

Harnessing SSL for Segmentation in 3D Microscopy with Noisy Labels and Hard Patches

Tony Xu^{1,2}, Ahmadreza Attarpour^{1,2,3}, Shruti Patel^{1,2}, Fengqing Yu⁴,
Matthew W. Rozak^{1,2}, Bojana Stefanovic^{1,2}, Anne L. Martel^{1,2}, and
Maged Goubran^{1,2}

¹ University of Toronto, Toronto ON, Canada

² Sunnybrook Research Institute, Toronto ON, Canada

³ University Health Network, Toronto ON, Canada

⁴ California Institute of Technology, Pasadena CA, USA

tonylt.xu@mail.utoronto.ca, maged.goubran@utoronto.ca

Abstract. Segmentation in 3D microscopy is challenging due to hard (to segment) patches, noisy labels resulting from the use of semi-automated labeling methods, image artifacts, off-target fluorescence, and lack of labeled data because of high labeling effort. In this work, we introduce a novel unified method to tackle these issues simultaneously in 3D microscopy. First, we introduce μ 3DINO, a 3D model pretrained on an ultra-large multimodal dataset of over 2 million microscopy volumes. We then create μ DivSeg, a segmentation pipeline that uses pretrained weights to detect noisy labels and hard patches to guide and correct segmentation training. We evaluate our methods on a toy dataset and 4 real-world datasets from light-sheet and two-photon microscopy with a variety of markers, in comparison to two state-of-the-art (SOTA) pipelines. Our methods outperform SOTA techniques on increasing levels of synthetic label perturbations and real-world data with diverse distributions.

Keywords: 3D Microscopy · Self-supervised learning · Noisy labels

1 Introduction

Recent advances in imaging hardware, computational resources, and data storage have enabled the acquisition of exceptionally large and high-resolution 3D microscopy datasets. Modalities such as two-photon excitation microscopy (TPEM) allow real-time, in vivo imaging, while light-sheet fluorescence microscopy (LSFM) combined with tissue clearing enables whole-brain imaging at cellular resolution. Confocal microscopy is capable of live-cell microscopic imaging, which can also yield large quantities of data. Together, such technological advances help facilitate the study of disease and injury in animal models with unprecedented detail, with single images often on the order of *teravoxels* in scale.

Image segmentation is a key task for analyzing 3D microscopy and is used to delineate structures like neurons and vessels, provide insights on the structural and functional connectivity of animal brains [18, 30], and perform tasks like cell

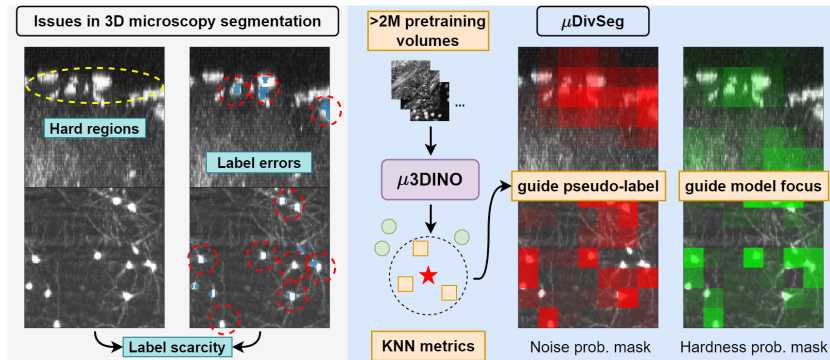


Fig. 1: To handle label scarcity, label error (red circle), and hard regions (yellow ellipse) in 3D microscopy segmentation, we pretrain μ 3DINO, and use it in our novel pipeline, μ DivSeg, to obtain KNN clustering-based probability masks to increase pseudo-labelling in noisy areas, and loss weighting in hard regions.

tracking [1, 27]. However, the sheer scale of 3D microscopy data makes manual segmentation infeasible. Deep learning (DL) has emerged as an automated approach for image segmentation in microscopy [3, 14, 35, 38], but its success typically depends on the availability of high-quality pixel-level annotations. In practice, generating dense 3D labels is prohibitively time-consuming, and annotations are often produced using semi-automated tools such as Ilastik [5]. With these tools, labeling can still take *days* for TPEM data and *weeks* for *LSFM* volumes, making them both scarce, and often noisy or inconsistent.

Thus, we identify three main challenges to the application of DL to microscopy image segmentation (Fig. 1). **1.** DL models can struggle with generalization due to the scarcity of detailed labels. **2.** DL models can be confounded by poor-quality labels that often contain (structured) noise. **3.** DL models can also struggle with “hard regions” that contain off-target fluorescence, bright objects, or imaging artifacts that are difficult to delineate from the true structure of interest. In 3D microscopy, label scarcity has been tackled using self-supervised learning (SSL) approaches [19, 28, 42, 44]. However, these methods are limited by being trained on a single modality, having limited pretraining dataset sizes, having weights that are not publicly released, or using simple pretext tasks that do not scale well. Many works applied to 2D image classification have tackled label noise [16, 21, 47, 49]. Some works have also been applied to medical image segmentation [22, 49] in modalities like CT and MRI. In such modalities, label noise is typically introduced by degrading label boundaries (*i.e.* via erosion or dilation). These methods may not translate directly to microscopy, where label noise often has different characteristics. Since segmentation in microscopy typically involves detecting many instances of the same structure, false positives (FPs) commonly arise from bright artifact objects, while false negatives (FNs) are more commonly caused by entire missing instances. Finally, methods like hard-example mining [34] or reweighting methods [24, 45] have been proposed

to tackle hard example learning in 2D image classification and object detection, but only a few works have explored both noisy labels and hard examples [11]. To our knowledge, the challenges of label scarcity, label noise, and hard regions are underexplored in 3D microscopy segmentation, leaving a pressing need for a framework that can operate effectively in this difficult training regime.

