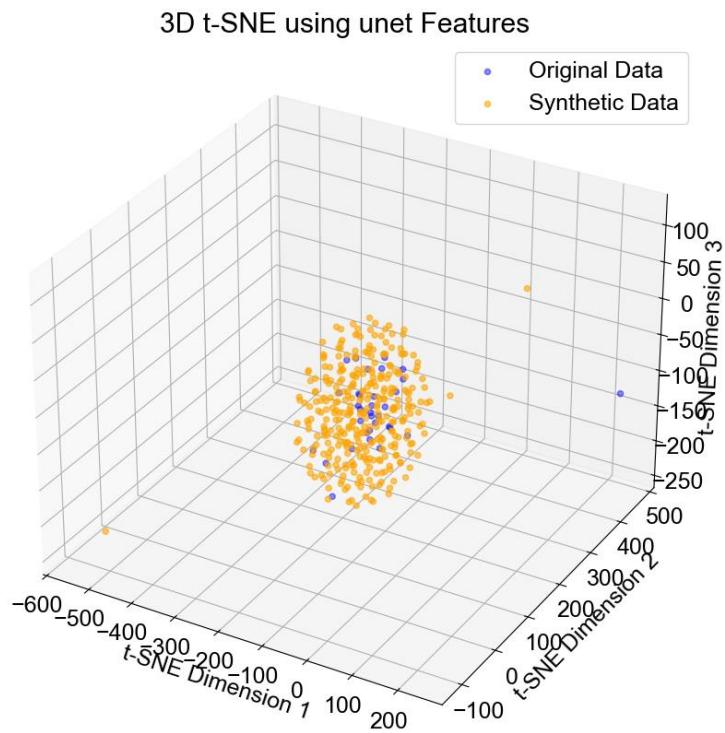
Supplementary Fig. 40. UMAP visualization of adapted synthetic data and original data.



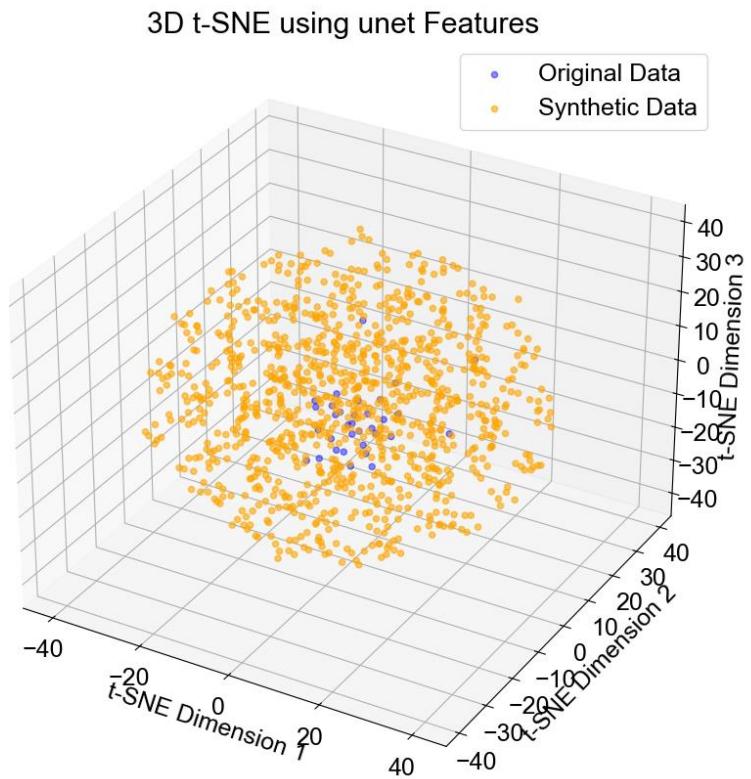
Supplementary Fig. 41. PCA visualization of raw synthetic data and original data.



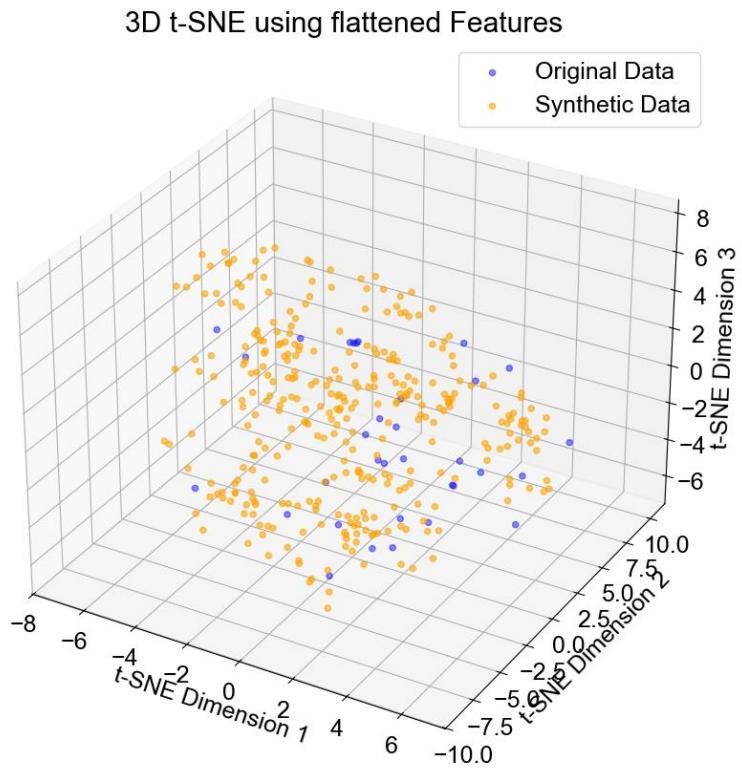
Supplementary Fig. 42. PCA visualization of adapted synthetic data and original data.



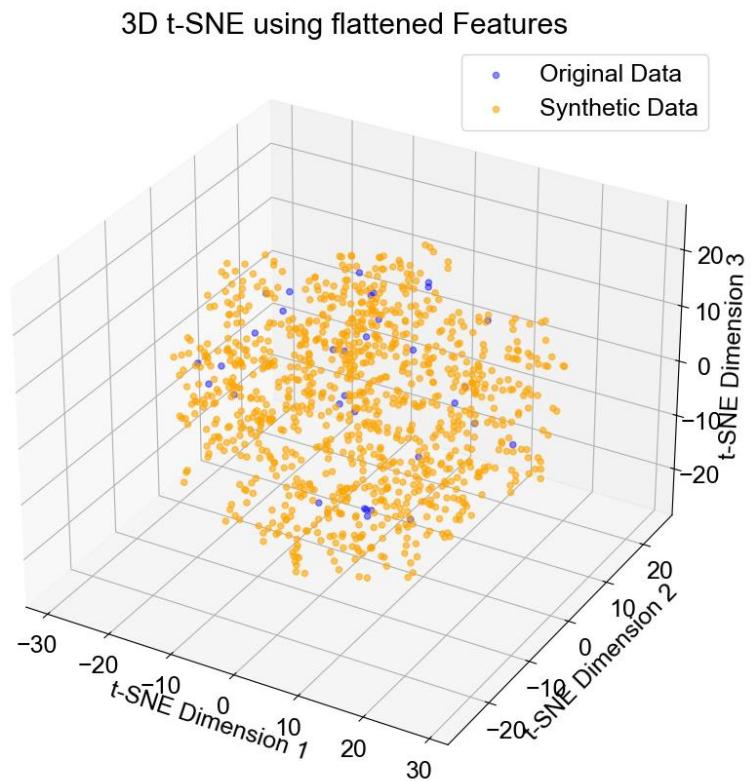
Supplementary Fig. 43. 3D t-SNE visualization of original synthetic data and original data by CBAM-UNet++.



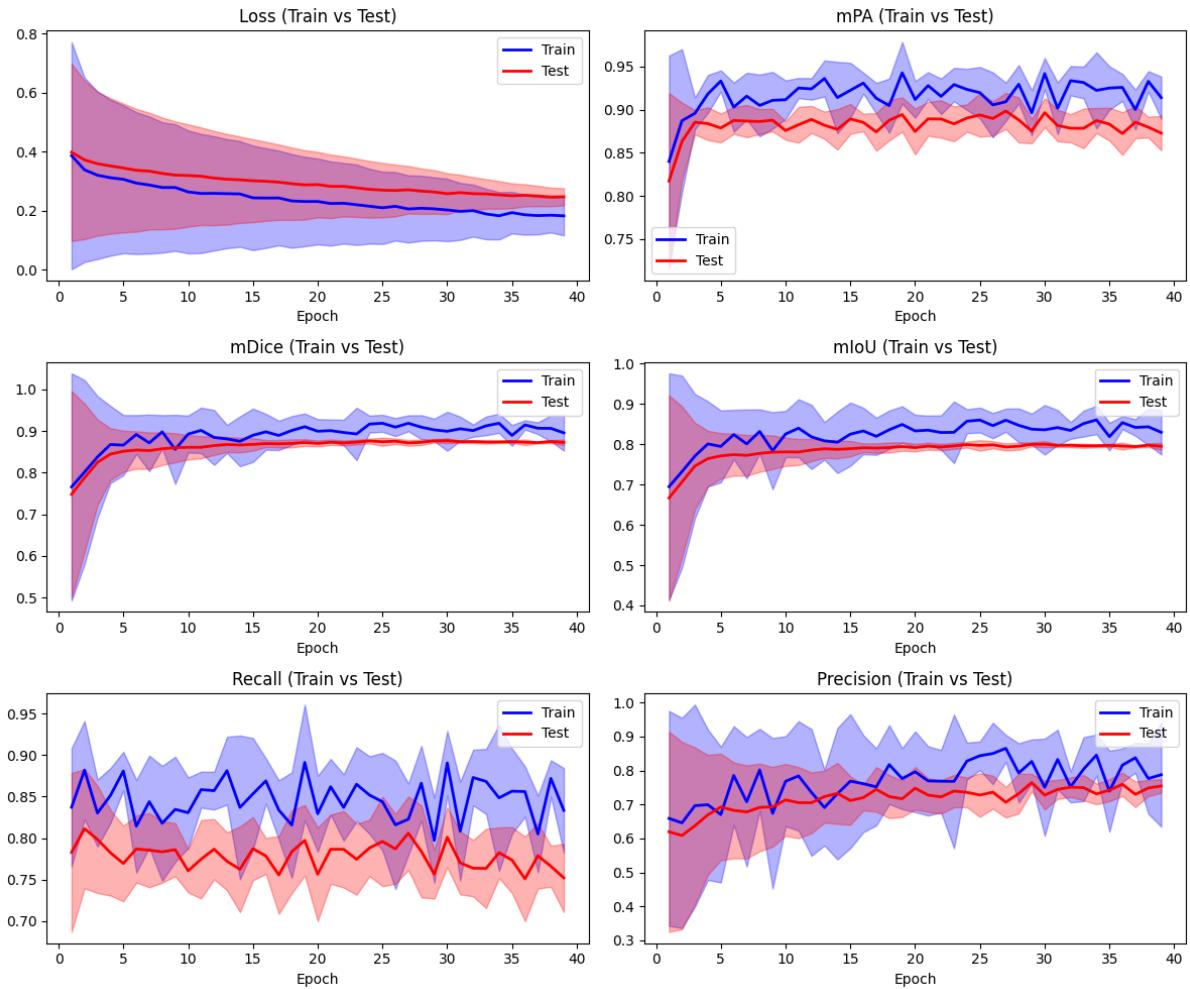
Supplementary Fig. 44. 3D t-SNE visualization of 1000 adapted synthetic data and original data by CBAM-UNet++.



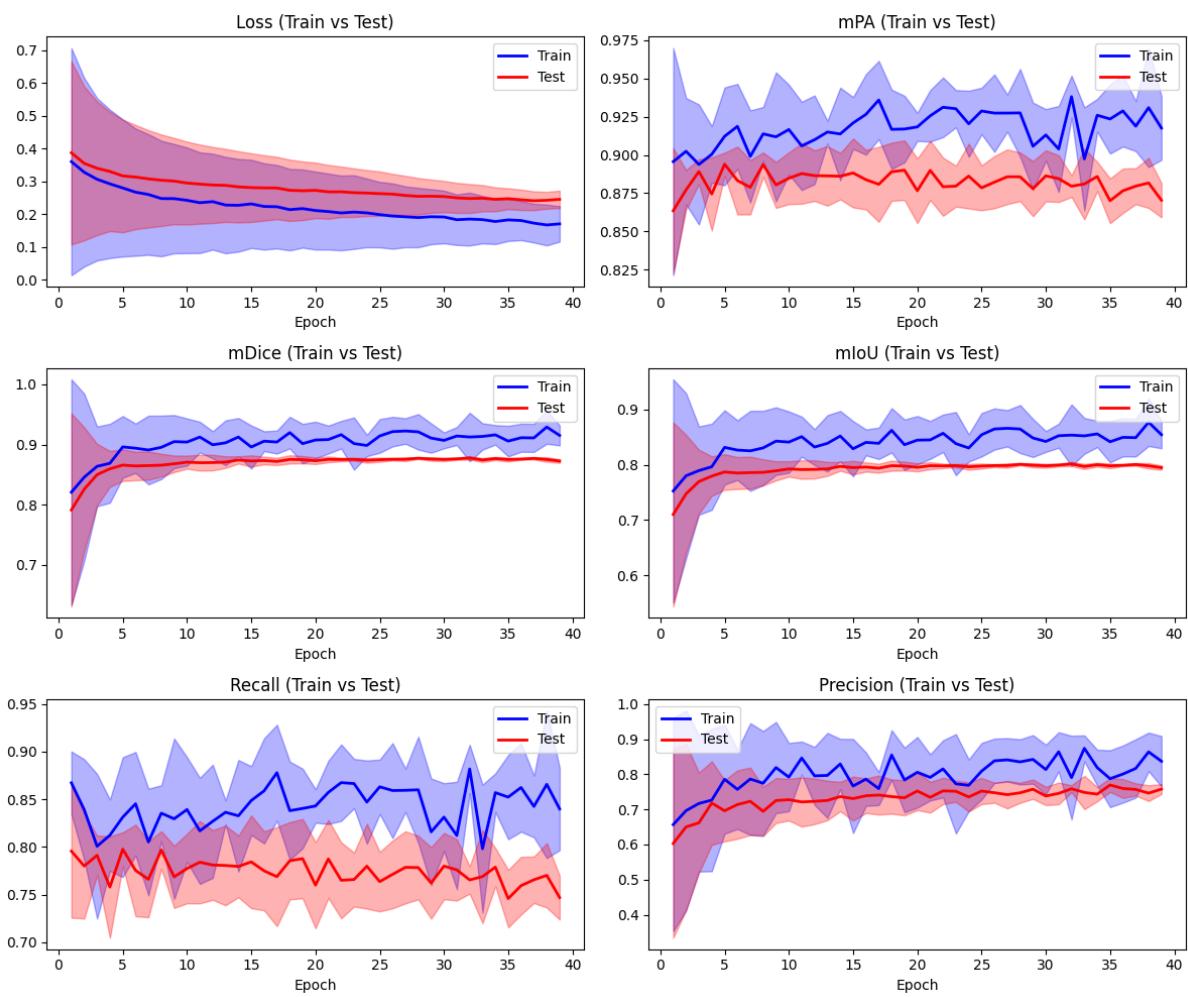
Supplementary Fig. 45. 3D t-SNE visualization of original synthetic data and original data by flattened vectoring.



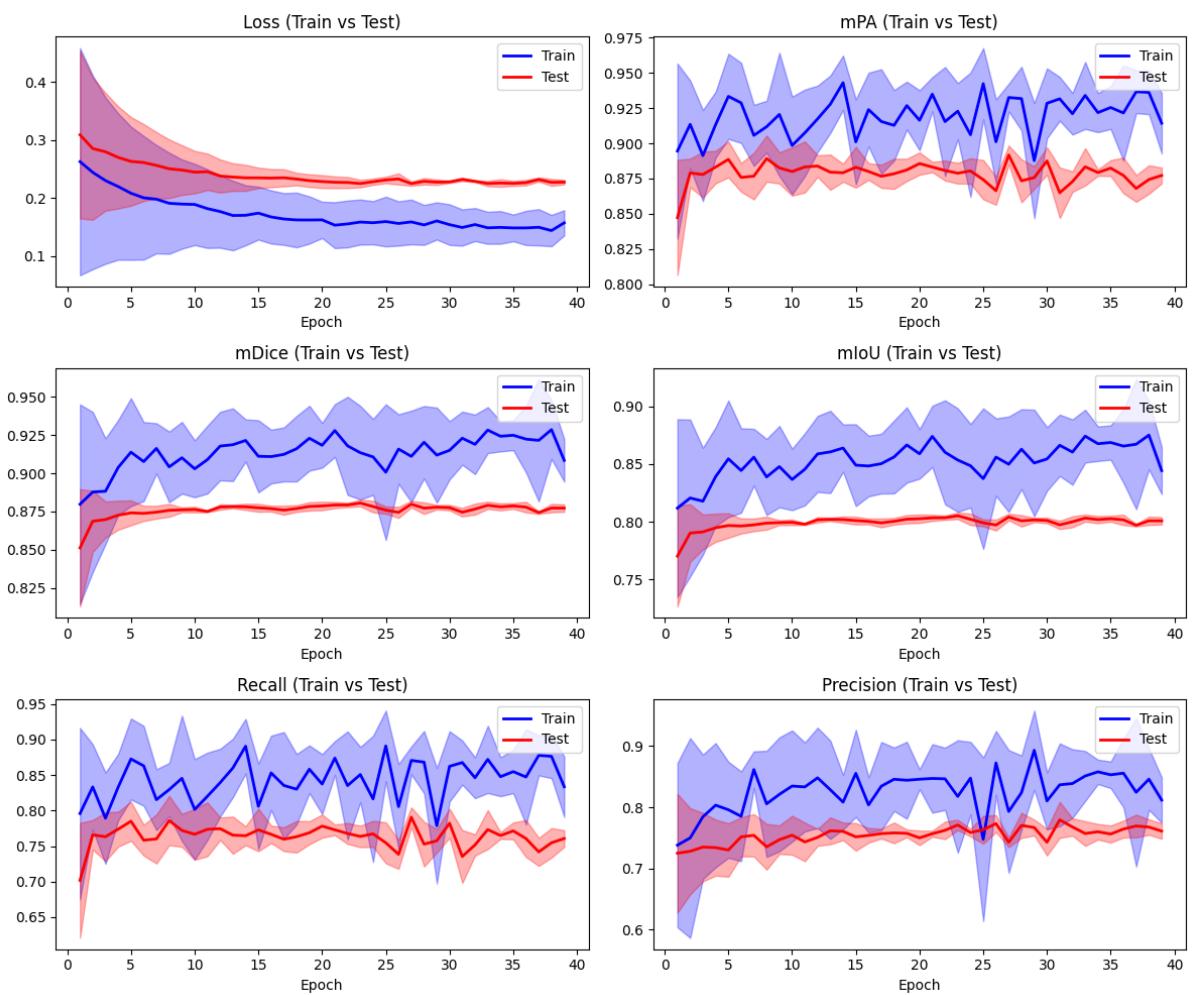
Supplementary Fig. 46. 3D t-SNE visualization of 1000 adapted synthetic data and original data by flattened vectoring.



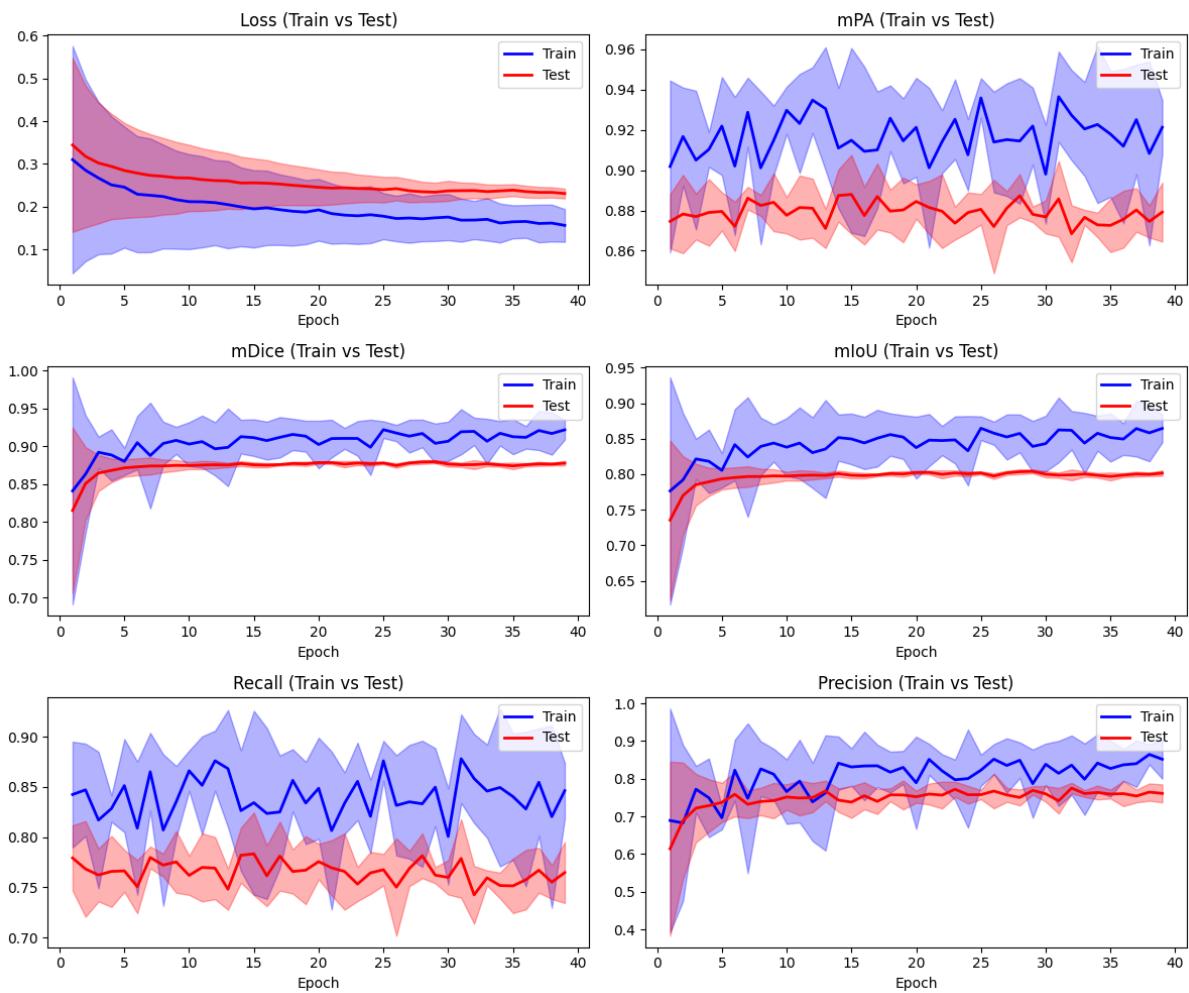
Supplementary Fig. 47. Loss and validation metrics of CBAM-UNet++ trained for on 33 manual-labeled images (error band from 5 times runs ensuring reproducibility).



