applied sciences

MDPI

Article

Sparse Point-to-Pseudo-Instance Supervision for Annotation-Efficient Nuclei Segmentation

Weichen Huang [1], Yun Song [1,*] and Haobo Li [2]

[1] School of Computer Science and Technology, Changsha University of Science and Technology, Changsha 410076, China; 24408060242@csust.edu.cn
[2] Bordeaux Population Health Research Center, Université de Bordeaux, 33076 Bordeaux, France; haobo.li@u-bordeaux.fr
* Correspondence: songyun@csust.edu.cn

Featured Application

The framework generates pseudo-instance labels from sparse center points for nuclei instance segmentation in digital pathology. These labels can train downstream segmentation models when tracing every nucleus in the training images is impractical.

Abstract

Accurate nuclei instance segmentation requires pixel-level boundary annotation, which is expensive and time-consuming in histopathology images. We propose a sparse point-to-pseudo-instance supervision framework that learns from limited center-point annotations. The framework first completes sparse nucleus centers through curriculum-based center recovery and then generates instance-level pseudo labels using point-prompted foundation-model proposals. For completed centers without valid proposals, point-derived fallback regions are introduced to recover missing supervision, and all generated regions are assembled into unified pseudo-instance labels. The generated labels are converted into the native training targets of different downstream segmentation models, including HoVerNet and StarDist, while inference relies only on the trained segmentation networks. Experiments on MoNuSeg, CPM17, and CoNSeP evaluate segmentation performance under limited point budgets and analyze the coverage–precision trade-off introduced by fallback recovery. Under a 10% point annotation budget, the framework achieves AJI and PQ scores of 0.6233 and 0.6036 on MoNuSeg and 0.6653 and 0.6626 on CPM17, respectively. Point budgets cover training; fully annotated validation subsets support model selection. Additional validation analyses quantify recovered nuclei and false additions, showing that fallback improves supervision coverage while its effectiveness depends on the proposal coverage and center accuracy. The results demonstrate that sparse point annotations can be effectively converted into transferable pseudo-instance supervision for nuclei instance segmentation.

Keywords: nuclei instance segmentation; sparse point supervision; pseudo-instance labels; segmentation foundation model; coverage recovery; annotation efficiency

Check for updates

Academic Editor: Thomas Lindner

Received: 14 October 2026
Revised: 4 November 2026
Accepted: 4 November 2026
Published: 8 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Nuclei instance segmentation is an essential step for quantitative analysis of histopathology images, enabling measurements of nuclear morphology, density, and spatial organization. However, accurate instance-level segmentation requires pixel-wise boundary annotations, which are expensive and time-consuming to obtain, especially in crowded and

visually ambiguous tissue regions. Representative fully supervised approaches, including Mask R-CNN, HoVerNet, and StarDist, have achieved strong segmentation performance by learning from dense annotations [1–3]. Nevertheless, the annotation burden limits their scalability for large-scale pathological datasets.

Point annotations provide a more efficient alternative by replacing detailed contour tracing with sparse localization clicks. Previous point-supervised methods have demonstrated that simplified annotations can provide useful supervision through geometric expansion, auxiliary boundary learning, and co-training strategies [4–7]. However, full-point supervision still requires annotating every nucleus, which remains labor-intensive. Partial-point supervision further reduces annotation costs by labeling only a subset of nuclei [8,9]. This setting introduces additional challenges because unlabeled regions may contain both background regions and missing nuclei, making incomplete supervision difficult to interpret and potentially causing true instances to be suppressed during training.

Existing partial-point approaches mainly focus on recovering missing nuclei locations or constructing approximate supervision from sparse annotations. However, recovered locations alone do not guarantee complete instance-level supervision because a recovered center may fail to obtain a reliable instance region. Recent advances in foundation segmentation models provide new opportunities for generating instance candidates from sparse prompts, but the generated proposals may still suffer from missing candidates, ambiguous boundaries, or incomplete coverage. Therefore, an effective sparse supervision framework should not only recover missing nuclei locations but also evaluate proposal availability and construct reliable instance-level pseudo labels.

In this work, we propose a sparse point-to-pseudo-instance supervision framework that converts limited center annotations into instance-level training supervision. The framework first completes missing nucleus centers through curriculum-based center recovery and then generates point-prompted instance candidates using a fixed-foundation segmentation model. For completed centers without valid proposals, point-derived fallback regions are introduced to recover missing supervision. The generated regions are further assembled into unified pseudo-instance labels, which can be converted into the native training targets required by different downstream segmentation architectures. Unlike approaches that adapt the foundation model itself, our framework treats the foundation model as a pseudo-label generation component and evaluates the generated supervision through downstream instance segmentation networks.

The main contributions of this work are summarized as follows:

(1) We propose a sparse point-to-pseudo-instance supervision framework that transforms limited center annotations into instance-level pseudo labels through center completion and foundation model-guided proposal generation.
(2) We introduce a fallback-assisted region recovery strategy for completed centers without valid proposals and analyze its influence by measuring recovered instances and additional false regions.
(3) We validate the generated supervision across two downstream architectures, HoVerNet and StarDist, demonstrating that the proposed pseudo-instance representation can support different instance segmentation models without modifying their original designs.

2. Related Work

2.1. Fully Supervised Nuclei Instance Segmentation

Fully supervised methods learn instance shapes and separation cues from complete labels. Mask R-CNN predicts a mask for each detected region [1]. HoVerNet uses nuclear foreground probabilities and horizontal and vertical offsets to separate adjacent nuclei [2],

whereas StarDist predicts star-convex polygons from radial distances [3]. We use the latter two models as complementary downstream validation settings while retaining their original architectures and target-generation procedures.

2.2. Point-Supervised and Partially Point-Supervised Nuclei Segmentation

Point-supervised nuclei segmentation addresses the missing region information in point labels in several ways. Qu et al. derived coarse labels around annotated points [5]. PseudoEdgeNet introduces auxiliary boundary cues [6], and SC-Net combines self-supervised representation learning with co-training [7]. Point-guided attention and pseudo-label correction methods address noise in the derived supervision [10,11]. These methods use point-derived supervision, but their annotation coverage and training protocols differ from the partial-point setting considered here.

For the partial-point setting, Qu et al. used Gaussian center supervision and self-training to recover missing nuclei before segmentation training [8]. A later curriculum-based method also recovers nuclei progressively and uses a separate boundary-mining segmentation procedure [9]. Our first stage uses the center completion idea to generate prompts; the subsequent pseudo-instance generation, fallback assembly, and downstream supervision follow a different pipeline.

InstaSAM constructs pseudo-instances from point annotations and adapts SAM for nuclei segmentation [12]. All-in-SAM likewise generates pixel labels from weak prompts and then fine-tunes SAM for prediction without manual prompts at inference [13]. Our framework keeps the generic proposal model fixed and uses the generated instance labels to train separate segmentation networks rather than adapting the proposal model itself.

2.3. Foundation Models for Instance-Region Generation

Segment Anything (SAM) provides a promptable foundation model for general image segmentation [14], and SAM 2 extends this paradigm with improved image and video segmentation capability [15]. Recent studies have investigated the applicability and generalizability of foundation models for nuclei segmentation under different prompting strategies, including point and box prompts [16]. Domain-adapted variants, such as Segment Anything for Microscopy and CellSAM, further investigate microscopy-specific adaptation or detector-guided prompting [17,18].

Several studies also exploited SAM-derived regions as training supervision. All-in-SAM uses these labels to adapt SAM itself [13]. Nakai and Hotta used SAM-derived pseudo labels to train successive U-Nets and introduce logical union operations to recover missed nuclei [19]. These workflows motivate using foundation-generated masks as training labels. We evaluate an integer pseudo-instance representation that separates foundation model-based label generation from downstream segmentation training and can be converted into the targets required by separate downstream models.

2.4. Point-Derived Fallback Regions

Voronoi and cluster-derived regions have been used as coarse point-based supervision [5,7,9]. In our pipeline, this mechanism is invoked only for completed centers for which SAM2 candidate filtering returns no valid mask. SAM2-selected and fallback regions are subsequently assembled by a common conflict resolver. Nakai and Hotta similarly reported that supplementing SAM pseudo-labels can recover nuclei but may also increase over-segmentation [19]. We therefore quantify both recovered instances and false additions. Section 3.4 describes how joint assembly can alter overlapping regions.