We propose a two-pronged approach to tackle these challenges. First, we introduce μ 3DINO, a Vision Transformer (ViT) [10] pretrained on the largest 3D microscopy dataset, comprised of over 2 million multimodal image volumes from LSFM, TPEM, and confocal microscopy. Then, we introduce **Divide and Segment** for microscopy (μ DivSeg), a segmentation method that *builds* upon and takes advantage of pretrained weights to perform noisy label and hard patch detection while self-correcting *online* during training. μ DivSeg makes use of the strong content-based retrieval (CBR) capabilities of SSL models to detect noisy and hard examples using a data-centric approach, which allows it to learn and often improve on noisy annotations without the need for cleanly labeled subsets or extensive hyperparameter tuning [50]. To summarize our contributions, we:

- Collect the largest multimodal 3D microscopy dataset comprised of over 2 million images, and pretrain a general-purpose ViT for 3D microscopy.
- Devise a self-correcting segmentation pipeline that performs patch-level noisy label learning and hard example detection simultaneously using an SSL model and cluster-based approaches.

We evaluate our methods on four real-world datasets with a variety of structures in TPEM and LSFM, and find that initializing with μ 3DINO performs 5.8% and 3.9% better than random initialization when training on a small subset of each dataset (10% or 2%) and training on the full (100%) dataset respectively. This further increases by 1.1% and 1.9% when using μ DivSeg instead of a standard segmentation pipeline. Compared to the next best SOTA baseline method that we reimplement for 3D microscopy, μ DivSeg performs an average of 6.4% better at 100% of data. We qualitatively evaluate μ DivSeg to show how results on datasets with noisier ground truth may be misleading, and that our method can improve *beyond* the ground truth more effectively than other methods. We also apply μ DivSeg to NIS3D [48], a clean toy dataset with LSFM and confocal modalities, where we show that it is more robust to increasing amounts of instance-based synthetic noise. Overall, we present the first large-scale multimodal pretrained model for 3D microscopy data and unified pipeline that leverages noisy label and hard patch detection to improve 3D segmentation in this difficult regime. Code for μ DivSeg and weights for μ 3DINO can be found at <https://github.com/AICONSlab/microDivSeg>.

2 Related Works

SSL in Microscopy: There have been a few studies exploring SSL in TPEM [28, 44], however the datasets used were unimodal and relatively small. Prior work explored using the shuffling pretext task for SSL in microscopy [19], though the

dataset was also small in scale, and it used a relatively simple pretext task that has been since superseded by more scalable multi-view representation methods [12]. Many studies have explored 2D representation learning in histopathology [7, 9, 26, 40, 41] and 2D microscopy [2, 13] and 3D SSL has been explored in medical imaging [36, 37, 43], with the SELMA3D challenge [8] exploring 3D SSL for LSFM. We build upon this line of work by combining LSFM, TPEM and confocal modalities, yielding the largest multimodal 3D microscopy image dataset, and pretrain and publicly release a general-purpose 3D model for microscopy.

Noisy Label and Hard Example Learning: Noisy label learning has been explored extensively in 2D image classification [16, 21, 47, 49], where methods have improved resilience to noisy labels by exploiting the tendency of DL models to fit clean labels before noisy ones. Other works have extended such methods to segmentation using pixel-level learning-based noise scores [22]. Semi-supervised segmentation is related to this regime, such as GuidedNet [46] which uses unlabeled data and co-training to improve label efficiency in 3D CT organ segmentation. However, no methods have been evaluated under synthetic label noise that is based on FP or FN *instances* rather than boundary noise. Noisy label learning when a dataset contains hard samples has been explored empirically on a small toy dataset [11]. A meta-learning approach, L2B [49], used a small, clean “meta set” to automatically weigh real labels *vs.* model generated pseudo-labels to improve segmentation training under label noise. By not constraining weights on these labels to sum to 1, this method can overweigh key regions, making it an implicit hard example learner. However, in L2B’s segmentation implementation, the labels of the meta-dataset must be clean, which can be difficult in microscopy. Preliminary work exists in noisy label [23], and hard example [20] settings in microscopy segmentation, though these issues were not explicitly addressed during training. Thus, limited works have explored noisy label and hard example learning methods in 3D microscopy segmentation, and none have investigated how to integrate such methods with SSL models.

3 Method

In this section, we introduce our methods (Fig. 2). We discuss μ 3DINO (Sec. 3.1), how we perform patch-level noisy label and hard region detection (Sec. 3.2), and how we incorporate these methods together to create μ DivSeg (Sec. 3.3).

3.1 μ 3DINO

Considering the success of DINOv2’s [29] application to a variety of medical tasks, including histopathology [7, 40], and adaptation to 3D medical images, we used the 3DINO [43] method to pretrain μ 3DINO. This ViT was trained on over 2 million unlabeled microscopy volumes (Sec. 4.1). We denote our final model as $f(\cdot)$. When applying μ 3DINO to a downstream labeled dataset, $\mathcal{D} = (X, Y)$, we sampled a single volume and segmentation mask label pair, (x, y) , $x \in \mathbb{R}^{S \times S \times S}$ with side length S . We extracted embeddings using our pretrained model, $f(x) =$