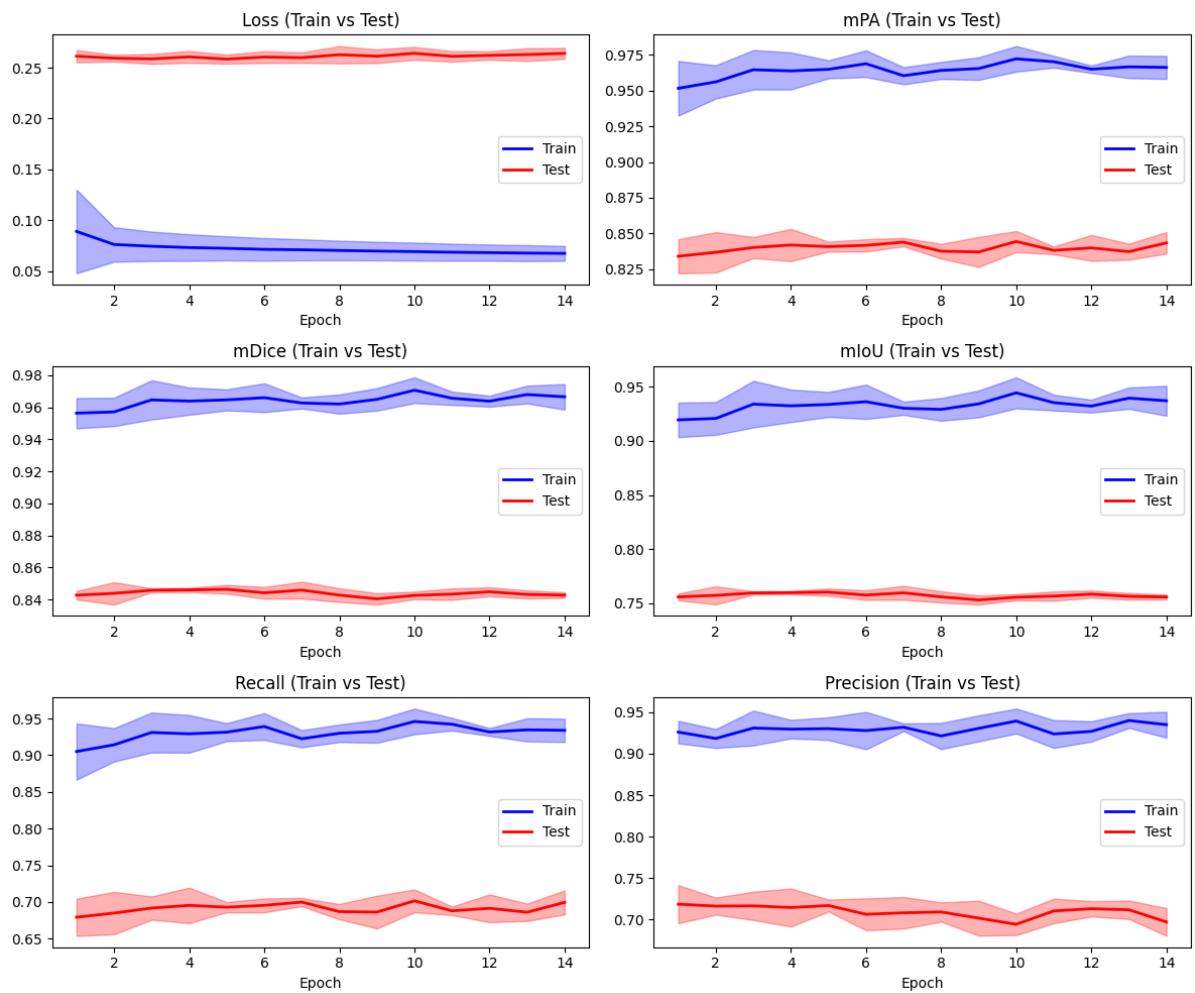
Supplementary Fig. 48. Loss and validation metrics of UNet++ trained for on 33 manual-labeled images.



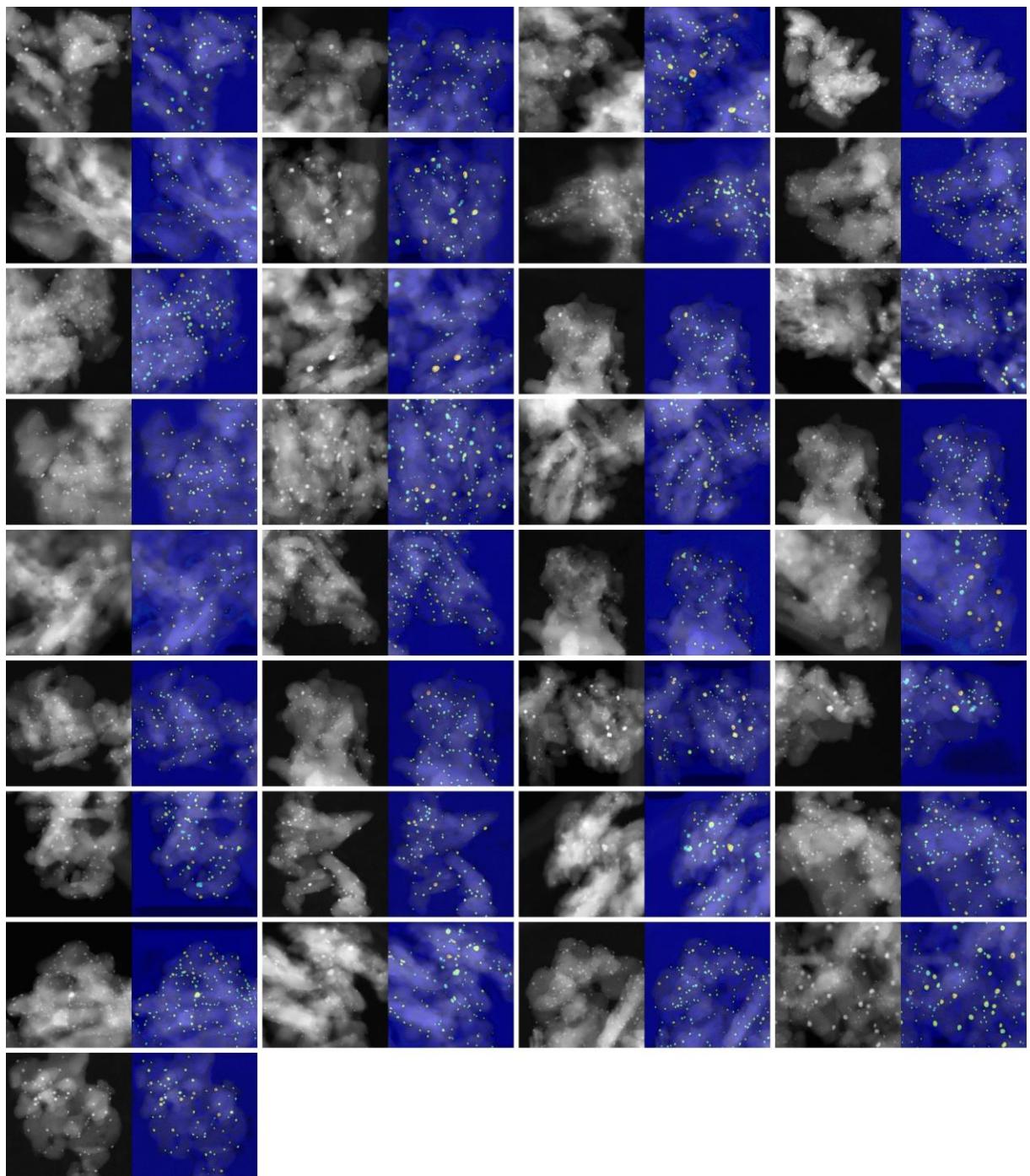
Supplementary Fig. 49. Loss and validation metrics of SE-UNet++ trained for on 33 manual-labeled images.



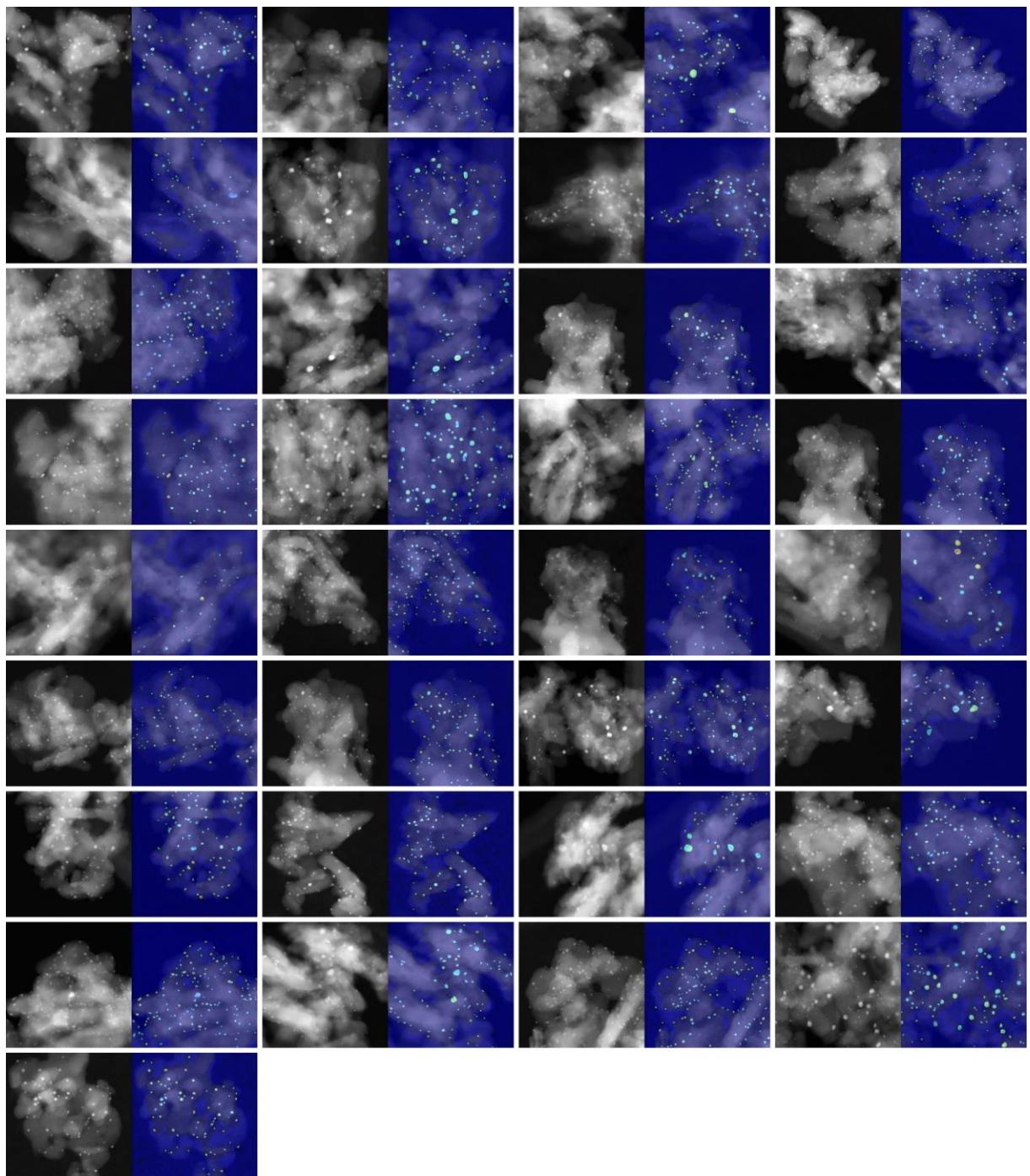
Supplementary Fig. 50. Loss and validation metrics of ECA-UNet++ trained for on 33 manual-labeled images.



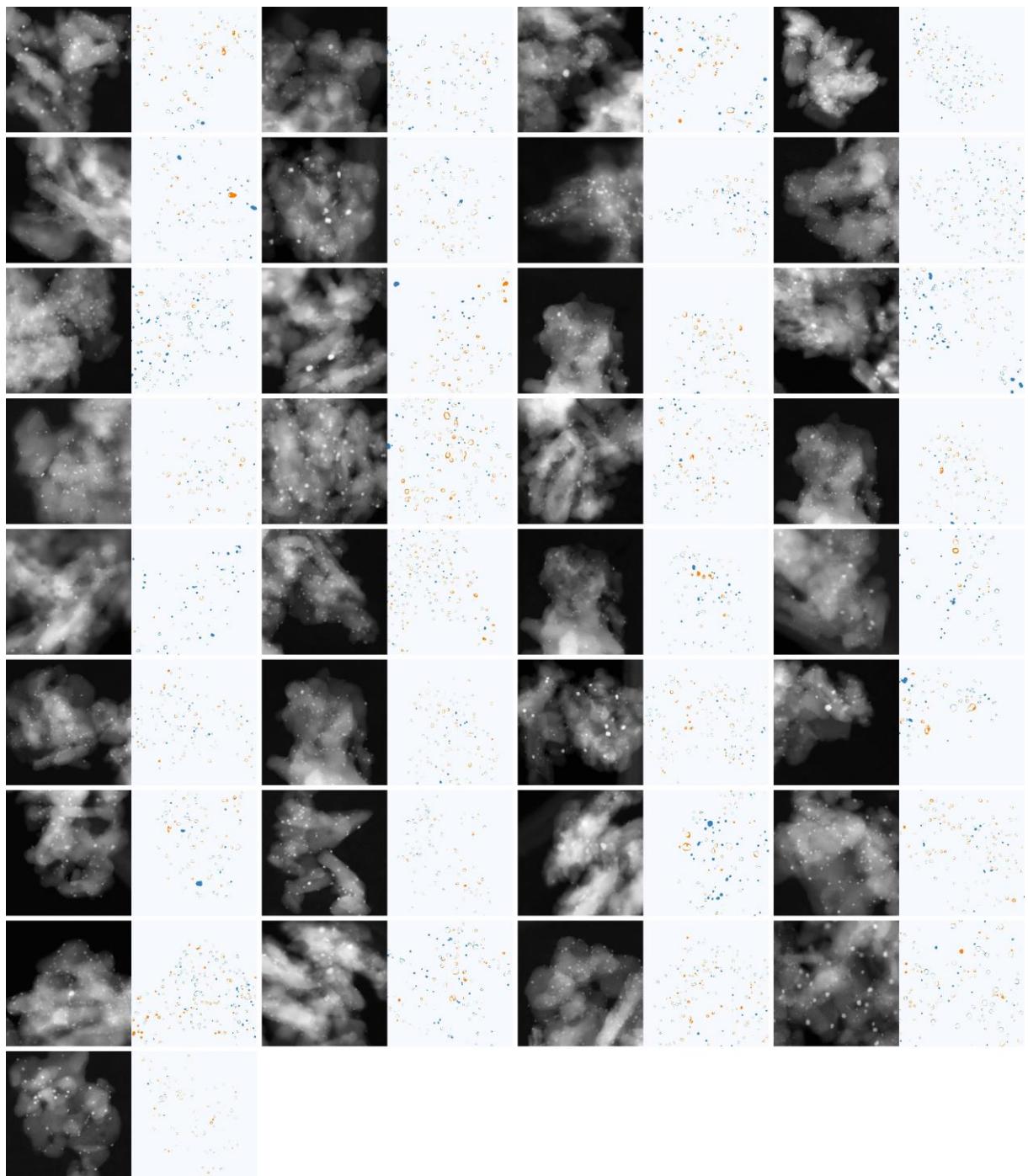
Supplementary Fig. 51. Loss and validation metrics of CBAM-UNet++ trained for on 5940 synthetic HAADF-STEM images.



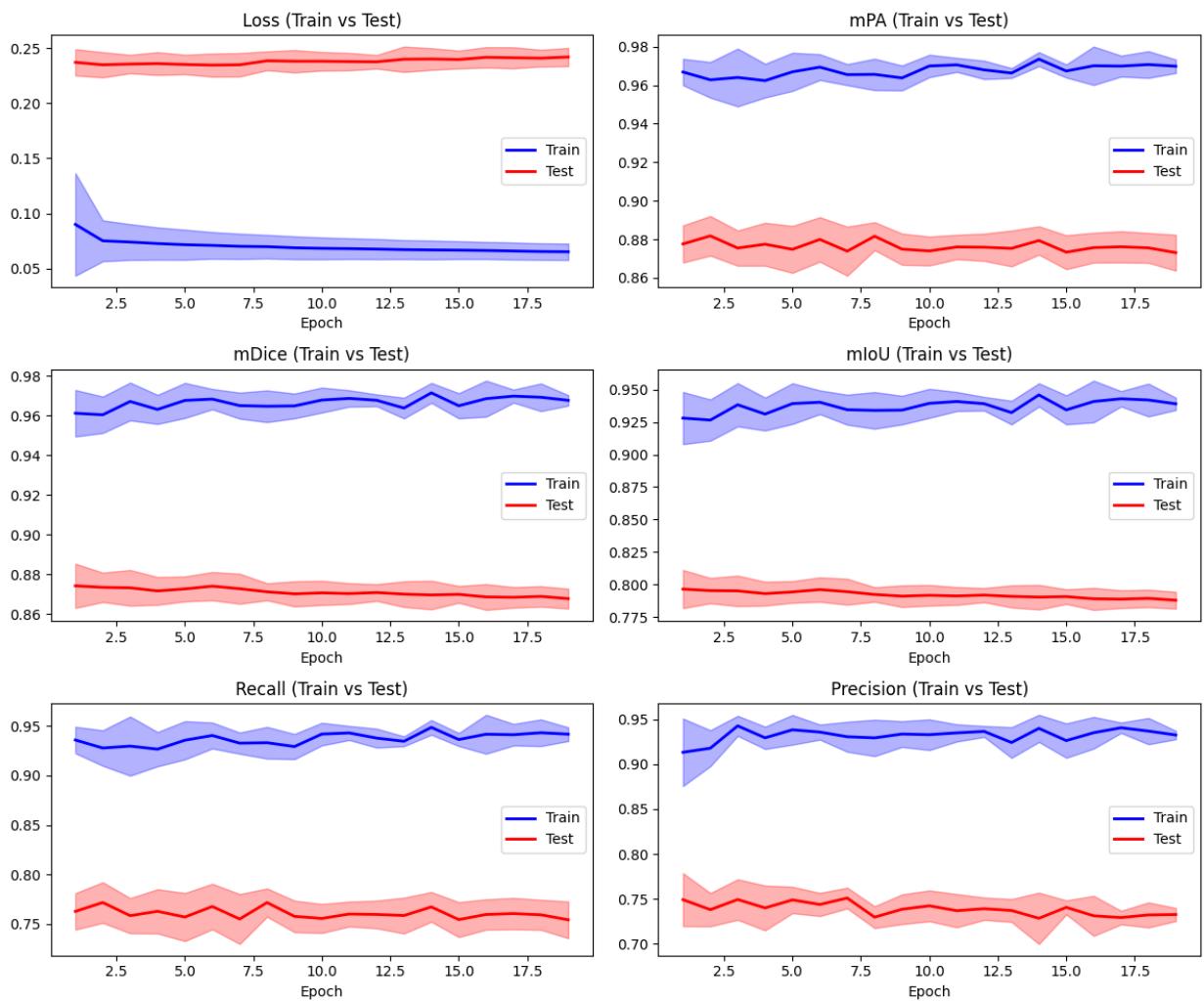
Supplementary Fig. 52. Grad-CAM of EMcopilot on 33 experimental images.



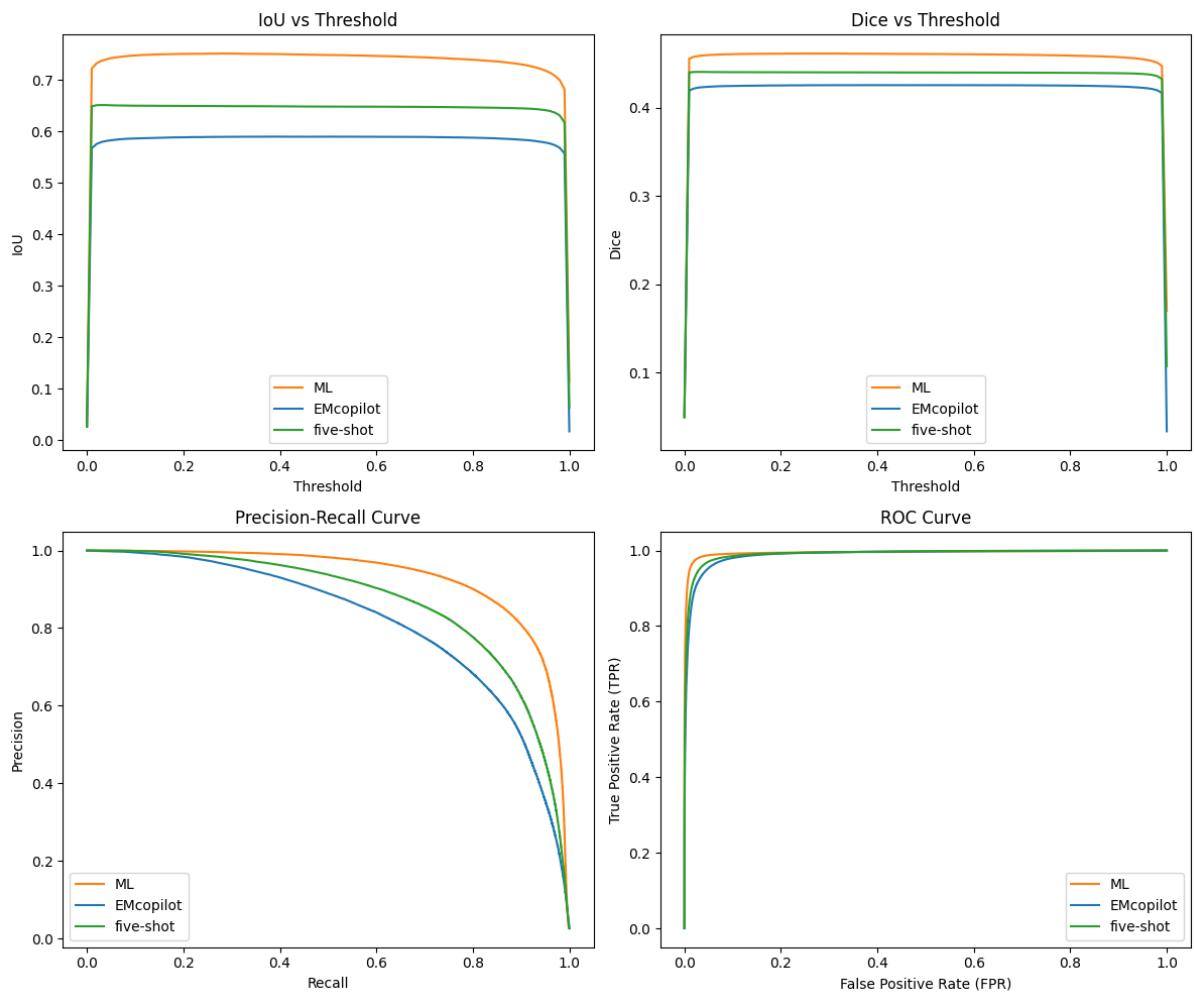
Supplementary Fig. 53. Grad-CAM of manual labeling trained-model on 33 experimental images.