3. Method

3.1. Problem Formulation and Overview

Let I: $\Omega \to R^3$ denote an RGB histology image on the pixel domain Ω. The available sparse manual center annotations are

$$P_s = \{p_i : i = 1, \ldots, N_s\}, \quad p_i \in \Omega \tag{1}$$

where p_i is the ith annotated nuclear center and N_s is the number of manual points. Stage 1 predicts an additional center set P_r. Manual points are immutable anchors, and the completed center set is

$$\hat{P} = P_s \cup P_r \tag{2}$$

The term completed refers to the union of observed and recovered centers; it does not assume that every true nucleus has been detected. Stage 2 converts $\hat{P}$ and I into an integer pseudo-instance map

$$\hat{M} : \Omega \to \{0, 1, \ldots, N_{inst}\} \tag{3}$$

where 0 denotes the background and each positive integer identifies one pseudo-nucleus. Figure 1 summarizes center completion, proposal generation with fallback recovery, and conversion to downstream supervision. The downstream models are alternative consumers of the generated supervision rather than sequential stages.

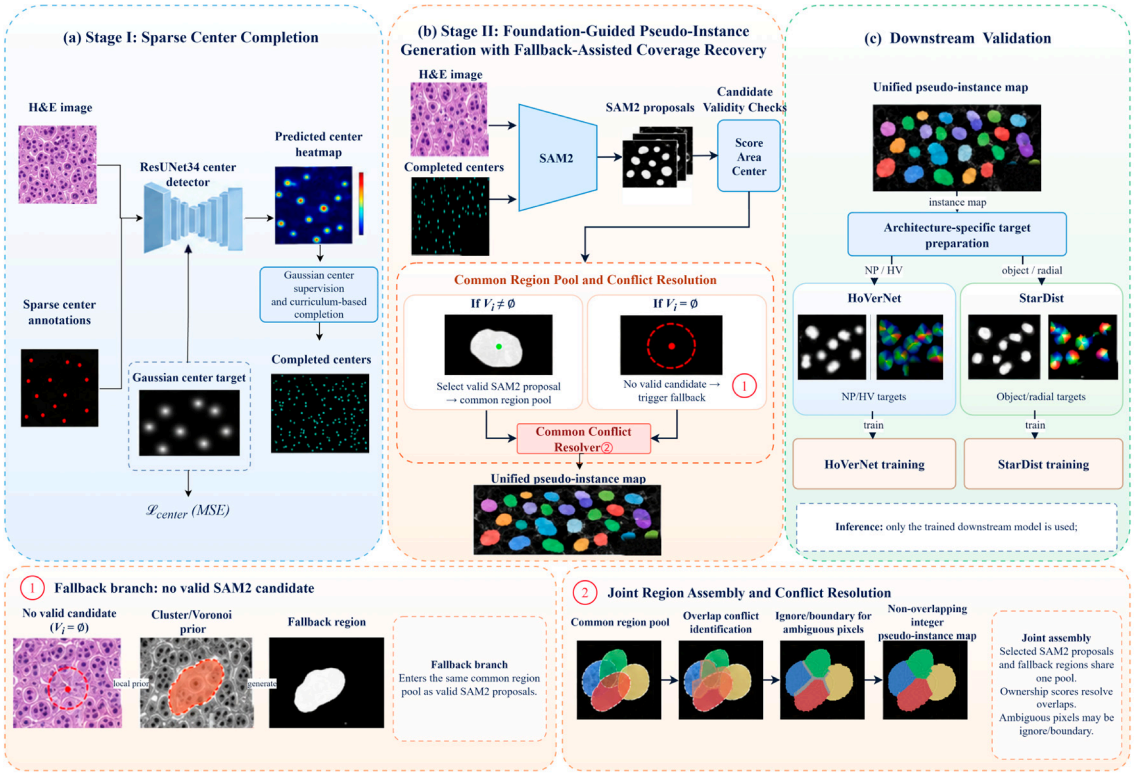


Figure 1. Overall framework for sparse point-to-pseudo-instance supervision.

Note that P_s denotes sparse manual centers, P_r denotes recovered centers, and $\hat{P}$ denotes their union. $\hat{M}$ is the final pseudo-instance map. The candidate index k is distinct from the final instance identifier.

3.2. Sparse Center Completion

3.2.1. Gaussian Center Supervision

Stage 1 uses ResUNet34, comprising a ResNet-34 encoder and a U-Net-style decoder with skip connections [20,21]. The encoder is initialized with ImageNet-pretrained

weights [22]. At curriculum round t, the current center-peak map A^t is convolved with a Gaussian kernel G_σ. The implemented target is

$$\begin{aligned} Z &= G_\sigma * A^t, \\ a &= \min\{Z(x) : A^t(x) > 0\}, \\ Y(x) &= \begin{cases} \min\left(1, \frac{Z(x)}{a}\right) & \text{if } \frac{Z(x)}{a} \geq 0.05 \\ 0 & \text{if } \frac{Z(x)}{a} < 0.05 \end{cases} \end{aligned} \tag{4}$$

We used a Gaussian standard deviation of $\sigma = 2.75$ for all three datasets. The initial supervision mask is constructed by dilating the Gaussian target peak regions with a disk with a radius of 22 pixels. During curriculum learning, regions with predicted center probabilities below 0.1 are additionally incorporated as background supervision. Positive target pixels receive a loss weight of 10, other supervised pixels receive a weight of 1, and pixels outside the supervision mask are ignored. Unannotated nuclei are not explicitly identified; their pixels contribute to the loss only when they fall within the current supervision mask.

The normalization value a is the smallest nonzero Gaussian response at a supervised peak. Normalized responses below 0.05 are discarded, values above 1 are clipped, and an empty peak map yields an all-zero target. Let $H_\theta(I)$ be the sigmoid center probability map predicted by the detector. Training minimizes the weighted squared error

$$e_\theta(x) = H_\theta(I)(x) - Y(x),\ L_{ctr} = [\Sigma_{x\in\Omega} w(x) e_\theta^2][\Sigma_{x\in\Omega} m(x)]^{-1} \tag{5}$$

where $m(x)$ is the binary supervision mask and $w(x) = 10$ at positive target pixels and is $w(x) = m(x)$ otherwise. The denominator counts supervised pixels; patches with an empty supervision mask are skipped.

3.2.2. Curriculum-Based Completion

Stage 1 builds on the curriculum-based center completion procedure of BoNuS [9,23], progressively incorporating confident detector predictions into center supervision. Sparse manual points remain fixed anchors, while eligible predictions are added during subsequent curriculum updates. Dense training instance masks are not used to generate curriculum targets. Complete validation annotations are used for model selection and validation-side analyses, as described in Sections 3.6 and 6.2.

The curriculum is updated every 30 epochs. Candidate centers are extracted from predictions using a confidence threshold of 0.65 and a maximum connected-component area of 150 pixels. Candidates overlapping existing detection labels are excluded. The remaining candidates are ranked using confidence, region area, and distance-related terms. For P existing centers and V candidates with nonzero ranking scores, at most $\lfloor Ve^{-0.5P/V} \rfloor$ ranked candidates are considered when $V > 0$; no candidates are considered when $V = 0$. Newly accepted centers must be at least 11 pixels away from existing or previously accepted centers in the same curriculum update.

3.2.3. Completed-Center Extraction

The final probability map is thresholded at τ. Connected components satisfying the selected area interval are represented by their centroids and filtered by the anchor-aware duplicate removal rule before they are merged with the manual points:

$$P_r = Extract(H_\theta; \tau, a_{min}, a_{max}, r_d) \tag{6}$$

Here, $a_{\min}$ and $a_{\max}$ are the component-area bounds and r_d is the duplicate removal radius. The extraction and merge therefore retain every manual center while adding only

nonduplicate recovered centers. The validation-selected center extraction settings (threshold, minimum area, and maximum area) were (0.65, 12, 150), (0.60, 8, 100), and (0.55, 8, 100) for MoNuSeg, CPM17, and CoNSeP, respectively, with a common manual center merge radius of 15 pixels.

3.3. Foundation-Guided Pseudo-Instance Generation

3.3.1. Completed Center Prompting

For each completed center p_i in $\hat{P}$, the fixed SAM2 model receives p_i as the positive point prompt and the configured neighboring centers as negative prompts. Multimask output is enabled, producing three binary candidates and their SAM2 model scores [15]:

$$C_i = \{(S_{ik}, s_{ik})\}_{k=1}^{K}, \quad K = 3 \tag{7}$$

Here, S_{ik} denotes the kth SAM2 candidate mask for center i, and s_{ik} denotes its model-provided quality score. These scores are not treated as calibrated IoU values, and no separate stability score is used.

