Supplementary Material

COIN: Confidence Score-Guided Distillation for Annotation-Free Cell Segmentation

Sanghyun Jo[1*] Seo Jin Lee[3*] Seungwoo Lee[1] Seohyung Hong[4]
Hyungseok Seo[3†] Kyungsu Kim[2,4,5†]
{shjo.april, vict.lee0}@gmail.com {seojinleee, hong.sh, h.seo, kyskim}@snu.ac.kr
[1]OGQ, Seoul, Korea [2]School of Transdisciplinary Innovations, Seoul National University, Korea
[3]Laboratory of Cell & Gene Therapy, Institute of Pharmaceutical Sciences, College of Pharmacy, Seoul National University, Korea
[4]Department of Biomedical Science and Medical Research Center, College of Medicine, Seoul National University, Korea
[5]Interdisciplinary Programs in Artificial Intelligence, Bioengineering, and Bioinformatics, Seoul National University, Korea

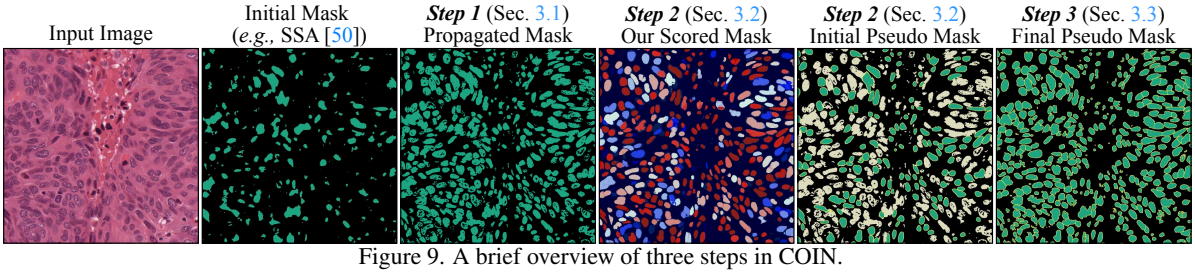

Figure 9. A brief overview of three steps in COIN.

A. Method Overview

Our approach is divided into three steps, and we assess the effect of each step in Tab. 10 on adjacent and non-adjacent cells (see Fig. 13 for more information). In *Step 1*, pixel-level cell propagation utilizing USS [16] was used to increase the sensitivity to detect all instances, resulting in a significant drop in false negative rate (FN). Specifically, FN decreases 2.3-fold, corresponding to Fig. 9 that illustrates how most cells become detected following *Step 1* (the second row). However, this was accompanied by an increase in the rate of false positives (FP), which was handled by incorporating optimal transport (OT) [48] for its ability to cluster minor pixel groups. In *Step 2*, to identify and use only error-free instances for recursive self-distillation, we introduce, for the first time, an instance-level confidence scoring approach to automatically select highly confident instances without depending on the ground truth (GT) annotations. This scoring approach measures the consistency between the baseline UCIS model [50] prediction and SAM-generated mask and selects only the instances close to GT (*i.e.,* instances with AJI scores close to 1). As shown in the table (the third row) significantly decreases FP, particularly threefold for non-adjacent cells. Here, consistency-based selection acts as implicit memory that preserves error-free masks with the dynamically adjusted threshold δ_k (Eq. (7)), preventing the accumulation of noisy labels. Additionally, we chose to decouple pseudo-label generation from training (USS once/image, SAM each epoch) to avoid end-to-end fine-tuning yet still double AJI without increasing the inference time (Tabs. 2, 4, and 9). Training slows by 50-75% compared to the baseline [50] (Tab. 10), and our modular design supports multiple UCIS (SSA [50], PSM [5]; Tabs. 2 and 4) and USS models [3, 9, 16, 44] (Tab. 12). Then, in *Step 3*, the selected instances are used for recursive self-distillation to expand the confidence, progressively increasing the number of highly confident instances each round. Notably, this last step results in a 1.9-fold improvement in AJI for adjacent cells (the fourth row), highlighting substantial advancements in our method's accuracy in cell instance segmentation, as depicted in Fig. 9.

Table 10. Performance comparison of the three steps of COIN on adjacent and non-adjacent cells on the MoNuSeg [28, 29] train set.

Method	Non-adjacent Cells				Adjacent Cells			
	AJI	IoU	FN	FP	AJI	IoU	FN	FP
SSA [50] MICCAI'20	0.201	0.547	0.254	0.199	0.176	0.546	0.267	0.188
+*Step 1* (Sec. 3.1)	0.300	0.435	0.110	0.456	0.252	0.471	0.142	0.387
+*Step 2* (Sec. 3.2)	0.341	0.615	0.235	0.149	0.211	0.529	0.325	0.146
+*Step 3* (Sec. 3.3)	**0.510**	**0.663**	**0.126**	**0.211**	**0.405**	**0.701**	**0.152**	**0.147**

Furthermore, in Tab. 11, we present a component-wise ablation study to individual modules of COIN and their

*These authors contributed equally.
†Corresponding author.

Table 11. Effect of key components in COIN on the MoNuSeg train set [28, 29] (Baseline: SSA [50], Extension of Tab. 5).

		COIN Components				Metrics			
		USS	OT	CRF	Watershed	AJI (↑)	IoU (↑)	FN (↓)	FP (↓)
(a)		✗	✗	✗	✗	0.001	0.305	0.536	0.159
		✓	✗	✗	✗	0.001	0.439	**0.020**	0.552
(b)		✓	✓	✗	✗	0.001	0.539	0.163	**0.303**
		✓	✗	✓	✗	0.001	0.443	0.022	0.549
(c)		✓	✓	✓	✗	0.001	0.543	0.157	0.301
	Ours/Step 1	✓	✓	✓	✓	**0.380**	0.543	0.157	0.301
	+USS: FN(↓)		+OT and +CRF: FP(↓)			+Watershed: AJI(↑)			

contribution to the cell segmentation performance on the MoNuSeg [28, 29] train set. Compared to the baseline [50] without CRF and watershed (the first row), USS incorporation significantly reduced FN from 0.536 to 0.020 (the second row), accompanied by an increase in FP from 0.159 to 0.552 (Tab. 11(a)). While the application of CRF [26] showed only a 0.4%p increase in IoU (the fourth row), OT alone reduced FP by 1.8 times more than did CRF alone (Tab. 11(b)). Notably, applying OT before CRF reduces the FP from 0.552 to 0.303, which is almost identical to the reduction seen when applying OT alone (from 0.552 to 0.301), suggesting that OT is the key factor in adjusting FP, while CRF has minimal impact. Lastly, as shown in Eq. (5), watershed algorithm [50] separates adjacent binary masks into distinct instances, a standard post-processing step in all UCIS baselines [5, 50]. Therefore, while IoU remained at 0.543, AJI increased from 0.001 to 0.380 (see Tab. 11(c)).

B. Method Details

B.1. Details of Unsupervised Semantic Segmentation

We are the first case to apply DINOv2 [44] and MAE [16] for analyzing pathological images (see Fig. 2). As shown in Fig. 3 and Tab. 5, the USS models [16, 44] group similar pixels (*e.g.,* cells) from UCIS seeds [50], resulting in more than 26× reduction in FN (see Tab. 11(b)). However, USS's pixel similarity-based grouping often fails to distinguish between cells and tissues of similar colors. As shown in Fig. 4, the USS output S^{us}_{θ} cannot differentiate cell activation from the background. We address this substantial increase in FP by incorporating optimal transport (OT) [48] (see Sec. 3.1, Tab. 5, and Fig. 4).

B.2. Class-level Average Pooling

Class-level average pooling (CAP) [22] is the modified version of the standard pooling technique (*i.e.,* global average pooling) in which the average of the grouped embedding vectors outputs class-specific centroids. In Sec. 3.1. the implementation of CAP to $M^{ucis}_{\theta}(I_k)$ yields class-specific USS centroids V^{us}. In our study, class denotes either cell or background.

B.3. Push Operation in Optimal Transport

The push operation T involved in Eq. (4) is the optimal-transport plan that redistributes the mass from the original pixel similarity distribution (S^{us}_{ij}) to the target distribution consisting of two distinct classes: foreground (cells) and background (tissue) [22, 33]. T_{ij} determines how much mass moves from pixel i to class j by minimizing $\sum_{i,j} T_{ij}(1 - S^{us}_{ij}) - \lambda H(T))$.

The computed T then pushes the original similarity map S^{us} to the refined mask S^{OT} by $S^{OT} = T \circ S^{us}$, sharpening pixel-wise foreground-background boundaries. We confirmed that this operation is robust to changes in λ (Fig. 11).

B.4. Watershed Algorithm

The watershed algorithm [50] is a classical image segmentation technique that is particularly effective for separating overlapping objects. Specifically, the image, treated like a topographic map, is turned into a grayscale that allows pixels to have distinctive values (0 to 255) with high intensity indicating peaks and low intensity denoting valleys. Imagine pouring water over this topographic map, where the valleys are flooded first and eventually merge as the water rises. Each valley contains different labels, and to prevent the labels from merging, the barriers are built at locations where water merges. This process continues until the peaks are all submerged underwater. Here, the barriers indicate the segmentation result. In previous work [50], the instance is obtained in the post-processing step which uses the inverse of the distance transform and the local maxima as markers (*i.e.,* labels) for the watershed algorithm (see Sec. 3.4 and Fig. 2 in [50]). Inspired by this, we utilize the watershed algorithm to obtain an initial instance mask $E^{i}_{\theta_1}(I_k)$ for N instances before training the edge decoder in Eq. (5).