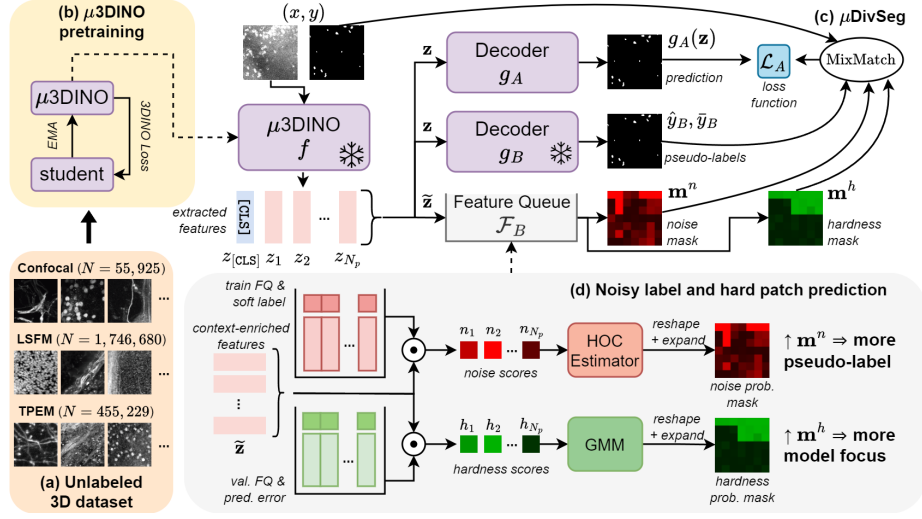


Fig. 2: Overview of main methods. An unlabeled multimodal pretraining dataset with over 2 million 3D microscopy volumes (a) was used to pretrain $\mu 3DINO$ (b). $\mu DivSeg$ transferred and froze $\mu 3DINO$ to extract features which were used for segmentation and to generate masks predicting patch noise and hardness levels that guide segmentation (c). A feature queue estimated noise and hardness masks during training (d). In this figure, model A (g_A) is being trained and model B (g_B) is used to pseudo-label.

$z = \{z_{[CLS]}, z_1, \dots, z_{N_p}\} \in \mathbb{R}^{(N_p+1) \times D}$, with feature size D and consisting of N_p patch features per image and a image-level representation, $z_{[CLS]}$.

3.2 Noisy Label and Hard Patch Detection in Segmentation

We took advantage of $\mu 3DINO$ to perform noisy label and hard patch detection in segmentation tasks. Specifically, we obtained a patch-level “noise probability” and “hardness probability” that could be used to inform training.

Noisy Label Detection: “Clusterability” defines a model’s ability to group features with the same true class together. Exploiting this property for $\mu 3DINO$, we detected label noise in a data-centric manner using SimiFeat [50] (Suppl. Material Sec. 1). SimiFeat was formulated for classification, so to adapt this method for segmentation, we converted segmentation labels into a soft classification label. To simplify notation because our datasets are all single-class, we define the one-hot label of an image patch as $y_i \in \{0, 1\}^{2 \times P \times P \times P}$ for a patch of size P and $i \in \mathcal{N}$, $|\mathcal{N}| = N$ for a dataset containing N total patches. To convert y_i into a soft classification label, we used a clipped threshold: $y_i^c = \min(s_i / Q_{0.9}(\{s_i\}_{i \in \mathcal{N}}), 1)$, where $s_i = \sum_p y_i^{(1)}[p]$ is the sum of all pixels, enumerated by p , in the foreground class of the patch label, and $Q_a(\cdot)$ obtains the a quantile.

We then obtained the KNN estimate of the soft patch label, $\hat{y}_i$, by performing a weighted average along the soft labels of the K^n nearest neighbors of the

patch in feature space based on cosine similarity. The noise score of the patch label was obtained by comparing it to its KNN estimate using Cross Entropy (CE), $n_i = H(\hat{y}_i, y_i^c)$, where a high score indicates that y_i^c and $\hat{y}_i$ are dissimilar. Intuitively, this would indicate patch i clustered near patches with dissimilar soft labels to its own. To convert from patch-level noise scores to probabilities, $m_i^n \in [0, 1]$, we employed the Higher-Order Consensus (HOC) estimator [51] (additional details in Suppl. Material Sec. 2).

Hard Patch Detection: We defined “hard” patches as those with a high prediction error. To decouple hard patch from noisy label detection, we used the CBR capabilities of μ 3DINO to detect patches in the training set that cluster near those with high error in the *validation* set. Prediction error per patch was computed by averaging the pixel-level CE loss between the model prediction and the patch label. Then, the patch-level “hardness score” for a patch, h_i , in the training set was the weighted average of the prediction errors of the K^h nearest validation patches. Again, similarity was defined in feature space using cosine similarity. Intuitively, a high hardness score would indicate that a training patch clustered near validation patches containing structures that were difficult to segment. Notably in real-world data, the validation set is not necessarily free of label noise. Thus, a patch with high prediction error could also have a noisy label rather than being a difficult example. While it is possible to detect and remove noisily labeled patches in the validation set using our method, we found it was not necessary when using a sufficiently high K^h . To convert from patch-level hardness scores to probabilities, $m_i^h \in [0, 1]$, we fit a 2-component Gaussian Mixture Model (GMM) [31] to values of h_i in the training set.

3.3 μ DivSeg

Our novel method, μ DivSeg, integrated noisy label and hard patch detection to guide and improve segmentation training *online* on 3D microscopy. Our pipeline was inspired by DivideMix [21], which self-corrects noisy classification labels.

Feature Queue: We used a feature queue with length Q to compute noise and hardness probabilities online. This running queue tracked the most recent training patch features and their respective soft labels, as well as the most recent validation patch features and their prediction errors. With sufficiently large Q , KNN-based noise and hardness scores can be computed against the features present in the queue. The queue can also be used to recompute quantiles and GMMs needed for detection at the start of every epoch.

Online Hard Patch Focusing and Label Correction: To apply μ 3DINO to segmentation, we froze it and passed its extracted patch features through a segmentation decoder, $g(\cdot)$, to predict pixel-level segmentations. Like DivideMix [21], we trained two decoders, g_A and g_B , and maintained two feature queues $\mathcal{F}_A$ and $\mathcal{F}_B$. g_A and $\mathcal{F}_A$ were used to generate pseudo-labels and predict label noise and patch hardness to train g_B , and vice versa to reduce biases. To start, we warmed up both models for a short period by training them using a standard segmentation pipeline against the real labels. This warm up period also permits us to fill the feature queues. After the warm up period, g_A and g_B

were trained one-by-one. To train g_A , features were extracted from an image, $\mathbf{z} = f(x)$. Then we used the [CLS] token as a “context vector” to inform the patch features. This was useful for when objects of interest were larger than patches (*e.g.* large vessels), and allowed context of the full image to inform KNN comparisons. This was done simply by summing every patch token with the [CLS] token: $\tilde{z}_j = z_j + z_{[\text{CLS}]}, j \in \{1, \dots, N_p\}$. These context-enriched features, $\tilde{\mathbf{z}} \in \mathbb{R}^{N_p \times D}$, were compared against $\mathcal{F}_B$ to obtain noise and hardness probabilities per patch, $\tilde{\mathbf{m}}^n, \tilde{\mathbf{m}}^h \in \mathbb{R}^{N_p}$ that we reshaped and expanded back to the same size as the input image to create probability masks $\mathbf{m}^n, \mathbf{m}^h \in \mathbb{R}^{S \times S \times S}$. We froze g_B and used it to pseudo-label the input, $\hat{y}_B = \sigma(g_B(\mathbf{z})/\tau), \hat{y}_B \in \mathbb{R}^{2 \times S \times S \times S}$, where g_B returns 2-class pixel-level logits, $\sigma(\cdot)$ is softmax, and τ is a sharpening temperature.