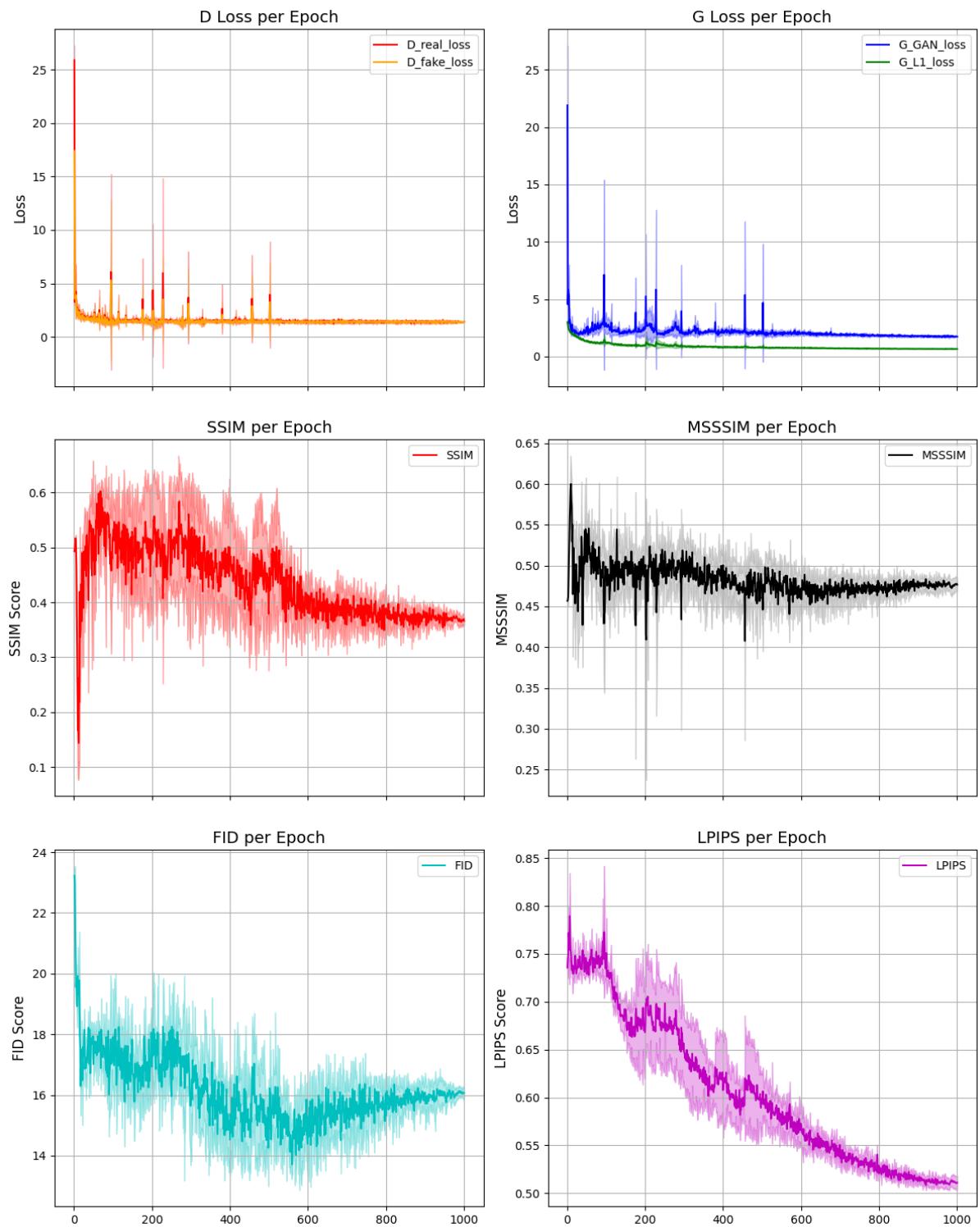
Supplementary Fig. 54. Prediction different between EMcopilot and manual labeling method on 33 experimental images.



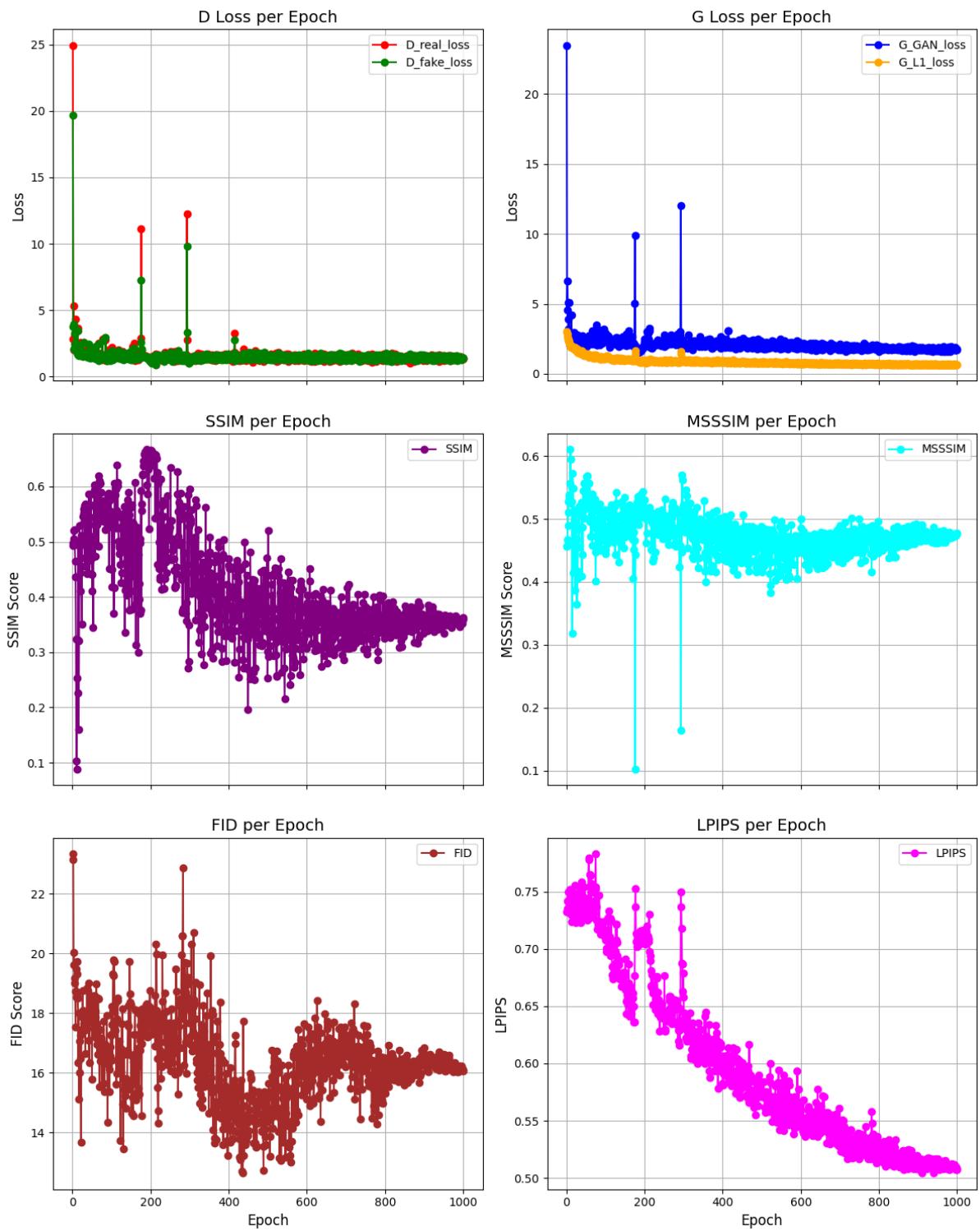
Supplementary Fig. 55. Loss and validation metrics of CBAM-UNet++ trained for 5940 synthetic images hybrid with 5 manual labeled images (randomly selected).



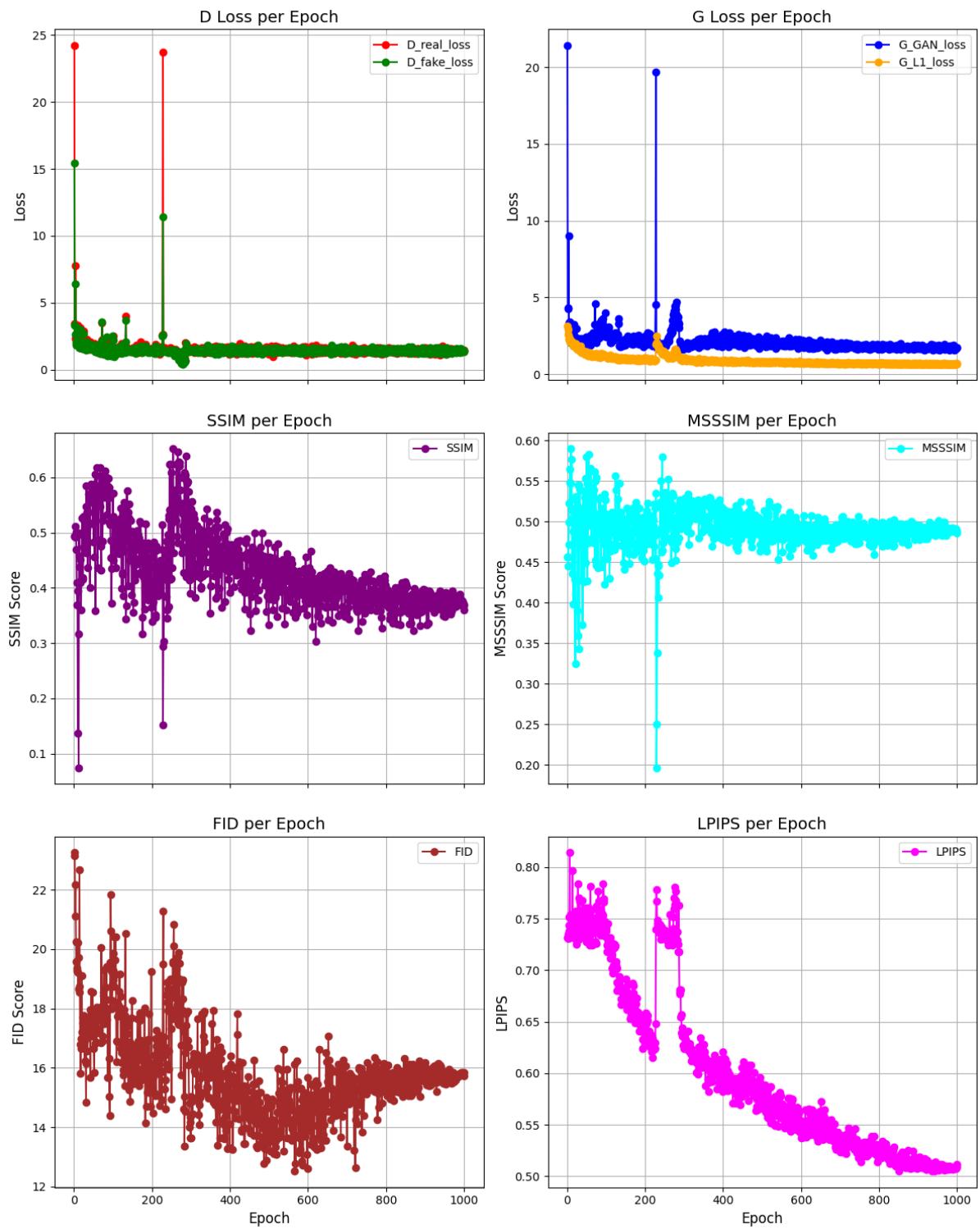
Supplementary Fig. 56. Segmentation metrics comparison of CBAM-UNet++ trained by ML, EMcopilot and five-shot learning.



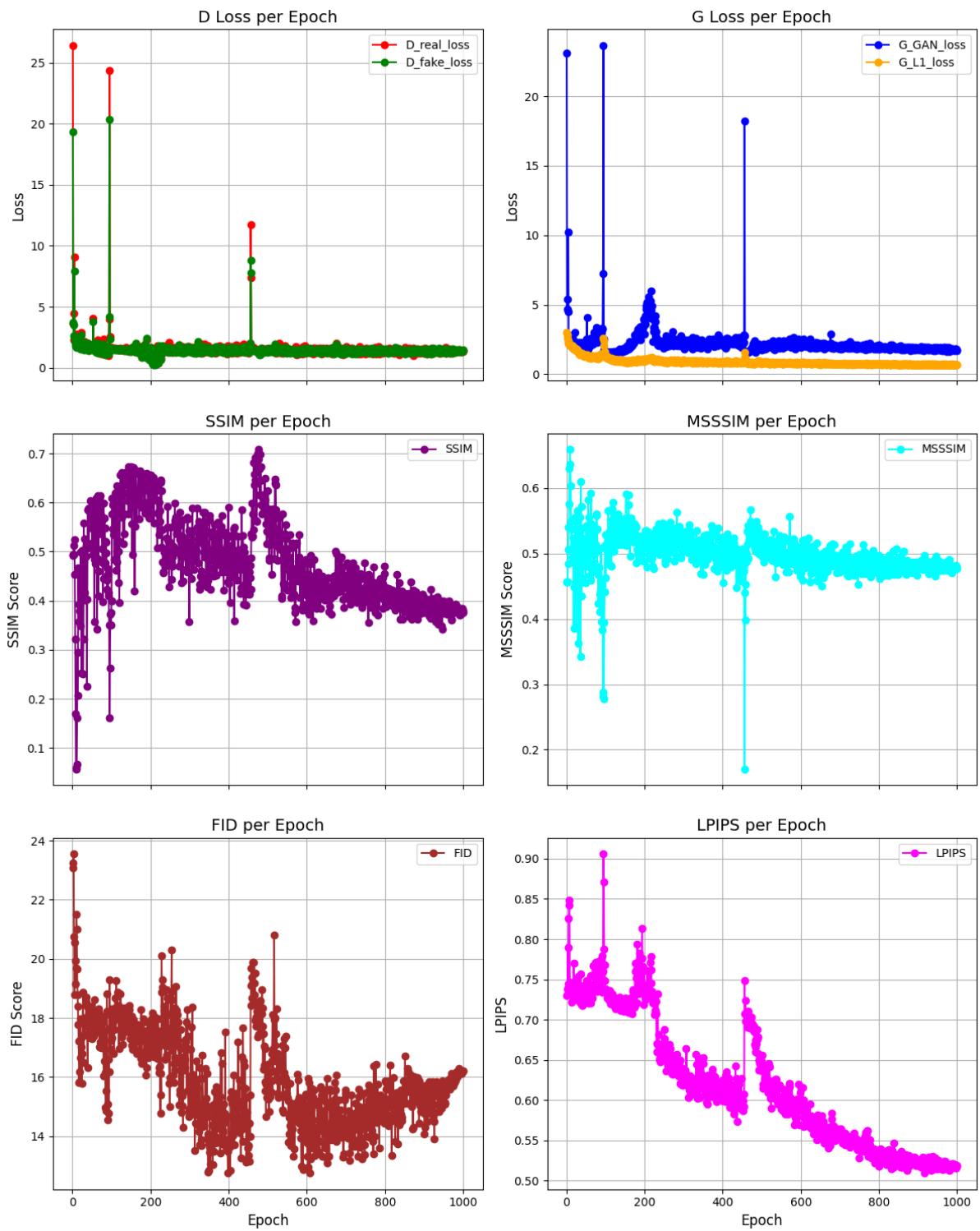
Supplementary Fig. 57. Loss and validation metrics of pix2pix trained for 1000 epochs of PtSn Clusters@Al₂O₃.



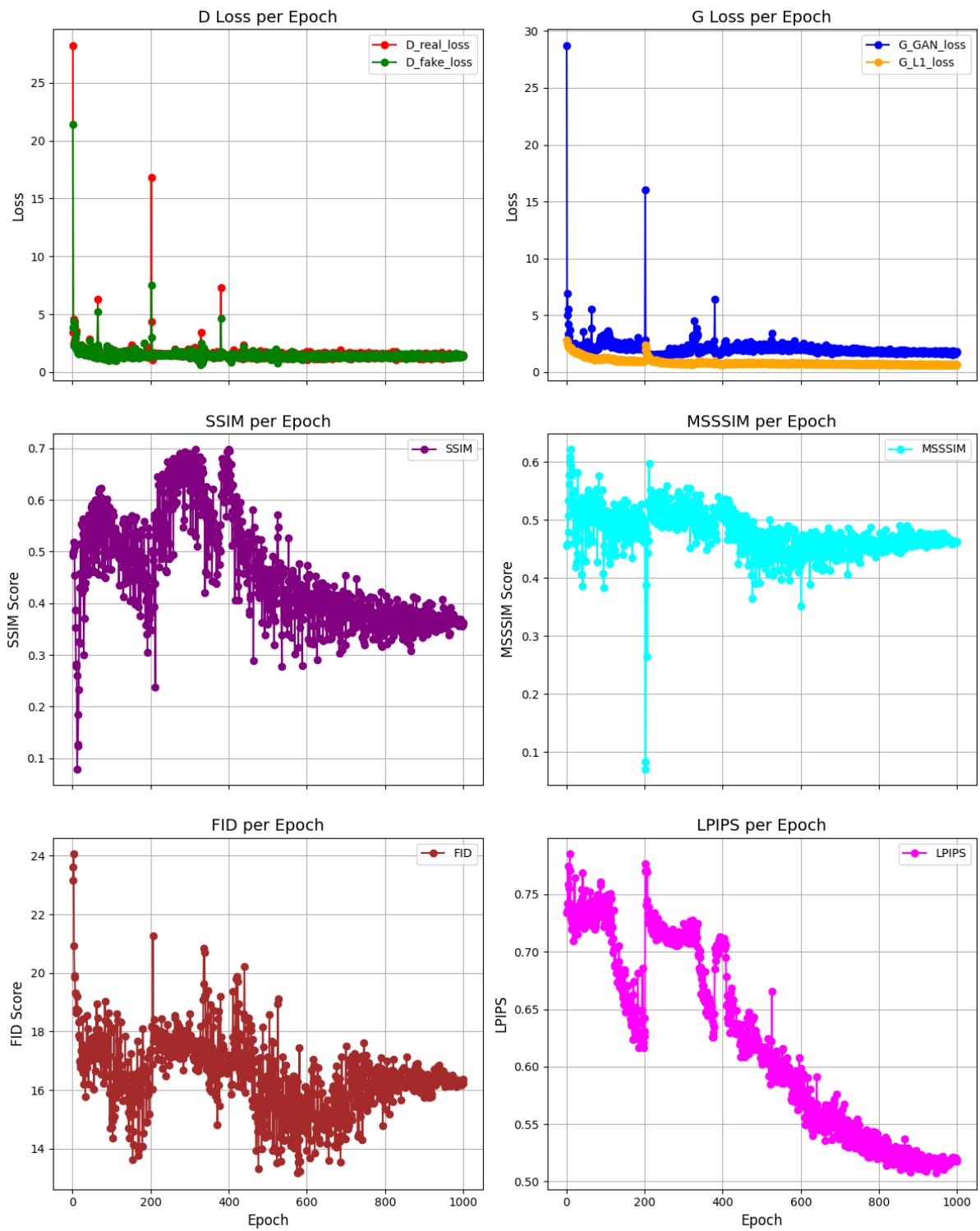
Supplementary Fig. 58. 1st round pix2pix training log of PtSn Clusters@Al₂O₃.



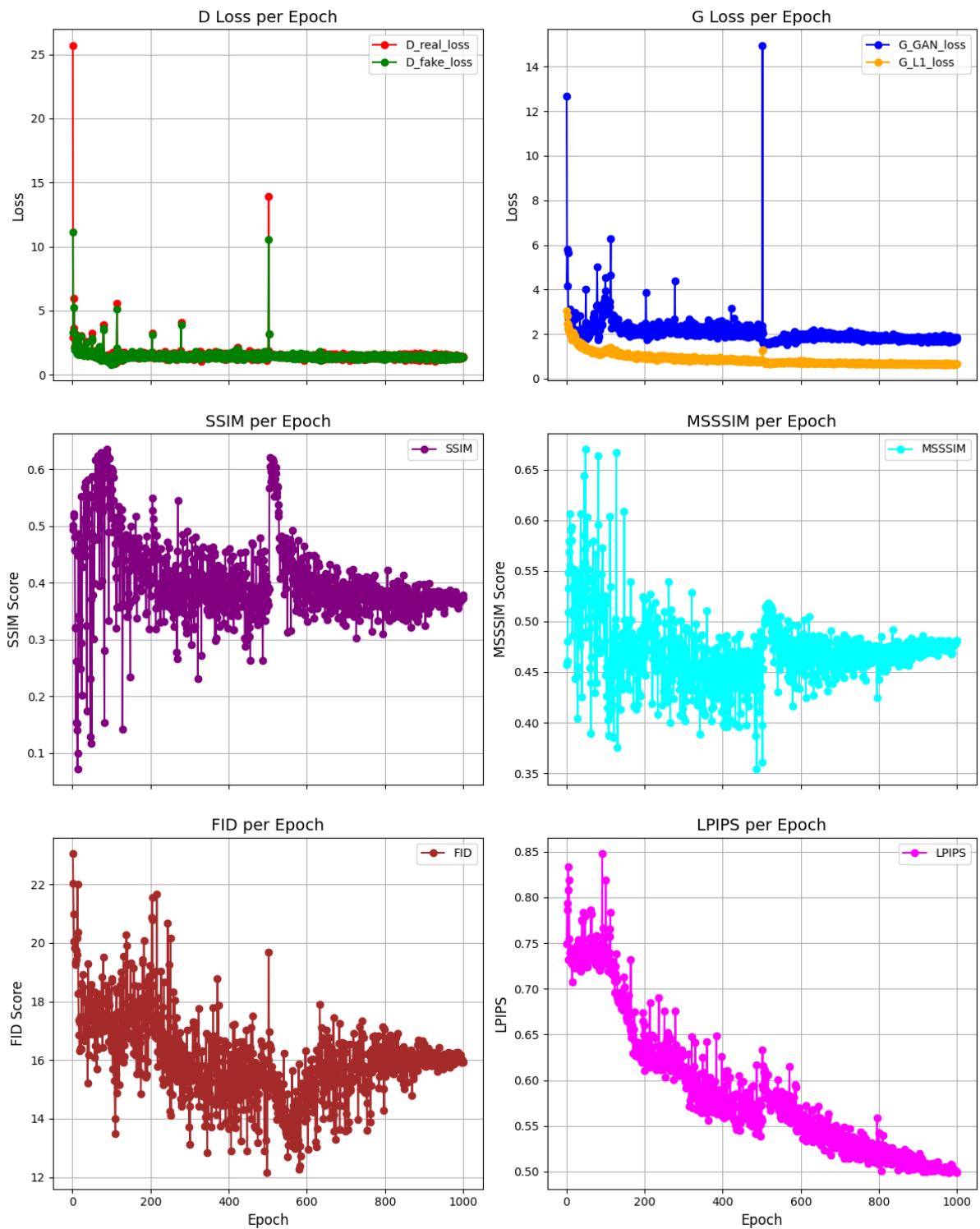
Supplementary Fig. 59. 2nd round pix2pix training log of PtSn Clusters@Al₂O₃.



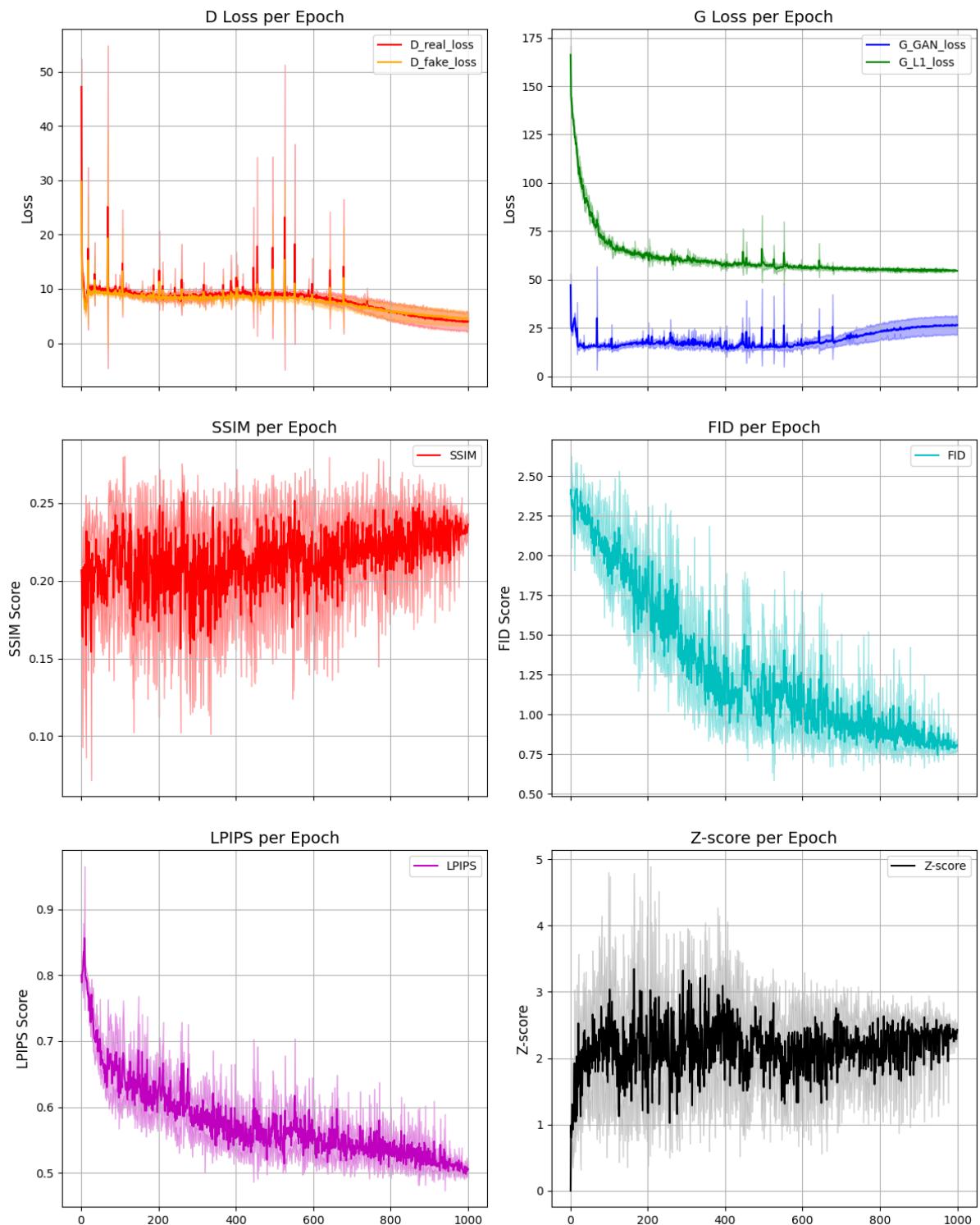
Supplementary Fig. 60. 3nd round pix2pix training log of PtSn Clusters@Al₂O₃.