B.5. Details of SAM Consistency

As shown in Fig. 16, we hypothesize and confirm that SAM [25] faithfully reconstructs an instance's shape only when the input prompt (*i.e.*, model-predicted mask) aligns closely with the ground truth, but when the prompt is noisy or incorrect, SAM often overgeneralizes and activates most of the surrounding pixels (Fig. 16; SAM Failure Cases). Thus, our method does not rely solely on SAM because the application of SAM to out-of-distribution data (*e.g.,* cell segmentation) itself introduces uncertainty. For example, when SAM randomly targets the background pixel, many pixels become overgeneralized as foreground, jeopardizing the segmentation performance (top right side of Fig. 16). Therefore, a high IoU between the input prompt (model-predicted mask) and SAM's output reliably flags error-free instances, and these top-scoring masks achieve AJI values nearly identical to those using ground-truth labels (Fig. 8). Specifi-

cally, COIN outputs low scores when either the UCIS baseline or SAM fails (right) and high scores when both succeed (left). Without our scoring approach, SAM would frequently assign multiple pixels in the background as cell instances, preventing the detection of individual cells. The high IoU scores corresponding to success cases for both predictions suggest that our instance-level confidence scoring method can automatically select highly confident instances for training without relying on ground truth annotations. Therefore, unlike the naïve application of SAM (the second row of Tab. 8), we observe a substantial performance improvement when our scoring method is applied (the third row of Tab. 8).

B.6. Canny Algorithm

In contrast to standard edge detection applications that process RGB images, we simply extract edges from binary masks (see Eq. (10)). Therefore, we utilize the traditional and well-known Canny algorithm [2]. Processing a 1000×1000 binary mask with this algorithm requires approximately seven milliseconds.

B.7. Details of Pseudo Masks and Edge Decoder

In *Step 3* (Sec. 3.3), two pseudo masks are generated based on the accepted indices $\mathcal{A}_\delta$ from Eq. (8). As depicted in Fig. 10, pseudo binary mask $\hat{M}^i_{bin}(t)$ from Eq. (9) refers to the pixels designated as foreground (*i.e.,* cell), which corresponds to high-scoring instances within the scored instances. Therefore, the low-scoring instances are omitted and not used for training. The pseudo edge mask $\hat{M}^i_{edge}(t)$ from Eq. (10) denotes the cell boundaries. The pseudo binary and edge masks in Fig. 10 are the decomposed representation of the pseudo mask at $t = 1$ from Fig. 6.

Unlike existing UCIS models [5, 50], our framework incorporates an edge decoder to train on pseudo edge masks. Inspired by recent studies [33, 34, 45], the edge decoder learns the boundaries between neighboring instances to address the challenge of distinguishing adjacent cells. Specifically, DeepSnake [45] trains on the loss from iterative contour deformation (refer to Eq. (4) at [45]), which iteratively deforms the initial contour to approach the actual object boundary, and Point2Mask [33] learns high-level boundary map by utilizing the mask affinity equivalence among the eight neighbor pixels (refer to Eq. (7) at [33]). PolyTransform [34] trains on the losses from the feature extraction network and deforming network for learning strong object boundaries and predicting the offset for each vertex, respectively (refer to Sec. 3.4 from [34]). Thus, including an edge decoder allows our approach to learn discriminative instance features during training, leading to significant improvements in segmentation accuracy (see Tab. 14).

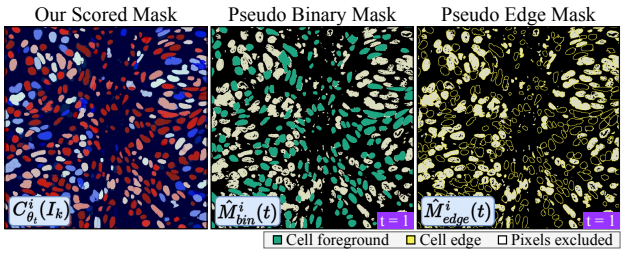

Figure 10. **Illustration of binary and edge pseudo masks.** Green represents the cell foreground, and yellow lines denote the cell edges. White indicate pixels excluded from training. Note that only high-scoring instances (red) are used to generate pseudo masks.

B.8. Datasets

Main experiments (Tab. 2) are conducted on MoNuSeg [28, 29] and TNBC [42] datasets. MoNuSeg contains multi-organ nuclei segmentation images that are H&E-stained and captured at 40x magnification. Specifically, it includes a total of 21,623 annotated nuclear boundaries. TNBC (Triple Negative Breast Cancer) dataset is generated at the Curie Institute and consists of 50 images with 4,022 annotated cells. BRCA [1] contains breast cancer H&E-stained images. CPM-17 [56] and PanNuke [12] are derived from multiple types of tissues, consisting of 205,343 and 7,750 annotated nuclei, respectively. CryoNuSeg [37] contains fully annotated H&E-stained nuclei instance segmentation images derived from frozen tissue samples (FS) of 10 human organs.

C. Additional Quantitative Results

C.1. OT Hyperparameters

The experiment on OT parameters, as shown in Fig. 11, demonstrates that the IoU values remain stable across varying λ values. Specifically, when λ is increased from 0.01 to 0.4, the IoU fluctuates only slightly, with the highest IoU observed at $\lambda = 0.1$ (0.543) and the lowest at $\lambda = 0.01$ (0.532). The difference between the maximum and minimum IoU is just 0.011, indicating that the model's performance is not significantly influenced by the parameter value.

C.2. Effect of Adaptive Thresholding

We train for 100 epochs as in all experiments (Tab. 9) and observe that on the MoNuSeg [28, 29] test set, IoU steadily improves and plateaus around 40% of training (2.4 hours; Fig. 12, pink box), regardless of threshold type. Notably, our non-parametric, adaptive threshold (green) consistently outperforms fixed parametric variants by about 3%p in IoU.

To further understand why δ_k adapts so effectively, we plot the standard deviation of consistency scores across

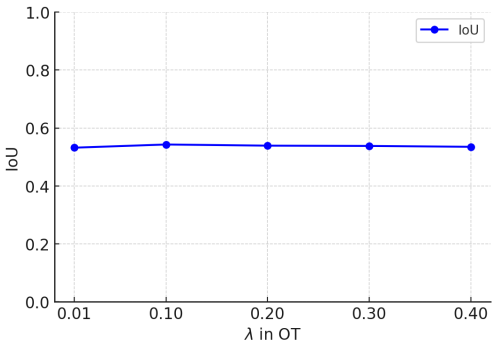

Figure 11. **Performance analysis with varying OT parameters.**

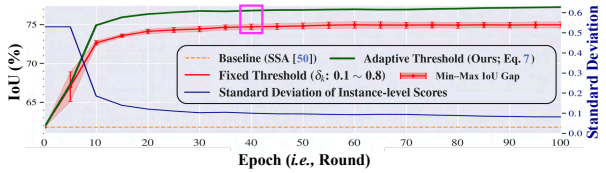

Figure 12. **Performance comparison between fixed and adaptive threshold.**

training epochs (Fig. 12, blue line). Since δ_k is characterized by dividing its standard deviation by the mean, and a high standard deviation implies a large variation in predicted instance scores (*i.e.,* greater uncertainty), we focus on how this uncertainty evolves during training. As shown in Fig. 12, the standard deviation is high early in the training process, indicating that model predictions vary significantly due to noisy or uncertain instances. However, as training progresses, the standard deviation steadily decreases, reflecting a stabilization of model predictions. This aligns with the notion of performance saturation (pink box). The adaptive nature of δ_k enables it to respond to these changes by filtering out early noise and adjusting as confidence solidifies. In contrast, the fixed thresholds fail to remove noisy instances effectively in the early stages of self-distillation, resulting in suboptimal performance compared to our adaptive threshold.

C.3. Model-agnostic Improvements with Various USS Backbones

We extensively evaluate our method by experimenting on different USS backbones [3, 9, 16, 44] in all metrics on TNBC [42] test set. In Tab. 12, we compare the performance of Masked Autoencoder (MAE) [16], which is leveraged for all other experiments, against DINOv1 [3], DINOv2 [44], and DINOv2-reg [9], which demonstrates that MAE outperforms all USS backbones across all metrics. Specifically, for instance segmentation performance, MAE outperforms DINOv1 by at least +2%p, DINOv2 by at least +1%p, and DINOv2-reg by at least +0.5%p. MAE also surpasses other USS backbones regarding semantic segmentation, with at least +3.3%p for DINOv1, +2.6%p for DINOv2, and +1.7%p for DINOv2-reg. Therefore, we select MAE as the USS backbone for all experiments.

Table 12. Comparison of four USS backbones [3, 9, 16, 44] on the TNBC [42] test set.