We used $\mathbf{m}^n$ and $\mathbf{m}^h$ to guide the training of g_A using two losses. Firstly, we computed a pixel-level CE loss, $\mathcal{L}_{CE}$. We weighted this loss using $\mathbf{m}^n$ to trade off focus on real labels (where probability is low), versus pseudo-labels (where probability is high). We weighted the loss function using $\mathbf{m}^h$ to increase focus on hard patches in the image, resulting in Eq. (1) to train g_A .

$$\mathcal{L}_{CE,A} = H(g_A(\mathbf{z}), y) \cdot (1 - \mathbf{m}^n) \cdot (1 + \lambda^h \mathbf{m}^h) + H(g_A(\mathbf{z}), \hat{y}_B) \cdot \mathbf{m}^n \quad (1)$$

Where λ^h is a parameter to control the enhancement strength of $\mathbf{m}^h$ and all operations are performed element-wise. Note, we did not enhance the pseudo-label CE loss because we expected pseudo-labels from hard patches to potentially be lower quality and confound training. The second loss is a confidence-based image-level Dice loss. Rather than weighting the loss function, we used $\mathbf{m}^n$ to weigh the labels themselves as in equation Eq. (2).

$$\bar{y}_B = \text{Sharpen}(y \cdot (1 - \mathbf{m}^n) + \sigma(g_B(\mathbf{z})) \cdot \mathbf{m}^n, \tau) \quad (2)$$

Where $\text{Sharpen}(a, \tau)$ takes $a^{1/\tau}$ element-wise, and normalizes by dividing by the sum over the class dimension per pixel in a . The Dice loss was then Eq. (3).

$$\mathcal{L}_{DSC,A} = DSC(\sigma(g_A(\mathbf{z})), \bar{y}_B) \quad (3)$$

And the overall loss function for training g_A was simply the sum of the CE and Dice-based losses: $\mathcal{L}_A = \mathcal{L}_{CE,A} + \mathcal{L}_{DSC,A}$. Before loss computation, we applied MixMatch [6] in-place to the computed tensors ($\mathbf{z}, \mathbf{m}^n, \mathbf{m}^h, y, \bar{y}_B$, and $\hat{y}_B$). For each sample, this method linearly interpolated these tensors with those of another randomly selected batch element using a mixing coefficient drawn from a β distribution, $\text{Beta}(\alpha, \alpha)$. After g_A was trained for an iteration, the extracted patch features, $\mathbf{z}$ and their soft labels were added to the feature queue, $\mathcal{F}_B$. After a training epoch, we froze g_A and used it along with $\mathcal{F}_A$ to train g_B similarly. We performed validation separately on g_A and g_B after both models are trained for an epoch, where we added features and their patch-level prediction errors on the validation set to their respective queues. Since μDivSeg uses a data-centric approach for noisy label detection and predicts hard patches using a training-based error metric on a separate validation set, these metrics are decoupled. By

allowing both to vary continuously to adapt segmentation training on-the-fly, we aimed to simultaneously tackle label noise and hard regions in 3D microscopy. Pseudocode is provided in Suppl. Material Sec. 3.

4 Experiments and Results

In this section we describe our pretraining and downstream segmentation datasets (Sec. 4.1), μ DivSeg (Sec. 4.2) and baseline (Sec. 4.3) setup and implementation, and real-world (Sec. 4.4) and toy dataset (Sec. 4.5) experiments and results. Experiments ablating the effect of our key modules (noise mask, hardness mask, SSL initialization) and other parameters can be found in Suppl. Material Sec. 4, and visualizations of noisy label and hard patch probability masks and example detected noisy and hard patches can be found in Suppl. Material Sec. 5 and 6.

4.1 Datasets

Pretraining Dataset: To pretrain μ 3DINO, we gathered an ultra-large, multimodal 3D dataset consisting of unlabeled in-house data, data from collaborators, and public resources including the Brain Image Library [17] and the SELMA3D challenge dataset [8]. This dataset contained a variety of fluorescently labeled markers and cells or cellular components across a wide range of resolutions and field of views from whole-brain data to regional scans. Volumes of size 128^3 were extracted from these datasets, resulting in 1,746,680 volumes from LSFM, 455,229 from TPEM, and 55,925 from confocal microscopy, for a total of 2,257,834 volumes. Additional information on these datasets including links to public resources can be found in Suppl. Material Sec. 7.

Downstream Real-World Datasets: We used four real-world datasets to evaluate our method. First, we employed TPEM vessel (TPEM_V) and neuron (TPEM_N) datasets from 25 Thy1-YFP mice with lectin stain to label the vasculature, with a train/val/test split of 24/10/8 volumes of size $256 \times 512 \times 512$ voxels [33]. We used an LSFM axon (LSFM_A) dataset from 6 mice injected with AAV8-CaMKIIa-eYFP-NRN to label axonal projections, with a split of 694/21/8 volumes of size $256 \times 256 \times 256$ voxels [4]. We employed an LSFM neuron (LSFM_N) dataset from 18 Tg TRAP2-Ai9 mice with the c-Fos promoter, with a split of 266/53/80 volumes of size $512 \times 512 \times 512$ voxels [3]. All splits were generated on the subject level. Notably, since these are real-world large 3D microscopy datasets, all of them rely on “silver”-standard ground truth generated by semi-automated tools, meaning they contain label noise, even in validation and test sets. TPEM_V and LSFM_N, however have *noisier* test labels, which can make it difficult to know if differences in quantitative metrics are due to improved performance or overfitting to noisy labels. Nonetheless, we report results on all datasets and perform qualitative analyses to evaluate our pipeline *beyond* the ground truth.

