Is PEFT Enough for Cell Segmentation? An Empirical No-Go Result on Frozen Foundation Models

Joonsuk Hwang [1] **Hyowon Woo** [1] **Hyobin Seok** [1] **Jangwon Seo** [2] **Jeong Hoon Lee** [1 3]

Abstract

In cell microscopy, downstream analysis increasingly requires precise cell instance segmentation rather than a high average segmentation score. Using Cellpose-SAM on LIVECell as a controlled case study, we examine whether various adaptation strategies—backbone adapters, decoder-side LoRA and partial unfreezing, reweighting losses, and cell-separation losses—together with full fine-tuning can further improve performance. Across all of them, performance stays within a narrow band around $\mathrm{AP}_{\mathrm{COCO}} \approx 0.56$: the best setting (a flow-repulsion separation loss) improves over the base model but still leaves missed and merged cells in dense regions, and even full fine-tuning ($\mathrm{AP}_{\mathrm{COCO}} = 0.5627$) does not escape this band. These results suggest that the observed ceiling is difficult to explain solely by PEFT capacity or the number of trainable parameters. Instead, they point to a representational limitation of existing cell-domain models, motivating structure-aware cell foundation models trained to encode microscopy-specific structural properties—dense cells, weak boundaries, touching instances, and flow-based separation—from the pretraining stage.

1. Introduction

Cell segmentation is one of the most fundamental steps in biological microscopy image analysis. Segmentation results are used as inputs to downstream biological analyses such as cell count, morphology, density, cell-cell interaction, and drug response analysis (Caicedo et al., 2017; Stringer et al., 2021). Therefore, errors in cell segmentation do not merely degrade mask quality, but directly affect the reliability of subsequent biological interpretation (Stringer et al., 2021; Edlund et al., 2021).

In particular, in cell microscopy, distinguishing foreground from background is not sufficient. Because the actual analysis is performed at the level of individual cell instances, errors in which two adjacent cells are merged into a single instance, or one cell is split into multiple instances, become important problems. Such instance-level errors may not be prominent under average pixel accuracy or at low IoU thresholds, but can lead to critical errors in morphology-based analysis or high-density cell analysis (Caicedo et al., 2017; Meijering, 2020).

In image segmentation, general-purpose foundation models such as SAM (Kirillov et al., 2023), together with broader vision-language foundation models (Xiao et al., 2025), have shown strong generalization across diverse image domains. Building on this progress, cell-domain foundation models such as CellSAM (Israel et al., 2025) and Cellpose-SAM (Pachitariu et al., 2025), trained for the cell domain, have emerged and become strong off-the-shelf tools for cell segmentation. However, applying such a cell-domain model to a new cell microscopy setting still requires additional adaptation, and because full fine-tuning is computationally costly and prone to overfitting in annotation-scarce settings, PEFT (parameter-efficient fine-tuning)—which freezes the backbone and trains only a small subset of parameters such as adapters or LoRA—is widely used as a realistic alternative (Houlsby et al., 2019; Hu et al., 2022; Wu et al., 2025; Chen et al., 2023). Figure 1 summarizes this landscape and the gap that our study highlights.

In this study, we use Cellpose-SAM on LIVECell (Edlund et al., 2021) as a case study to examine whether various parameter-efficient and loss-based adaptation methods, as well as full fine-tuning, can further improve cell instance segmentation performance.

The contributions of this study can be summarized as follows.

- Using Cellpose-SAM on LIVECell, we systematically

[1]KU-KIST Graduate School of Converging Science and Technology, Korea University, Seoul, Republic of Korea [2]Department of Mechanical Engineering, Korea University, Seoul, Republic of Korea [3]Integrative Energy Engineering, Korea University, Seoul, Republic of Korea. Correspondence to: Jeong Hoon Lee <hello-jhlee@korea.ac.kr>.

Proceedings of the ICML 2026 3rd Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences, Seoul, Korea. 2026. Copyright 2026 by the author(s).

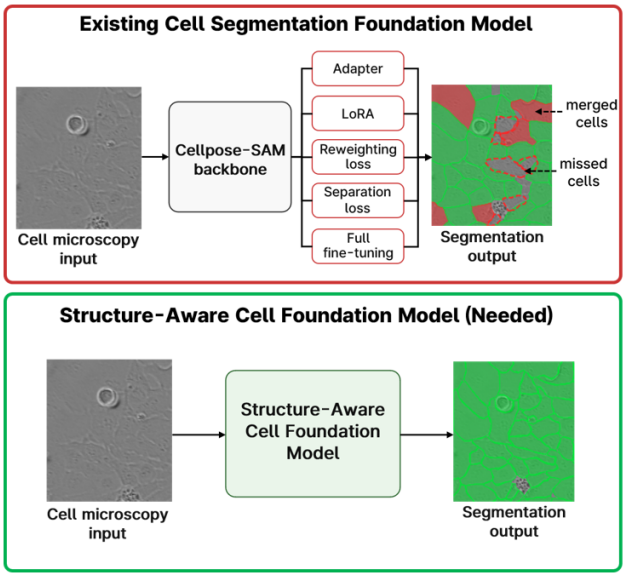

Figure 1. Overview of our argument. (Top) Existing approaches adapt a cell-domain foundation model (Cellpose-SAM) via adapters, LoRA, separation losses, or full fine-tuning, yet still produce merged and missed cells in dense regions. (Bottom) This motivates a structure-aware cell foundation model that encodes microscopy-specific structural properties—dense cells, weak boundaries, touching instances, and flow-based separation—from the pretraining stage.

compare diverse adaptation strategies—backbone adapters, decoder-side LoRA, partial unfreezing, reweighting losses, and cell-separation losses—alongside full fine-tuning under a single protocol.

- We show that these strategies, including full fine-tuning, provide only marginal and similar gains, while missed and merged cells remain in dense regions.
- These results suggest that the limitation is not simply about adaptation capacity; precise instance separation instead calls for a structure-aware cell foundation model that encodes these microscopy-specific structural properties—dense cells, weak boundaries, touching instances, and flow-based separation—from the pretraining stage.

2. Method

This study evaluates the cell instance segmentation performance of Cellpose-SAM on the LIVECell dataset (Edlund et al., 2021). LIVECell consists of label-free phase-contrast microscopy images and includes diverse cell types, cell densities, weak boundaries, and touching-cell conditions. Because of these characteristics, LIVECell is suitable for evaluating instance-level cell separation ability rather than simple foreground segmentation.