Backbone	Instance Segmentation		Semantic Segmentation	
	AJI	PQ	IoU	Dice
DINOv1 [3]	0.534	0.519	0.764	0.721
DINOv2 [44]	0.558	0.528	0.771	0.733
DINOv2-reg [9]	0.563	0.533	0.780	0.754
MAE [16]	**0.568**	**0.540**	**0.797**	**0.774**

Table 13. Performance evaluation of COIN on adjacent and non-adjacent cells on the MoNuSeg [28, 29] test set.

Method	Non-adjacent Cells		Adjacent Cells	
	AJI (↑)	IoU (↑)	AJI (↑)	IoU (↑)
SSA [50] MICCAI'20	0.288	0.583	0.235	0.632
SSA + COIN (Ours)	**0.602**	**0.729**	**0.528**	**0.750**
Δ_{ssa}	+0.314	+0.146	+0.293	+0.118
PSM [5] MICCAI'23	0.498	0.695	0.408	0.660
PSM + COIN (Ours)	**0.601**	**0.725**	**0.527**	**0.748**
Δ_{psm}	+0.103	+0.030	+0.119	+0.088

Δ_{ssa}: Performance gap between SSA [50] and our proposed method.
Δ_{psm}: Performance gap between PSM [5] and our proposed method.

Table 14. Effect of the edge decoder on the MoNuSeg [28, 29] test set.

Edge Decoder	Non-adjacent Cells		Adjacent Cells	
	AJI (↑)	IoU (↑)	AJI (↑)	IoU (↑)
✗	0.594	0.712	0.493	0.738
✓	**0.602**	**0.729**	**0.528**	**0.750**
Δ_{edge}	+0.008	+0.017	+0.035	+0.012

Δ_{edge}: Performance enhancement made by training the edge decoder.

C.4. Performance on Adjacent Cells

To validate our method's performance in distinguishing adjacent cells, we specifically categorize non-adjacent cells and adjacent cells in the ground truth image of MoNuSeg [28, 29], as depicted in Fig. 13. Following the dilation of ground truth cell edges and connected component labeling (CCL) [49], we identify cells that are connected to two or more cells as adjacent cells. Tab. 13 demonstrates that our approach consistently enhances the performances of existing UCIS models across both adjacent and non-adjacent cell types. For non-adjacent cells, our model achieves +31.4%p in AJI and +14.6%p in IoU with SSA and +10.3%p in AJI and +3%p in IoU with PSM. Notably, a similar pattern of performance improvement occurs with adjacent cells, with +29.3%p in AJI and +11.8%p in IoU with SSA and +11.9%p in AJI and +8.8%p in IoU with PSM. These results highlight COIN's ability to accurately separate instances, validating performance improvements in cell instance segmentation.

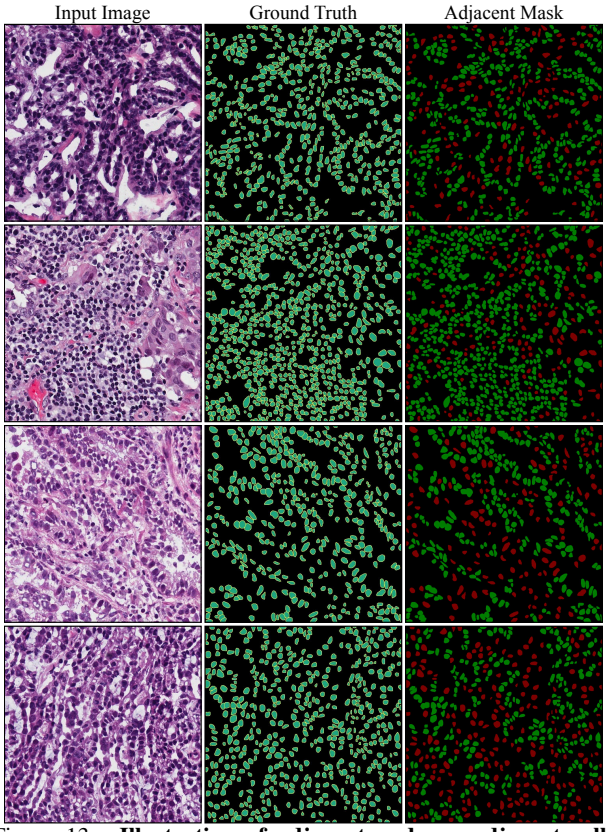

Figure 13. **Illustration of adjacent and non-adjacent cells.** Green represents the adjacent cells that are connected to at least two other cells, and red indicates non-adjacent cells that are not connected to any other cells.

C.5. Effect of Edge Decoder

In Tab. 14, we assess the impact of incorporating an edge decoder on the MoNuSeg [28, 29] test set. The application of our edge decoder enhances segmentation performance for both adjacent and non-adjacent cells. Notably, for adjacent cells, the performance improves by $+1.7$%p in AJI and $+3.5$%p in IoU when the edge decoder is present, demonstrating its capability to effectively distinguish cell boundaries and enhance overall segmentation results.

D. Additional Qualitative Results

D.1. Comparison of UIS Methods and Ours

We compare the qualitative performance of our method against UIS baselines [32, 59] in Fig. 15. As demonstrated in Tab. 2, our proposed method substantially outperforms all UIS models [3, 16, 44] that have low AJI scores.

D.2. Examples of SAM-based Instance-level Confidence Scoring

Previous methods [39, 63] that leverage SAM [25] rely on outputs generated by SAM from manual annotations (*e.g.,* points) to create pseudo labels. Their dependency on such annotations indicates that the annotation burden is persistent. In contrast, our work utilizes SAM for confidence measurement and confident instance selection without requiring SAM-based image-related manual annotations (see Sec. 3.2). The proposed scoring process is completely unsupervised and automatic, and it is the first-ever case to leverage SAM for confidence score-related tasks. Refer to Fig. 14 for example visualizations of SAM-based scoring. Notably, our scoring approach separates adjacent cells effectively even when the pseudo mask doesn't distinguish individual cells (the fourth row).

D.3. Limitations of Recursive Self-distillation

When we tracked IoU across each self-distillation iteration in Fig. 12, we noticed that a small number of noisy pseudo-labels persist in rare cases when the initial USS propagation fails (*e.g.,* transparent cells). This particular case is a persistent challenge faced by prior UCIS approaches [5, 50] including ours, but nonetheless, these cases are rare and represent only a small fraction of our datasets, exerting limited influence on the overall accuracy. We demonstrate example failure cases in Fig. 17.

D.4. Additional State-of-the-art Qualitative Results

In Figs. 18 and 19, we provide additional qualitative comparisons between our COIN method, two image-related annotation-driven models [11, 63], and one image-related annotation-free model [50]. As confirmed by the improvements across all metrics in Tab. 2, these visual examples further highlight that the output of our method is not only comparable but often surpasses the performance of supervised models that depend on image-related annotations. It is noteworthy given that COIN achieves high-quality segmentation without depending on such labor-intensive and time-consuming annotations.

D.5. Model-agnostic Improvements with Various UCIS Models

Fig. 20 illustrates the model-agnostic performance improvement by COIN on two different UCIS baselines [5, 50]. Our framework notably improves the segmentation performance for both SSA [50] and PSM [5], demonstrating its model-agnostic nature. Specifically, COIN significantly improves SSA's missed and chunky predictions and PSM's incomplete edges, demonstrating the flexibility of our method.

D.6. Consistent Improvements on Multiple Datasets

In Figs. 21, 22, 23, 24, and 25, we further validate the scalability of our method by comparing qualitative improvements against the UCIS baseline (*e.g.,* SSA [50]) on multiple datasets, including BRCA [1], CPM-17 [56], CryoNuSeg [37], and PanNuke [12]. As demonstrated in Tab. 3,

our model combined with SSA substantially improves semantic and instance segmentation performances throughout multiple datasets.