Downstream Toy Datasets: To evaluate the benefit of μ DivSeg in a controlled manner, we needed a clean, fully annotated dataset. We employed the publicly

available NIS3D dataset [48] as it fulfills these requirements. NIS3D only consists of a pair each of annotated *Drosophila* (NIS3D_D), *MusMusculus* (NIS3D_M) and *Zebrafish* (NIS3D_Z) embryo images. Importantly, NIS3D is also annotated for labeler confidence, which we used as a surrogate for “hardness”. For each dataset, we used one of the images as training, and the other as validation. To add synthetic label noise to NIS3D training images, we randomly masked out *instances* of cells to introduce FNs, and thresholded the image based on intensity to introduce FPs. We did this for three levels of noise, L_1 , L_2 , and L_3 . Using confidence annotations along with the clean segmentation label, we obtained quantitative metrics for our pipeline’s ability to detect noisy labels and hard patches. Results from this evaluation and more details and visualizations on synthetic noise can be found in Suppl. Material Sec. 8. NIS3D_Z contains more low confidence cells, and so in this dataset, we sampled FNs in two ways: 1. fully randomly, and 2. by oversampling cells with low annotator confidence ($L_{lc,1}$, $L_{lc,2}$, $L_{lc,3}$). This enabled us to evaluate how methods perform when FNs in the ground truth are also hard examples. Several images in NIS3D contain large “blobs” of unconfident regions. We ignored these regions when computing metrics, and instead investigated performance qualitatively. Since these are present in both NIS3D_M images, we did not add synthetic label noise to this dataset.

4.2 μ DivSeg Setup & Implementation

Our μ 3DINO encoder (f) was a ViT-Large [10] with an image and patch size of 112^3 and 16^3 . For noisy label and hard patch detection, we permitted patch sizes that are a multiple of the ViT patch size by averaging patch-level features (*e.g.* a patch size of 32^3 averaged a cube of 2^3 patch features). For TPED and LSFMA, we used a patch size of 32^3 ; for LSFMA_N and NIS3D, we used a patch size of 16^3 . These were chosen to ensure each patch has sufficient context to perform detection. We froze μ 3DINO and added a UNETR-like decoder (g) [15] to obtain pixel-level segmentations. Rather than splitting data to train g_A and g_B , we used randomly cropped and augmented versions of the same dataset. In practice, this sufficiently diverged g_A and g_B to reduce pseudo-label bias. On real-world data, we used a learning rate (LR) of $5e-4$, a batch size of 8, and we trained for 30,000 iterations using AdamW [25]. The standard segmentation warm up period was 2,000 iterations. On NIS3D, we adjusted the LR to $1e-4$, training iterations to 2,500, and warm up period to 100 iterations. All experiments were performed on an NVidia A100-40GB GPU, and real-world experiments took ~ 12 hours. Additional hyperparameters are in Suppl. Material Sec. 9.

4.3 Baseline Setup & Implementation

We compared against a standard segmentation pipeline with a randomly initialized ViT and the same UNETR decoder as μ DivSeg, where the ViT was trained (‘Rand.’), and UNet [32] (‘UNet’). We also initialized with μ 3DINO, froze it, and trained with a standard pipeline (‘ μ 3DINO’). These experiments used the same relevant parameters as μ DivSeg, except the LR was $1e-4$. We then use

Table 1: Quantitative results on TPEN neuron (TPEN_N) and vessel (TPEN_V), and LSFMA axon (LSFM_A) and neuron (LSFM_N) real-world datasets. Data sizes in percentages of full dataset (# of volumes). Performance in mean (std.) of Dice scores (%) across all volumes in the test dataset. * μ DivSeg performs significantly better ($p < 0.05$) at that dataset size based on a Wilcoxon signed rank test on test volumes.

Data size	TPEN _N			TPEN _V			LSFM _A			LSFM _N		
	10% (2)	25% (6)	100% (24)	10% (2)	25% (6)	100% (24)	2% (14)	5% (35)	100% (694)	10% (27)	25% (66)	100% (266)
Rand.	67.4* (8.5)	68.2 (13.4)	73.2* (8.7)	64.6* (22.1)	74.9* (12.5)	77.4* (10.9)	65.1* (4.1)	66.6* (4.3)	71.2* (3.1)	72.6* (19.4)	73.9* (18.6)	74.3* (18.3)
UNet	68.5* (6.8)	70.6 (12.2)	73.4* (8.4)	65.8* (19.8)	76.3* (10.2)	77.3* (11.3)	63.8* (4.9)	66.1* (4.4)	73.8* (2.7)	73.6* (18.8)	74.2* (18.3)	74.5* (18.2)
μ SAM	65.9* (8.2)	69.1* (9.4)	70.7* (10.3)	52.1* (22.8)	62.4* (18.6)	64.1* (20.3)	68.7 (3.6)	73.5 (3.2)	74.7 (2.7)	69.9* (19.6)	71.5* (18.3)	71.8* (18.1)
GuidedNet	58.4* (8.9)	64.0* (13.7)	70.0* (8.2)	56.6* (23.7)	64.1* (20.5)	75.5* (12.0)	66.7* (3.9)	71.6 (3.3)	71.3* (2.8)	68.9* (18.9)	69.8* (18.1)	69.7* (18.1)
GuidedNet _{μ}	—	—	74.6* (6.4)	—	—	74.3* (13.6)	—	—	76.0 (3.0)	—	—	69.8* (18.0)
L2B	59.0* (7.8)	63.3* (7.7)	65.1* (6.2)	37.9* (20.8)	51.5* (21.6)	73.0* (13.2)	71.5 (3.0)	74.1 (2.1)	74.7 (2.1)	70.4* (18.1)	69.7* (17.7)	68.1* (18.1)
L2B _{μ}	—	—	71.7* (4.8)	—	—	72.4* (17.6)	—	—	74.5* (2.2)	—	—	68.1* (18.0)
μ 3DINO	70.9 (5.9)	73.1 (10.2)	77.1* (7.1)	70.0* (14.9)	78.1 (9.7)	81.2* (7.6)	69.9 (2.3)	71.6 (4.3)	73.3 (3.5)	74.5 (18.9)	75.3 (18.4)	76.1 (18.5)
μ DivSeg	71.5 (5.6)	74.9 (8.0)	79.5 (4.1)	73.5 (12.2)	80.0 (8.5)	82.0 (7.4)	69.4 (3.0)	72.9 (3.6)	76.8 (3.5)	74.1 (18.8)	74.8 (18.5)	75.2 (18.4)

the same pipeline with $\text{LR}=5e-4$ to train the simple 3D wrapper on the 2D μ SAM model [2] to compare against 2D pretrained encoders (‘ μ SAM’).