As the base model, we use the official Cellpose-SAM pretrained model (Pachitariu et al., 2025). Cellpose-SAM is based on SAM-family visual representations and, through Cellpose-style outputs, predicts a cell probability map and a 2D flow field. It then recovers individual cell instances using flow-based dynamics. All experiments are conducted on the same LIVECell split and the same evaluation pipeline, and we use AP_{COCO} as the primary metric.

In this study, we divide the ways of adapting Cellpose-SAM to cell microscopy into four axes for comparison.

Backbone-side adapter. We compare the Bottleneck adapter (Houlsby et al., 2019), the Med-SA-style adapter (Wu et al., 2025), and the ViT-Adapter-style module (Chen et al., 2023). The Bottleneck adapter inserts small trainable modules inside the ViT block, and by adjusting the adapter capacity we examine the effect of the number of trainable parameters on performance. The Med-SA-style adapter places a similar adapter module as a parallel branch on the MLP path, and the ViT-Adapter-style module adds a spatial prior branch outside the backbone to perform feature adaptation. Through this, we analyze whether performance changes vary greatly with adapter capacity, placement, or adapter architecture (detailed structures in Appendix A).

Decoder-side adaptation. We compare LoRA (Hu et al., 2022) and partial unfreezing. LoRA is applied by adding low-rank updates to the neck or output head, and partial unfreezing is performed by directly training part of the output head or neck. This experiment checks whether, when backbone-side adapters alone are not sufficient, additional degrees of freedom on the decoder side can lead to performance gains.

Reweighting loss. We compare losses that give a larger training signal to hard regions. Specifically, we add a Tversky loss (Salehi et al., 2017) to the probability prediction, and apply a boundary-band flow MSE and an inter-cell boundary flow MSE to the flow prediction (Ronneberger et al., 2015). The boundary-band flow MSE emphasizes the region around cell boundaries, while the inter-cell boundary flow MSE focuses on the boundaries where different cell instances touch. We also evaluate a consistency loss that matches the predictions before and after weak augmentation. This axis checks whether the missed-cell or merged-cell problems can be explained by a lack of training signal in hard boundary regions (equations in Appendix B).

Cell-separation loss. We add a direct separation signal to the flow field so that adjacent cells are separated into different instances. As a representative example, the flow repulsion loss enforces that the flow vectors of touching boundary pixel pairs belonging to different cell instances

point in opposite directions. Unlike reweighting losses that raise the importance of a particular region, this approach explicitly strengthens instance separation itself at the cell-cell boundary. We additionally compare a dynamics separation loss that guides different cell pairs toward different centroid basins after flow dynamics (equations in Appendix B).

In addition, to check whether the performance limitation stems from adapter capacity, we also perform full fine-tuning. All training is conducted on a single NVIDIA GeForce RTX 4090 GPU, and full fine-tuning uses a smaller batch size than the PEFT settings due to memory constraints. In the main text, we report only representative settings among all experiments to summarize the main trends, and detailed ablations are presented in Appendix D (training protocol in Appendix C).

3. Results

3.1. Best Settings Across Adaptation Axes

In this study, we collected the representative setting that achieved the highest performance in each adaptation axis and directly compared the best performance range of each method group observed in our experiments.

Table 1 summarizes Base Cellpose-SAM and the best representative setting of each adaptation axis. Base Cellpose-SAM recorded $AP_{COCO} = 0.5447$. For backbone-side adapters, BN-r1, the largest-capacity setting of the Bottleneck adapter, was the highest at $AP_{COCO} = 0.5600$. In the adapter type and placement comparison, the BN-r4 serial setting was the highest at $AP_{COCO} = 0.5592$. For decoder-side adaptation, BN-r4 + LoRA(neck) was the highest at $AP_{COCO} = 0.5599$. For reweighting losses, the BN-r4 + Tversky + boundary setting recorded $AP_{COCO} = 0.5582$, which was in fact lower than the BN-r4 baseline.

Overall, most settings—including adapter capacity, adapter placement, decoder-side LoRA, partial unfreezing, and reweighting losses—remained around AP_{COCO} of approximately 0.558–0.562. Even when full fine-tuning is included, the best performance stayed at $AP_{COCO} = 0.5627$, and the overall results were still within a narrow performance range. That is, even when selecting only the best setting in each axis for comparison, the performance differences were small, and neither the adapter structure, the loss design, nor full fine-tuning produced a clear difference in performance.

3.2. Cell-Separation Loss Improves Performance but Remains Limited

Among the PEFT and loss-based adaptation axes, the highest performance was achieved by the flow repulsion setting, which corresponds to the cell-separation loss. BN-r2 + LoRA(out) + flow repulsion recorded $AP_{COCO} = 0.5620$,

Table 1. Best representative setting per adaptation axis on LIVECell test.

Method *(axis)*	AP_{COCO}	AP@0.9
Base Cellpose-SAM	0.5447	0.1187
+ BN-r1 *(capacity)*	0.5600	0.1454
+ BN-r4 serial *(placement)*	0.5592	0.1458
+ BN-r4, LoRA(neck) *(decoder)*	0.5599	0.1456
+ BN-r4, tv+b *(reweighting)*	0.5582	0.1451
+ BN-r2, LoRA(out), rep *(separation)*	0.5620	**0.1516**
Full fine-tuning	**0.5627**	0.1499

BN: bottleneck adapter (r: reduction rank); tv: Tversky; b: boundary-band flow MSE; rep: flow repulsion.

an improvement of $+1.73$ pp over Base Cellpose-SAM, and its AP@0.9 was also the highest at 0.1516. This suggests that explicitly separating the flow directions of different instances at touching cell boundaries can provide a useful training signal for cell instance segmentation.

However, this improvement is also difficult to regard as clearly breaking through the performance ceiling. Although flow repulsion showed the highest performance among the PEFT and cell-separation settings, its difference from the other representative PEFT settings was very small in terms of AP_{COCO}. In the qualitative results (Figure 2), the missed-cell problem of failing to detect cells in dense cell regions remained, and the merged-cell problem in which adjacent cells are merged into a single instance was also consistently observed.

Therefore, although flow repulsion was the most effective additional signal in the current pipeline, simply adding a separation loss on top of an existing cell-domain foundation model is insufficient to fundamentally solve the dense cell instance separation problem.

3.3. Full Fine-Tuning Shows Similar Performance

The full fine-tuning result, which trains all model parameters, recorded $AP_{COCO} = 0.5627$. This is higher than the $AP_{COCO} = 0.5447$ of Base Cellpose-SAM, but it does not greatly deviate from the 0.558–0.562 performance range reached by the various PEFT and cell-separation settings.

This result suggests that the observed performance limitation is difficult to explain solely by a particular adapter structure or an insufficient number of trainable parameters, and that the existing model may not sufficiently capture fine-grained cell microscopy structures such as the dense cell boundaries, weak contrast, and touching-instance separation of LIVECell.