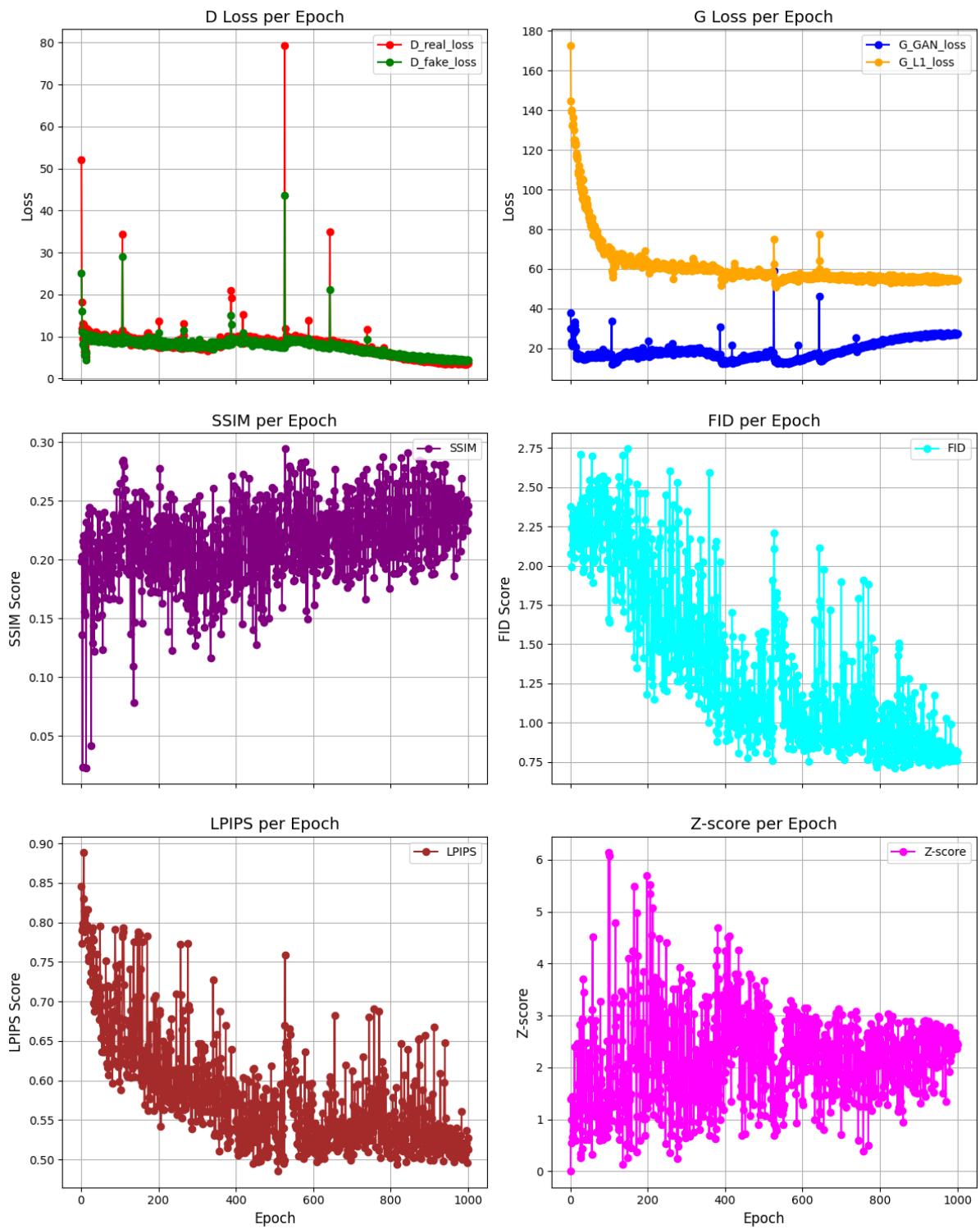
Supplementary Fig. 61. 4th round pix2pix training log of PtSn Clusters@Al₂O₃.



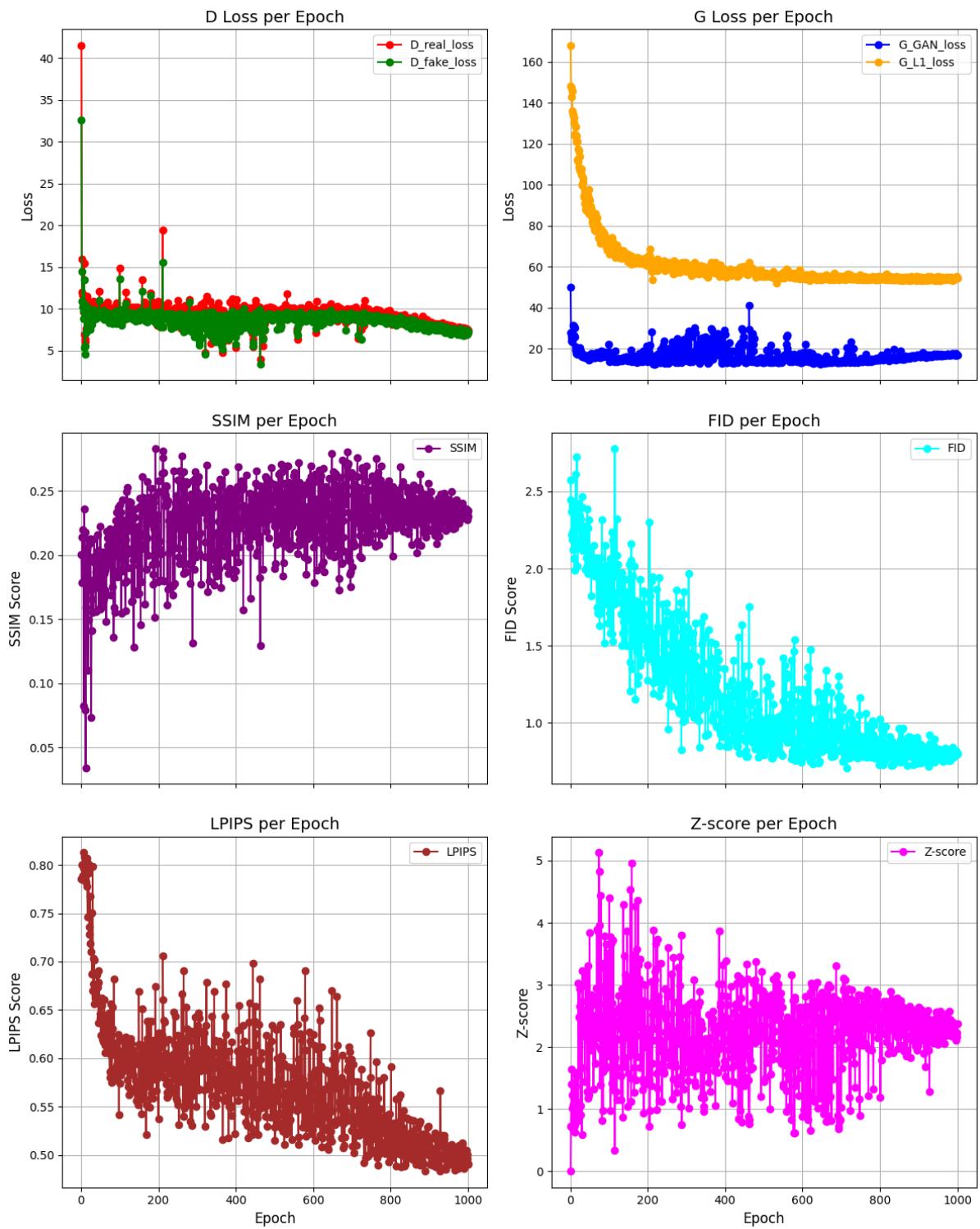
Supplementary Fig. 62. 5th round pix2pix training log of PtSn Clusters@Al₂O₃.



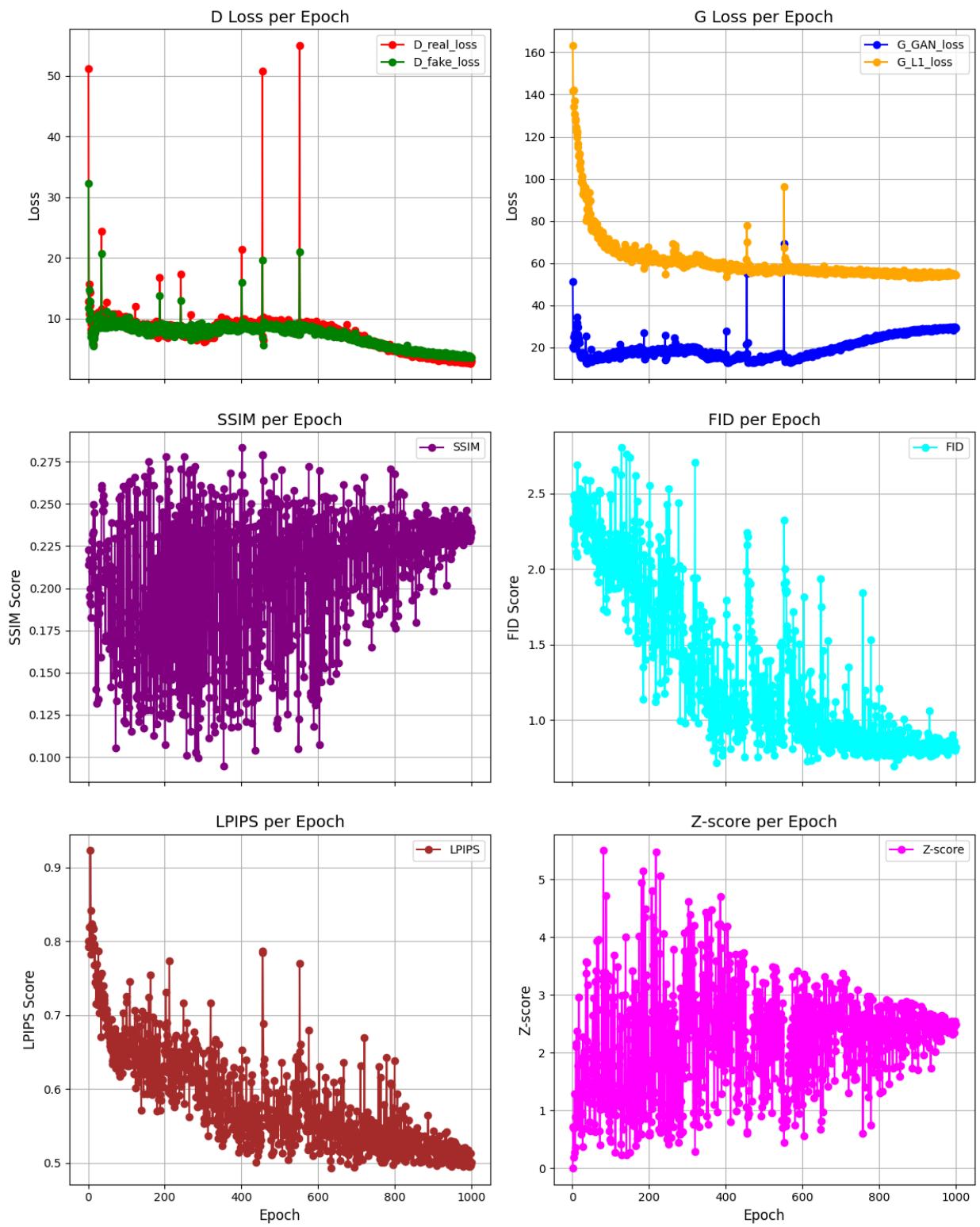
Supplementary Fig. 63. Loss and validation metrics of pix2pix trained for 1000 epochs of SACs.



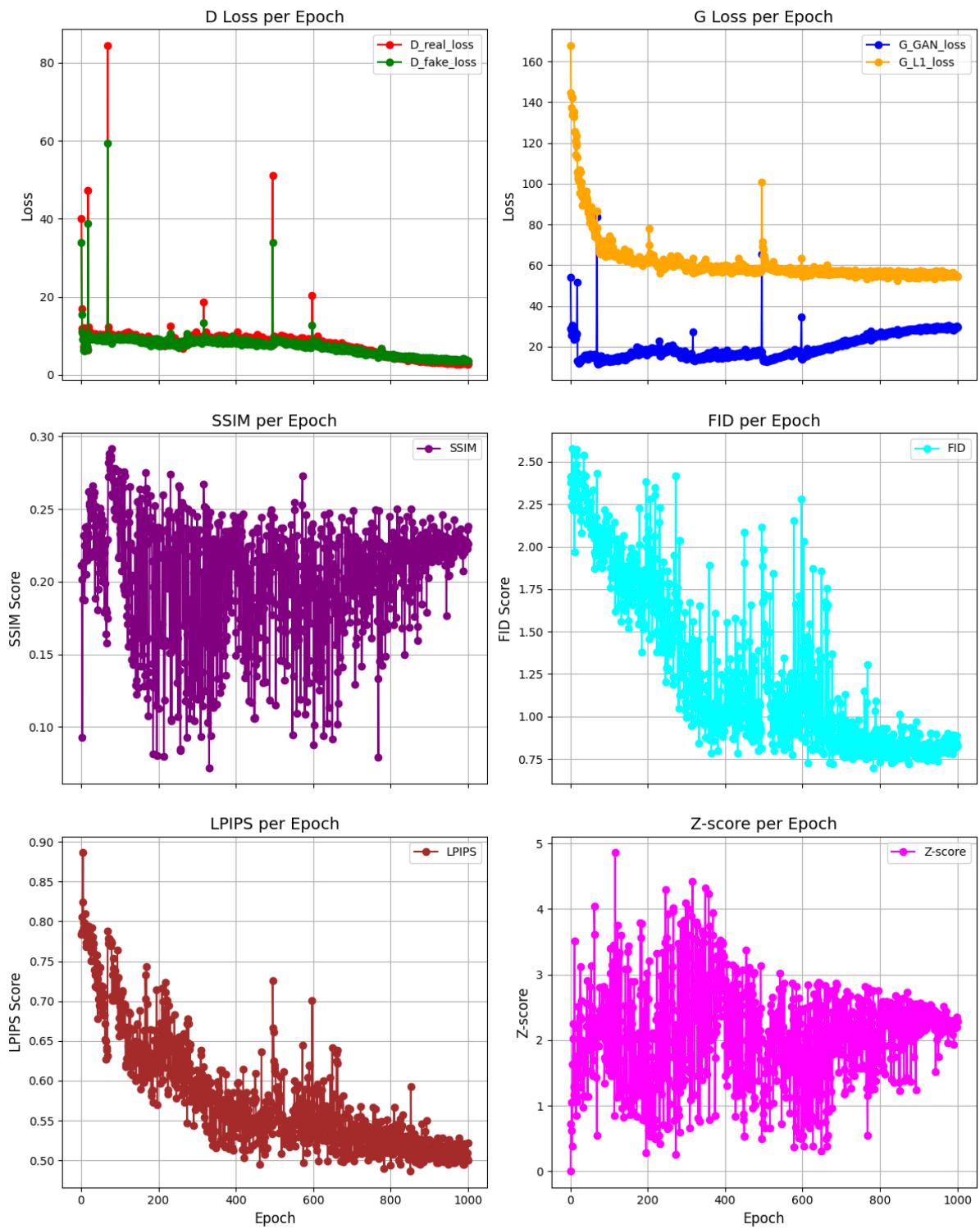
Supplementary Fig. 64. 1st round pix2pix training log of SACs.



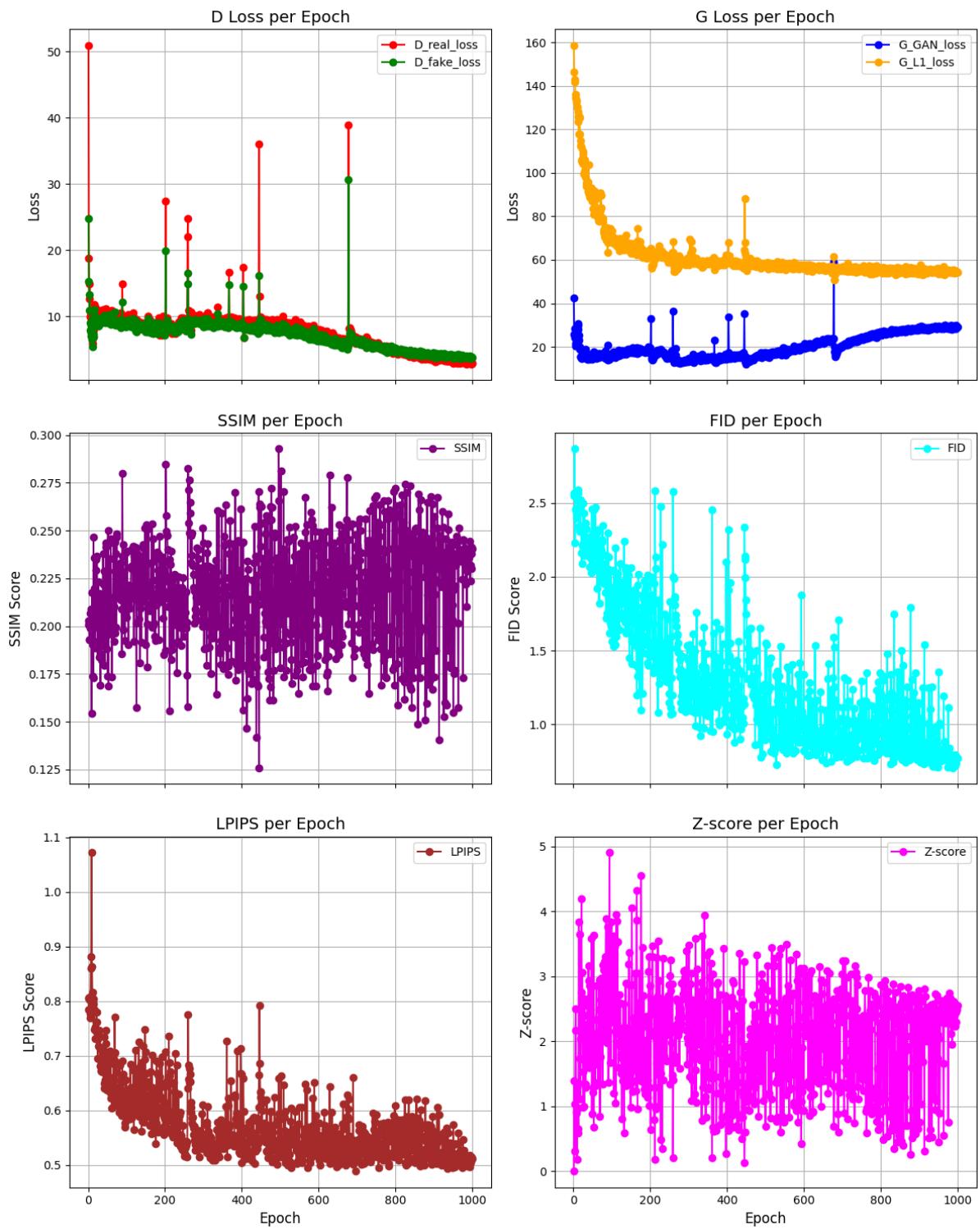
Supplementary Fig. 65. 2nd round pix2pix training log of SACs.



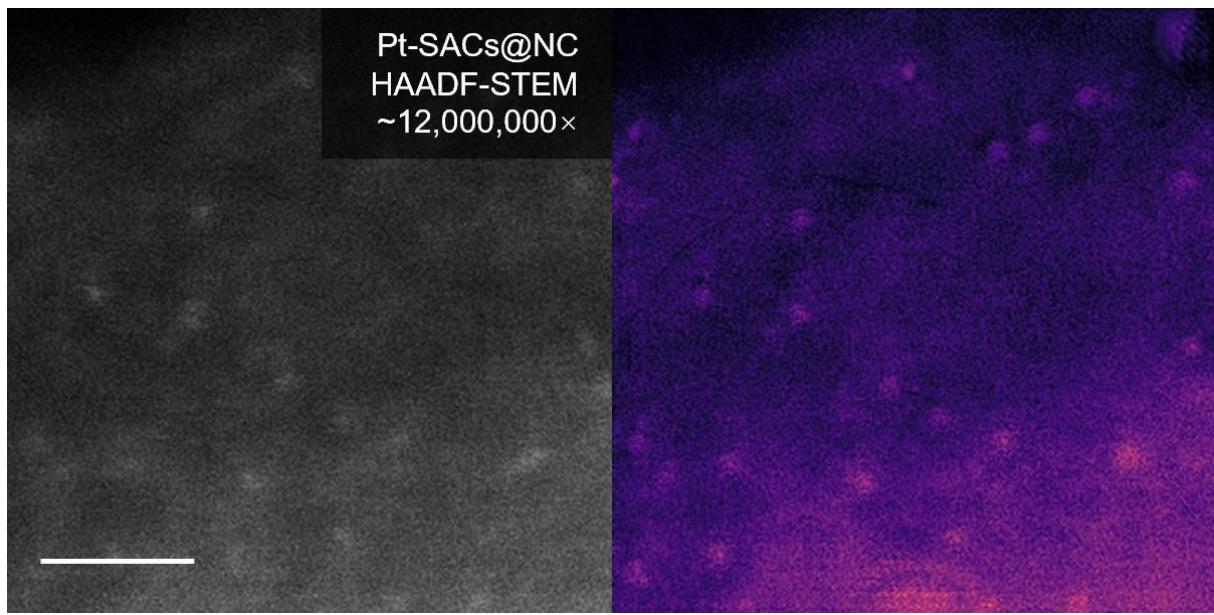
Supplementary Fig. 66. 3nd round pix2pix training log of SACs.