3.3.2. Candidate Validity and Selection

A candidate is valid only if its SAM2 score meets or exceeds the selected threshold, its area lies within the selected interval, it contains its own center, and it does not contain another completed center. These checks are represented by

$$v_{ik} = (s_{ik} \geq \tau_s) \wedge (a_{min} \leq |S_{ik}| \leq a_{max}) \wedge (p_i \in S_{ik}) \wedge \left(n_{ik}^{other} = 0\right) \tag{8}$$

where n_{ik}^{other} is the number of other completed centers contained by the candidate after the configured optional exclusion radius dilation. For the matched SAM2-only and fallback-assisted operating point, $\tau_s = 0.65$, $a_{\min} = 20px^2$, and $a_{\max} = 3000px^2$. The Voronoi IoU, cell coverage, and outside ratio statistics are recorded for analysis but are inactive in the default candidate selection procedure.

The valid candidate index set and its availability indicator are

$$V_i = \{k : v_{ik} = 1\}, \quad c_i = 1\left[V_i \neq \emptyset\right] \tag{9}$$

Here, V_i denotes the set of valid SAM2 candidate indices for center i, and c_i is a binary indicator of whether this set is nonempty.

When V_i is nonempty, candidates are ranked first by descending SAM2 model score and then, on a score tie, by closeness of their area to a reference area of 200 px^2:

$$k_i^* = \arg\max_{k \in V_i} s_{ik}, \quad S_i^* = S_{i\,k_i^*} \tag{10}$$

The selected region S_i^* enters the common region pool. The method does not apply geometry-based candidate gating or boundary refinement in its default configuration.

3.4. Fallback-Assisted Joint Proposal Assembly

3.4.1. Fallback Trigger and Region Construction

Fallback is triggered at candidate selection time, before the regions are jointly assembled. Specifically, a completed center is eligible for fallback when no candidate survives the validity checks:

$$U = \{i : V_i = \emptyset\} \tag{11}$$

Here, U denotes the set of completed-center indices for which no SAM2 candidate satisfies the validity criteria. For each i ∈ U, we attempt to obtain a point-derived fallback region F_i using established cluster- and Voronoi-based weak label construction [5,7,9]. The

pixel feature vector is $z(x) = \left[I_R(x)/10, I_G(x)/10, I_B(x)/10, \min(d(x, \hat{P}), 20)\right]$. K-means uses three clusters (random state 0; 10 initializations). The nucleus cluster was the cluster with the greatest overlap with the completed point map; among the remaining clusters, the background cluster had the least overlap with a radius-5 dilation of the point map. Objects smaller than 30 pixels were removed. Within Voronoi cells, the nucleus foreground was dilated by a radius of 5, hole-filled, and eroded by a radius of 7. The refined foreground was then partitioned among completed centers by nearest-center assignment.

The fallback lookup uses the cluster-derived instance map and snaps each seed to the nearest positive region within an 8-pixel search window. If no positive region is found, then that center contributes no region. Generated fallback regions are not filtered again by the SAM2 score or candidate-area criteria; they proceed directly to the common conflict resolver. Its effects on coverage and false additions are evaluated in Section 6.2.

3.4.2. Common Region Pool and Conflict Resolution

For each completed center, a valid SAM2 proposal is used when available; otherwise, fallback is attempted. Only selected SAM2 proposals or successfully generated fallback regions enter the common region pool. Let R_i denote this source region and σ_i denote its source score:

$$R_i = c_i S_i^* + (1 - c_i) F_i, \sigma_i = c_i s_{ik_i^*} \tag{12}$$

Equation (12) represents conditional source selection and applies only to centers with an available source region; terms corresponding to the inactive branch are not evaluated. Centers for which both proposal selection and fallback construction fail are excluded from the common region pool.

SAM2 regions retain their model scores, whereas fallback regions receive a score of 0. All source regions are assembled jointly. For pixel x, let $J(x)$ be the indices of covering regions and define the ownership score as follows:

$$J(x) = \{i : x \in R_i\},\ u_i(x) = \sigma_i + \lambda_d/(1 + \|x - p_i\|_2),\ \lambda_d = 0.03 \tag{13}$$

Here, $\lambda_d = 0.03$ provides a small preference for regions whose centers are closer to the pixel. For pixels covered by at least two regions, $\Delta u(x)$ denotes the difference between the highest and second-highest ownership scores. Under the frozen winner policy, pixels covered by a single region are assigned directly. In overlapping regions, the highest-scoring source wins unless the two best ownership scores differ by less than $\delta = 0.015$; such ambiguous pixels are assigned background and marked as ignore or boundary pixels:

$$i^*(x) = \underset{i \in J(x)}{\operatorname{argmax}} u_i(x),$$
$$\hat{M}(x) = \begin{cases} i^*(x) & \text{if} |J(x)| = 1 \vee \Delta u(x) \geq \delta \\ 0 & \text{otherwise} \end{cases}, \ \delta = 0.015 \tag{14}$$

When the covering set is empty, where $(J(x) = \emptyset)$, the pixel is assigned to background, where $(M(x) = 0)$, without evaluating the ownership maximum. The score difference $\Delta u(x)$ is evaluated only when at least two source regions cover the pixel. After ownership assignment, a one-pixel object boundary is computed and dilated by one pixel. Foreground pixels in this boundary are also marked in a separate boundary or ignore map. Positive instance identifiers are finally relabeled contiguously. Because SAM2-selected and fallback regions share this resolver, overlap pixels may be reassigned or ignored. An unchanged-mask audit consequently describes the measured outcome for a given split and configuration rather than a theoretical guarantee.

3.4.3. Output Representation

The output of Stage 2 is the integer map $\hat{M}$ in Equation (3), together with separate boundary and ignore maps that record ambiguous and boundary pixels. This representation separates pseudo-instance generation from the architecture-specific targets used by a downstream model.

3.5. Downstream Supervision Conversion

The integer pseudo-instance representation is converted into the native targets required by each downstream architecture. Let A_H and A_S denote the respective preparation operators, including patch extraction and any model-specific consistency handling. HoVerNet receives its standard nuclear-pixel and horizontal or vertical targets [2]:

$$T_H(A_H(\hat{M}, \hat{P})) = (Y_{NP}, Y_{HV}) \tag{15}$$

StarDist receives its standard object-probability and radial-distance targets [3]:

$$T_S(A_S(\hat{M}, \hat{P})) = (Y_{obj}, Y_{rad}) \tag{16}$$

The separate Stage 2 ignore map is not passed as an explicit loss mask in the evaluated downstream training pipelines. HoVerNet generates NP and HV targets from the integer instance map, whereas StarDist uses an adapted integer instance label map. Pixels assigned a value of zero are handled through the resulting labels according to the respective model-specific training procedures. Although both architectures share the same sparse-center-to-instance generation protocol, their target preparation procedures may produce different effective training labels. Therefore, cross-architecture validation assesses the transferability of the pseudo-instance generation protocol rather than identical training tensors. SAM2 is used only during pseudo-label generation, and each trained downstream model performs image-only inference.

3.6. Annotation Access and Leakage Control

In controlled experiments, sparse center-point annotations are simulated by sampling from the available training instance annotations. Training uses RGB images, these budgeted center points, and generated pseudo-instance targets. The complete training instance masks are used only to derive sparse center points; they are not used for Stage 1 optimization, pseudo-label construction, or downstream weakly supervised optimization. A small, fully annotated validation subset is used for predeclared selection of the Stage 1 checkpoint and center extraction operating point, the Stage 2 candidate operating point, the downstream checkpoint, and downstream postprocessing and pruning settings. The same validation annotations are also used for the diagnostic analyses in Section 6.2 but not as training prompts or gradient targets.

Specifically, the validation subsets are Val6 for MoNuSeg, Val4 for CPM17, and Val5 for CoNSeP. Test annotations are used only for final evaluation of the frozen models and are not used for training, model selection, or hyperparameter tuning. The split-specific annotation access policy is summarized in Table 1.

The nominal 10% annotation budget refers to center points in the training split. It does not include the fully annotated validation subsets (6 images for MoNuSeg, 4 images for CPM17, and 5 images for CoNSeP) or the benchmark test labels. These dense labels are disclosed separately because they are needed for validation-based selection and final evaluation, even though they do not enter training optimization or pseudo-label construction.