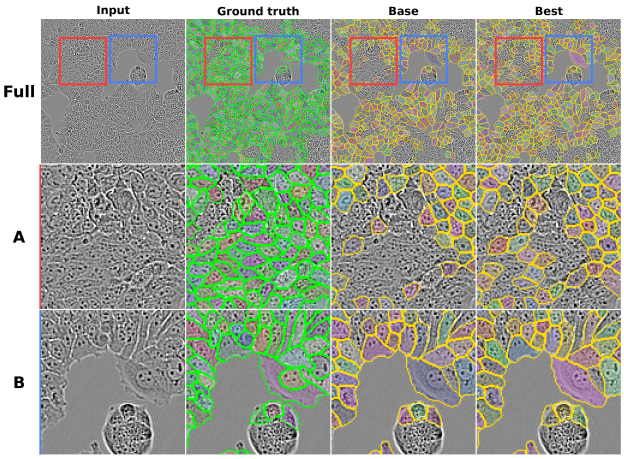

Figure 2. Failure modes that remain even after the best PEFT adaptation. On MCF7 cells from the LIVECell test set, we compare Input, Ground truth, Base, and Best (BN-r2 + LoRA(out) + rep λ_{rep}=0.5). *Row 1* shows the whole region; *Row 2 (A)* and *Row 3 (B)* are 2× zoom-ins of the red/blue boxes, showing missed cells in the dense region (A) and merged cells (B).

4. Conclusion

Using Cellpose-SAM on LIVECell as a controlled case study, this study analyzed whether various adaptation methods can further improve cell instance segmentation performance. To this end, we compared backbone-side adapters, adapter capacity, adapter placement, decoder-side LoRA and partial unfreezing, reweighting losses, and cell-separation losses, and also analyzed the full fine-tuning result.

Even when selecting the best setting in each method group for comparison, most results remained within a narrow performance range in terms of AP_{COCO}. Backbone adapters and decoder-side adaptation stayed around 0.558–0.562, and reweighting losses did not exceed the representative PEFT baseline. Among the PEFT and loss-based adaptations, the flow repulsion loss showed the highest performance at $\text{AP}_{\text{COCO}} = 0.5620$, but the missed-cell and merged-cell problems in dense cell regions still remained. Full fine-tuning also remained within a similar performance range compared to all the adaptation settings.

The fact that this ceiling recurs across all adaptation axes—rather than being tied to one PEFT or loss design—indicates that it is difficult to explain solely by PEFT capacity or the number of trainable parameters, and that adding various adaptations on top of an existing model is not sufficient to surpass it. This suggests that the limitation may lie in whether the model representation sufficiently encodes the structural properties of cell microscopy. We therefore argue that breaking this ceiling requires a structure-aware cell foundation model that encodes these microscopy-specific structural properties—dense cells, weak boundaries, touching instances, and flow-based separation—from the pretraining stage.

This study has several limitations. All experiments were conducted on a single dataset (LIVECell), and the backbone was also limited to Cellpose-SAM. Therefore, we cannot conclude that these results represent all microscopy datasets or that the same trends appear for other biomedical vision backbones. Nevertheless, since LIVECell includes diverse cell morphology, density variation, weak boundaries, and touching-cell conditions, it is a benchmark well suited for evaluating the important difficulties in cell instance segmentation. Thus, our results are meaningful as a controlled case study showing the limitations that existing cell-domain foundation models may face in dense instance separation.

5. Impact Statement

This study shows that, when applying foundation models in biological microscopy image analysis, careful examination is needed for the approach of expecting sufficient performance from simple parameter-efficient adaptation alone. Because cell segmentation forms the basis of diverse downstream biological analyses such as cell count, morphology, density, cell-cell interaction, and drug response analysis, instance-level errors such as missed cells or merged cells can directly affect the reliability of actual analysis results.

This study emphasizes the need for a structure-aware cell foundation model, together with a pretraining objective and an evaluation protocol that reflect the structural properties of cell microscopy. In particular, we suggest that not only the average segmentation score but also metrics such as high-IoU instance separation, dense-region failure, and cell-cell boundary preservation should be considered together. Furthermore, by clearly showing the limitations of current foundation model adaptation, we aim to promote the development of more reliable structure-aware cell foundation models.

References

Caicedo, J. C., Cooper, S., Heigwer, F., Warchal, S., Qiu, P., Molnar, C., Vasilevich, A. S., Barry, J. D., Bansal, S., Kraus, O., et al. Data-analysis strategies for image-based cell profiling. *Nature Methods*, 14(9):849–863, 2017.

Chen, Z., Duan, Y., Wang, W., et al. Vision transformer adapter for dense predictions. In *International Conference on Learning Representations*, 2023.

Edlund, C., Jackson, T. R., Khalid, N., et al. Livecell—a large-scale dataset for label-free live cell segmentation. *Nature Methods*, 18(9):1038–1045, 2021.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrill, B.,

Schultz, Q., Sharma, H., Sharma, A., et al. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Hu, E. J., Shen, Y., Wallis, P., et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Israel, U., Marks, M., Dilip, R., et al. Cellsam: a foundation model for cell segmentation. *Nature Methods*, 22(12): 2585–2593, 2025.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Meijering, E. A bird's-eye view of deep learning in bioimage analysis. *Computational and Structural Biotechnology Journal*, 18:2312–2325, 2020.

Pachitariu, M., Rariden, M., and Stringer, C. Cellposesam: Superhuman generalization for cellular segmentation. *bioRxiv*, 2025.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241. Springer, 2015.

Salehi, S. S. M., Erdogmus, D., and Gholipour, A. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *International Workshop on Machine Learning in Medical Imaging*, pp. 379–387. Springer, 2017.

Stringer, C., Wang, T., Michaelos, M., and Pachitariu, M. Cellpose: a generalist algorithm for cellular segmentation. *Nature Methods*, 18(1):100–106, 2021.

Wu, J., Wang, Z., Hong, M., et al. Medical sam adapter: Adapting segment anything model for medical image segmentation. *Medical Image Analysis*, 102:103547, 2025.

Xiao, X., Zhang, Y., Nguyen, T.-H., Lam, B.-T., Wang, J., Zhao, L., Hamm, J., Wang, T., Li, X., Wang, X., Xu, H., Liu, T., and Xu, M. Describe anything in medical images. In *ICML 2025 Workshop on Foundation Models for Life Sciences (FM4LS)*, 2025. arXiv:2505.05804.

A. Adapter Module Details

This study compares three ways of using trainable modules inside/outside the ViT block, none of which touch the weights of the frozen ViT block.