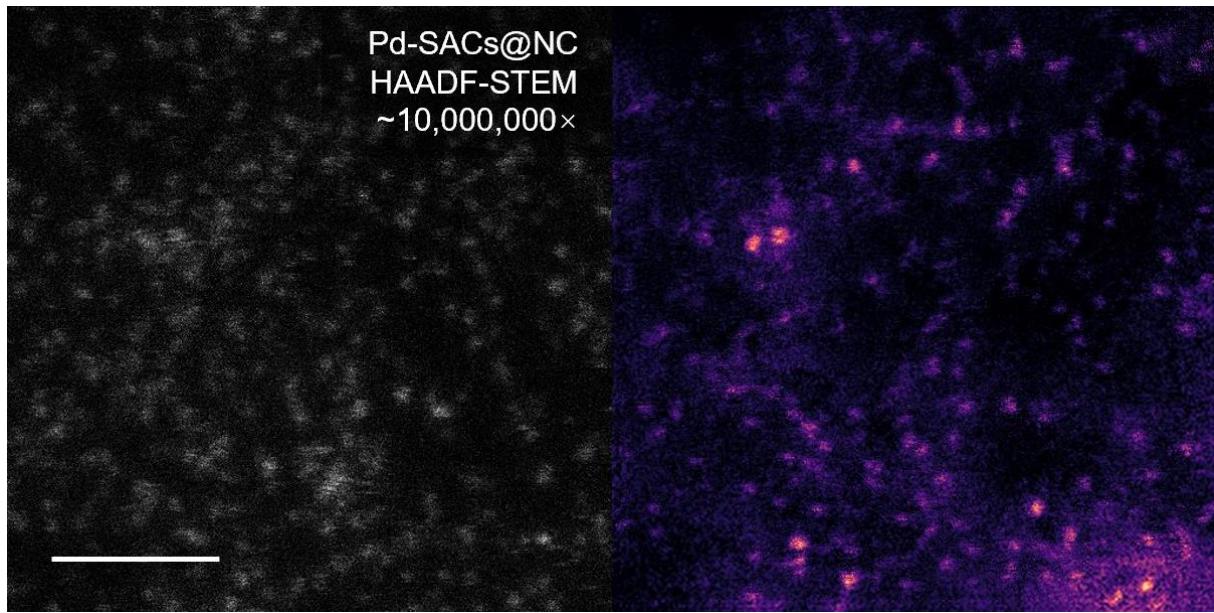
Supplementary Fig. 67. 4th round pix2pix training log of SACs.



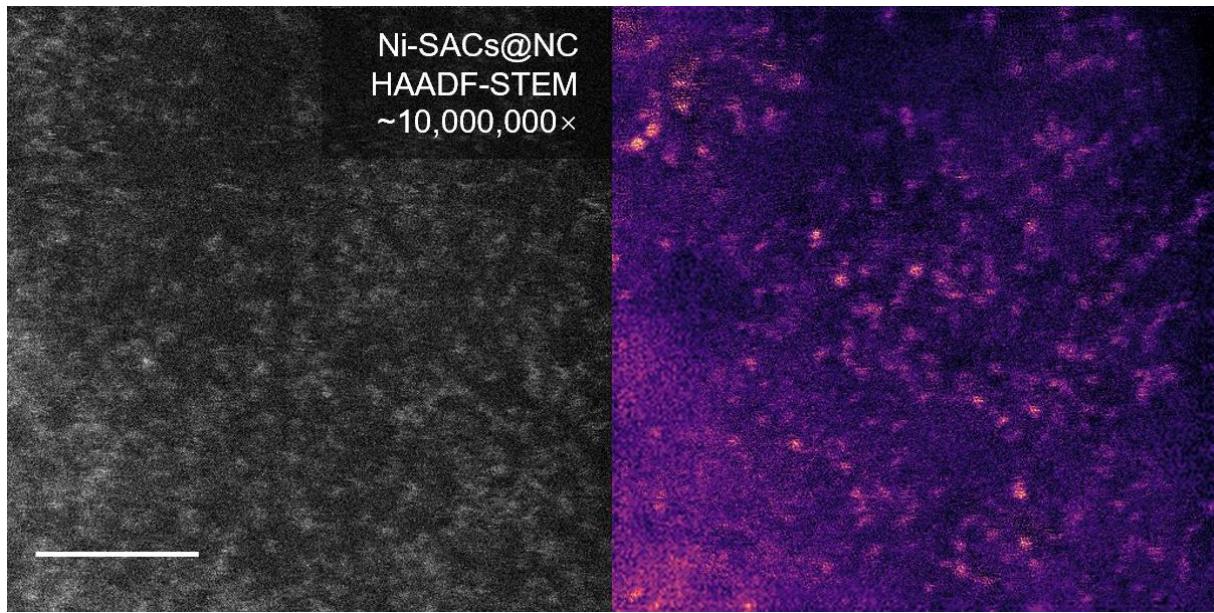
Supplementary Fig. 68. 5th round pix2pix training log of SACs.



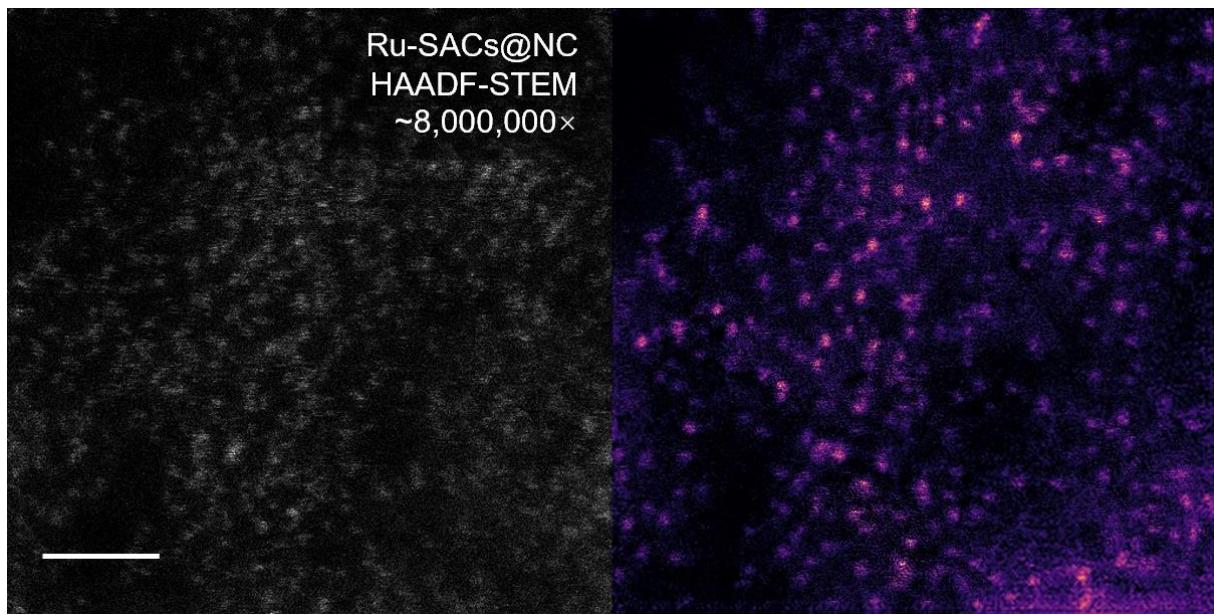
Supplementary Fig. 69. Experimental HAADF-STEM image (gray) and EMcopilot synthetic image (magma) of Pt single-atom catalysts (SACs) supported by nitrogen-doped amorphous carbon (NC) substrate (Pt-SACs@NC), scale bar 2 nm. The image is obtained from the public dataset associated with the work⁹ by Mitchell's group.



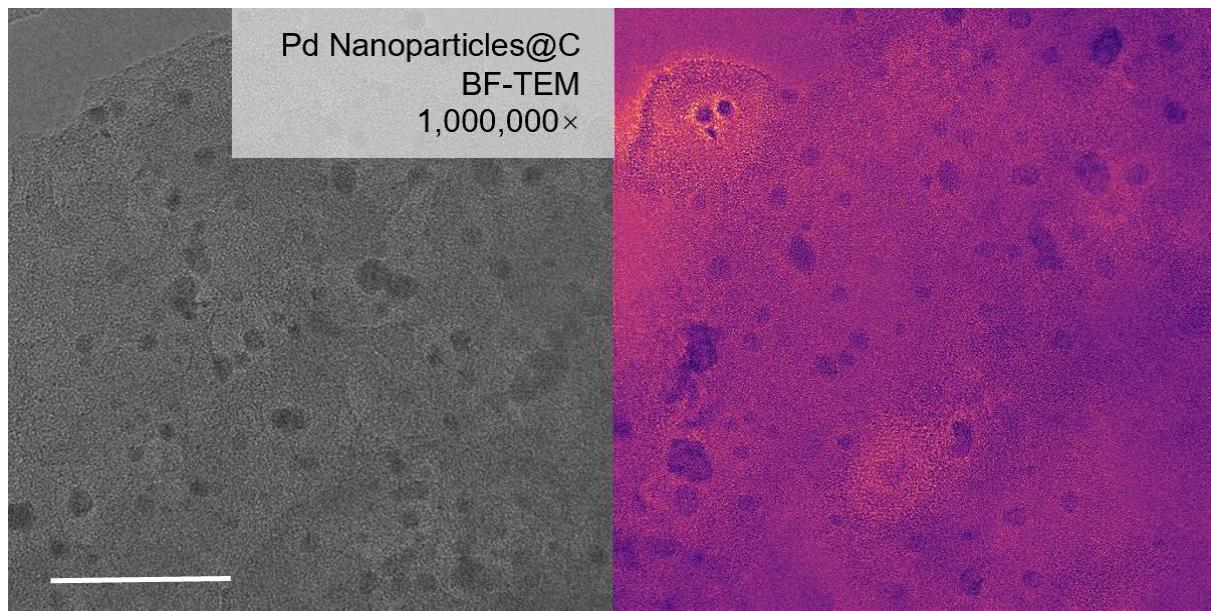
Supplementary Fig. 70. Experimental HAADF-STEM image (gray) and EMcopilot synthetic image (magma) of Pd-SACs@NC, scale bar 2 nm. The image is obtained from the public dataset associated with the work⁹ by Mitchell's group.



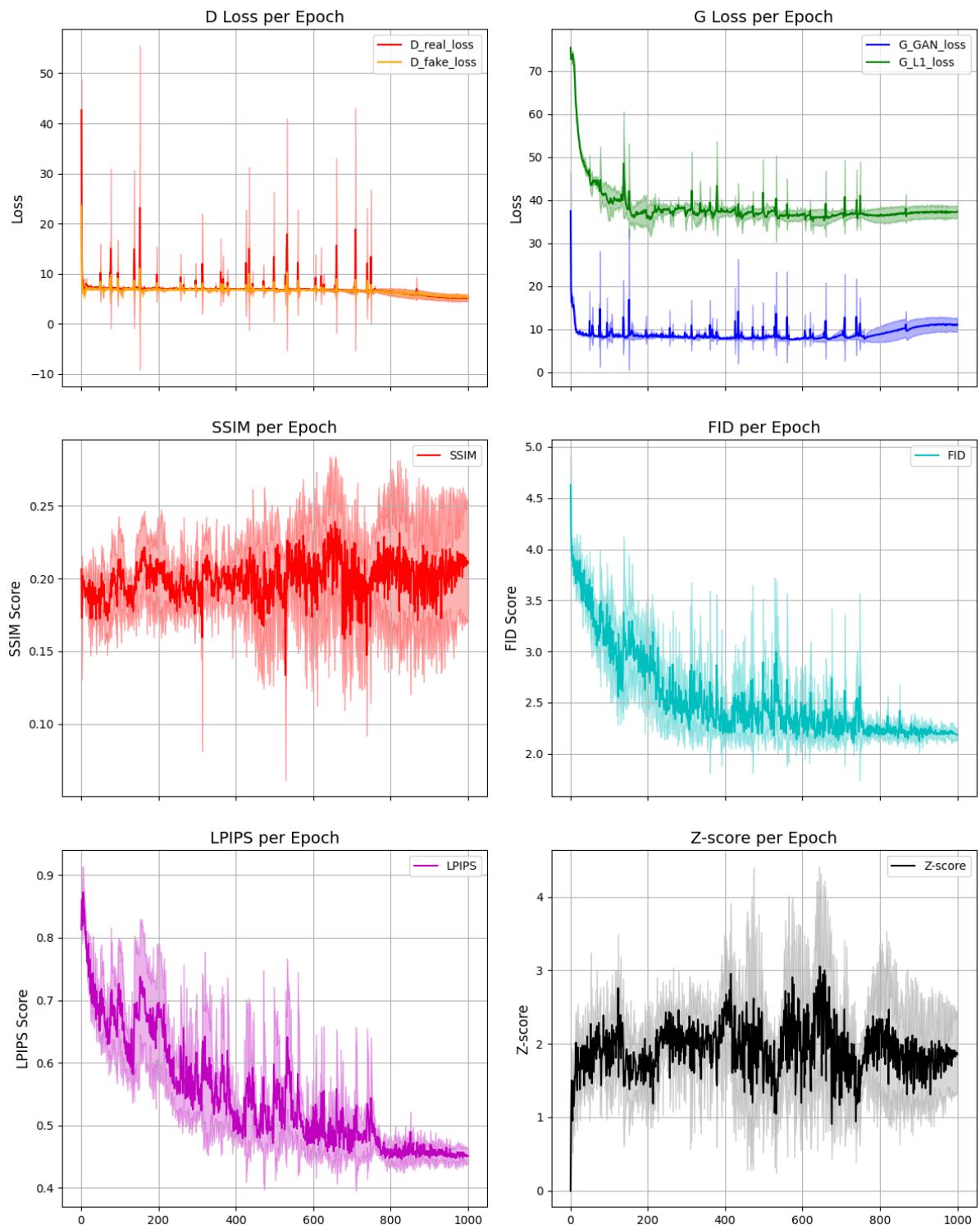
Supplementary Fig. 71. Experimental HAADF-STEM image (gray) and EMcopilot synthetic image (magma) of Ni-SACs@NC, scale bar 2 nm. The image is obtained from the public dataset associated with the work⁹ by Mitchell's group.



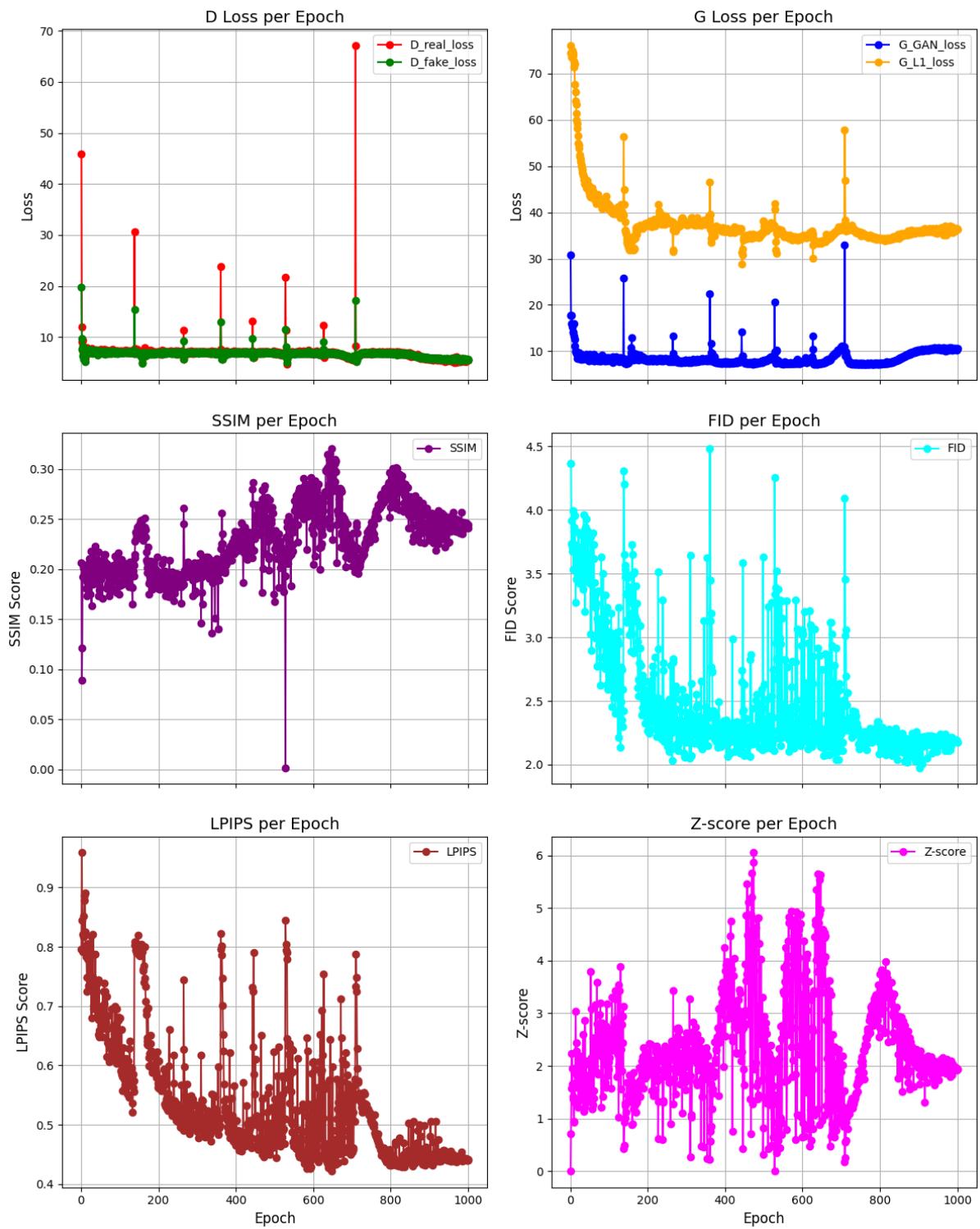
Supplementary Fig. 72. Experimental HAADF-STEM image (gray) and EMcopilot synthetic image (magma) of Ru-SACs@NC, scale bar 2 nm. The image is obtained from the public dataset associated with the work⁹ by Mitchell's group.



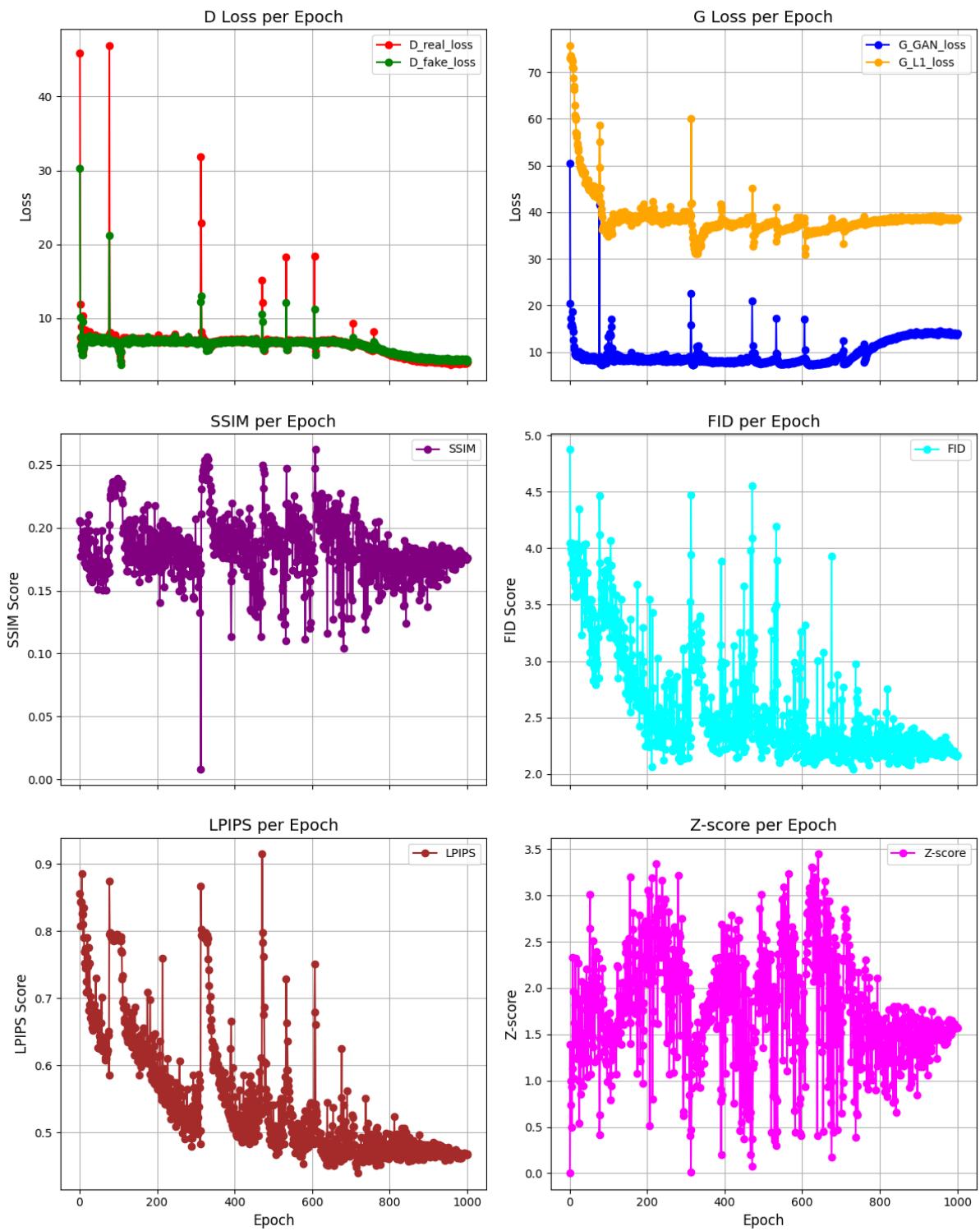
Supplementary Fig. 73. Experimental BF-TEM image (gray) and EMcopilot synthetic image (magma) of Pd nanoparticles supported on carbon (Pd@C), scale bar 50 nm. The image is obtained from the public dataset associated with the work¹⁰ by Kirkland's group.



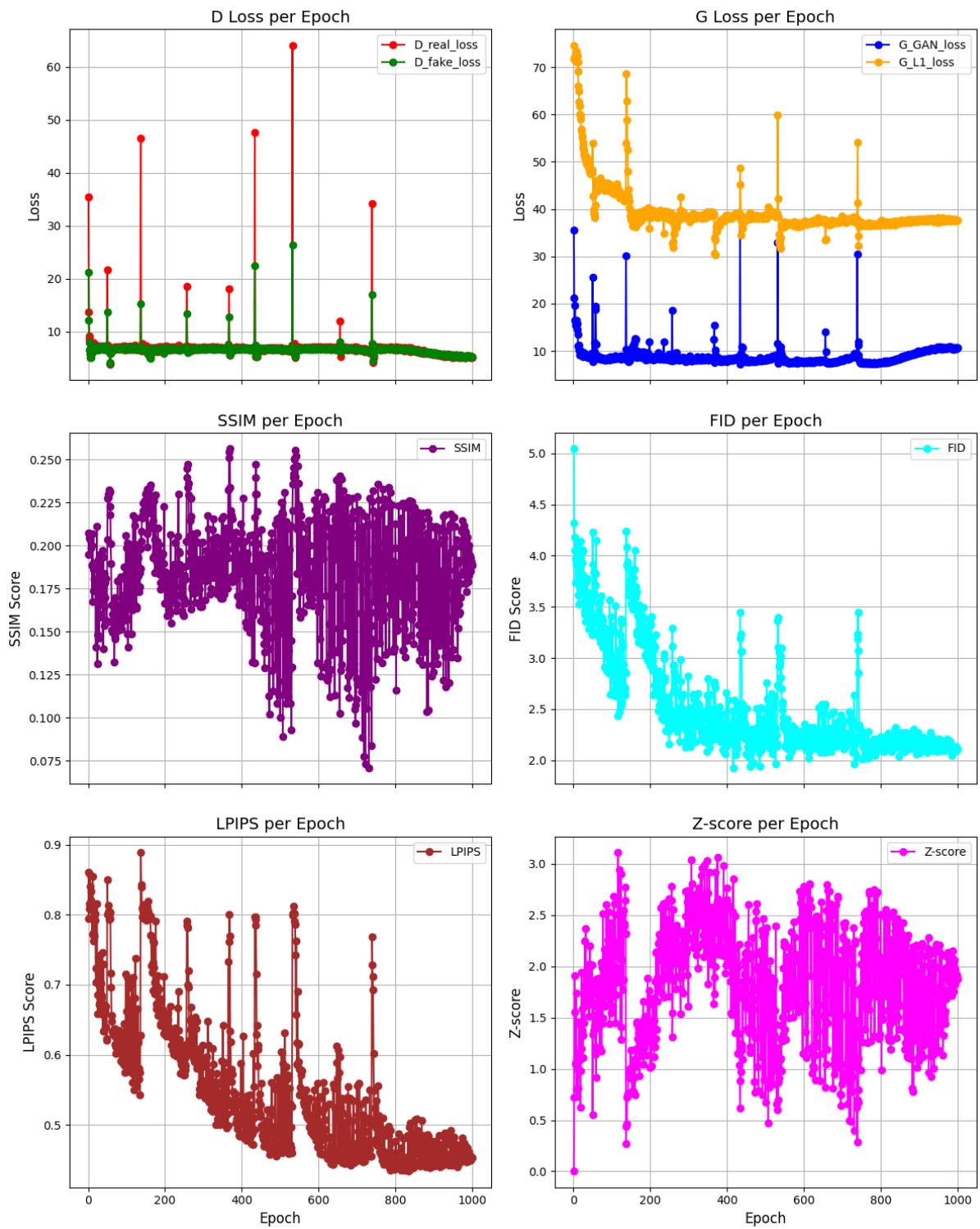
Supplementary Fig. 74. Loss and validation metrics of pix2pix trained for 1000 epochs of Au@ZSM-5.