Table 1. Information access by data split.

Split	Accessible Annotation	Use
Training	Sparse center points and generated pseudo targets; no dense-mask model supervision	Stage 1 and downstream optimization
Validation	Full instance masks in a small held-out subset	Checkpoint, operating point, and postprocessing selection; validation diagnostics
Test	Full instance masks	Final evaluation of frozen models only

4. Experimental Set-Up

4.1. Datasets and Splits

MoNuSeg provided H&E images from multiple organs; its challenge release contained 30 training and 14 test images 1000 × 1000 pixels in size [24,25]. Our split allocated 24, 6, and 14 images to training, validation, and test sets, respectively. CPM17 was evaluated with 28, 4, and 32 images [26]. Image-level partitions were fixed before patch extraction. Table 2 gives the dataset partitions, patch protocol, and point budgets. CoNSeP comprised 41 H&E-stained histopathology images 1000 × 1000 pixels in size, which were partitioned into 22 training, 5 validation, and 14 test images in our experiments [2]. The official training and test partitions were retained, with validation subsets selected from the respective training partitions. Although all splits were disjoint at the image level, patient- or slide-level grouping could not be independently verified for every dataset because the required identifiers were not consistently available.

Table 2. Dataset partitions, patch extraction, and annotation protocol.

Dataset	Train/Val/Test Images	Original Resolution	Patch/Stride	Budget Analysis
MoNuSeg	24/6/14	1000 × 1000	540 × 540/164 × 164	5/10/15%
CPM17	28/4/32	500 × 500/600 × 600	540 × 540/164 × 164	5/10/15%
CoNSeP	22/5/14	1000 × 1000	540 × 540/164 × 164	5/10/15%

4.2. Sparse Annotation Protocol

The 10% point ratio is the primary operating budget; 5% and 15% were used to study the annotation–performance relationship. Sparse center points were sampled independently within each training image rather than globally across the training split. For all three datasets, an image containing $N_i > 0$ eligible centers contributed $\max(1, \lfloor rN_i \rfloor)$ annotated points at ratio r using sampling seed 7. The 5% and 15% annotation sets were constructed as subsets and supersets, respectively, of the frozen 10% point set. The resulting nested point sets satisfied

$$P_{5\%}{}^{s} \subset P_{10\%}{}^{s} \subset P_{15\%}{}^{s} \quad (17)$$

For each dataset and annotation budget, locally reproduced methods evaluated under the same budget used the same frozen training point files. The same per-image evaluator was used for all locally reproduced methods. Each result was identified by its available run or aggregation level.

4.3. Implementation Details

Stage 1 was trained for a fixed 100 epochs without early stopping on all three datasets, with an initial learning rate of (1×10^{-4}). It used the ResUNet34 architecture described in

Section 3.2, with the encoder initialized from ImageNet-pretrained weights [20–22]. The local training routine used Adam with betas (0.9, 0.99) and the weighted MSE in Equation (5). Checkpoints and operating points were selected using the validation-only procedures in Section 3.6.

Pseudo-instance labels were generated using SAM 2.1 Hiera Base+ [15] with the sam2.1_hiera_base_plus.pt checkpoint and sam2.1_hiera_b+.yaml configuration, together with the point-derived fallback described in Section 3.4. SAM2 remained frozen and generated proposals from point prompts without fine-tuning on the target datasets. HoVerNet and StarDist retained their original architectures and were trained using NP and HV targets and object-probability and radial-distance targets, respectively [2,3]. Downstream checkpoints and postprocessing settings were selected on the declared validation subsets.

4.4. Baselines and Comparison Policy

We locally reproduced WeaklySegPartialPoints [8], SC-Net [7], SPN+IEN [27], and BoNuS [9] using the same image-level splits and evaluation code. At each annotation budget, the reproduced point-supervised methods used the same sampled point files. Model and parameter selection followed the validation procedures described in Section 3.6. Table 3 includes these results alongside fully supervised HoVerNet and StarDist [2,3] for comparison.

Table 3. Comparison of nuclei instance segmentation methods under different annotation supervision settings.

Method	Supervision	Budget	MoNuSeg			CPM17			CoNSeP		
			Dice	AJI	PQ	Dice	AJI	PQ	Dice	AJI	PQ
Fully supervised methods											
HoVerNet	Full mask	100% masks	0.7924	0.6215	0.6029	0.8727	0.7066	0.6942	0.8420	0.5671	0.5112
StarDist	Full mask	100% masks	0.7921	0.6281	0.6239	0.8498	0.6848	0.6837	0.7874	0.5339	0.5039
Full point-supervised methods											
BoNuS	Point	100% points	0.7618	0.5888	0.5423	0.7846	0.6098	0.5717	0.6132	0.3756	0.2756
SC-Net	Point	100% points	0.7282	0.5317	0.4874	0.7448	0.5138	0.4452	0.7004	0.2701	0.2298
SPN+IEN	Point	100% points	0.7173	0.5134	0.4414	0.7669	0.5857	0.4882	0.7408	0.4161	0.2819
Partial point-supervised methods (10%)											
WSPP	Partial point	10% points	0.7558	0.5746	0.5359	0.7194	0.5371	0.5078	0.6423	0.3850	0.3093
BoNuS	Partial point	10% points	0.7424	0.5718	0.5427	0.7511	0.5567	0.5116	0.4962	0.2800	0.2049
Ours	Partial point	10% points	0.7969	0.6233	0.6036	0.8386	0.6653	0.6626	0.6590	0.3863	0.3370

Table 4 lists the published results from EOFD-Net [28], DoNuSeg [29], PLCUD [11], Constraint-Attention [30], and Sparsely Guided [31]. These results followed the original experimental settings, whose annotation budgets, data splits, and evaluation protocols may differ from ours. The SAM2-only setting was also included in the supervision-source ablation.

Table 4. Comparison with recently published nuclei segmentation methods under reported experimental settings.

Method	Supervision	MoNuSeg			CPM17			CoNSeP		
		Dice	AJI	PQ	Dice	AJI	PQ	Dice	AJI	PQ
EOFD-Net	100% point	0.7412	0.5237	-	-	-	-	-	-	-
DoNuSeg	100% point	-	-	-	-	-	-	0.6020	0.3639	0.2395
PLCUD	100% point	0.7444	0.5314	-	-	-	-	-	-	-
Constraint-Attention	1/16 partial point	0.7390	-	-	-	-	-	-	-	-

Table 4. *Cont.*

Method	Supervision	MoNuSeg			CPM17			CoNSeP		
		Dice	AJI	PQ	Dice	AJI	PQ	Dice	AJI	PQ
Sparsely Guided	100% point	0.7821	0.6176	0.6038	0.7913	0.6236	0.5974	0.6517	0.4248	0.3528
Ours	10% partial point	0.7969	0.6233	0.6036	0.8386	0.6653	0.6626	0.6590	0.3863	0.3370

4.5. Metrics

We report the foreground Dice coefficient, AJI, PQ, DQ, SQ, instance precision, recall, F1 score, and predicted-to-ground-truth count ratio. AJI evaluates the aggregate instance overlap [24], and PQ combines detection and matched-mask segmentation quality [32], with the nuclei evaluation conventions specified alongside HoVerNet [2]. All reproduced rows use the same evaluator. The Dice coefficient, AJI, PQ, DQ, SQ, precision, recall, and F1 score are reported on a 0–1 scale. The improvements described in Section 6.1 are expressed in percentage points, whereas the metric differences in the validation audit are reported as absolute differences on the original scale.

5. Main Results

5.1. Main Comparison at the 10% Point Budget

Table 3 compares the proposed framework with partial point-supervised methods at the 10% point budget. Fully supervised and 100% point-supervised methods provided references for different annotation requirements. In the reported single run on the MoNuSeg test split, our method achieved a Dice score of 0.7969, an AJI of 0.6233, and a PQ of 0.6036.

Figure 2 shows segmentation examples across the evaluated datasets and point annotation protocols.