Bottleneck (BN) adapter. It inserts two bottleneck modules per ViT block—one after the attention output and one after the MLP output. The inner dimension is $r' = \max(d/r_{\text{reduction}}, 16)$, and to control capacity we sweep the reduction rank $r_{\text{reduction}} \in \{1, 2, 4, 8\}$.

Med-SA-style adapter. The internal module structure is identical to BN, but the module on the MLP path is placed as a parallel branch. Compared to BN, the module structure is the same and only the MLP-path placement differs.

ViT-Adapter. Unlike BN/Med-SA which use modules inside the ViT block, it does not touch the forward of the ViT block and places a separate trainable branch outside the ViT. This branch consists of a spatial prior module (SPM) and four interaction blocks, each containing an injector that uses deformable cross-attention. The setting follows the original paper default.

Decoder LoRA / partial unfreezing. Since the neck and output head of Cellpose-SAM are simpler modules than the backbone and are not suitable for backbone-style adapters, we use LoRA and partial unfreezing. Decoder LoRA learns only low-rank updates for the neck/output head, and depending on the applied range we compare (a) output head only, (b) neck only, and (c) both, with $r_L = 4$. Partial unfreezing keeps the backbone frozen and unfreezes only the decoder-side modules.

B. Loss Function Formulations

The original Cellpose training objective is

$$L_{\text{cp}} = \tfrac{1}{2}\,\|\hat{f} - s\,f^*\|_2^2 + \text{BCE}(\hat{p}, p^*), \tag{1}$$

and all terms below are added as separate terms without modifying L_{cp}.

Tversky loss. To make the learning of the probability channel lean more toward recall, a Tversky term is added alongside BCE:

$$L_{\text{tv}} = 1 - \frac{TP + \varepsilon}{TP + \alpha\,FP + \beta\,FN + \varepsilon}, \tag{2}$$

where $TP = \sum \hat{q}g,\ FP = \sum \hat{q}(1-g),\ FN = \sum(1-\hat{q})g$, and we set $\alpha = 0.3, \beta = 0.7$ to increase the FN penalty.

Boundary-band masked flow MSE. We add an additional flow-MSE weighting over the region M_∂ within $w = 3$ pixels of all boundaries of the GT label:

$$L_{\text{b}} = \frac{1}{|M_\partial|} \sum_{(i,j)\in M_\partial} \tfrac{1}{2}\,\|\hat{f}_{ij} - s\,f^*_{ij}\|_2^2. \tag{3}$$

The **inter-cell band masked flow MSE** (L_{ic}) has the same form but narrows the mask M_{ic} to cell-cell touching boundaries only.

Consistency loss. The MSE between the original prediction $\hat{p}_o$ and the prediction $\hat{p}_d$ for a weakly augmented version of the same input:

$$L_{\text{cons}} = \text{MSE}(\hat{p}_o,\, \hat{p}_d). \tag{4}$$

The total reweighting objective is

$$L = L_{\text{cp}} + \lambda_{\text{tv}} L_{\text{tv}} + \lambda_{\text{b}} L_{\text{b}} + \lambda_{\text{ic}} L_{\text{ic}} + \lambda_{\text{cons}} L_{\text{cons}}, \tag{5}$$

which we ablate by varying the weights.

Flow repulsion (L_{rep}). Cellpose flow is trained so that pixels inside a cell point toward their own center, but the per-point MSE of L_{cp} does not directly prevent the flow from being averaged to near zero at touching boundaries. To address this, we define a set $\mathcal{P}$ of touching pairs—among 4-connected neighbor pairs, those whose two pixels belong to different instances—and enforce that the two unit flow vectors point in opposite directions:

$$L_{\text{rep}} = \frac{1}{|\mathcal{P}_m|} \sum_{(p,q)\in\mathcal{P}_m} \big(1 + \cos(\hat{f}_p, \hat{f}_q)\big), \quad \mathcal{P}_m = \{(p,q) \in \mathcal{P} \mid \cos(\hat{f}_p, \hat{f}_q) > m\}. \tag{6}$$

We sweep the weight $\lambda_{\text{rep}} \in \{0.3, 0.4, 0.5, 1.0\}$ with margin $m = -0.3$.

Dynamics separation (L_{dyn}). It enforces that endpoints obtained by moving $K = 4$ steps along the flow reach the centroid basins of different cells. With $z^{(k+1)} = z^{(k)} + \eta\, \hat{f}(z^{(k)})/s$ to obtain the endpoint $z^{(K)}$, letting c_i be the GT centroid of the cell that pixel i belongs to and $d_{ij} = |z_i^{(K)} - c_j|$,

$$\begin{aligned} L_{\text{dyn}}^{(p,q)} = \lambda_{\text{att}} \sum_{i\in\{p,q\}} \text{SmoothL1}(z_i^{(K)}, c_i) + \lambda_{\text{sep}} \Big[\text{ReLU}(\delta - \|z_p^{(K)} - z_q^{(K)}\|)\Big]^2 \\ + \sum_{i\in\{p,q\}, j\neq i} \text{ReLU}(d_{ii} - d_{ij} + \delta_c). \end{aligned} \tag{7}$$

The three terms act, respectively, as a weak attractor toward each cell's own centroid, a separation term between the two endpoints, and a cross-instance margin.

C. Training and Evaluation Protocol

Each run is trained on LIVECell train, and the checkpoint of the epoch with the highest val AP_{COCO} is selected and evaluated once on the test set. The backbone uses the official Cellpose-SAM ViT-L weights. The optimizer is AdamW (weight decay 0.1) for all runs, the seed is fixed to 42, and training is done on a single RTX 4090 (24GB) with batch size 8–12 depending on memory. Evaluation uses AP, the standard metric for instance segmentation; the primary metric is AP_{COCO} (0.5–0.95, image-level mean), with AP@0.5/0.75/0.9 reported as auxiliary.

Training uses 30 epochs (capacity sweeps and some long-train ablations use 50 or 100 epochs), 0–3 warmup epochs, and either a cosine or the Cellpose default scheduler, with a learning rate of 1×10^{-5} or 1×10^{-4}.

D. Full Ablation Tables

Table A1. Bottleneck adapter rank sweep ($\eta = 1 \times 10^{-5}$, 50 epochs).

Setting	Trainable (M)	AP	AP@.5	AP@.75	AP@.9
BN-r1	100.76	**0.5600**	**0.8416**	**0.6382**	0.1454
BN-r2	50.41	0.5596	0.8410	0.6377	**0.1459**
BN-r4	25.23	0.5592	0.8405	0.6370	0.1458
BN-r8	12.64	0.5585	0.8398	0.6365	0.1447

BN: bottleneck adapter; r: reduction rank ($r' = \max(d/r, 16)$).