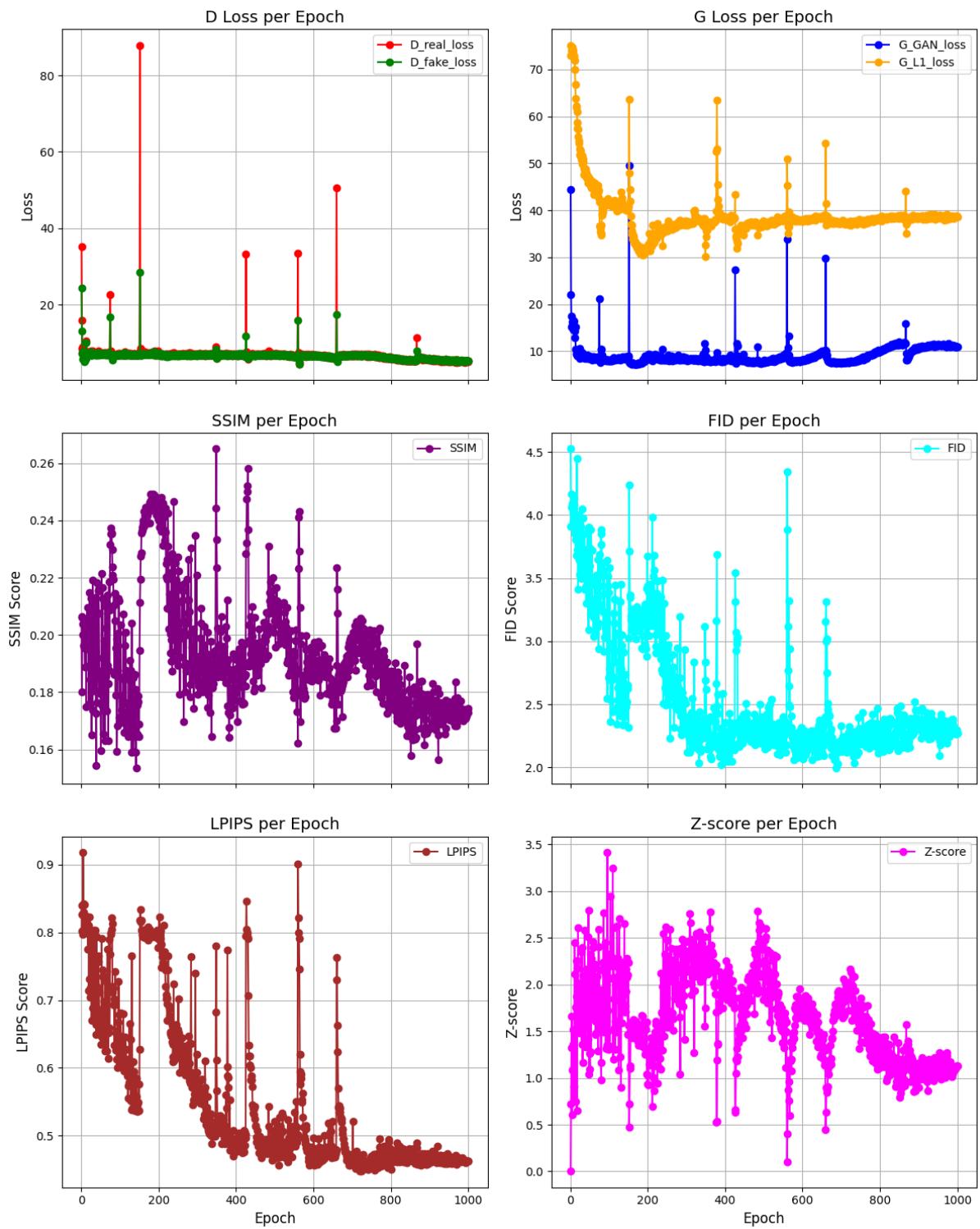
Supplementary Fig. 75. 1st round pix2pix training log of Au@ZSM-5.



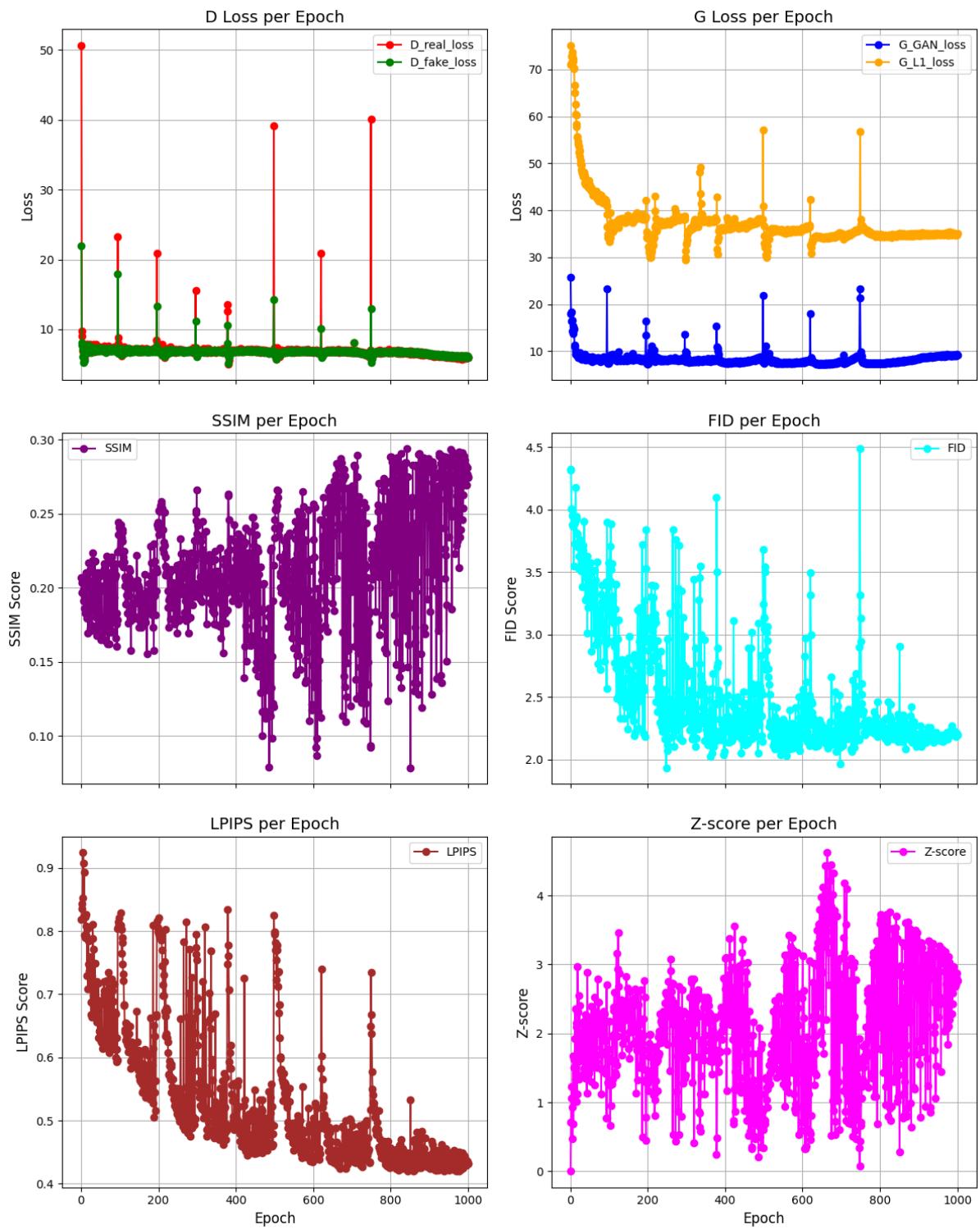
Supplementary Fig. 76. 2nd round pix2pix training log of Au@ZSM-5.



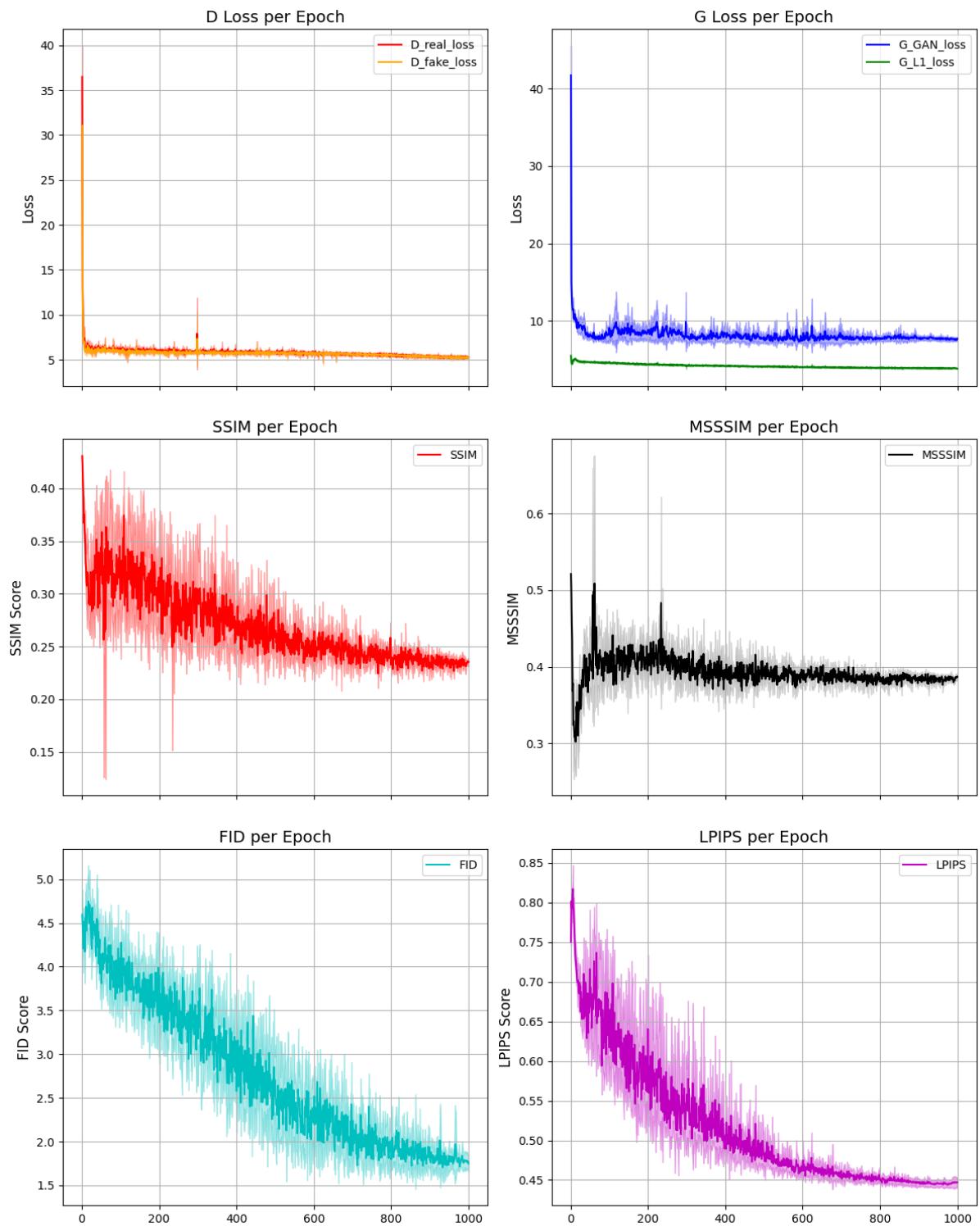
Supplementary Fig. 77. 3nd round pix2pix training log of Au@ZSM-5.



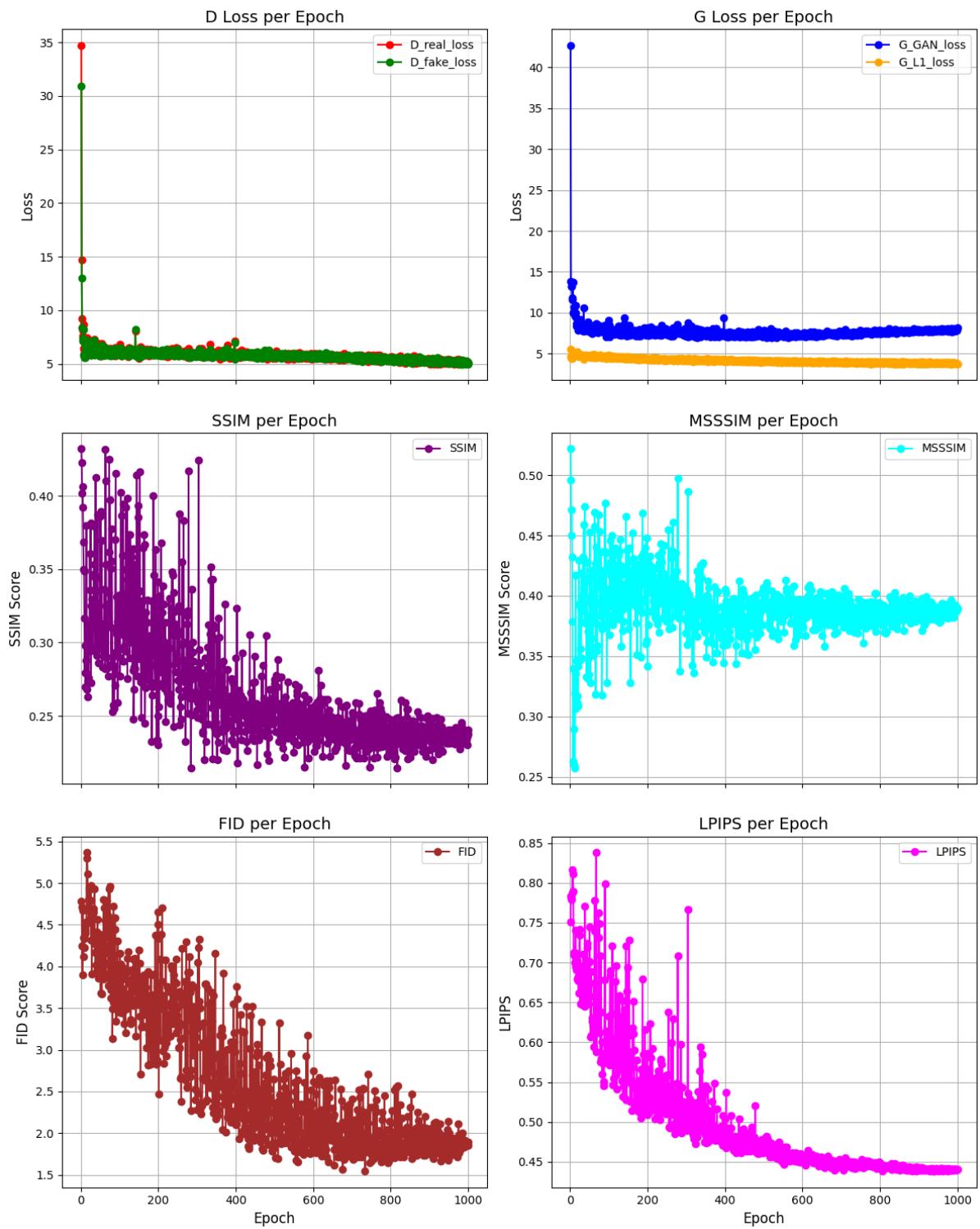
Supplementary Fig. 78. 4th round pix2pix training log of Au@ZSM-5.



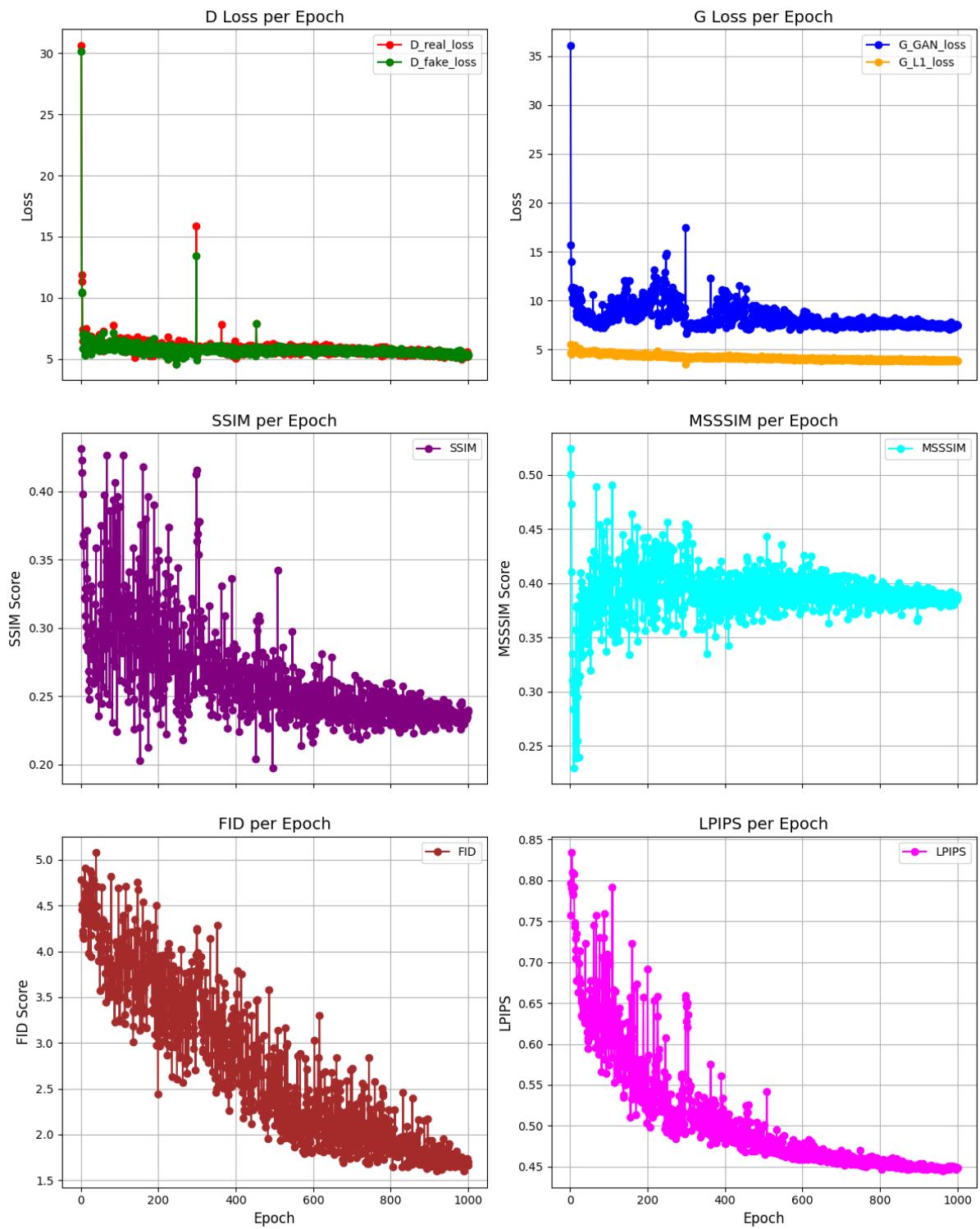
Supplementary Fig. 79. 5th round pix2pix training log of Au@ZSM-5.



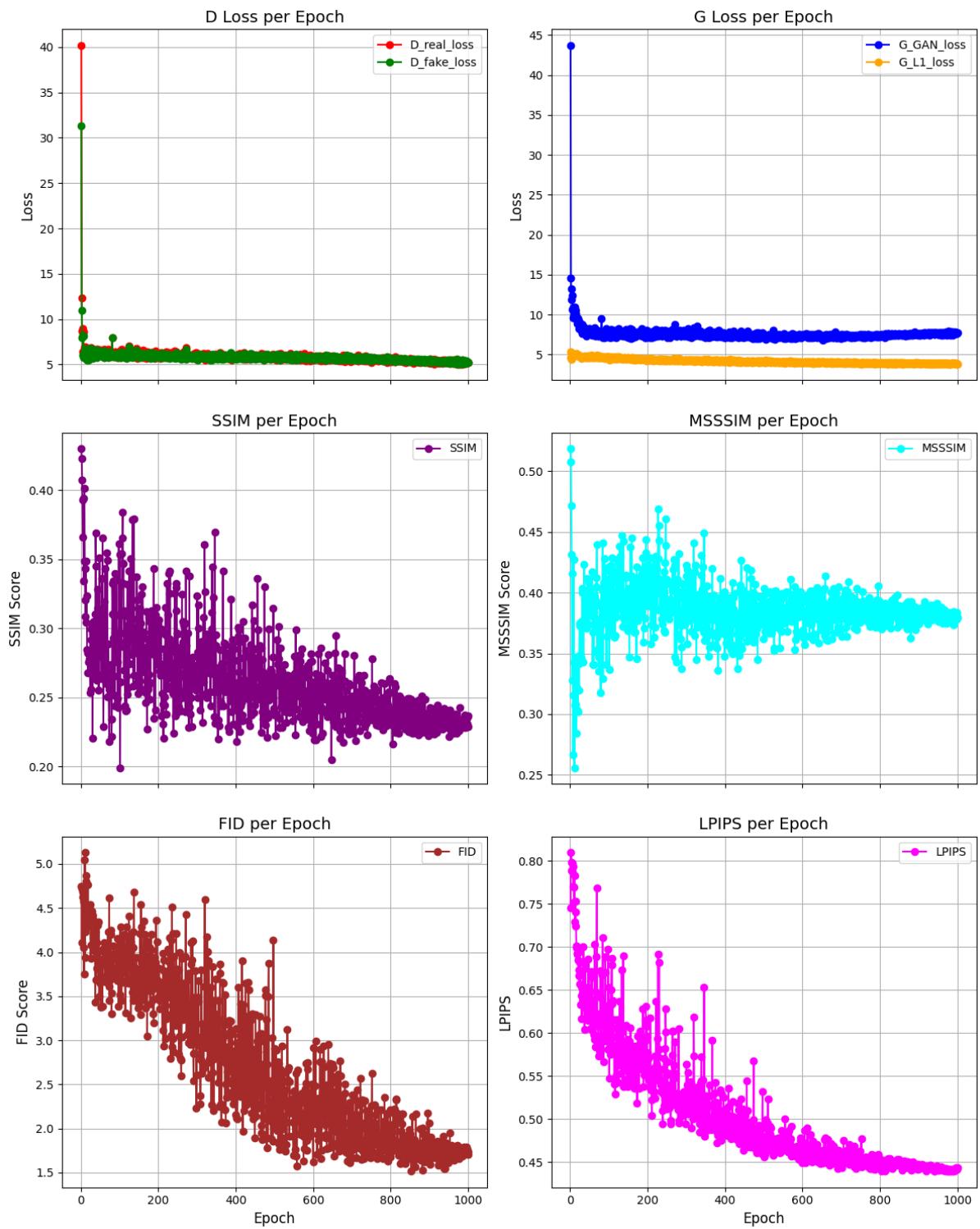
Supplementary Fig. 80. Loss and validation metrics of pix2pix trained for 1000 epochs of Pd@C.