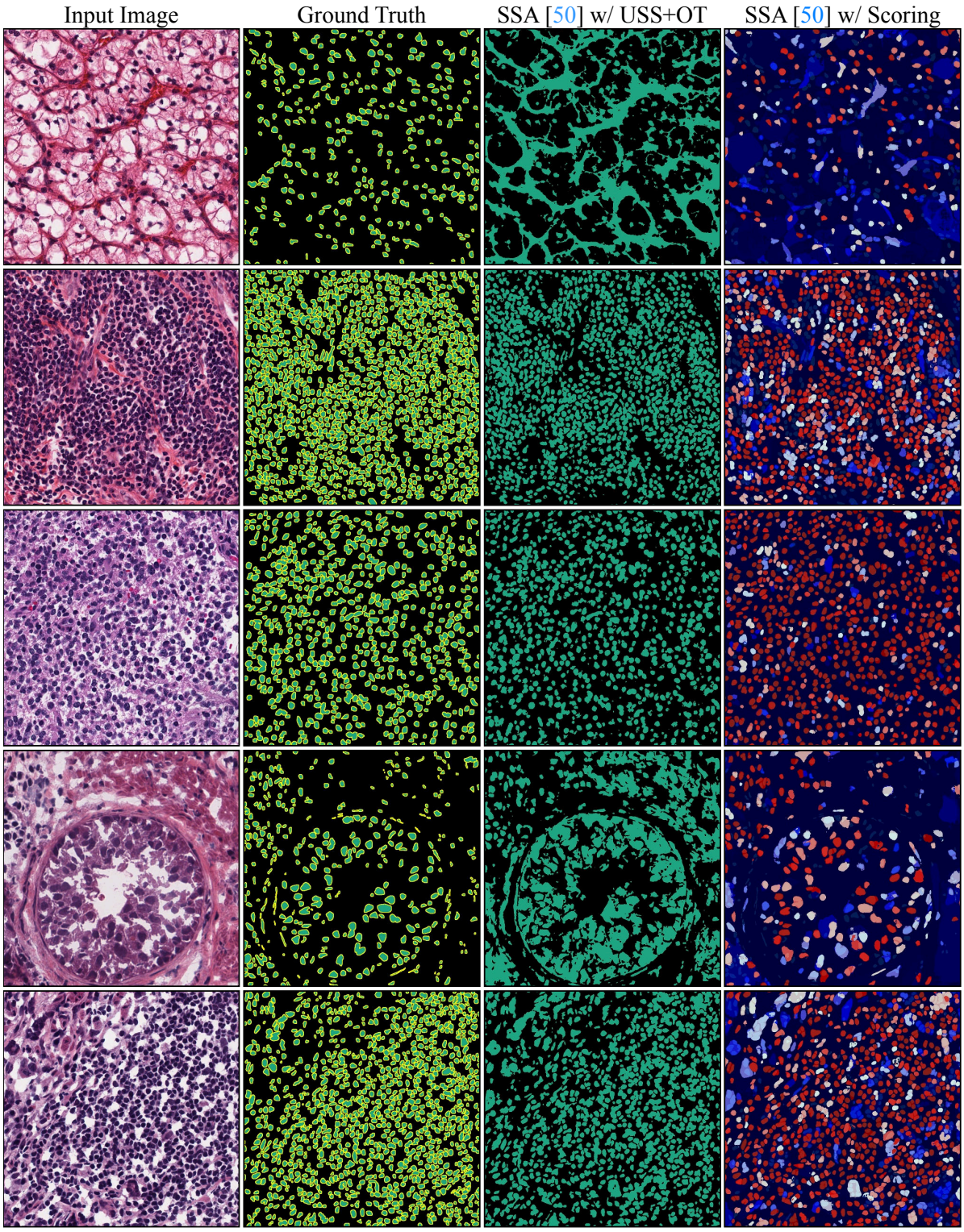

Figure 14. Qualitative examples of our instance-level confidence scoring based on SAM [25].

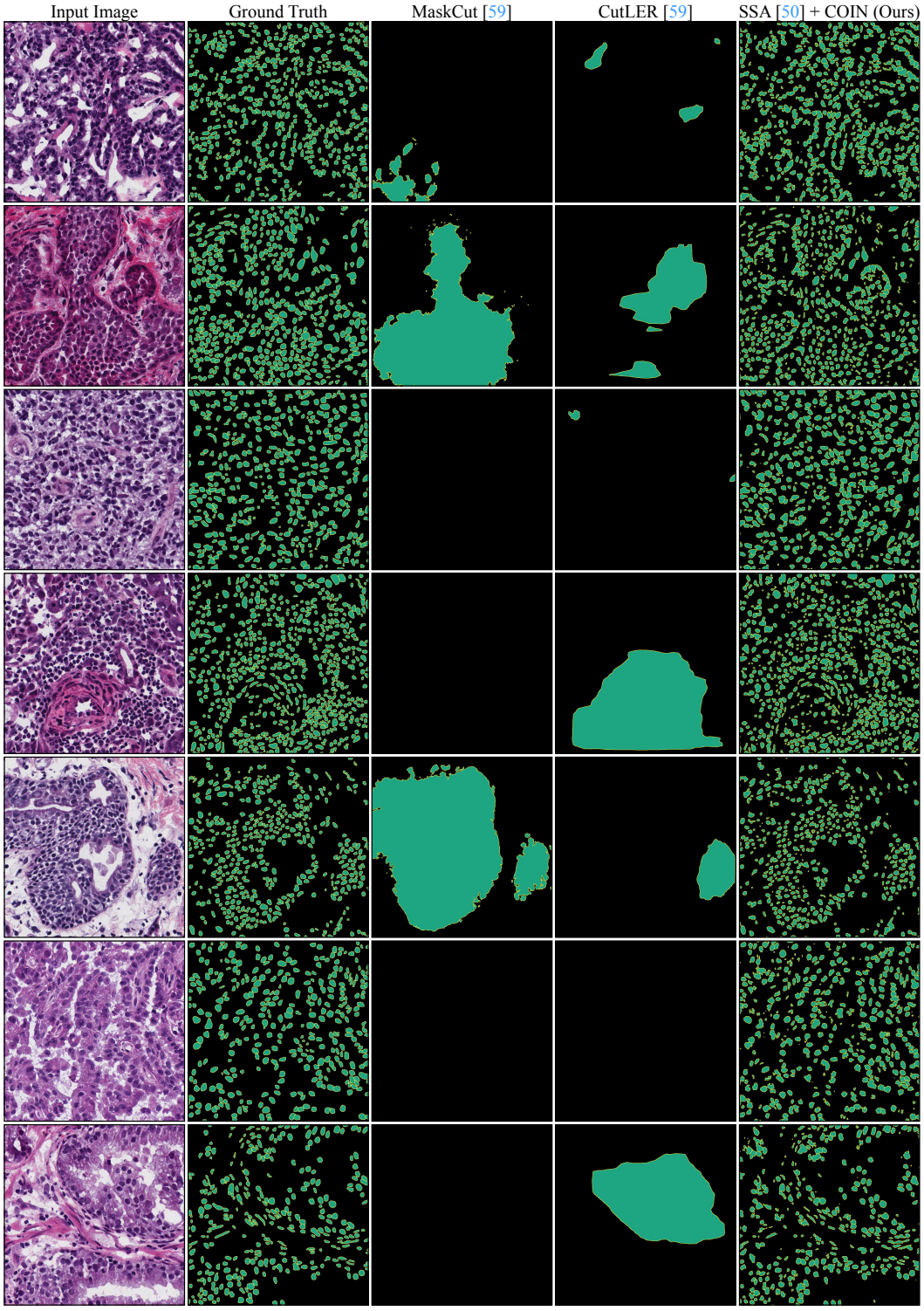

Figure 15. Qualitative comparison of UIS [59] and COIN combined with SSA [50] on the MoNuSeg [28, 29] test set.