Table A2. Adapter type and placement comparison ($r = 4$).

Setting	Trainable (M)	AP	AP@.5	AP@.75	AP@.9
BN-r4 (Serial)	25.23	0.5592	0.8405	0.6370	0.1458
Med-SA-r4 (Parallel)	25.23	0.5591	0.8404	0.6370	0.1454
ViT-Adapter-r4	19.03	0.5574	0.8398	0.6350	0.1423

Serial: BN modules inserted in sequence after the attention/MLP output. Parallel (Med-SA): the MLP-path module placed as a parallel branch. ViT-Adapter: an external spatial-prior branch with deformable cross-attention.

Table A3. Decoder-side LoRA and partial unfreezing (all on the BN-r4 backbone).

Setting	Trainable (M)	AP	AP@.5	AP@.75	AP@.9
BN-r4 + LoRA(neck)	25.24	**0.5599**	0.8418	**0.6378**	**0.1456**
BN-r4 + LoRA(out)	25.23	0.5589	**0.8419**	0.6362	0.1435
BN-r4 + LoRA(neck+out)	25.24	0.5587	0.8417	0.6358	0.1434
BN-r4 + Unfreeze(out)	25.28	0.5587	0.8417	0.6362	0.1432
BN-r4 + Unfreeze(out+neck)	26.13	0.5577	0.8415	0.6353	0.1405

LoRA(neck/out/neck+out): LoRA ($r_L = 4$) on the corresponding module. Unfreeze(out/out+neck): the corresponding module is directly unfrozen.

Table A4. Generic loss reweighting results.

Setting	λ_{tv}	λ_{b}	λ_{ic}	λ_{cons}	AP	AP@.5	AP@.9
BN-r4	–	–	–	–	0.5592	0.8405	0.1458
BN-r4 + tv + b	1.0	1.0	–	–	0.5582	0.8407	0.1451
BN-r4 + tv + b (h)	1.0	2.0	–	–	0.5578	0.8403	0.1442
BN-r4 + tv + b + cons	1.0	2.0	–	1.0	0.5447	0.8363	0.1187
BN-r4 + Unfreeze(out)+b+ic	–	0.1	0.3	–	0.5552	0.8398	0.1397

tv: Tversky. b: boundary-band flow MSE. ic: inter-cell band flow MSE. cons: consistency loss. (h): boundary weight λ_{b} strengthened from 1.0 to 2.0.

Table A5. Flow repulsion weight sweep.

Setting	λ_{rep}	m	AP	AP@.5	AP@.75	AP@.9
Base	–	–	0.5447	0.8358	0.6190	0.1187
BN-r2 + LoRA(out) + rep	0.3	-0.3	0.5614	0.8434	0.6385	0.1503
BN-r2 + LoRA(out) + rep	0.4	-0.3	0.5605	0.8431	0.6372	0.1497
BN-r2 + LoRA(out) + rep	**0.5**	-0.3	**0.5620**	**0.8437**	**0.6398**	**0.1516**
BN-r2 + LoRA(out) + rep	1.0	-0.3	0.5609	0.8411	0.6393	0.1509

rep: flow repulsion loss. λ_{rep}: weight. m: margin.

Table A6. Dynamics separation.

Setting	λ_{dyn}	λ_{ic}	AP	AP@.5	AP@.9
Base	–	–	0.5447	0.8358	0.1187
BN-r2 + dyn + ic	0.05	0.3	0.5542	0.8392	0.1420

dyn: dynamics separation loss. ic: inter-cell band flow MSE.

E. Qualitative Comparison per Ablation Table

Each figure selects one scene from each of five cell types in the LIVECell test set, and places the input image, the GT instance mask, and each model's prediction in the corresponding table side by side.