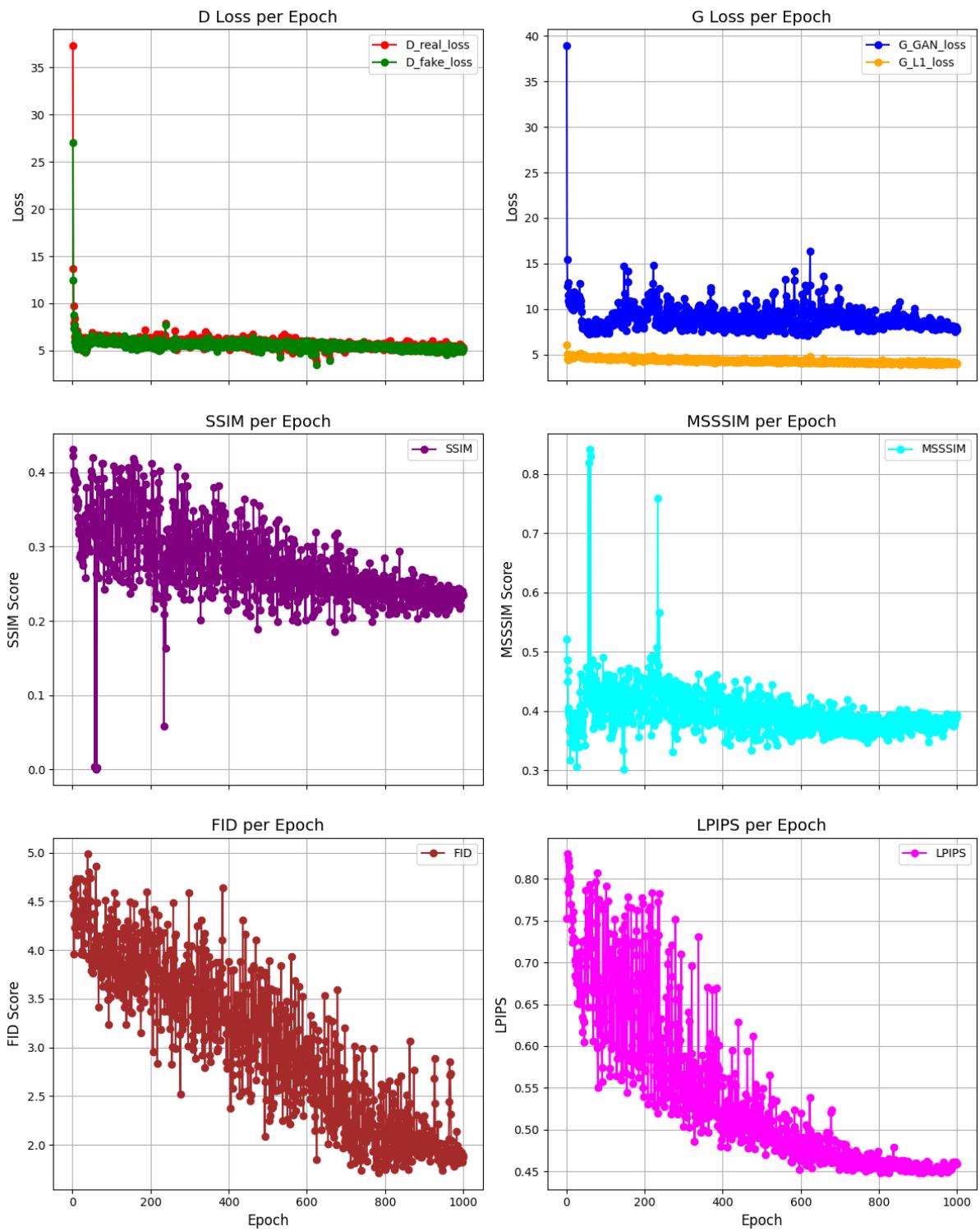
Supplementary Fig. 81. 1st round pix2pix training log of Pd@C.



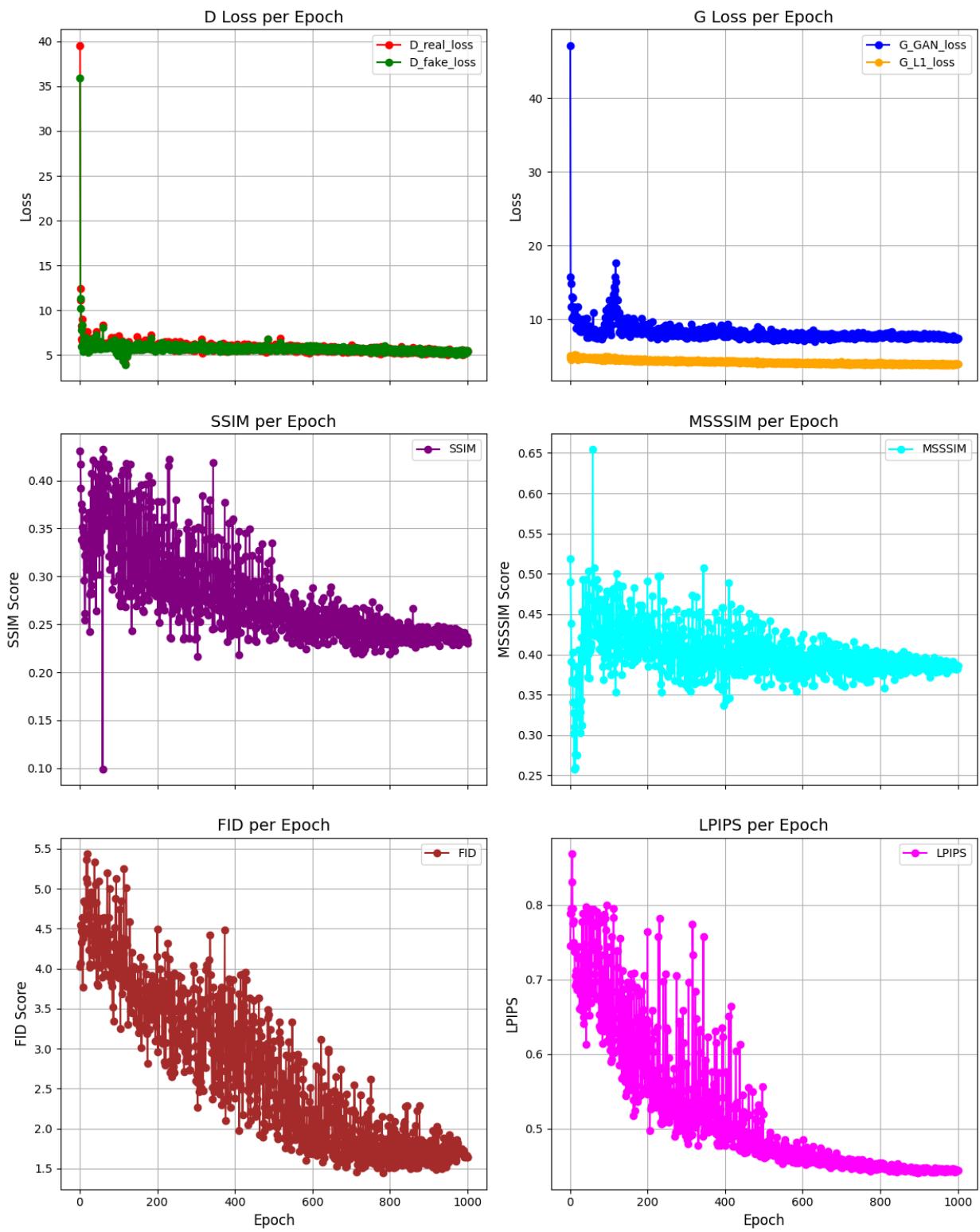
Supplementary Fig. 82. 2nd round pix2pix training log of Pd@C.



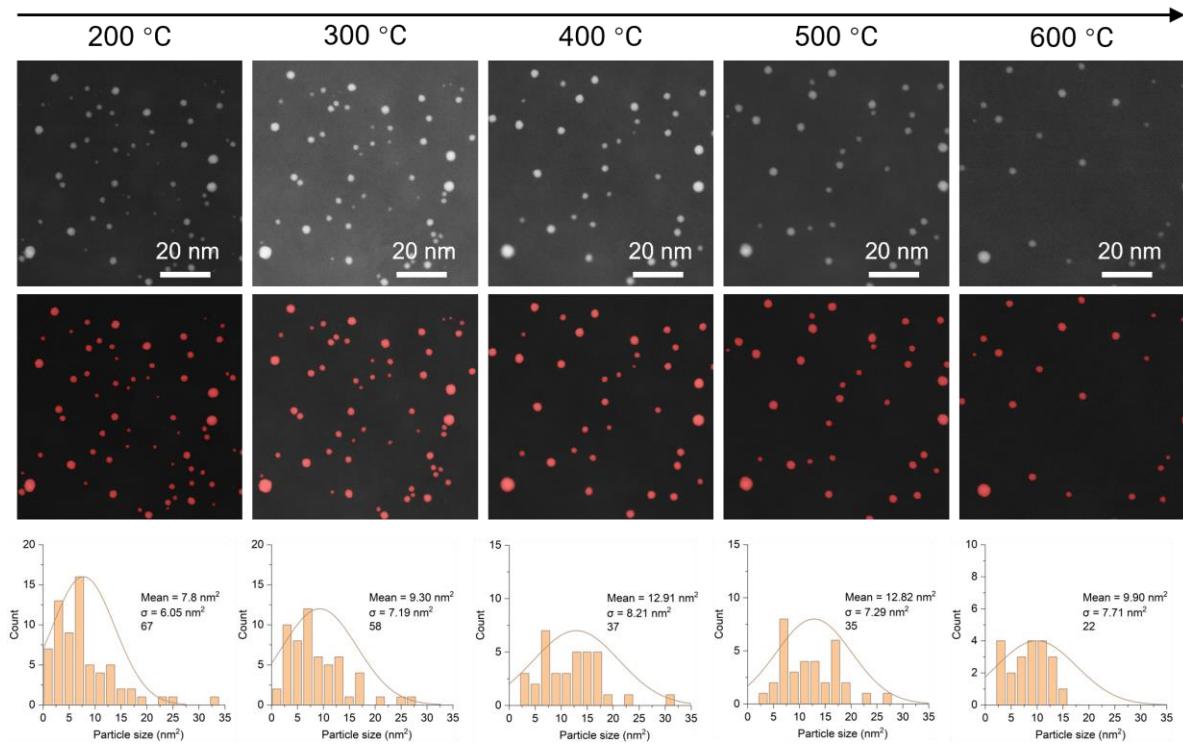
Supplementary Fig. 83. 3nd round pix2pix training log of Pd@C.



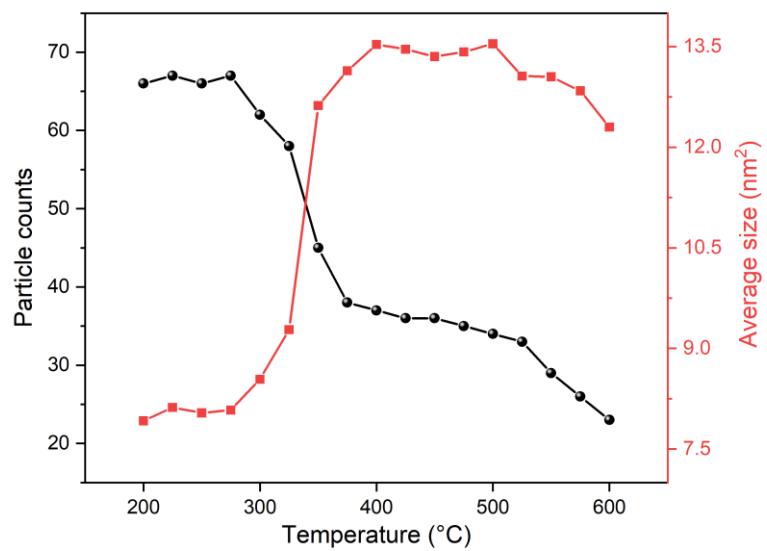
Supplementary Fig. 84. 4th round pix2pix training log of Pd@C.



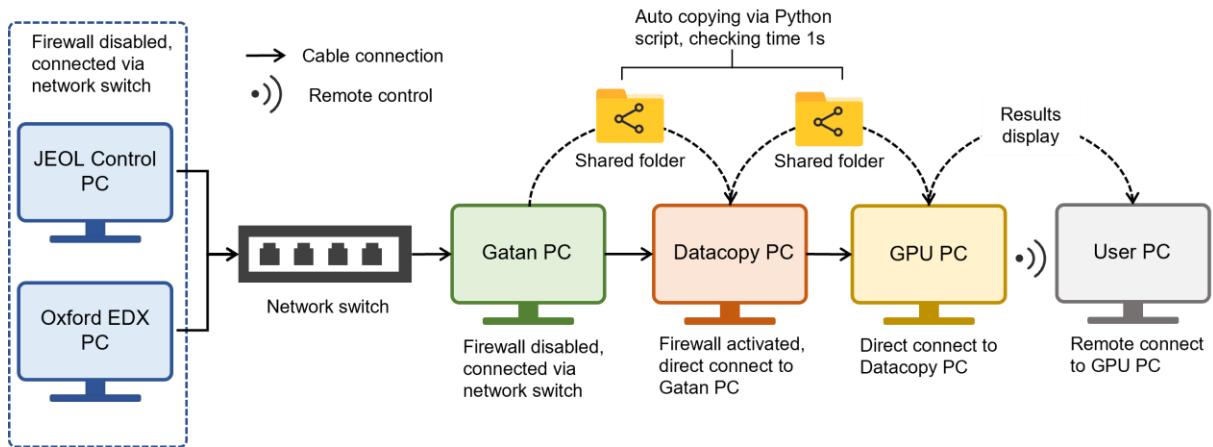
Supplementary Fig. 85. 5th round pix2pix training log of Pd@C.



Supplementary Fig. 86. Visualization of identifying nanoparticles dynamics and analyzing PSD during in situ experiments (Au@C).



Supplementary Fig. 87. Evolution of particle counts and size during in situ experiments.



Supplementary Fig. 88. Schematic illustration of the PC connections of ARM200F system for the in-situ experiments.

Supplementary Algorithms

Algorithm 1 SAM Mask Preprocessing and Filtering

```
1: Input: Raw mask images  $M$ , corresponding original images  $I$ 
2: Output: Processed masks with noise removed and edge filtering applied
3: Initialize input folders  $F_{mask}$  and  $F_{image}$ , output folder  $F_{processed}$ , and visualization folder  $F_{vis}$ .
4: Set pixel intensity threshold  $\theta$  for valid particles.
5: for each mask  $M_i$  in  $F_{mask}$  do
6:   Read  $M_i$  and corresponding image  $I_i$ .
7:   Resize both to  $512 \times 512$ .
8:   Convert  $M_i$  to binary.
9:   Step 1: Vacuum and Support Removal
10:  Find all particle contours in  $M_i$ .
11:  for each contour  $C_j$  do
12:    Compute mean intensity  $\mu_j$  in  $I_i$ .
13:    if  $\mu_j < \theta$  then continue (remove noise)
14:    end if
15:    Keep  $C_j$  in processed mask.
16:  end for
17:  Step 2: Edge Particles Filtering
18:  Label connected components in the mask.
19:  for each particle  $P_k$  do
20:    if  $P_k$  is near the image boundary then
21:      Remove  $P_k$ 
22:    end if
23:  end for
24:  Save final mask to  $F_{processed}$ .
25:  Step 3: Visualization
26:  Overlay original and processed masks onto  $I_i$ .
27:  Save visualization to  $F_{vis}$ .
28: end for
29: Print "Processing and visualization completed."
```

Supplementary Algorithm 1. Processing binary masks obtained from SAM-AMG.

Algorithm 2 Particle Analysis of SAM Masks

- 1: **Input:** Folder containing mask images (PNG format)
- 2: **Output:** Statistical summary and visualization of particle features
- 3: **Initialize** input folder F_{in} , output folder F_{out} , and create F_{out} if not exist.
- 4: Set visualization parameters for Matplotlib.
- 5: **for** each mask image M in F_{in} **do**
- 6: Read M as grayscale image.
- 7: Label particles using connected component analysis.
- 8: Initialize empty lists for particle statistics and contours.
- 9: **for** each particle P_i in M **do**
- 10: Extract particle contour.
- 11: **if** particle area $A_i < 10$ **then**
- 12: **continue** (skip small particles)
- 13: **end if**
- 14: Compute geometric features:
 - Perimeter (P_i)
 - Eccentricity (E_i)
 - Aspect Ratio (AR_i)
 - Circularity ($C_i = \frac{4\pi A_i}{P_i^2}$)
 - Solidity (S_i)
- 15: Store results in statistical summary.
- 16: Save contours as JSON.
- 17: **end for**
- 18: Save individual mask statistics to CSV.
- 19: **end for**
- 20: Merge all individual statistics into a global summary CSV.
- 21: Generate KDE histograms for Area, Perimeter, Eccentricity, Aspect Ratio, Circularity, and Solidity.
- 22: Save visualization as PNG.

Supplementary Algorithm 2. Extracting morphology priors from processed SAM masks.

Algorithm 3 Random Mask Generation from Morphology Priors

- 1: **Input:** JSON folder F_{json} containing particle contours
- 2: **Output:** New mask images and statistical summaries
- 3: Initialize output folders $F_{summary}$ and F_{masks} .
- 4: Set parameters: $N = 100$ (number of new masks per original), image size 512×512 , shift range, and rotation limits.
- 5: **for** each JSON file J in F_{json} **do**
- 6: Extract original contours $C = \{C_1, C_2, \dots, C_n\}$ from J .
- 7: **for** $i = 1$ to N **do**
- 8: Create a blank mask M .
- 9: Initialize an empty list C_{placed} for storing placed contours.
- 10: **for** each C_j in C **do**
- 11: Apply random shift within $\pm \text{SHIFT_RANGE}$.
- 12: Rotate by a random angle in $[0, 360^\circ]$.
- 13: Check for overlap with C_{placed} , if overlap exists, **continue**.
- 14: Add transformed contour to C_{placed} and draw on M .
- 15: **end for**
- 16: Remove boundary artifacts.
- 17: Compute geometric properties for valid particles:
 - Area (A), Perimeter (P), Eccentricity (E), Aspect Ratio (AR)
 - Circularity ($C = \frac{4\pi A}{P^2}$), Solidity (S)
- 18: Save new mask as an image file.
- 19: Append statistics to summary file.
- 20: **end for**
- 21: **end for**
- 22: Generate KDE histograms for statistical analysis.
- 23: Save results to CSV and PNG.

Supplementary Algorithm 3. Generating reorganized masks based on morphology priors.

Algorithm 4 Totally Random Mask Generation without Priors

```
1: Input: Image size  $S$ , base diameter  $D_0$ , max images  $N$ 
2: Output: Generated mask images stored in folder
3: Initialize parameter ranges:
4:     - Particle count  $P \in \{10, 20, 30, 40, 60, 80, 100\}$ 
5:     - Size variability  $\sigma_s \in \{0.4, 0.6, 0.8, 1\}$ 
6:     - Ellipse ratio  $\rho \in \{0.2, 0.5, 0.8\}$ 
7:     - Max ellipticity  $\epsilon_{max} \in \{0.3, 0.5, 0.7, 0.9\}$ 
8:     - Ellipticity variability  $\sigma_\epsilon \in \{0.05, 0.1, 0.2, 0.4, 0.6, 0.8\}$ 
9: for  $i = 1$  to  $N$  do
10:    Randomly select parameters  $P, \sigma_s, \rho, \epsilon_{max}, \sigma_\epsilon$ 
11:    Adjust mean diameter  $D$  based on  $P$ :
12:        
$$D = \min(D_0, \sqrt{\frac{S}{P}})$$

13:    Initialize empty image  $M$ 
14:    Compute  $P_{ellipse} = \rho P$ ,  $P_{circle} = (1 - \rho)P$ 
15:    for each shape type (circle or ellipse) do
16:        for each particle do
17:            Generate random radius  $r$  with variation  $\sigma_s$ 
18:            Compute ellipticity  $\epsilon = 1 - \sigma_\epsilon(1 - \epsilon_{max})$ 
19:            Generate ellipse mask with random rotation  $\theta$ 
20:            if no overlap with existing particles then
21:                Add particle to mask  $M$ 
22:            end if
23:        end for
24:    end for
25:    Save mask  $M$  as image file
26: end for
```

Supplementary Algorithm 4. Generating totally random masks without morphology priors.

Algorithm 5 pix2pix cGAN Training

- 1: **Input:** Training dataset D_{train} , validation dataset D_{val} , model parameters θ_G, θ_D
- 2: **Output:** Trained generator G^* with optimal performance
- 3: Initialize learning rate α , batch size B , total epochs E
- 4: Define loss functions: Adversarial Loss (\mathcal{L}_{GAN}), L1 Loss (\mathcal{L}_{L1}), SSIM Loss (\mathcal{L}_{SSIM})
- 5: Set dataset size sequence $S = \{s_1, s_2, \dots, s_n\}$ for progressive training
- 6: **for** each dataset size s_i in S **do**
- 7: Create training subset $D_{train}^{s_i} \subset D_{train}$ with s_i samples
- 8: **for** run $r = 1$ to 5 **do**
- 9: Initialize models G and D with parameters θ_G, θ_D
- 10: Initialize optimizers $\mathcal{O}_G, \mathcal{O}_D$ and schedulers
- 11: **for** epoch $e = 1$ to E **do**
- 12: **Training Phase (on $D_{train}^{s_i}$)**
- 13: **for** each batch $(X, Y) \in D_{train}^{s_i}$ **do**
- 14: Generate fake samples: $\hat{Y} = G(X)$
- 15: Compute discriminator losses:
16: $\mathcal{L}_{D_{real}} = E[\log D(X, Y)]$
17: $\mathcal{L}_{D_{fake}} = E[\log(1 - D(X, \hat{Y}))]$
18: Update θ_D using $\mathcal{L}_D = \frac{1}{2}(\mathcal{L}_{D_{real}} + \mathcal{L}_{D_{fake}})$
19: Compute generator losses:
20: $\mathcal{L}_{GAN} = E[\log D(X, \hat{Y})]$
21: $\mathcal{L}_{L1} = \|Y - \hat{Y}\|_1$
22: $\mathcal{L}_{SSIM} = 1 - SSIM(Y, \hat{Y})$
23: Update θ_G using $\mathcal{L}_G = \mathcal{L}_{GAN} + \lambda_{L1}\mathcal{L}_{L1} + \lambda_{SSIM}\mathcal{L}_{SSIM}$
- 24: **end for**
- 25: **Validation Phase (on D_{val})**
- 26: Compute validation metrics: SSIM, MSSSIM, FID, LPIPS
- 27: **if** metric improves **then**
- 28: Save best generator G^*
- 29: **end if**
- 30: Update learning rate
- 31: **end for**
- 32: **end for**
- 33: Compute statistical summary for dataset size s_i
- 34: **end for**
- 35: Return best trained generator G^*

Supplementary Algorithm 5. Training and validating the generator.