Since no related baselines existed for 3D microscopy, we reimplemented and tuned two SOTA methods with public code. For fairness, these baselines used the same randomly initialized ViT and data loading code. To compare against semi-supervised segmentation in 3D medical imaging, we used GuidedNet [46]. We trained this method for 30,000 iterations, used a labeled and unlabeled batch size of 4, a LR of $1e-2$, and the SGD optimizer. When training on 100% of training data available for a downstream dataset, the labeled and unlabeled datasets were two randomly augmented versions of the full dataset. Otherwise, unused data formed the unlabeled dataset. For a meta-learning-based self-correction method that also permits hard region focusing, we adapted L2B [49] to 3D microscopy data. L2B was trained for 30,000 iterations, used a batch size and meta-set batch size of 3, a LR of $1e-2$ and the SGD optimizer. Both SOTA comparisons used the same optimizer as their original implementations, with a logarithmic search on LR to determine the best value. To provide a fair comparison against μ DivSeg, which detects hard patches using the validation set, we employed a randomly augmented version of the validation set as the “meta set”. In the original implementation, L2B trained their model *directly* on the meta set, but we did not do this for fairness. GuidedNet used the same parameters on NIS3D and trained for 2,500 iterations. L2B was also trained for 2,500 iterations, used a LR of $1e-3$, and required a short warm up period of about 250 iterations of standard segmentation training to avoid collapse. To conduct methods-focused comparisons, we also train baselines with μ 3DINO initialization with a frozen ViT encoder (‘GuidedNet _{μ} ’ and ‘L2B _{μ} ’).

4.4 Real-World Dataset Experiments

Quantitative Analysis: We trained and evaluated our method on real-world datasets at three different dataset sizes and used the best performing model on the validation set to infer on the test set. We found μ 3DINO improved over Rand. with an average of 5.8% and 3.9% Dice similarity coefficient improvement at the

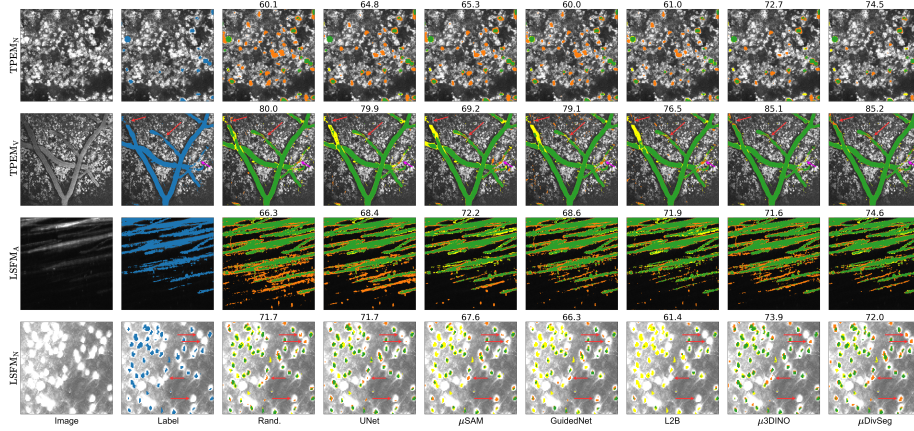


Fig. 3: Qualitative results on real-world test datasets. Displays image, label overlaid in blue, predictions with TP (green), FP (orange), and FN (yellow). All visualizations are maximum intensity projections (MIPs) with varying depth; values above predictions are Dice scores (%) on the full volume. In TPEN_V , red arrows show errors in ground truth and magenta arrows show where μDivSeg predicted the correct vessel radius where most baselines overpredicted off-target fluorescence. In LSFM_N , red arrows show large neurons that are undersegmented by other methods that μDivSeg detects. Our method comparatively reduces FPs and FNs and often performs better than ground truth.

smallest and full dataset sizes, respectively (Tab. 1). We found μSAM underperformed, likely because it was primarily pretrained for cell/nuclear segmentation, and does not have cross-slice interaction due to being a 2D encoder. Notably, we found SSL to be more beneficial at smaller dataset sizes, while μDivSeg performed better with a moderate amount of data. This is likely because μDivSeg used KNN-based predictions to compute noise and hardness masks, as well as pseudo-labeling to correct noisy labels, which both improve with data size. When scaling to the full dataset, μDivSeg outperformed all other baselines on datasets with cleaner test sets (TPEN_N , LSFM_A), with an average of 2.9% Dice improvement over the next best method. On the datasets with test sets that have noisier labels (TPEN_V , LSFM_N), the benefits relative to a standard segmentation pipeline were more subdued (1.0% better, 1.2% worse at 100% data). However, we expect this was due to poor ground truth quality, and explore this qualitatively below. Comparing methods, we found that μDivSeg was much better than baselines (avg. +6.4% *vs.* GuidedNet $_{\mu}$, +9.4% *vs.* L2B $_{\mu}$ at 100% data). These SOTA baselines severely underperformed when using smaller datasets, which was partially recovered when using μ3DINO initialization (avg. +2.9% for GuidedNet, +2.3% for L2B). For L2B, this degradation likely stemmed from low-quality pseudo-labels compounded by unreliable reweighting due to label noise. GuidedNet performed slightly better in the low-data regime, but its lack of reweighting meant that noisy pseudo-labels were incorporated without correction. As a result, both methods with random initialization performed worse

Table 2: Quantitative results on NIS3D. Noise levels are no synthetic noise (Clean), increasing noise (L_1 , L_2 , L_3), and increasing noise oversampling low-confidence cells ($L_{lc,1}$, $L_{lc,2}$, $L_{lc,3}$). Performance in mean (std.) of Dice scores (%) across three runs.