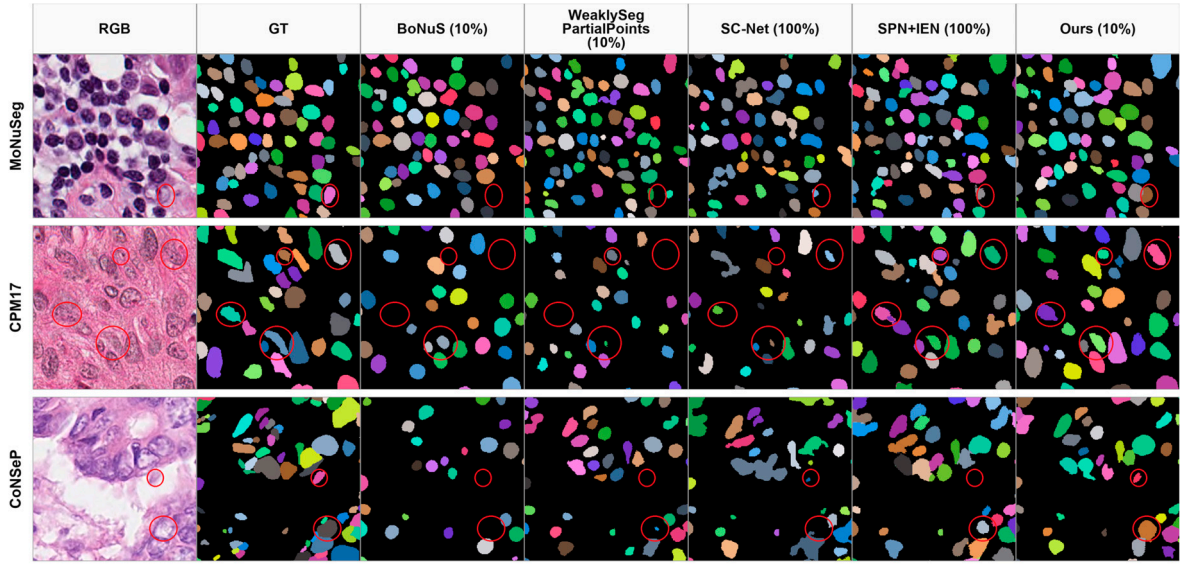


Figure 2. Qualitative comparison of nuclei instance segmentation results under different point annotation protocols. Different colors represent individual nuclear instances. Red circles highlight representative regions where nuclei are missed or incompletely segmented by one or more comparison methods but are successfully recovered by the proposed method.

Table 4 lists additional published results under their original annotation protocols. These methods are included to provide a broader literature context, whereas Table 3 focuses on comparisons under the controlled experimental protocol used in this study.

5.2. Performance Versus Point Annotation Budget

Table 5 reports the test set performance at 5%, 10%, and 15% point annotation budgets using the nested sampling protocol described in Section 4.2. Figure 3 plots the corresponding Dice, AJI, and PQ scores at the three point annotation budgets. Each dataset and budget configuration was evaluated using a single training run.

Table 5. Segmentation performance across nested point budgets.

Dataset	Point Ratio	Dice	AJI	DQ	SQ	PQ
MoNuSeg	5%	0.8037	0.6283	0.8139	0.7471	0.6088
MoNuSeg	10%	0.7969	0.6233	0.8020	0.7518	0.6036
MoNuSeg	15%	0.7996	0.6253	0.8053	0.7505	0.6053
CPM17	5%	0.8214	0.6382	0.8148	0.7880	0.6437
CPM17	10%	0.8386	0.6653	0.8325	0.7937	0.6626
CPM17	15%	0.8350	0.6646	0.8342	0.7842	0.6566
CoNSeP	5%	0.6323	0.3666	0.4545	0.6941	0.3168
CoNSeP	10%	0.6590	0.3863	0.4822	0.6952	0.3370
CoNSeP	15%	0.6830	0.3951	0.4741	0.6948	0.3309

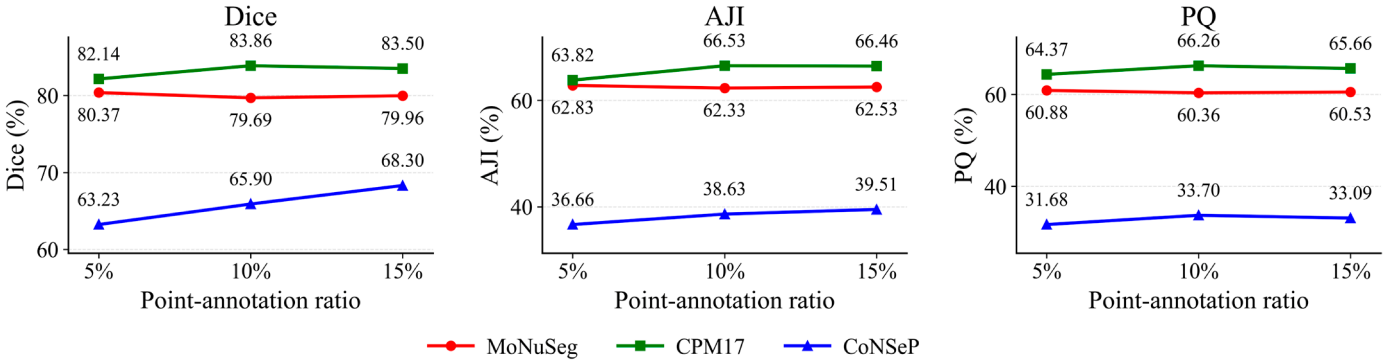


Figure 3. Dice, AJI, and PQ scores under different point-annotation budgets.

Across the nested point budgets, performance varied by dataset and metric. On MoNuSeg, changes across 5–15% points were modest: the Dice, AJI, DQ, and PQ scores were highest at 5%, whereas the SQ was highest at 10%. On CPM17, the Dice, AJI, SQ, and PQ scores peaked at 10%, while the DQ increased slightly further at 15%. On CoNSeP, the Dice score and AJI increase with the point budget, while the DQ and PQ peaked at 10% and the SQ remained nearly unchanged. Increasing the point budget did not consistently improve segmentation performance across the three datasets. Although the point subsets were nested, additional annotated centers did not necessarily translate into proportionally more valid pseudo-instance regions. Candidate availability, fallback reliability, and differences between independently trained downstream models may have limited the gains from additional supervision. However, the single-run results did not establish a specific saturation mechanism or its causal contribution.

6. Ablation and Additional Analyses

6.1. Ablation of Pseudo-Instance Supervision Sources

We evaluated four pseudo-instance supervision variants: Point-Disk, Voronoi–Cluster, SAM2-only, and our model (Table 6). Point-Disk generates local disk-shaped regions around completed centers. Voronoi–Cluster constructs regions through point-derived clustering and Voronoi partitioning. SAM2-only uses valid SAM2 proposals without fallback. Our model combines valid SAM2 proposals with point-derived fallback regions through joint

conflict resolution. All four variants used the same completed centers and followed the same downstream training and validation selection protocol.

To assess the contribution of fallback within the SAM2-based pipeline, we compared the SAM2-only model and our model under matched upstream conditions: completed centers, the SAM2 model, point prompts, and candidate filtering settings. Each supervision variant was used to train a separate HoVerNet model, with its checkpoint and postprocessing settings selected independently under the same validation protocol.

Table 6. Ablation of pseudo-instance supervision sources using HoVerNet at a 10% point budget.

Variant	Disk	VC	SAM2	FB	MoNuSeg		CPM17		CoNSeP	
					AJI	PQ	AJI	PQ	AJI	PQ
Point-Disk	✓	—	—	—	0.5129	0.4558	0.4549	0.4038	0.3077	0.2020
Voronoi–Cluster	—	✓	—	—	0.5649	0.5262	0.4876	0.4536	0.3561	0.2738
SAM2-only	—	—	✓	—	0.5600	0.5710	0.6548	0.6620	0.2161	0.2327
Ours	—	—	✓	✓	0.6233	0.6036	0.6653	0.6626	0.3863	0.3370

Note: VC = Voronoi–Cluster; FB = fallback. Checkmarks indicate the supervision components used by each variant.

In Table 6, fallback-assisted joint assembly improved the AJI and PQ over SAM2-only supervision by 6.33 and 3.26 percentage points on MoNuSeg and by 17.02 and 10.43 percentage points on CoNSeP, respectively. On CPM17, the gains were smaller: 1.05 percentage points for the AJI and 0.06 percentage points for the PQ, with the PQ remaining nearly unchanged. The validation coverage analysis in Section 6.2 is consistent with missing region recovery contributing to the downstream gains, although it does not isolate their cause. CPM17 had a smaller validation coverage deficit, leaving a smaller proportion of completed centers eligible for fallback under the specified candidate validity criteria.