Algorithm 6 Noise and Contrast Adjustment for Domain Adaptation

- 1: **Input:** Grayscale images I from input folder
- 2: **Output:** Augmented images with various noise and contrast transformations
- 3: Initialize input folder F_{in} and output folder F_{out} .
- 4: Define **three types of noise** transformations:
 - 5: 1. **Shot noise** (N_s): Poisson noise with intensity $\{150, 250\}$
 - 6: 2. **Gaussian noise** (N_g): Normal distribution noise with std $\{0.03, 0.05\}$
 - 7: 3. **Scan noise** (N_{scan}): Sine-wave distortion with $\sigma_{jitter} = 2, freq = 0.001$
- 8: Define **six combined noise strategies**:
 - 9: (a) $N_s(150)$
 - 10: (b) $N_s(250)$
 - 11: (c) $N_g(0.03)$
 - 12: (d) $N_g(0.05)$
 - 13: (e) $N_s(250) + N_g(0.03)$
 - 14: (f) $N_s(250) + N_g(0.03) + N_{scan}$
- 15: Define **three contrast adjustment strategies** using sigmoid function:
 - 16: (i) $S(k = 2, i_0 = 0.3)$
 - 17: (ii) $S(k = 5, i_0 = 0.3)$
 - 18: (iii) No contrast adjustment (original)
- 19: The **total number of transformations** is $6 \times 3 = 18$.
- 20: **for** each image I_j in F_{in} **do**
- 21: Read and normalize image to range $[0, 1]$.
- 22: **for** each noise strategy N_i in $\{(a), (b), \dots, (f)\}$ **do**
- 23: Apply N_i to obtain noisy image \hat{I}_j .
- 24: **for** each contrast strategy S_m in $\{(i), (ii), (iii)\}$ **do**
- 25: Apply contrast adjustment $S_m(\hat{I}_j)$.
- 26: Save transformed image to F_{out} .
- 27: **end for**
- 28: **end for**
- 29: **end for**
- 30: Print "Processing completed."

Supplementary Algorithm 6. Augmenting the raw generated images by adding noise and contrast variation.

Algorithm 7 CBAM-UNet++ Hybrid Training

```
1: Input: Training dataset  $D_{train}$ , validation dataset  $D_{val}$ , model parameters  $\theta$ 
2: Output: Trained segmentation model  $M^*$  with optimal performance
3: Initialize learning rate  $\alpha$ , batch size  $B$ , total epochs  $E$ 
4: Define loss functions: Focal Loss ( $\mathcal{L}_{FL}$ ) and Jaccard Loss ( $\mathcal{L}_{JD}$ )
5: Define dataset size sequence  $S = \{s_1, s_2, \dots, s_n\}$  for progressive training
6: for each dataset size  $s_i$  in  $S$  do
7:   Create training subset  $D_{train}^{s_i} \subset D_{train}$  with  $s_i$  samples
8:   for run  $r = 1$  to  $5$  do
9:     Initialize model  $M$  with parameters  $\theta$ 
10:    for epoch  $e = 1$  to  $E$  do
11:      Training Phase (on  $D_{train}^{s_i}$ )
12:      for each batch  $(X, Y) \in D_{train}^{s_i}$  do
13:        Generate prediction:  $\hat{Y} = M(X)$ 
14:        Compute loss:
15:           $\mathcal{L} = \mathcal{L}_{FL}(\hat{Y}, Y) + \mathcal{L}_{JD}(\hat{Y}, Y)$ 
16:        Update  $\theta$  using gradient descent
17:      end for
18:      Validation Phase (on  $D_{val}$ )
19:      Compute evaluation metrics: mPA, mDice, mIoU, Recall, Precision
20:      if mIoU improves then
21:        Save best model  $M^*$ 
22:      end if
23:    end for
24:  end for
25:  Compute statistical summary for dataset size  $s_i$ 
26: end for
27: Return best trained model  $M^*$ 
```

Supplementary Algorithm 7. Training and validating the segmentor.

Supplementary Tables

Supplementary Table 1. Summary of reported methods and comparison with this work.

No. ^{a)}	Methods classification	Material system(s)	Imaging mode	Manual labeling necessity	Simulation necessity	Task type (s)	Baseline model (s)	Training set size (image counts)	Metrics and performance	Reference
1	Thresholding	Pt-NPs@C ^{b)}	BF-TEM	No	No	Segmentation	-	-	-	¹¹
2		Pt-SACs@Al ₂ O ₃	HAADF-STEM	No	No	-	-	-	-	¹²
3		Au-NPs@SiN	ETEM	No	No	Segmentation	U-Net ¹³	1,680	-	¹⁴
4	Supervised learning (via manual labeling)	Pt-SACs@NC, Fe-SACs@C ₃ N ₄	HAADF-STEM	Yes	No	Object detection	Customized CNN ¹⁵	21	Recall ≈ 0.900, F1 ^{c)} = 0.671	¹⁵
5		Pd-NPs@C	SEM	Yes	No	Segmentation	U-Net	14	IoU = 0.39~0.72	¹⁶
6		Pt-SACs@NC, Ni-SACs@NC, Pd-SACs@NC, Ru-SACs@NC, NiPd-SACs@NC, NiPdPt-SACs@NC	HAADF-STEM	Yes	No	Object detection	Customized CNN ⁹	49	Recall ≈ 0.54~0.90, Precision ≈ 0.48~0.90, F1 ≈ 0.57~0.88	⁹
7		NiCu-NPs@SiO ₂ , CuPd-NPs@SiO ₂	HAADF-STEM	Yes	No	Segmentation	U-Net	130	mAP = 0.787	¹⁷
8		PtCo-NPs@C, Cu-NPs@SiO ₂ , Ru-NPs@Al ₂ O ₃	HAADF-STEM and BF-TEM	Yes	No	Object detection and segmentation	YOLOv8 and SAM ¹⁸	25	Precision = 0.88~0.93, Recall = 0.90~0.94, F1 = 0.90~0.92, IoU = 0.81~0.84	¹⁹
9		Cu-NPs@C, CdSe-NPs@C	HRTEM	Yes	No	Segmentation	U-Net	285	F1 = 0.773	²⁰
10		Co ₃ O ₄ -NPs@C	HRTEM	Yes	No	Segmentation	U-Net	992	-	²¹
11		PtCo-NPs@C	ABF-STEM	Yes	No	Segmentation	U-Net	20	AP50 = 0.952~0.833	²²
12	Transfer learning ^{e)} (via simulations)	Graphene ^{d)} , WSe ₂ , SrTiO ₃	HAADF-STEM	No	Yes	Segmentation	FCN	107	Precision = 0.22~0.98, Recall = 0.40~0.91, F1 = 0.20~0.82,	²³

13		Pd-SACs@MgO	HAADF-STEM	No	Yes	Segmentation	U-Net	-	Precision= 0.858~0.830, Recall = 0.982~0.981, AUC = 0.997~0.989	²⁴
14		Pt-NPs@CeO ₂	HAADF-STEM	No	Yes	Segmentation	U-Net	7,500	F1 = 0.968~0.947	²⁵
15		Pd-NPs@Al ₂ O ₃	ETEM	No	Yes	Segmentation	U-Net	2,599	-	²⁶
16		Au-NPs@Ge, Pd-NPs@C	HRTEM	No	Yes	Object detection and segmentation	YOLOv5 and SegNet ²⁷	3,500	F1 = 0.967~0.253	¹⁰
17		W-SACs@KB, Pt-SACs@CN, Ir-SACs@MgO	HAADF-STEM	No	Yes	Object detection	Customized CNN ²⁸	100,000	Precision = 0.911~0.866, Recall = 0.975~0.864	²⁸
18		Pt-NPs@CeO ₂	HAADF-STEM	No	Yes	Segmentation	U-Net	4,000	-	²⁹
19		Pt-NPs@CeO ₂	HAADF-STEM	No	Yes	Segmentation	3D U-Net	4,000	-	³⁰
20		MoS ₂ , WSe ₂ ,	HAADF-STEM	No	Yes	Segmentation	U-Net	90	Precision = 0~1, Recall = 0~1, Recall = 0~1,	³¹
21	Self-supervised learning	PtSn-NPs@Al ₂ O ₃ , PtSn-Clusters@Al ₂ O ₃ , Au-NPs@ZSM5	HAADF-STEM and BF-TEM	No	No	Segmentation	U-Net++ ³²	33	AUC = 0.99 ^{f)} , AP = 0.82, Dice = 0.43, IoU = 0.59	This work

Note: ^{a)} this summary is not ensured to be exhaustive, and is mostly based on the representative works in recent five years;

^{b)} NPs refers to “nanoparticles”, SACs refers to “single-atom catalysts”, “-” refers to the linkage between compositional elements and catalyst type, and @ refers to “supported on”;

^{c)} for binary segmentation, F1 is equal is Dice;

^{d)} in this table, Graphene, WSe₂, SrTiO₃ and MoS₂ are all treated as 2D materials.

^{e)} “transfer” refers to the process in which the model is first trained on simulated images (sometimes with further enhancements), and then applied to experimental images—*i.e.*, a simulation-to-reality (sim-to-real) transfer.

^{f)} although our model does not achieve the best performance in this table, this outcome can be attributed to two main factors: (1) the evaluations are conducted on different materials and under varying experimental conditions, which can significantly impact model performance; and (2) as discussed in Section 2.5 of the main text, our model is capable of identifying additional true particles that are not included in the manually labeled masks. The presence of these unlabeled yet valid particles may also negatively influence the reported performance metrics.

Supplementary Table 2. Summary of datasets, validation objectives, evaluation metrics, and result references.

Stage	Objective	Input data	Evaluation Type	Metric(s)/Method(s)	Result location
SAM inference	Obtain SAM-generated masks	33 experimental images of PdSn-NPs@Al ₂ O ₃ (unlabeled)	-	-	Section 2.2; Fig. 2b; Supplementary Figs. 1-11;
Mask generation	Obtain reorganized masks	33 SAM-generated masks	-	-	Section 2.3; Fig. 2c; Supplementary Figs. 12
Mask comparison	Access the morphological feature extraction capabilities of SAM	17420 nanoparticles from 330 reorganized masks; 2860 nanoparticles from 33 manual-labeled masks; 1901 nanoparticles from 33 SAM-generated masks; 23317 nanoparticles from 330 random generated masks	Quantitative	Area, perimeter, eccentricity, circularity, aspect ratio, and solidity	Section 2.2; Fig. 2f; Supplementary Figs. 13-15, 19 Supplementary Algorithms 2-4
				NND	Section 2.2; Supplementary Fig. 17
				PDF	Section 2.2; Supplementary Fig. 18
		330 reorganized masks; 33 manual-labeled masks; 33 SAM-generated masks	Qualitative	UMAP	Section 2.2; Fig. 2g
pix2pix training	Monitor the training process and prevent overfitting	33 experimental images of PdSn-NPs@Al ₂ O ₃ and 33 synthetic images generated from the corresponding manual-labeled masks	Quantitative	SSIM	Section 2.3; Fig. 2h; Supplementary Figs. 20-25; Supplementary Algorithm 5
				LPIPS	
				MSSSIM	
				FID	
		23 experimental images of PdSn-Clusters@Al ₂ O ₃ and 23 synthetic images generated from the corresponding manual-labeled masks	Quantitative	SSIM	Section 2.6; Fig. 5b; Supplementary Figs. 57-62; Supplementary Algorithm 5
				LPIPS	
				MSSSIM	
				FID	
		25 experimental images of SACs (Pt-SACs@C, Ni-SACs@NC, Ru-SACs@NC, Pd-SACs@NC, and Pt-SACs@NC) and 25 synthetic images generated from the corresponding manual-labeled masks	Quantitative	SSIM	Section 2.6; Fig. 5c; Supplementary Figs. 63-72; Supplementary Algorithm 5
				LPIPS	
				MSSSIM	
				FID	
		110 experimental images of Au-NPs@ZSM-5 and 110 synthetic images generated from the corresponding manual-labeled masks	Quantitative	SSIM	Section 2.6; Fig. 5d; Supplementary Figs. 73-79; Supplementary Algorithm 5
				LPIPS	
				MSSSIM	
				FID	
		96 experimental images of Pd-NPs@C and 96 synthetic images generated from the corresponding manual-labeled masks	Quantitative	SSIM	Section 2.6; Supplementary Figs. 80-85; Supplementary Algorithm 5
				LPIPS	
				MSSSIM	
				FID	
		Compare the fine-detail quality of generated images of best SSIM model and best LPIPS model	Quantitative	DISTS	Section 2.3; Fig. 2i
				LPIPS	
		Compare the fidelity of generated images of best SSIM model and best LPIPS model	Qualitative	VIF	Section 2.3; Fig. 2j
		PSNR			

	Validate that the effect of imperfections of SAM masking have negligible impact on pix2pix training	33 experimental images of PdSn-NPs@Al ₂ O ₃ and 33 synthetic images generated from the corresponding manual-labeled masks		SSIM LPIPS MSSSIM FID	Section 2.3; Supplementary Figs. 30-35
pix2pix inference	Obtain pix2pix generated images	330 random generated masks from SAM	-	-	Section 2.3; Fig. 2d
Domain adaptation	Validate that the distribution of domain-adapted images is closer to that of experimental images	33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images	Qualitative	t-SNE	Section 2.4; Fig. 3c-e
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		UMAP	Section 2.4; Supplementary Figs. 39-40
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images		PCA	Section 2.4; Supplementary Figs. 41-42
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		t-SNE	Section 2.4; Fig. 3c-e
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images		UMAP	Section 2.4; Supplementary Figs. 39-40
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		PCA	Section 2.4; Supplementary Figs. 41-42
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images	Quantitative	NND	Section 2.4; Fig. 3f
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		PDF	Section 2.4; Fig. 3g
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images		NND	Section 2.4; Fig. 3f
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		PDF	Section 2.4; Fig. 3g
Assess the impact of domain adaptation on data diversity	Assess the impact of domain adaptation on data diversity	33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images	Qualitative	3D t-SNE	Section 2.4; Supplementary Figs. 43-46
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		Overlap volume ratio of 3D t-SNE	
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images	Quantitative	MD	
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		VIF	
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images		PSNR	
	Assess the fidelity of domain adapted images	33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images			Section 2.4; Fig. 3i
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images			
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images			
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images			
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images			

	Assess the fine-detail quality of domain adapted images	33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images	DISTS LPIPS FID KID	Section 2.4; Fig. 3j
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images		
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images		
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images		
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		
CBAM-UNet++ training	Assess the distributional consistency of domain adapted images with experimental images	33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images	FID	Section 2.4; Fig. 3k
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and with 330 raw synthetic images		
		33 experimental images of PdSn-NPs@Al ₂ O ₃ and 1,000 images randomly selected from 5,940 domain-adapted images		
		26 manual-labeled images for training (80% of 33), and 7 manual-labeled images for test (20% of 33)		
	Monitor the training process	4,752 synthetic images (80% of 5,940), 1,188 synthetic images for test (20% of 5,940)	mPA mIoU mDice Precision Recall	Section 2.5; Supplementary Figs. 47-50
		26 manual-labeled images for training (80% of 33), and 7 manual-labeled images for test (20% of 33)		
		4,752 synthetic images (80% of 5,940), 1,188 synthetic images for test (20% of 5,940)		
		4,752 synthetic images (80% of 5,940), 1,188 synthetic images for test (20% of 5,940)		
		4,752 synthetic images (80% of 5,940), 1,188 synthetic images for test (20% of 5,940)		
CBAM-UNet++ inference	Assess the segmentation performance	33 experimental images of PdSn-NPs@Al ₂ O ₃	Qualitative	Section 2.5; Fig. 4a, e
		33 experimental images of PdSn-NPs@Al ₂ O ₃		
		33 experimental images of PdSn-NPs@Al ₂ O ₃		
		33 experimental images of PdSn-NPs@Al ₂ O ₃		
		33 experimental images of PdSn-NPs@Al ₂ O ₃		
	Quantitative	AUC	Grad-CAM	Section 2.5; Fig. 4h, e
		AP		
		Dice		
		IoU		
		AUC		

			Qualitative	Grad-CAM	Section 2.5; Fig. 4h, e
--	--	--	-------------	----------	----------------------------

Supplementary Table 3. Settings of ablation studies.

Stage(s) removed	Remaining stage(s)	Experiment implemented	Corresponding setup	Result location	Remarks
1 ^{a)}	2+3	Yes	Totally random generated masks	Section 2.2; Supplementary Fig. 19	-
2	1+3	No	-	-	This will be no generated data in this case.
3	1+2	No	-	-	There will be no segmentator to be trained in this case, the generated data will be useless.
2+3	1	Yes	SAM-AMG	Section 2.5; Fig. 4a-e	-
1+3	2	No	-	-	There will be no segmentator in this case, performance cannot be tested.
1+2	3	Yes	ML	Section 2.5; Fig. 4a-e	-

Note: ^{a)} stage 1 is coarse segmentation, stage 2 is image generation, and stage 3 is segmentor training.

Supplementary Table 4. Statistics of training and inference time.

Model	Stage	Data ^{b)}	Run No.	Total time ^{a)} (s)	Avg. time
pix2pix	Training (1000 epochs)	33 SAM-masked images	1	10,849.70	10.83 s run ⁻¹ epoch ⁻¹
			2	10,871.98	
			3	10,860.27	
			4	10,787.63	
			5	10,803.81	
pix2pix	Inference (GPU) ^{c)}	5,940 SAM-masked images	-	82.68	0.01 s image ⁻¹
pix2pix	Inference (CPU) ^{d)}	5,940 SAM-masked images	-	2,014.85	0.34 s image ⁻¹
CBAM-Unet++	Training (40 epochs)	33 experimental images	1	154.65	3.76 s run ⁻¹ epoch ⁻¹
			2	149.35	
			3	150.07	
			4	148.85	
			5	149.67	
CBAM-Unet++	Training (15 epochs)	5,940 synthetic images	1	3,856.90	254.62 s run ⁻¹ epoch ⁻¹
			2	3,828.25	
			3	3,807.65	
			4	3,816.02	
			5	3,787.49	
CBAM-Unet++	Inference (GPU) ^{c)}	33 experimental images	-	0.50	0.02 s image ⁻¹
CBAM-Unet++	Inference (CPU) ^{d)}	33 experimental images	-	12.38	0.38 s image ⁻¹

Note: ^{a)} the size of all images is 512 × 512 pixels;

^{b)} the reported time includes not only the training process but also the evaluation and recording of performance metrics in every epoch. Therefore, the actual training time is shorter than the reported value;

^{c)} RTX 3080 (10 GB);

^{d)} 12th Gen Intel(R) Core(TM) i7-12700 (20 cores).

Reference

1. Kirillov, A. *et al.* in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4015-4026.
2. Vaswani, A. Attention is all you need. *NeurIPS* (2017).
3. Archit, A. *et al.* Segment anything for microscopy. *bioRxiv*, 2023.2008. 2021.554208 (2023).
4. Ma, J. *et al.* Segment anything in medical images. *Nat. Commun.* **15**, 654 (2024).
5. Osco, L. P. *et al.* The segment anything model (sam) for remote sensing applications: From zero to one shot. *International Journal of Applied Earth Observation and Geoinformation* **124**, 103540 (2023).
6. Li, C., Han, X., Yao, C. & Ban, X. MatSAM: Efficient Materials Microstructure Extraction via Visual Large Model. *arXiv preprint arXiv:2401.05638* (2024).
7. Larsen, R., Villadsen, T. L., Mathiesen, J. K., Jensen, K. M. & Bojesen, E. D. NP-SAM: Implementing the Segment Anything Model for Easy Nanoparticle Segmentation in Electron Microscopy Images. (2023).
8. Nichols, K. *et al.* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8120-8124.
9. Rossi, K. *et al.* Quantitative Description of Metal Center Organization and Interactions in Single-Atom Catalysts. *Adv. Mater.* **36**, e2307991 (2024).
10. Treder, K. P. *et al.* nNPipe: a neural network pipeline for automated analysis of morphologically diverse catalyst systems. *npj Comput. Mater.* **9**, 18 (2023).
11. Groom, D. J. *et al.* Automatic segmentation of inorganic nanoparticles in BF TEM micrographs. *Ultramicroscopy* **194**, 25-34 (2018).
12. Liu, S. *et al.* Identify the Activity Origin of Pt Single-Atom Catalyst via Atom-by-Atom Counting. *J. Am. Chem. Soc.* **143**, 15243-15249 (2021).
13. Ronneberger, O., Fischer, P. & Brox, T. in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. 234-241 (Springer).
14. Horwath, J. P., Zakharov, D. N., Mégret, R. & Stach, E. A. Understanding important features of deep learning models for segmentation of high-resolution transmission electron microscopy images. *npj Comput. Mater.* **6**, 108 (2020).
15. Mitchell, S. *et al.* Automated Image Analysis for Single-Atom Detection in Catalytic Materials by Transmission Electron Microscopy. *J. Am. Chem. Soc.* **144**, 8018-8029 (2022).
16. Eremin, D. B. *et al.* Toward Totally Defined Nanocatalysis: Deep Learning Reveals the Extraordinary Activity of Single Pd/C Particles. *J. Am. Chem. Soc.* **144**, 6071-6079 (2022).
17. Liu, S. *et al.* Dynamics of catalyst nanoparticles quantified from in situ TEM video. *Nano Today* **59**, 102505 (2024).
18. Kirillov, A. *et al.* Segment anything. *ICCV*, 3992-4003 (2023).
19. Genc, A. *et al.* A versatile machine learning workflow for high-throughput analysis of supported metal catalyst particles. *Ultramicroscopy* **271**, 114116 (2025).
20. Sytwu, K., Rangel DaCosta, L. & Scott, M. C. Generalization Across Experimental Parameters in Neural Network Analysis of High-Resolution Transmission Electron Microscopy Datasets. *Microsc. Microanal.* **30**, 85-95 (2024).
21. Cho, M. G. *et al.* Size-Resolved Shape Evolution in Inorganic Nanocrystals Captured via High-Throughput Deep Learning-Driven Statistical Characterization. *ACS Nano* **18**, 29736-29747 (2024).
22. Yang, S.-H. *et al.* Deep learning morphological distribution analysis of metal alloy catalysts in proton exchange membrane fuel cells. *Mater. Today Energy* **36**, 101348 (2023).
23. Khan, A., Lee, C. H., Huang, P. Y. & Clark, B. K. Leveraging generative adversarial networks to create realistic scanning transmission electron microscopy images. *npj Comput. Mater.* **9**, 85 (2023).
24. Aniceto-Ocaña, P. *et al.* Direct quantitative assessment of single-atom metal sites supported on powder catalysts. *Commun. Mater.* **5**, 206 (2024).

25. Eliasson, H. & Erni, R. Localization and segmentation of atomic columns in supported nanoparticles for fast scanning transmission electron microscopy. *npj Comput. Mater.* **10**, 168 (2024).
26. Faraz, K., Grenier, T., Ducottet, C. & Epicier, T. Deep learning detection of nanoparticles and multiple object tracking of their dynamic evolution during in situ ETEM studies. *Sci. Rep.* **12**, 2484 (2022).
27. Badrinarayanan, V., Kendall, A. & Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**, 2481-2495 (2017).
28. Ni, H. *et al.* Quantifying Atomically Dispersed Catalysts Using Deep Learning Assisted Microscopy. *Nano Lett.* **23**, 7442-7448 (2023).
29. Eliasson, H. *et al.* Precise Size Determination of Supported Catalyst Nanoparticles via Generative AI and Scanning Transmission Electron Microscopy. *Small Methods* **9**, 2401108 (2025).
30. Eliasson, H. & Erni, R. Improving Nanoparticle Size Estimation from Scanning Transmission Electron Micrographs with a Multislice Surrogate Model. *Nano Lett.* **25**, 2474-2479 (2025).
31. Li, K. *et al.* Single-Image-Based Deep Learning for Precise Atomic Defect Identification. *Nano Lett.* **24**, 10275-10283 (2024).
32. Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N. & Liang, J. in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings* 4. 3-11 (Springer).