Noise level	NIS3D _Z							NIS3D _D				NIS3D _M
	Clean	L_1	L_2	L_3	$L_{lc,1}$	$L_{lc,2}$	$L_{lc,3}$	Clean	L_1	L_2	L_3	Clean
GuidedNet _{μ}	Best 78.3 (0.7)	77.9 (1.3)	76.3 (1.6)	72.5 (1.6)	76.9 (0.7)	76.2 (1.6)	72.4 (1.8)	67.9 (3.9)	68.4 (3.6)	69.7 (3.8)	59.3 (9.7)	40.1 (1.3)
	Last 79.1 (0.4)	77.9 (1.3)	74.9 (2.3)	68.5 (2.7)	76.8 (0.8)	74.8 (2.8)	68.5 (1.0)	65.8 (4.0)	64.3 (3.9)	62.1 (10.7)	28.0 (15.0)	40.5 (1.1)
L2B _{μ}	Best 80.3 (0.8)	74.4 (3.4)	74.1 (6.1)	67.1 (7.6)	76.6 (3.8)	73.9 (2.3)	71.9 (10.7)	78.8 (2.8)	79.4 (3.1)	80.0 (3.1)	82.1 (0.7)	36.7 (1.3)
	Last 80.4 (0.9)	71.0 (10.8)	70.4 (9.8)	66.6 (8.1)	77.5 (2.3)	75.4 (4.5)	72.0 (10.2)	78.1 (3.2)	79.1 (3.3)	79.8 (3.1)	81.2 (1.3)	33.7 (2.5)
μ 3DINO	Best 82.2 (0.8)	76.2 (2.9)	74.3 (0.4)	71.9 (3.4)	79.4 (0.6)	75.6 (1.2)	71.5 (1.4)	84.6 (1.5)	83.2 (0.5)	81.1 (2.4)	74.6 (1.2)	40.2 (0.2)
	Last 82.0 (0.5)	72.2 (1.8)	66.8 (2.3)	59.6 (2.7)	75.2 (1.3)	67.2 (1.3)	60.3 (1.4)	85.3 (0.2)	79.5 (0.9)	68.5 (2.2)	47.2 (0.5)	38.2 (0.3)
μ DivSeg	Best 83.5 (0.2)	82.5 (0.4)	80.1 (0.7)	75.5 (1.5)	82.3 (0.0)	79.5 (1.3)	75.8 (1.5)	88.8 (0.3)	87.3 (0.4)	86.8 (0.1)	81.8 (1.6)	42.7 (0.2)
	Last 83.2 (0.1)	79.8 (0.5)	72.3 (2.3)	66.0 (1.5)	80.2 (0.5)	71.9 (1.9)	65.6 (1.1)	88.0 (0.4)	83.8 (0.7)	75.8 (1.3)	62.8 (1.0)	41.4 (0.2)

than a standard segmentation pipeline, highlighting the importance of the robust pseudo-label guidance in μ DivSeg. Although LSF_{M_N} contains more training data, the instance-based nature of the label noise which is also present in validation data likely limited the effectiveness of the SOTA baselines. These baselines were competitive on LSF_{M_A} which also contains more data, and more *boundary-based* label error, though μ DivSeg also appeared to generalize well to this type of label noise and improves over baselines on the full dataset.

Qualitative Analysis: Fig. 3 compares μ DivSeg and baselines trained on 100% of data across real-world test sets. In TP_{EM_N}, neurons at low depth produced challenging FPs that should be excluded from downstream analyses. SSL improved discrimination of these cells, with μ DivSeg further reducing FPs. TP_{EM_V} had noisier labels and severe off-target bleed-through from the neuron to vessel channel in some volumes. Red arrows in the figure show regions where labels are degraded. In these regions, μ DivSeg improved on μ 3DINO, which employs a standard segmentation pipeline, *beyond* the silver ground truth. Despite having similar Dice scores, μ DivSeg corrected FNs in the ground truth that were present in the μ 3DINO prediction and was the only method able to connect the disjointed ground truth in the top left of the image. μ DivSeg was also able to avoid overpredicting off-target fluorescence as foreground, making it better at predicting proper vessel radii. In LSF_{M_A}, μ DivSeg better delineated axon boundaries while correcting faint FNs and avoided the round FP artifacts seen in other baselines. Finally on the LSF_{M_N} dataset, we observed that the dataset contains larger neurons that constitute TP for downstream analysis; however, due to their large volume compared to the majority of neurons in the dataset, they were often undersegmented or missed by silver ground truth. While μ DivSeg does not capture all FNs, it detected them more consistently while more accurately predicting neuron sizes. Interestingly, this improvement beyond the label actually led to our method quantitatively *underperforming* versus standard μ 3DINO on this test case, which could explain why μ DivSeg benefits were reduced on LSF_{M_N}. We show more predictions in Suppl. Material Sec. 10 to demonstrate this further.