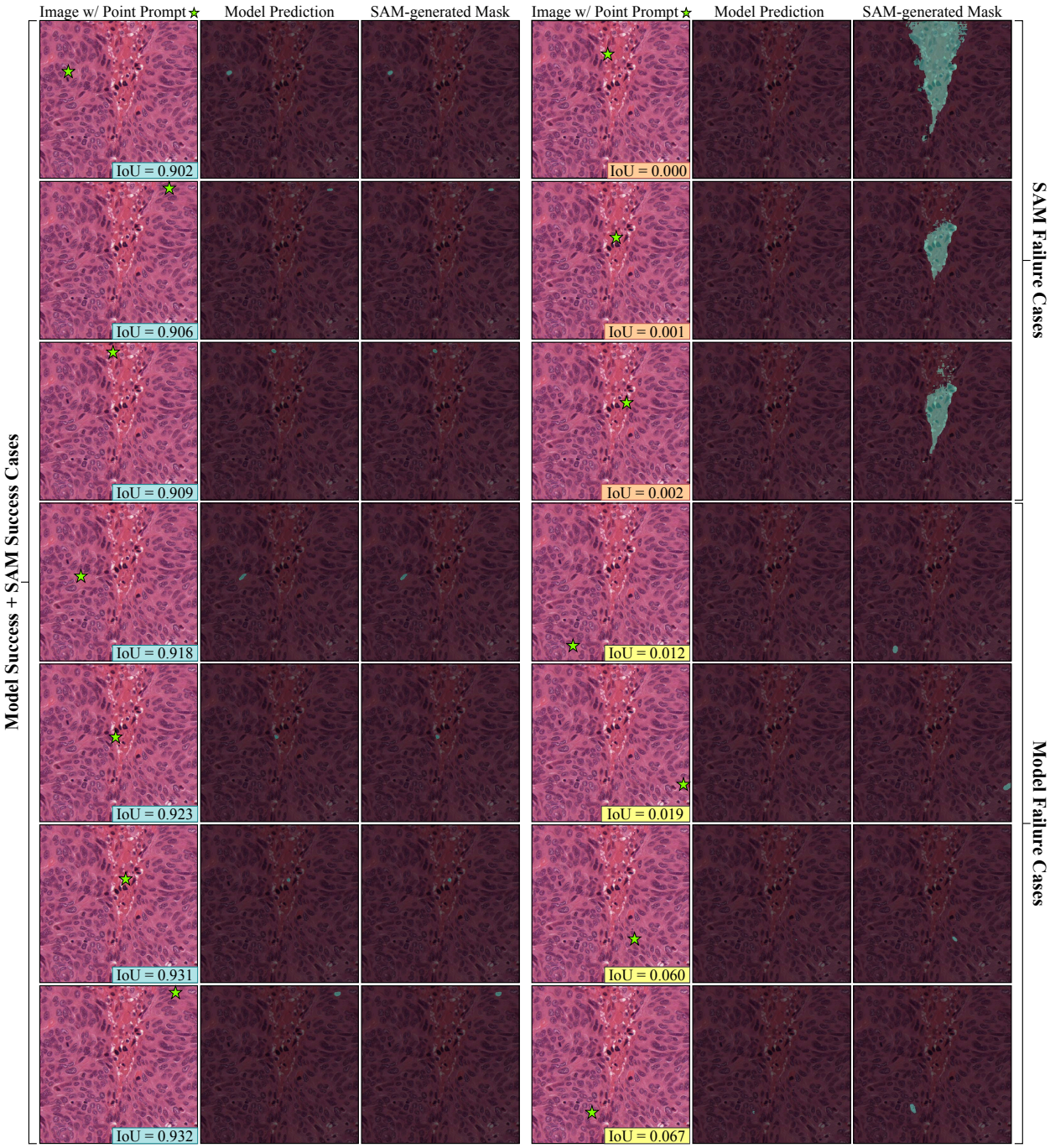

Figure 16. Visualization of success and failure cases for our propagated masks and their corresponding SAM-refined masks [25].

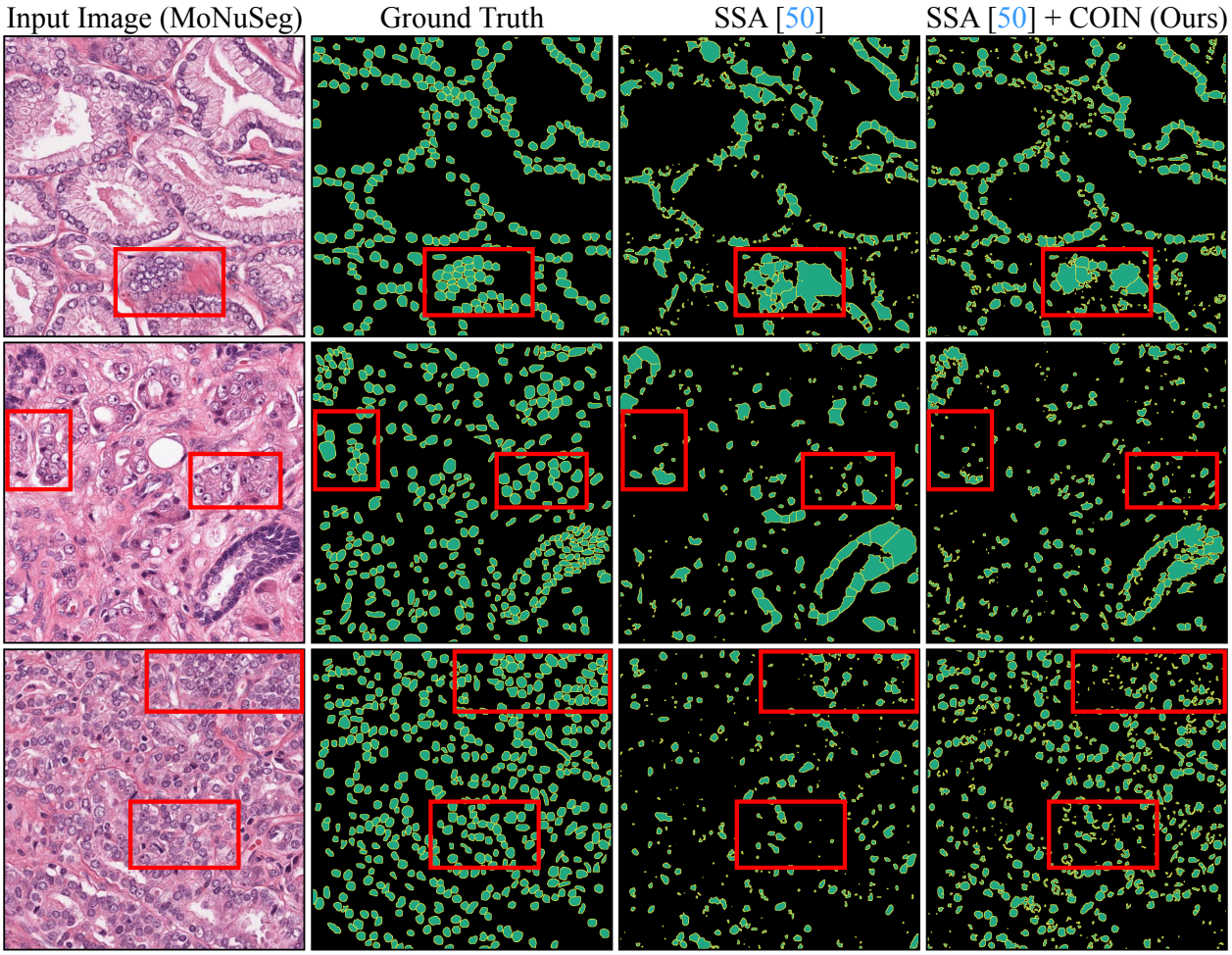

Figure 17. Visualization of failure cases for recursive self-distillation on the MoNuSeg [28, 29] train set.

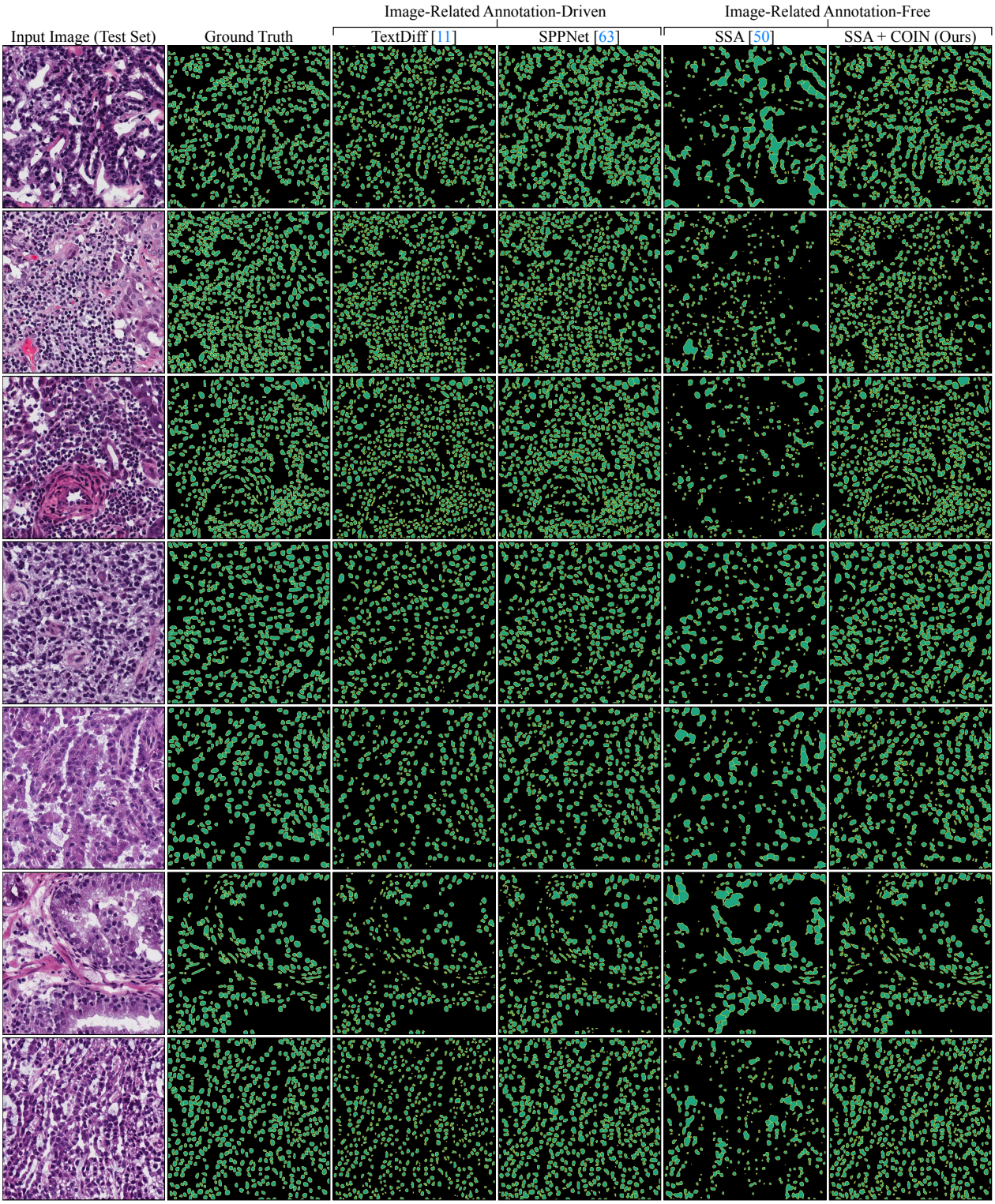

Figure 18. Qualitative comparison of annotation-driven and -free methods [11, 50, 63] on the MoNuSeg [28, 29] test set.