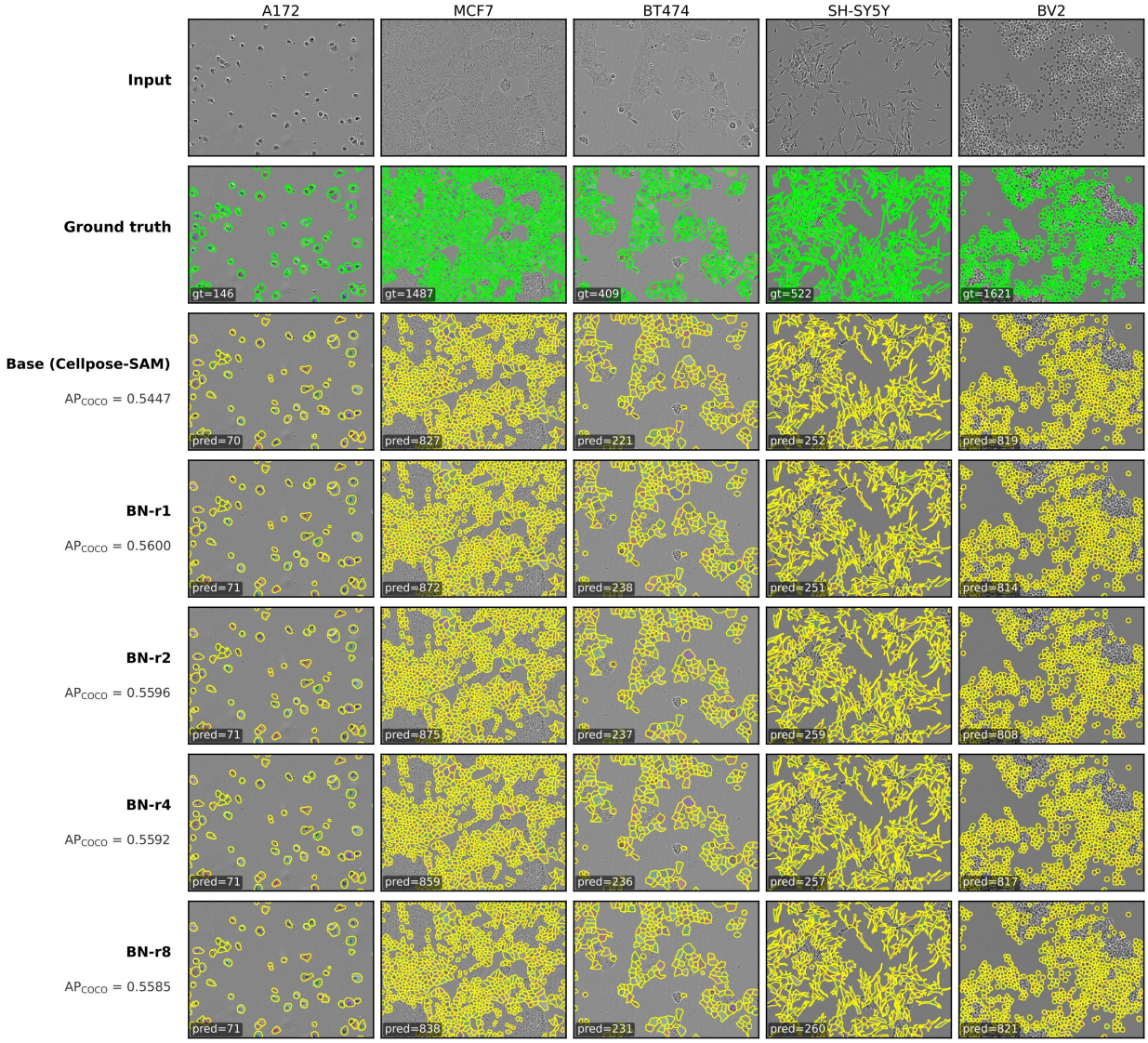

Figure A1. Qualitative comparison for Table A1 (PEFT capacity). Predictions of Base and BN serial $r \in \{1, 2, 4, 8\}$ models.

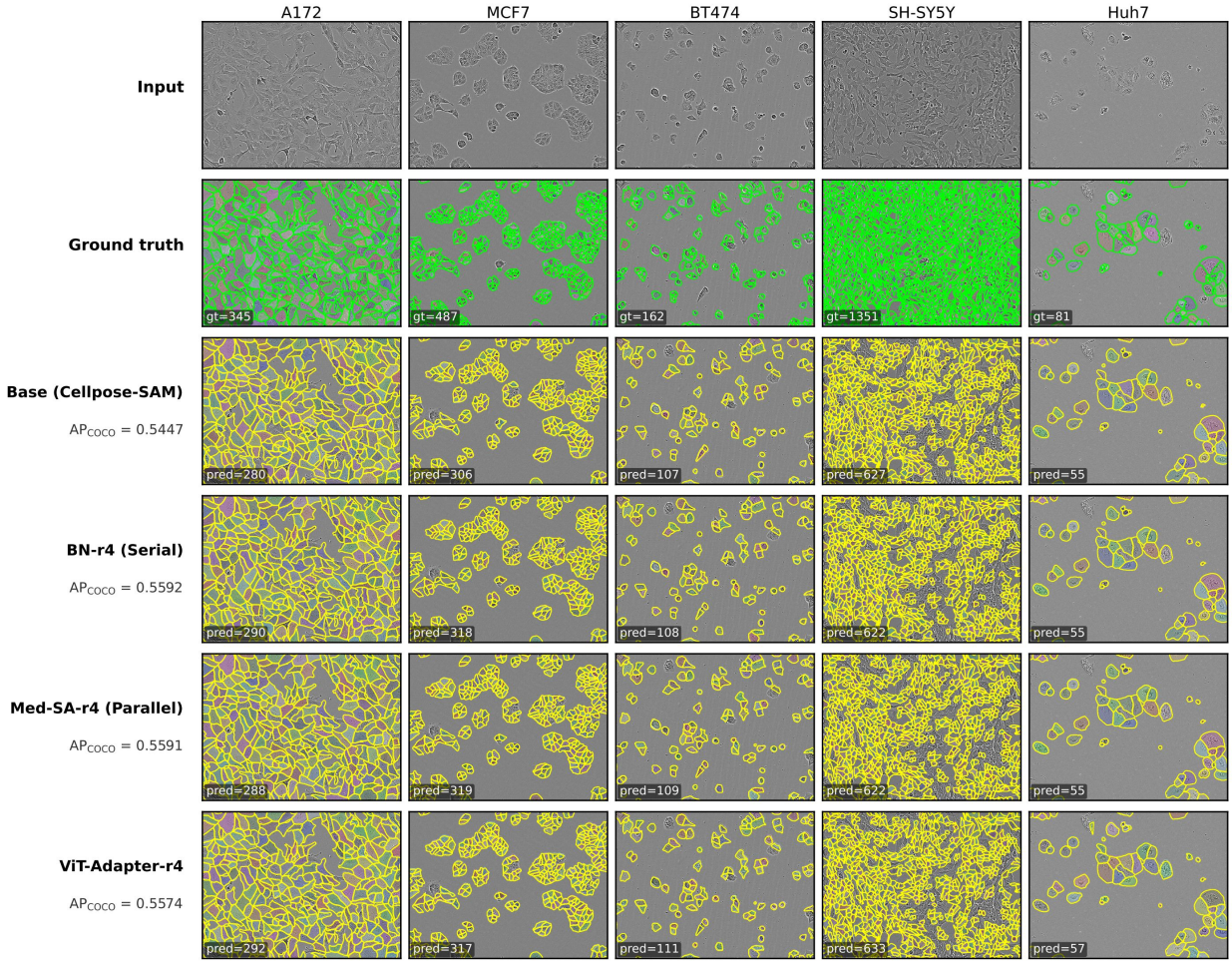

Figure A2. Qualitative comparison for Table A2 (adapter type/placement).