6.2. Coverage Recovery and Fallback Analysis

We measured the recovered nuclei and false additions to assess the effect of fallback on the validation pseudo-instance maps.

Table 7 reports the validation pseudo-map coverage and error measurements for MoNuSeg, CPM17, and CoNSeP using the definitions above. Figure 4 compares the coverage deficit and fallback ratio across the three validation splits. On MoNuSeg Val6, the audit reported a 37.59% coverage deficit and a 36.84% fallback-instance ratio. Among the added fallback instances, 541 were identified as useful fallback additions matched to ground-truth nuclei, corresponding to a useful fallback fraction of 47.04%.

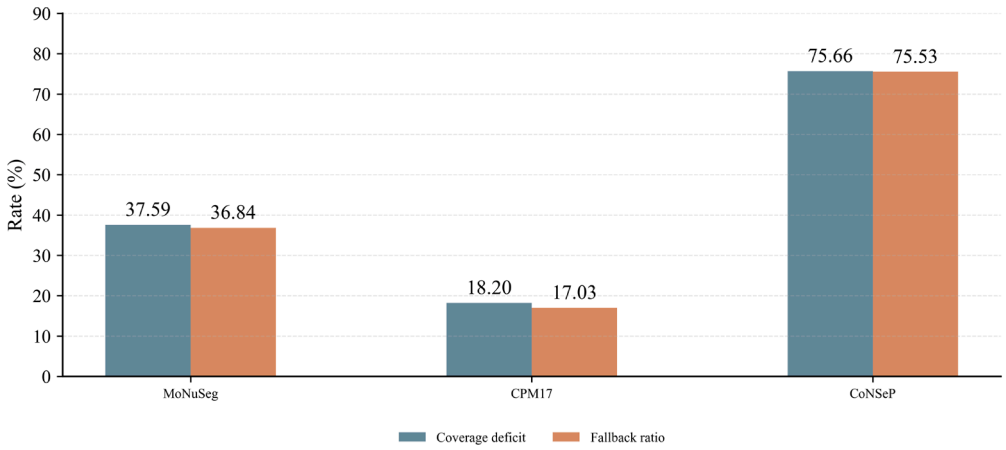


Figure 4. Coverage deficit and fallback ratio across validation datasets.

Table 7. Validation-split pseudo-instance map coverage and error analysis.

Dataset	Audit Split	Coverage Deficit	Fallback Ratio	Useful Fallback	NEW_TP	FP _LIKE	Delta Recall	Delta AJI	Delta PQ	Delta Prec
MoNuSeg	Val6	37.59%	36.84%	47.04%	541	365	+0.2506	+0.1056	+0.0706	−0.0374
CPM17	Val4	18.20%	17.03%	27.59%	32	69	+0.0803	+0.0249	−0.0203	−0.0762
CoNSeP	Val5	75.66%	75.53%	34.70%	686	1096	+0.4362	+0.1924	+0.0898	−0.1370

Note: Let U denote the set of indices of completed centers without a valid SAM2 candidate, $\hat{P}$ denote the completed-center set, and N_{FB} denote the number of added fallback instances. Coverage deficit $|U|/\left|\hat{P}\right|$ and fallback ratio $N_{\mathrm{FB}}/|P|$ are image-macro means. The useful fallback fraction $\mathrm{NEW}_{\mathrm{TP}}/N_{\mathrm{FB}}$ and category counts are pooled. FP_LIKE denotes an unreliable addition under the audit matching rule, not a downstream false-positive prediction. Metric deltas are image-macro differences between fallback-assisted and SAM2-only validation pseudo-instance maps. All statistics are validation-based, not test results.

A separate MoNuSeg validation audit assigned each ground-truth nucleus to one mutually exclusive error category. It identified 1002 nuclei with a nearby completed center but insufficient proposal overlap and 87 covered nuclei that failed the final match after excluding merge and fragmentation cases. These are counts of the ground-truth nuclei and not counts of the SAM2 candidates. Their ratio was 11.52. The diagnostic used an IoU match threshold of 0.5, a center radius of 15 pixels, substantial coverage thresholds of 0.10 IoU or 0.25 ground-truth coverage, and a boundary tolerance of two pixels.

The CPM17 Val4 audit contained 557 completed centers. Of these, 119 had no SAM2 candidate passing the validity criteria, and 116 fallback instances were added. The image-macro coverage-deficit and fallback ratios were 18.20% and 17.03%, respectively. The additions included 32 NEW_TP, 69 FP_LIKE, 15 merge and ambiguous cases with no duplicates. The useful fraction was 32/116 = 27.59%; this is not a precision estimate for the entire segmentation map. For this Val4 audit and its recorded configuration, all 438 baseline masks were observed to be unchanged, and the 397 common matched instances had zero change in the IoU and boundary F1 score. The audit's image-macro pseudo-map deltas were +0.0803 recall, +0.0249 AJI, −0.0203 PQ, and −0.0762 precision.

Figure 5 presents a MoNuSeg example to illustrate how fallback supplemented the regions absent from the SAM2-only proposal map.

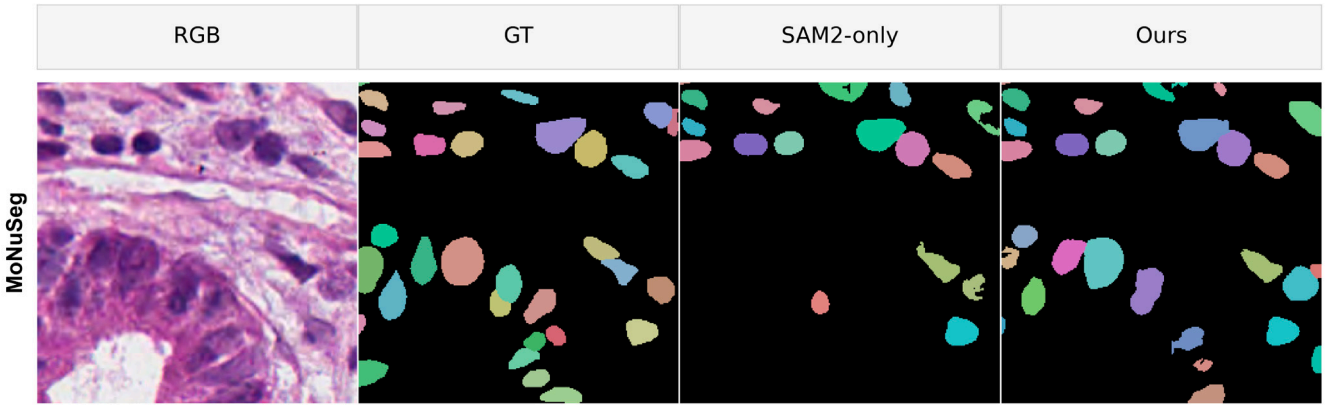


Figure 5. Qualitative visualization of fallback-based coverage recovery. Different colors denote individual nuclear instances for visual distinction.

Compared with the SAM2-only baseline, fallback-assisted assembly added regions that were previously missing, providing a visual illustration of the coverage gains reported in Table 7. Together with the precision decreases observed in the validation audits, these results indicate that fallback expanded supervision coverage but did not guarantee that every added region corresponded to a true nucleus.

6.3. Computational Overhead

We measured the offline computational overhead of the proposed framework on MoNuSeg, CPM17, and CoNSeP using an NVIDIA GeForce RTX 5060 Ti GPU (16 GB; NVIDIA Corporation, Santa Clara, CA, USA). Measurements used the frozen configurations and a 10% point annotation budget. The runtime benchmarks for all three datasets used Python 3.10.21, PyTorch 2.8.0+cu129, and torchvision 0.23.0+cu129. We processed the complete training split of each dataset, with a Stage 1 inference batch size of 16.

Table 8 reports the processing time and peak GPU memory for Stage 1 center completion and Stage 2 pseudo-instance generation. Stage 1 required 0.8501, 0.2511, and 0.7826 s per image on MoNuSeg, CPM17, and CoNSeP, respectively. The network-only inference latency ranged from 1.231 to 1.359 ms per 224 × 224 tile. Stage 2 required 20.3869, 5.8887, and 15.8614 s per image, respectively, excluding shared preprocessing.

Table 8. Computational overhead of offline pseudo-instance generation at the 10% point annotation budget.