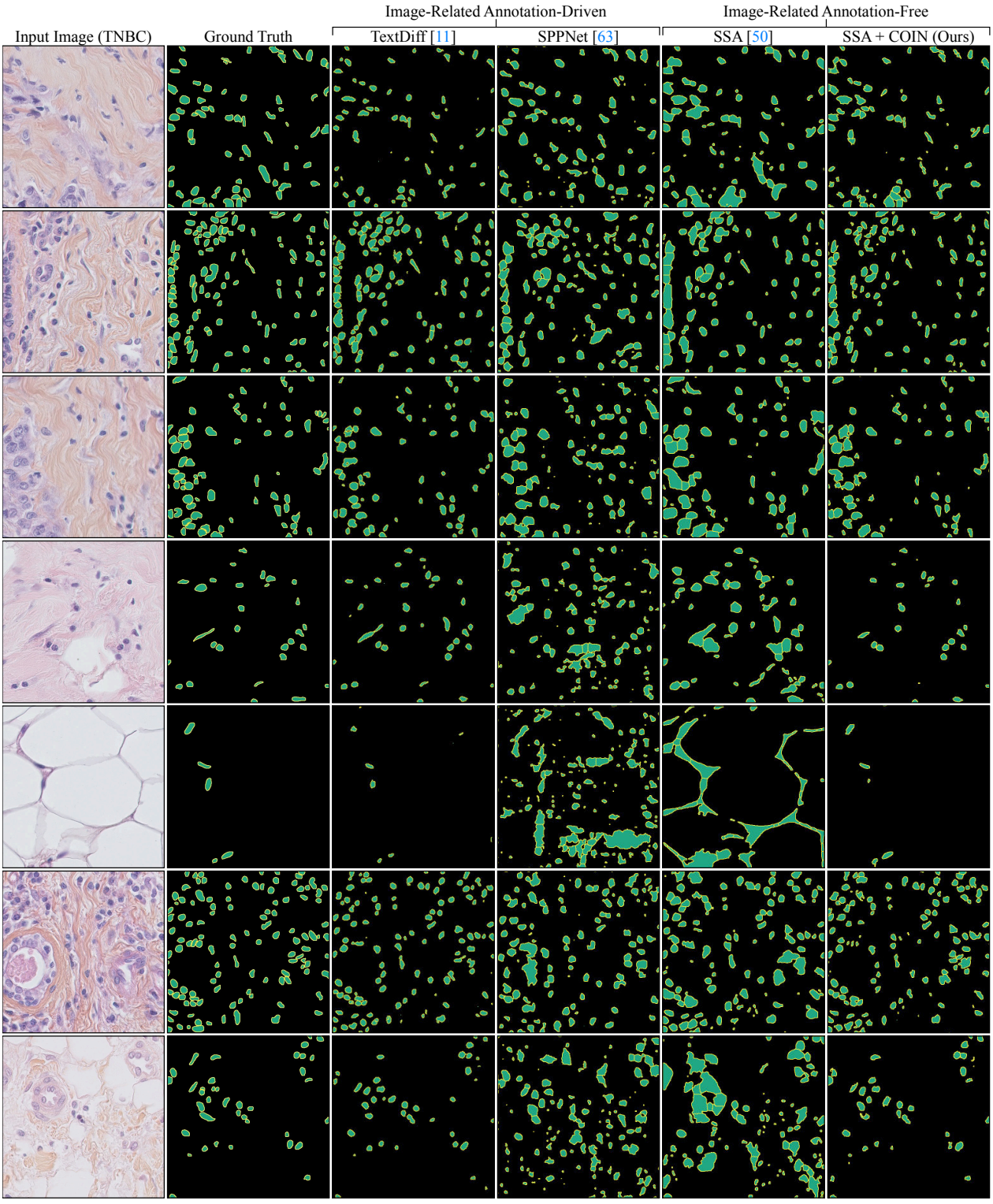

Figure 19. Qualitative comparison of annotation-driven and -free methods [11, 50, 63] on the TNBC [42] test set.

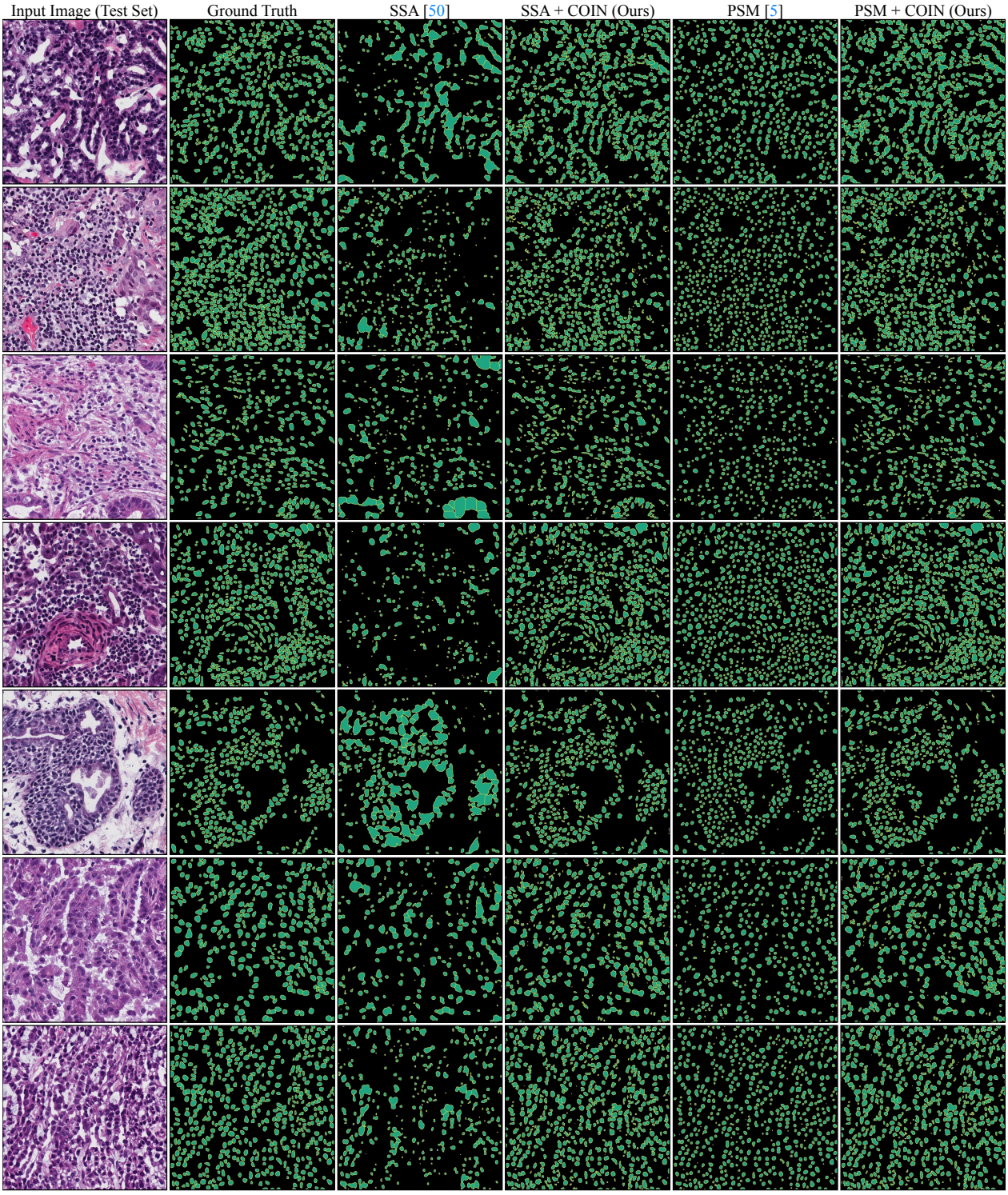

Figure 20. Model-agnostic qualitative comparison of two UCIS models [5, 50] on the MoNuSeg [28, 29] test set.