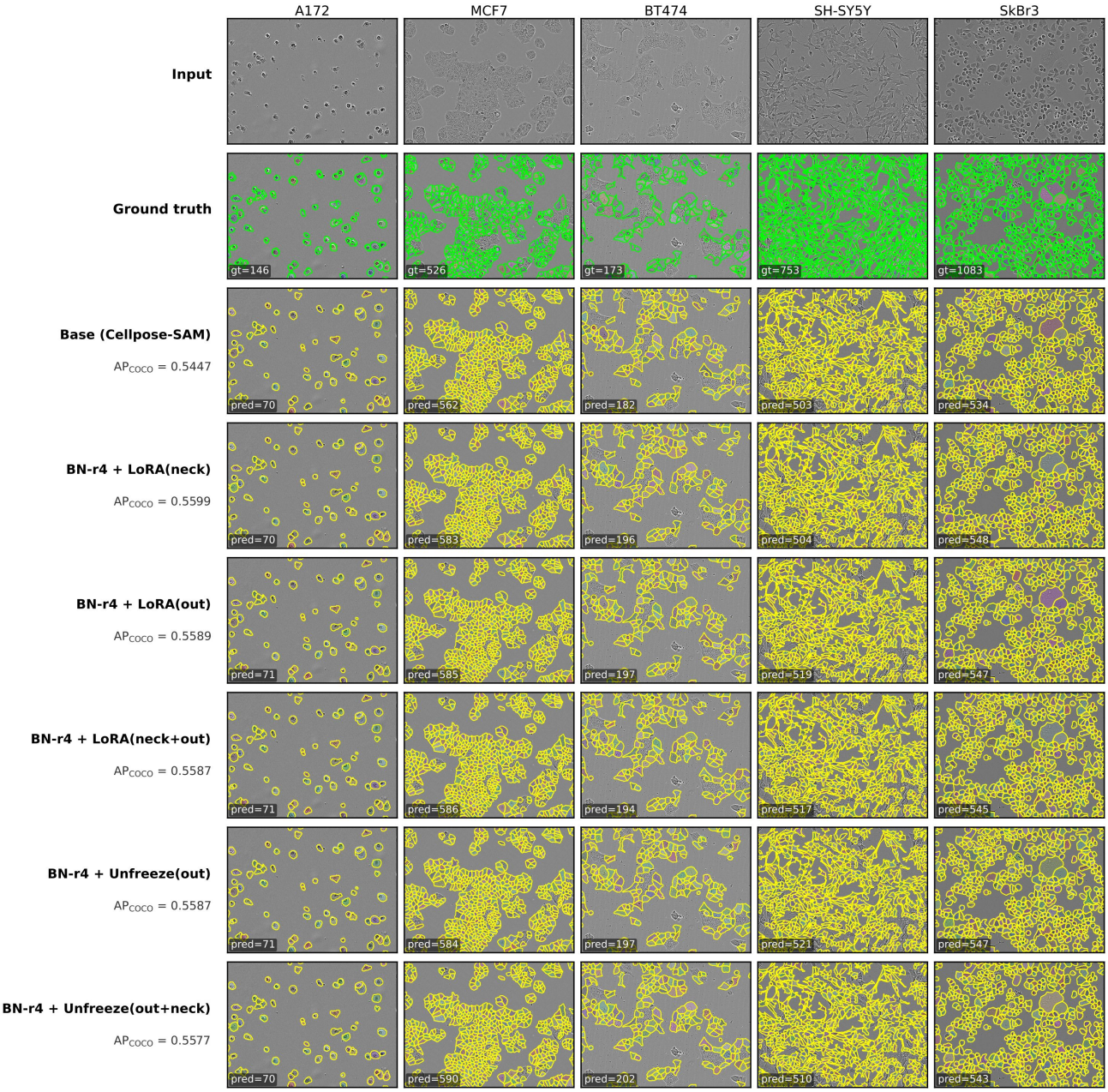

Figure A3. Qualitative comparison for Table A3 (decoder-side LoRA/partial unfreezing).

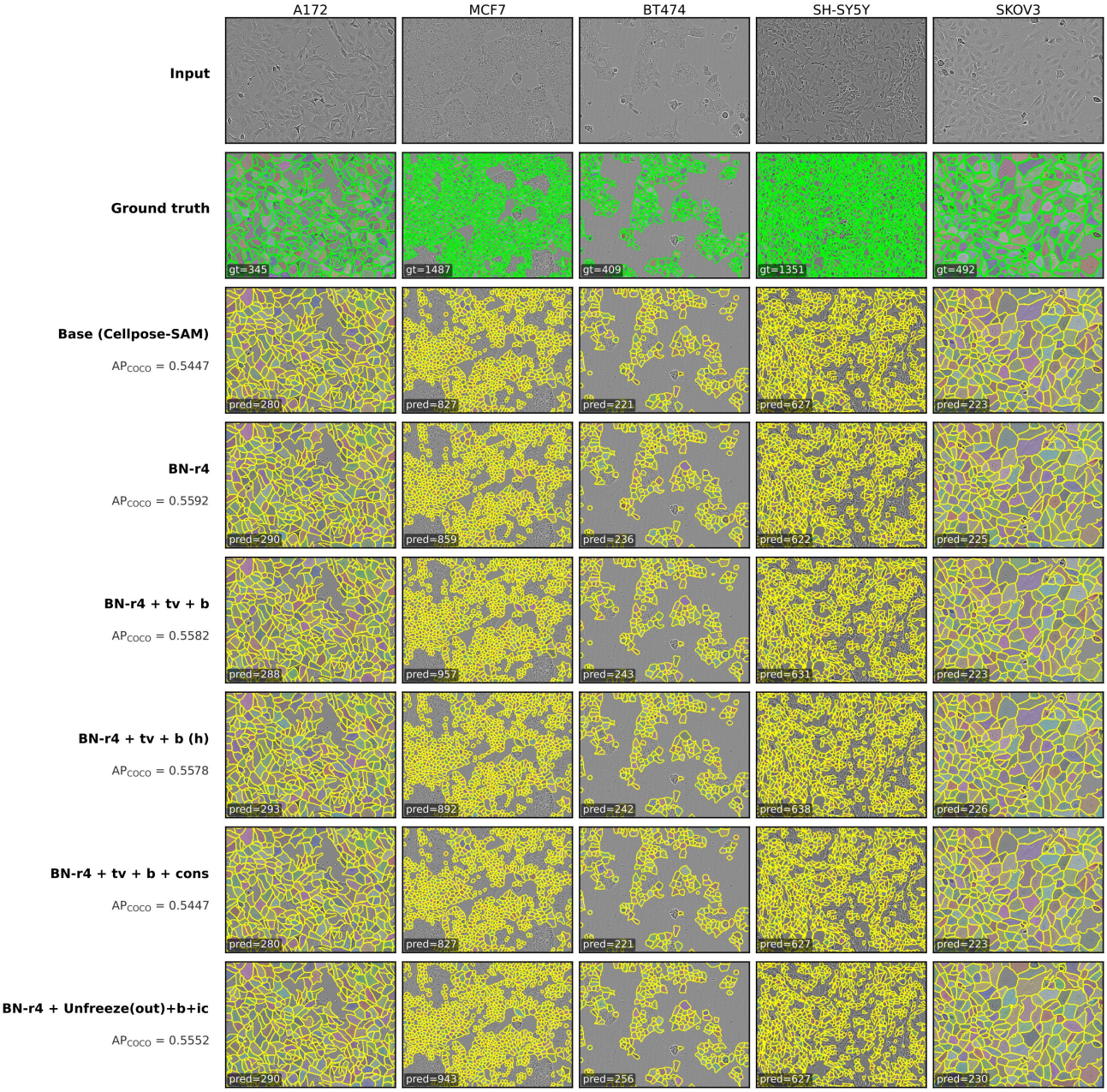

Figure A4. Qualitative comparison for Table A4 (reweighting loss).