Dataset	Stage 1 (s/Image)	Stage 1 (ms/Tile)	Stage 1 VRAM (GiB)	Stage 2 (s/Image)	Stage 2 VRAM (GiB)	Total (s)
MoNuSeg	0.8501	1.359	0.403	20.3869	0.788	1250.43
CPM17	0.2511	1.283	0.403	5.8887	0.788	256.69
CoNSeP	0.7826	1.231	0.403	15.8614	0.788	995.56

Note: Measurements cover the complete training splits (MoNuSeg = 24 images; CPM17 = 28; CoNSeP = 22). Stage 1 used 224 × 224 tiles with an inference batch size of 16. Its per-image time includes complete center recovery, while the per-tile time measures network inference only. The Stage 2 per-image time excludes shared Voronoi–cluster preprocessing, which was included once in the total processing time. VRAM refers to the peak GPU memory allocated by PyTorch (version 2.8.0+cu129), measured separately for each stage, including model weights. Reported times exclude Stage 1 and downstream segmentation network training.

Total offline pseudo-label generation took approximately 1250.43, 256.69, and 995.56 s for the complete MoNuSeg, CPM17, and CoNSeP training splits, respectively, including the required shared preprocessing. Stage 2 accounted for most of the measured offline generation time. This cost occurs during pseudo-label construction and does not affect downstream segmentation inference, which uses only the trained segmentation network.

These measurements reflect the implemented offline pseudo-label generation workflow. No additional inference-specific optimization, such as cross-image SAM2 model reuse or prompt batching, was applied.

7. Downstream Validation

Validation with HoVerNet and StarDist

Table 9 reports the results for HoVerNet and StarDist trained with the same pseudo-instance generation protocol. HoVerNet uses foreground and center-offset targets, while StarDist uses object probabilities and radial distances [2,3]. The architecture-specific preparation is described in Section 3.5. These experiments evaluated whether the generated supervision could be transferred across different downstream architectures.

Table 9. Downstream validation using the same pseudo-instance generation protocol with architecture-specific target preparation at a 10% training point budget.

Dataset	Student	Target Preparation	Point Budget	Dice	AJI	DQ	SQ	PQ
MoNuSeg	HoVerNet	Pseudo-instance labels → NP and HV targets	10%	0.7969	0.6233	0.8020	0.7518	0.6036
MoNuSeg	StarDist	Pseudo-instance labels → object and radial targets	10%	0.7792	0.6100	0.8092	0.7141	0.5787
CPM17	HoVerNet	Pseudo-instance labels → NP and HV targets	10%	0.8386	0.6653	0.8325	0.7937	0.6626
CPM17	StarDist	Pseudo-instance labels → object and radial targets	10%	0.8282	0.6437	0.8442	0.7659	0.6480

8. Discussion

8.1. Proposal Coverage as the Observed Bottleneck

The MoNuSeg validation diagnostic contained more ground-truth nuclei without sufficient proposal coverage than nuclei assigned to covered-boundary failure. This pattern supports studying fallback-assisted coverage recovery while retaining an explicit account of false additions. Joint conflict resolution can reassign or ignore overlap pixels, as described in Section 3.4. Nakai and Hotta also reported a recovery-versus-over-segmentation trade-off when supplementing SAM pseudo-labels [19]; our diagnostic examined this trade-off for center-triggered instance additions. The category counts depended on the diagnostic thresholds; boundary refinement may still benefit covered nuclei.

8.2. Dataset-Dependent Effect of Fallback

On MoNuSeg Val6, the no-valid-candidate coverage deficit and fallback ratio were near 37%; fallback produced 541 newly matched instances. On CPM17 Val4, the no-valid-candidate deficit was near 18%; fallback produced 32 new true positives and 69 FP-like instances. These audits link fallback gains to recovered nuclei while also documenting false additions. The benefit depended on whether centers eligible for fallback corresponded to true nuclei. On CoNSeP Val5, the coverage-deficit and fallback ratios were higher (approximately 76%), but only 34.70% of the fallback-added instances were classified as newly matched true positives (Table 7), illustrating that fallback reliability varied across the datasets. The substantially higher coverage-deficit ratio on CoNSeP indicates that a larger proportion of completed centers lacked valid SAM2 proposals under the selected filtering criteria. Together with the relatively low useful fallback fraction, this suggests that limited proposal coverage and unreliable fallback additions may constrain downstream performance. However, these validation audits do not establish the independent contribution of individual stages or specific tissue characteristics to the observed performance gap.

Future work could explore an adaptive gating mechanism based on center confidence and local region consistency to suppress unreliable fallback regions, particularly in cluttered backgrounds. Such a mechanism would require validation-based calibration and independent evaluation before application to held-out test data.

8.3. Future Improvement of Proposal Selection

Future work could evaluate geometry-aware selection among foundation model candidates. Such selection is outside the reported pipeline and would require separate validation.

8.4. Annotation Efficiency and Downstream Use

The experiments evaluated segmentation under a small point annotation budget and the use of the generated supervision in HoVerNet and StarDist. Each model received its native targets after architecture-specific preparation. SAM2 added computational cost during pseudo-label generation, while downstream inference used images alone.

8.5. Limitations

Fallback addresses missing proposals but cannot correct every center detection error, merged SAM2 proposal, displacement, or boundary error. The useful fallback fractions were 47.04% on MoNuSeg, 27.59% on CPM17, and 34.70% on CoNSeP. Many fallback additions were therefore not classified as newly matched true positives.

The evaluation covered three datasets and two downstream architectures, which limits conclusions about other tissues, staining conditions, and model families. Cross-dataset coverage differences are conditional on the present pipeline and cannot establish an intrinsic ordering of dataset difficulty.

The training split required only sparse center point annotations. A small, fully annotated validation subset was used for the predeclared checkpoint, operating-point, and hyperparameter selection procedures described in Section 3.6, while test annotations were used to score frozen outputs. Dense instance masks did not enter training optimization or pseudo-label construction. SAM2 introduced a one-time computational cost during pseudo-label generation but was not required for downstream inference.

9. Conclusions

In this study, we proposed a sparse point-to-pseudo-instance supervision framework for annotation-efficient nuclei instance segmentation. The framework converts sparse center annotations into instance-level pseudo-supervision through curriculum-based center completion, foundation model-guided proposal generation, and fallback-assisted region recovery. The generated pseudo-instance labels can be transformed into architecture-specific training targets for different segmentation models, as demonstrated with HoVerNet and StarDist.

Experiments on MoNuSeg, CPM17, and CoNSeP demonstrated that the proposed framework can support nuclei instance segmentation under limited point annotation budgets. The cross-architecture validation further showed that the generated supervision can be transferred to different downstream segmentation networks. In addition, the validation analysis revealed that fallback recovery improved supervision coverage while introducing additional false regions, indicating that its effectiveness depends on balancing proposal availability, center accuracy, and precision preservation.

Overall, this work demonstrates the feasibility of generating transferable pseudo-instance supervision from sparse point annotations and provides a practical direction for reducing annotation requirements in histopathology image analysis. Future work will investigate more adaptive proposal selection and robust recovery strategies to further improve pseudo-label quality and generalization across diverse tissue conditions.

Author Contributions: Conceptualization, W.H.; methodology, W.H.; software, W.H.; validation, W.H.; formal analysis, W.H.; investigation, W.H. and H.L.; data curation, W.H. and H.L.; visualization, W.H. and H.L.; writing—original draft preparation, W.H.; writing—review and editing, W.H., Y.S., and H.L.; supervision, Y.S.; project administration, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The MoNuSeg, CPM17, and CoNSeP datasets analyzed in this study were obtained from their respective official challenge distribution channels and are subject to the original access terms. The derived data supporting the findings of this study are available from the corresponding author upon reasonable request. The source code, experimental configurations for the proposed method and the locally reproduced baseline methods, and reproduction instructions are publicly available at https://github.com/vermouth26/sparse-point-nuclei-segmentation (accessed on 3 November 2026).