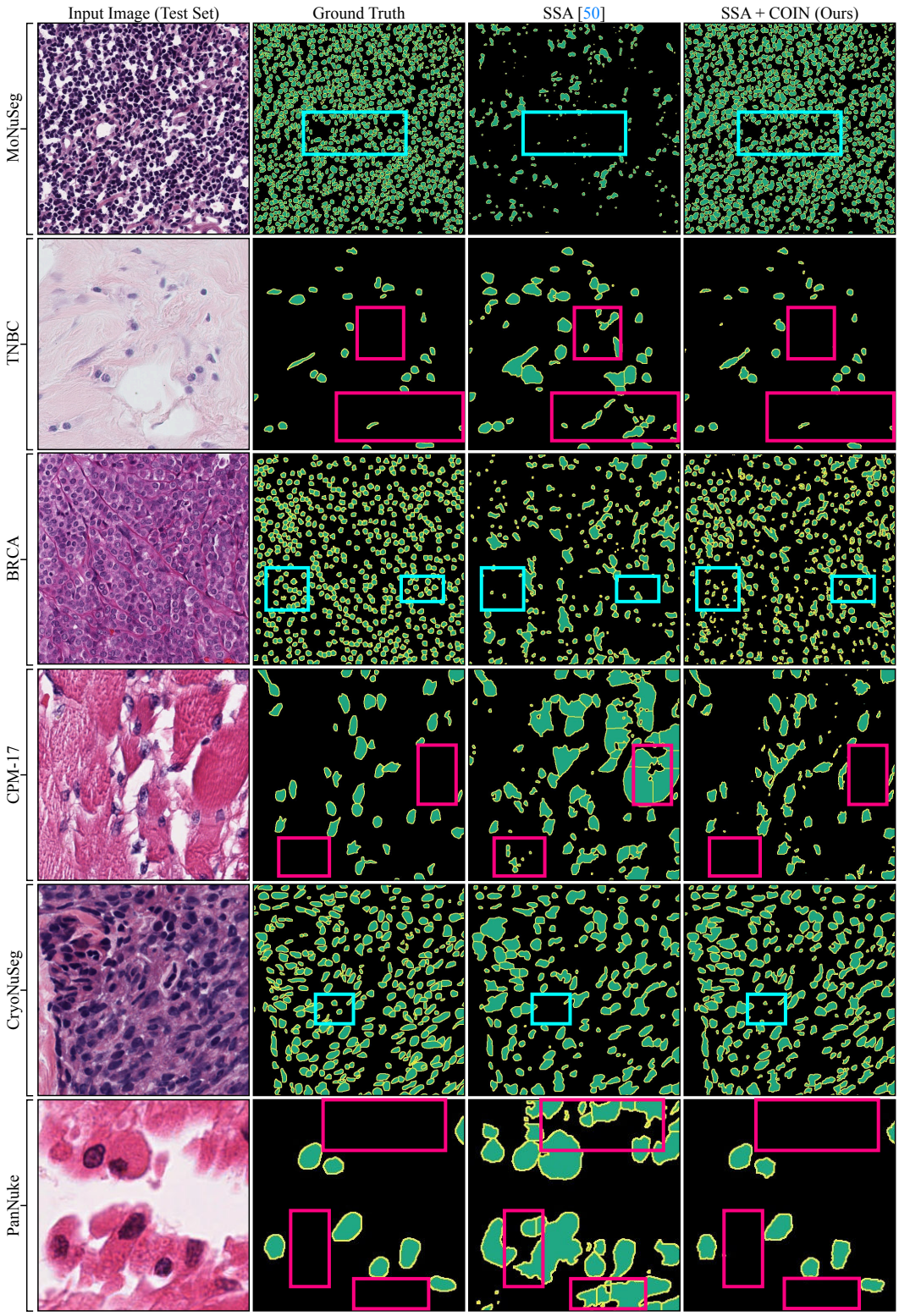

Figure 21. **Visualization of qualitative improvements on five benchmarks [1, 12, 37, 42, 56]**. Regions marked with pink boxes represent false positives identified by the baseline [50], while cyan boxes indicate false negatives.

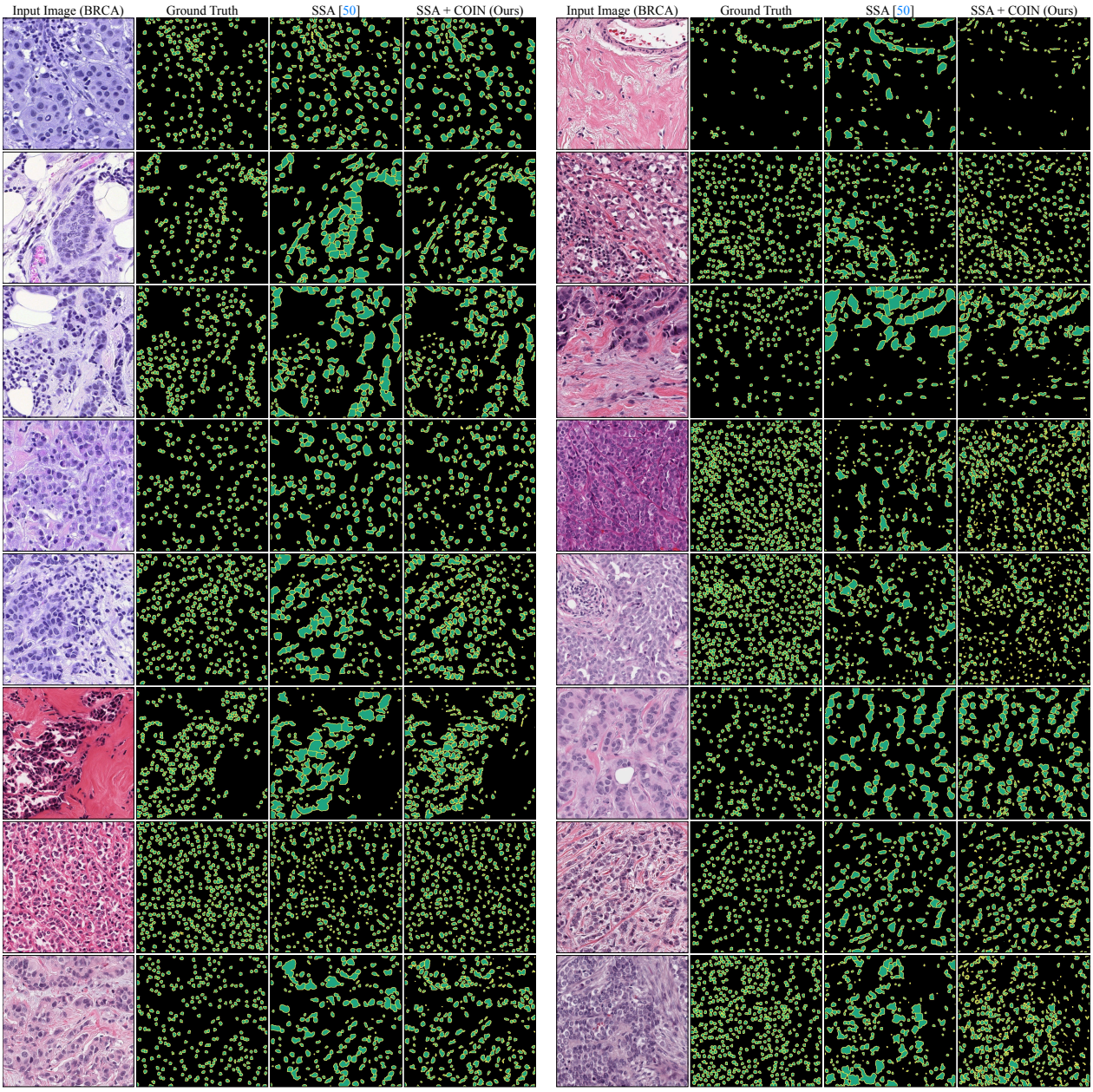

Figure 22. Qualitative examples on the BRCA [1] test set.