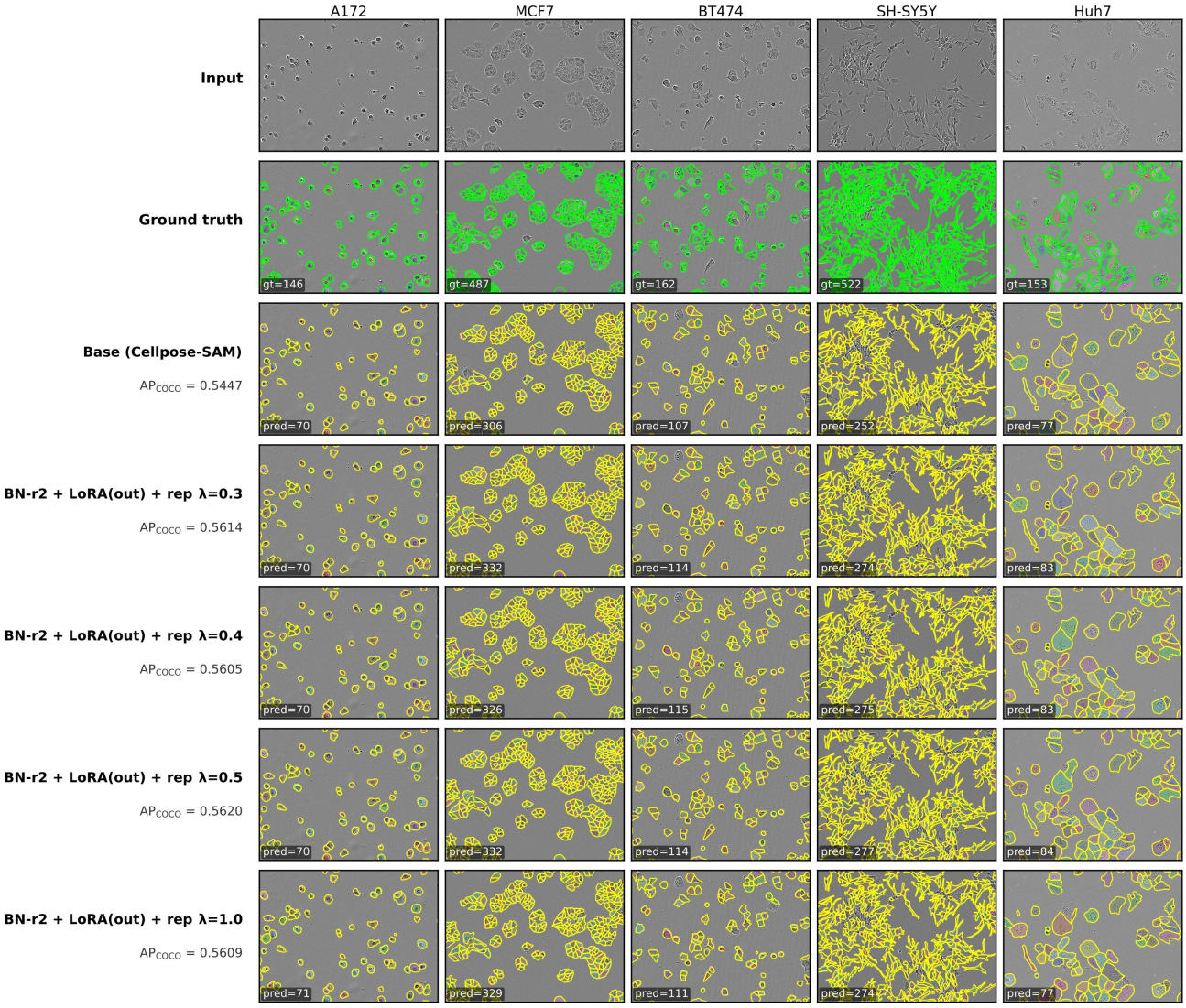

Figure A5. Qualitative comparison for Table A5 (flow repulsion weight sweep).

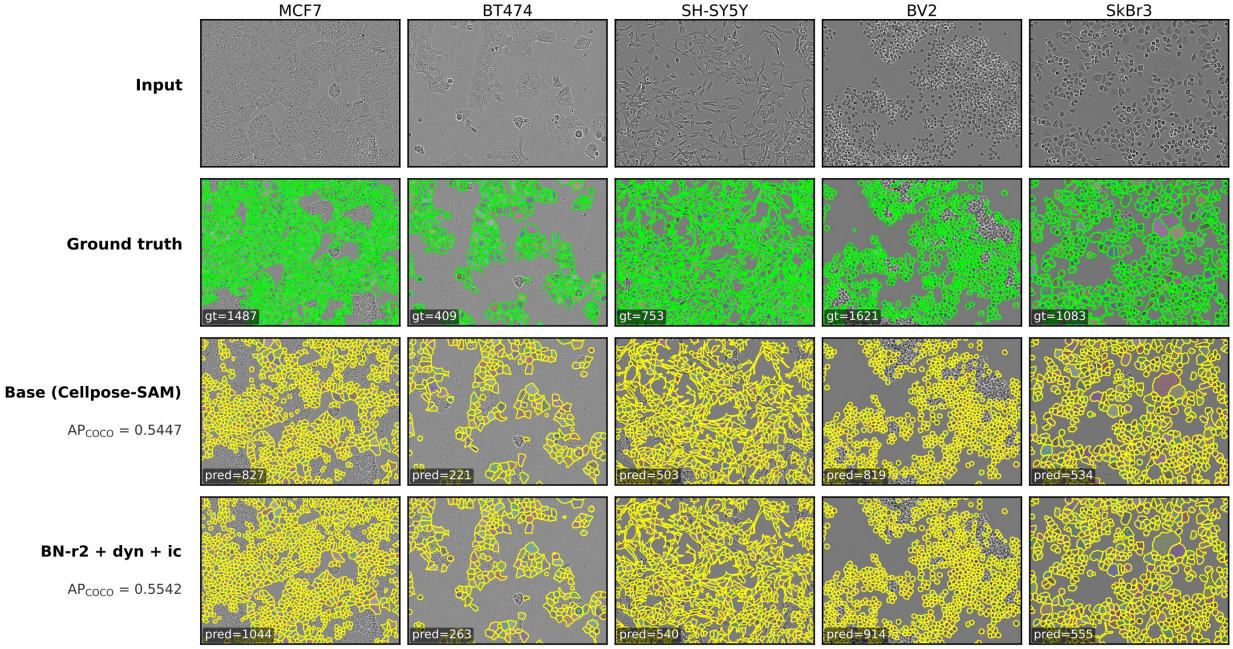

Figure A6. Qualitative comparison for Table A6 (dynamics separation).