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT (GPT-5.6, OpenAI) solely for Chinese-to-English translation and English-language editing. The authors reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988. [CrossRef]
2. Graham, S.; Vu, Q.D.; Raza, S.E.A.; Azam, A.; Tsang, Y.W.; Kwak, J.T.; Rajpoot, N. Hover-Net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Med. Image Anal.* **2019**, *58*, 101563. [CrossRef]
3. Schmidt, U.; Weigert, M.; Broaddus, C.; Myers, G. Cell Detection with Star-Convex Polygons. In Proceedings of the MICCAI, Granada, Spain, 16–20 September 2018; LNCS 11071. pp. 265–273. [CrossRef]
4. Bearman, A.; Russakovsky, O.; Ferrari, V.; Fei-Fei, L. What's the Point: Semantic Segmentation with Point Supervision. In Proceedings of the ECCV, Amsterdam, The Netherlands, 11–14 October 2016; LNCS 9911. pp. 549–565. [CrossRef]
5. Qu, H.; Wu, P.; Huang, Q.; Yi, J.; Riedlinger, G.M.; De, S.; Metaxas, D.N. Weakly Supervised Deep Nuclei Segmentation using Points Annotation in Histopathology Images. In Proceedings of the 2nd International Conference on Medical Imaging with Deep Learning (MIDL), London, UK, 8–10 July 2019; PMLR. Volume 102, pp. 390–400. Available online: https://proceedings.mlr.press/v102/qu19a.html (accessed on 3 November 2026).
6. Yoo, I.; Yoo, D.; Paeng, K. PseudoEdgeNet: Nuclei Segmentation only with Point Annotations. In Proceedings of the MICCAI, Shenzhen, China, 13–17 October 2019; LNCS 11764. pp. 731–739. [CrossRef]
7. Lin, Y.; Qu, Z.; Chen, H.; Gao, Z.; Li, Y.; Xia, L.; Ma, K.; Zheng, Y.; Cheng, K.-T. Nuclei Segmentation with Point Annotations from Pathology Images via Self-Supervised Learning and Co-Training. *Med. Image Anal.* **2023**, *89*, 102933. [CrossRef]
8. Qu, H.; Wu, P.; Huang, Q.; Yi, J.; Yan, Z.; Li, K.; Riedlinger, G.M.; De, S.; Zhang, S.; Metaxas, D.N. Weakly Supervised Deep Nuclei Segmentation Using Partial Points Annotation in Histopathology Images. *IEEE Trans. Med. Imaging* **2020**, *39*, 3655–3666. [CrossRef]
9. Lin, Y.; Wang, Z.; Zhang, D.; Cheng, K.-T.; Chen, H. BoNuS: Boundary Mining for Nuclei Segmentation With Partial Point Labels. *IEEE Trans. Med. Imaging* **2024**, *43*, 2137–2147. [CrossRef]
10. Mo, Y.; Chen, L.; Zhang, L.; Zhao, Q. Weakly Supervised Nuclei Segmentation with Point-Guided Attention and Self-Supervised Pseudo-Labeling. *Bioengineering* **2025**, *12*, 85. [CrossRef]
11. Pan, X.; Song, S.; Liu, Z.; Wang, H.; Li, L.; Lu, H.; Lan, R.; Luo, X. Weakly supervised nuclei segmentation based on pseudo label correction and uncertainty denoising. *Artif. Intell. Med.* **2025**, *164*, 103113. [CrossRef]
12. Nam, S.; Namgung, H.; Jeong, J.; Luna, M.; Kim, S.; Chikontwe, P.; Park, S.H. InstaSAM: Instance-aware Segment Any Nuclei Model with Point Annotations. In Proceedings of the MICCAI, Marrakesh, Morocco, 6–10 October 2024; LNCS 15004. pp. 232–242. [CrossRef]
13. Cui, C.; Deng, R.; Liu, Q.; Yao, T.; Bao, S.; Remedios, L.W.; Landman, B.A.; Tang, Y.; Huo, Y. All-in-SAM: From Weak Annotation to Pixel-wise Nuclei Segmentation with Prompt-based Finetuning. *J. Phys. Conf. Ser.* **2024**, *2722*, 012012. [CrossRef]
14. 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 ICCV, Paris, France, 1–6 October 2023; pp. 4015–4026. Available online: https://openaccess.thecvf.com/content/ICCV2023/html/Kirillov_Segment_Anything_ICCV_2023_paper.html (accessed on 3 November 2026).
15. Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. SAM 2: Segment Anything in Images and Videos. In Proceedings of the ICLR, Singapore, 24–28 April 2025. Available online: https://openreview.net/forum?id=Ha6RTeWMd0 (accessed on 3 November 2026).
16. Xu, K.; Goetz, L.; Rajpoot, N. On generalisability of segment anything model for nuclear instance segmentation in histology images. *arXiv* **2024**, arXiv:2401.14248. [CrossRef]

17. Archit, A.; Freckmann, L.; Nair, S.; Khalid, N.; Hilt, P.; Rajashekar, V.; Freitag, M.; Teuber, C.; Spitzner, M.; Tapia Contreras, C.; et al. Segment Anything for Microscopy. *Nat. Methods* **2025**, *22*, 579–591. [CrossRef] [PubMed]
18. Marks, M.; Israel, U.; Dilip, R.; Li, Q.; Yu, C.; Laubscher, E.; Iqbal, A.; Pradhan, E.; Ates, A.; Abt, M.; et al. CellSAM: A foundation model for cell segmentation. *Nat. Methods* **2025**, *22*, 2585–2593. [CrossRef]
19. Nakai, R.; Hotta, K. Unsupervised Nuclei Segmentation by Improving Pseudo Labels from Segment Anything Model. In Proceedings of the ICCV Workshops, Honolulu, HI, USA, 19–20 October 2025; pp. 1036–1044. [CrossRef]
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the CVPR, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
21. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the MICCAI, Munich, Germany, 5–9 October 2015; LNCS 9351. pp. 234–241. [CrossRef]
22. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the CVPR, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
23. Bengio, Y.; Louradour, J.; Collobert, R.; Weston, J. Curriculum Learning. In Proceedings of the ICML, Montreal, QC, Canada, 14–18 June 2009; pp. 41–48. [CrossRef]
24. Kumar, N.; Verma, R.; Sharma, S.; Bhargava, S.; Vahadane, A.; Sethi, A. A Dataset and a Technique for Generalized Nuclear Segmentation for Computational Pathology. *IEEE Trans. Med. Imaging* **2017**, *36*, 1550–1560. [CrossRef] [PubMed]
25. Kumar, N.; Verma, R.; Anand, D.; Zhou, Y.; Onder, O.F.; Tsougenis, E.; Chen, H.; Heng, P.-A.; Li, J.; Hu, Z.; et al. A Multi-Organ Nucleus Segmentation Challenge. *IEEE Trans. Med. Imaging* **2020**, *39*, 1380–1391. [CrossRef]
26. Vu, Q.D.; Graham, S.; Kurc, T.; To, M.N.N.; Shaban, M.; Qaiser, T.; Koohbanani, N.A.; Khurram, S.A.; Kalpathy-Cramer, J.; Zhao, T.; et al. Methods for Segmentation and Classification of Digital Microscopy Tissue Images. *Front. Bioeng. Biotechnol.* **2019**, *7*, 53. [CrossRef]
27. Liu, W.; He, Q.; He, X. Weakly Supervised Nuclei Segmentation via Instance Learning. In Proceedings of the 2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI), Kolkata, India, 28–31 March 2022; pp. 1–5. [CrossRef]
28. Pan, X.; Hou, F.; Liu, Z.; Feng, S.; Lan, R. EOFD-Net: Edge Optimization and Feature Denoising for Weakly Supervised Deep Nuclei Segmentation with Point Annotations. In Proceedings of the ICASSP, Seoul, Republic of Korea, 14–19 April 2024; pp. 2180–2184. [CrossRef]
29. Wang, Z.; Zhang, Y.; Wang, Y.; Cai, L.; Zhang, Y. Dynamic Pseudo Label Optimization in Point-Supervised Nuclei Segmentation. In Proceedings of the MICCAI, Marrakesh, Morocco, 6–10 October 2024; LNCS 15008. pp. 220–230. [CrossRef]
30. Wan, M.; Lin, L.; Wu, X.; Zhong, J.; Shi, P. A constraint-attention enhancement network for clinical nuclei segmentation. *Biomed. Signal Process. Control* **2025**, *109*, 107917. [CrossRef]
31. Chen, Y.; Huang, Q.; Wang, Z.; Geng, M. Sparsely guided adaptive pseudo-supervised learning for nucleus segmentation. *Expert Syst. Appl.* **2026**, *319*, 132018. [CrossRef]
32. Kirillov, A.; He, K.; Girshick, R.; Rother, C.; Dollár, P. Panoptic Segmentation. In Proceedings of the CVPR, Long Beach, CA, USA, 16–20 June 2019; pp. 9404–9413. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.