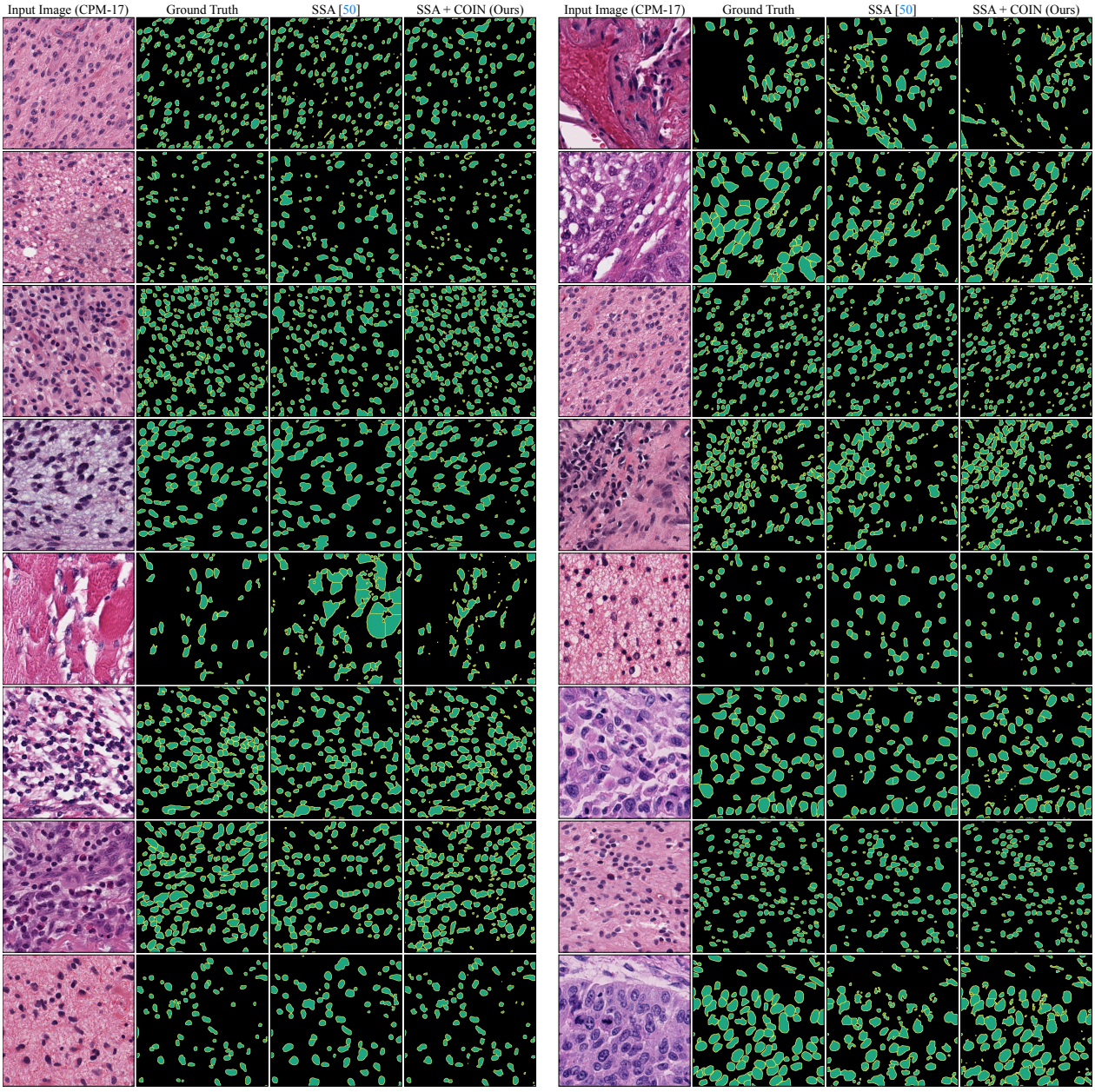

Figure 23. Qualitative examples on the CPM-17 [56] test set.

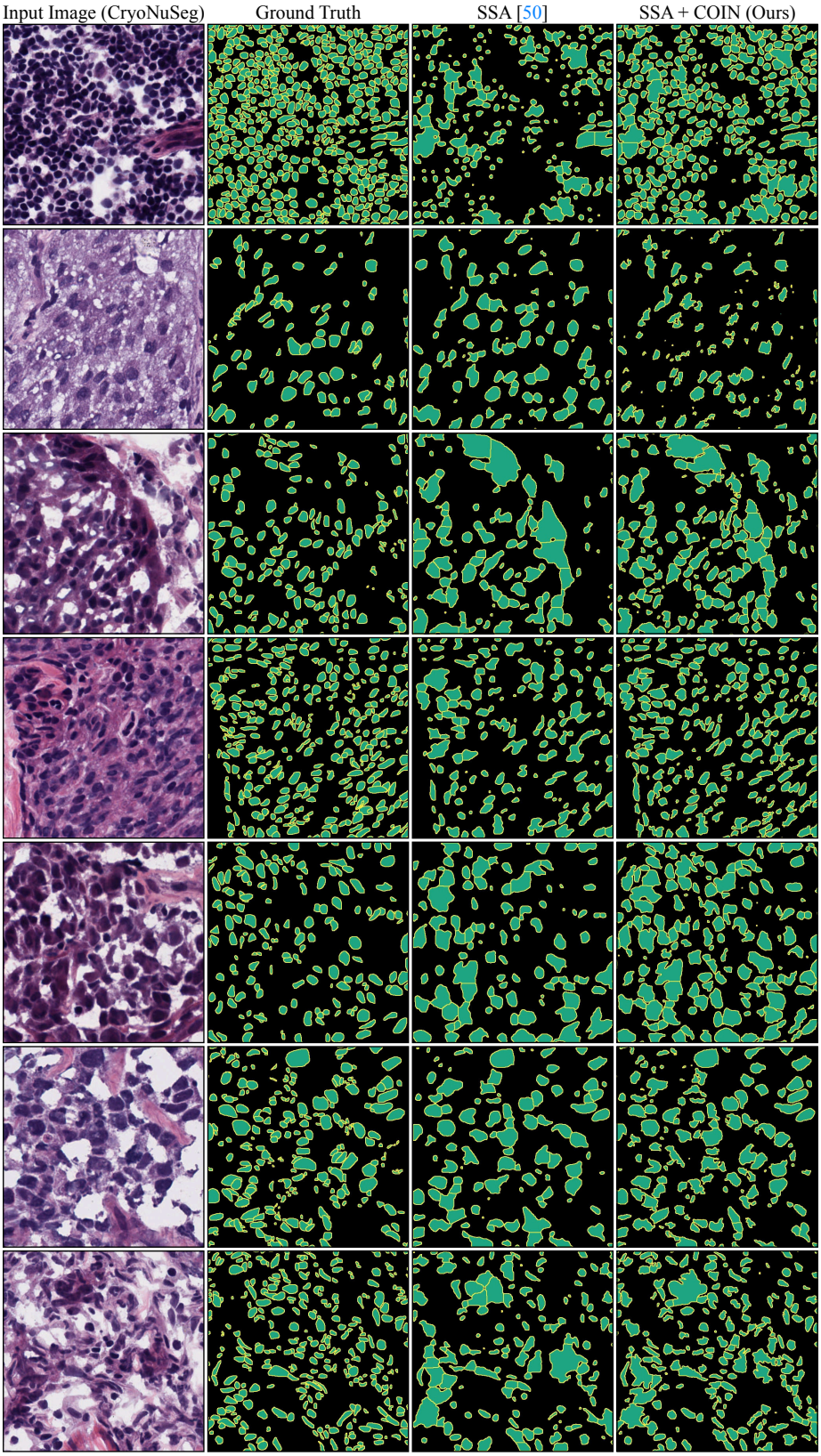

Figure 24. Qualitative examples on the CryoNuSeg [37] test set.

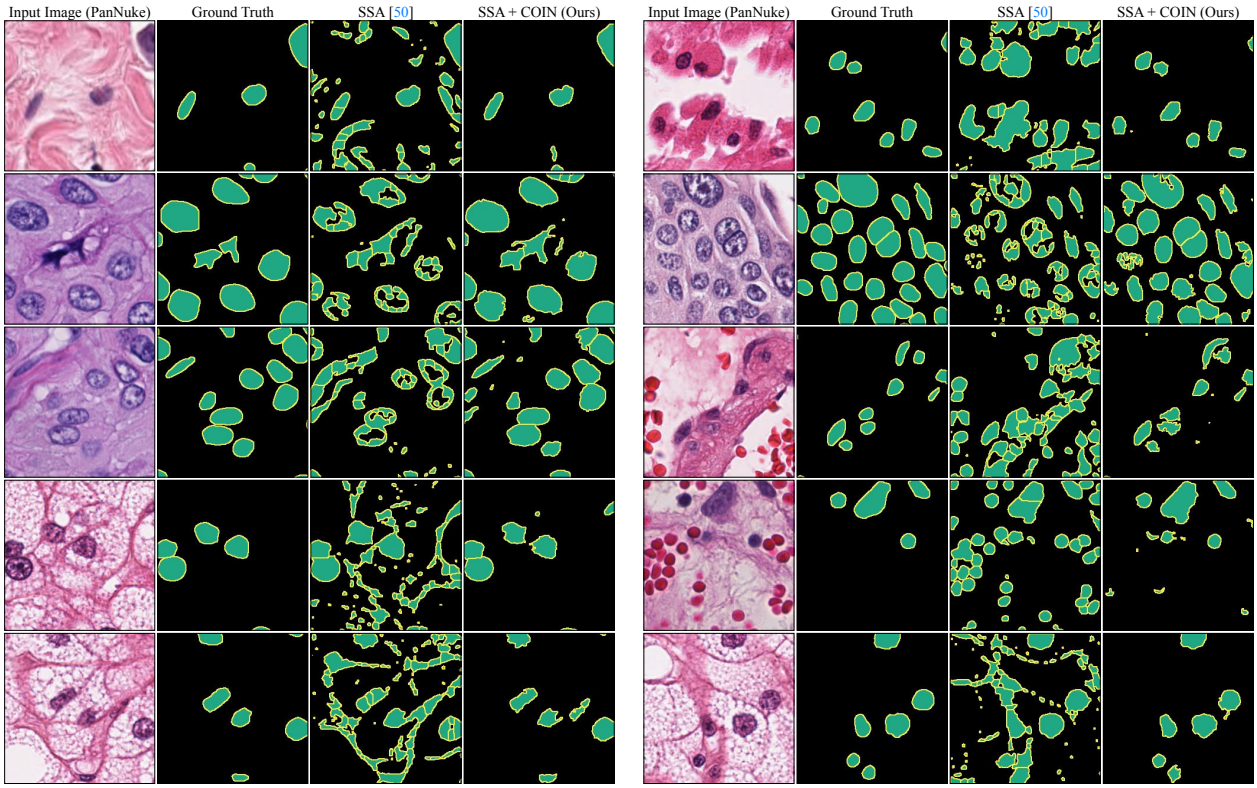

Figure 25. Qualitative examples on the PanNuke [12] test set.