4.5 Toy Dataset Experiments

Quantitative Analysis: We trained and evaluated on the NIS3D_Z, NIS3D_D and NIS3D_M datasets using 1. clean datasets, and 2. when adding synthetic

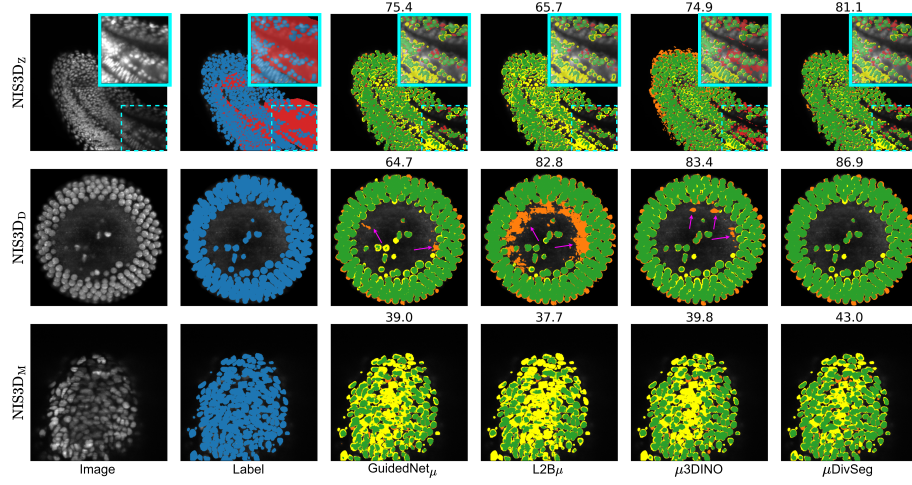


Fig. 4: Qualitative results on NIS3D. Labels and predictions follow Fig. 3 except red in labels marks unconfident regions excluded from quantitative analysis, while red in predictions indicates positives in those regions. Visualizations are MIPs with varying depth; numbers above predictions denote Dice (%) for a single seed on the full volume. The inset on NIS3D_Z highlights a hard region where μ DivSeg, GuidedNet, and L2B perform well. Arrows on NIS3D_D indicate FPs in other pipelines. Our method is more robust to real and synthetic label noise, reducing FPs and FNs.

noise (Tab. 2). We evaluated the “Best” and “Last” models on the validation image. Last models can perform better than Best models because we validated during training using a sliding window approach with a stride equal to image size (112), and performed final evaluations with 1/4th the stride (28) to match real-world dataset experiment setup. On the Best model, our method outperformed all other baselines at all noise levels with the exception of NIS3D_D L_3 , where it was slightly worse. Furthermore, the Best model performance for μ DivSeg degraded much slower with increasing label noise than a standard segmentation pipeline, demonstrating the effectiveness of our method for handling the instance-based noise present in 3D microscopy. L2B performs meta-learning on the same validation image that results are reported on. Thus, we posit that L2B performed well on NIS3D_D because its validation label is very clean, allowing for high quality meta-learning reweighting. In contrast, the NIS3D_Z and NIS3D_M validation labels are noisier and contain large uncertain regions. Despite masking these areas out, L2B performance was unstable, especially with larger amounts of synthetic noise. Notably, while μ DivSeg outperformed on Best models, on NIS3D_Z, GuidedNet degraded slower than μ DivSeg, and both baselines were closer when comparing Best *vs.* Last performance. We hypothesize this could be because the baselines performed suboptimal pixel-level pseudo-labelling for instance-based synthetic noise, but may be better at maintaining the pseudo-label during training to reduce overfitting to noisy labels.

Qualitative Analysis: The qualitative performance of the Best model for methods trained on L_2 noise strength for NIS3D_Z and NIS3D_D, and ‘Clean’ for NIS3D_M on the validation image are shown in Fig. 4. We note μ DivSeg predicted fewer FNs than L2B and GuidedNet and fewer FPs than standard μ 3DINO on NIS3D_Z. In unconfident regions, μ DivSeg generated better predictions than standard μ 3DINO, which demonstrates how the hardness mask can help improve model focus on low-confidence regions. The other baselines, and especially L2B which can similarly focus on hard image regions, also seemed to predict more accurate cell boundaries. The reason why GuidedNet underperformed on NIS3D_Z is because it erroneously predicted background noise as foreground for lower depths of the volume (not shown in the example visualization). In NIS3D_D, μ DivSeg balanced FPs and FNs better, and did not overfit to synthetic FPs in the ground truth like other methods. In NIS3D_M, we observed that μ DivSeg reduced FNs present in other baselines.

5 Conclusion

Despite its strong performance relative to baselines, our work also has limitations. Firstly, the pretraining dataset for μ 3DINO was imbalanced, and performance could be improved by improving diversity with data collection or curation methods [39]. Additionally, due to using fixed patch size, larger patch sizes improve context (and hence cluster quality) but reduce noise/hardness mask resolution; future work could investigate smoothing, smarter feature combination methods, or hierarchical clustering from larger to smaller patches to improve localization without sacrificing context. Finally, our feature queue was updated in a first-in-first-out manner, and a different approach taking into account feature diversity could benefit noisy label and hard patch prediction.

In this work, we introduce μ 3DINO, a general-purpose multimodal model pretrained on an ultra-large 3D microscopy dataset, and μ DivSeg, a pipeline leveraging SSL weights to detect noisy labels and hard patches to guide and improve segmentation in these images. To our knowledge, our methods are the first to tackle the core issues of label noise, hard regions, and label scarcity in 3D microscopy simultaneously. Our model and pipeline improved performance in real-world conditions across several downstream tasks in TPEM and LSFM with varying data sizes, and for increasing amounts of instance-based synthetic noise in a toy dataset. Using qualitative analyses, we also show how our framework is able to improve *beyond* silver ground truth in datasets with more label noise. This study, along with our code and model weights, can be used advance image analysis in 3D microscopy in neuroscience and beyond.

Acknowledgements

This work was supported by funding from the Natural Sciences and Engineering Research Council (NSERC) Discovery grant (RGPIN-2021-03728), the Canadian Institute of Health Research (PJT178059, PJT376309 and PJT156179),

Canada Foundation for Innovation (CFI) (40206), CFI JELF (43890), and the Ontario Research Fund. T.X. is funded by the NSERC PGS-D award and Google PhD Fellowship. M.G. is funded by the Canada Research Chairs program award (CRC-2021-00374). A.L.M. is supported by the Tory Family Chair in Oncology. We are grateful to the Digital Research Alliance of Canada (alliancecan.ca) for their allocation of computing resources used in parts of this research. We are grateful for support from the Harquail Centre for Neuromodulation.

References

1. Amat, F., Lemon, W., Mossing, D.P., McDole, K., Wan, Y., Branson, K., Myers, E.W., Keller, P.J.: Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. *Nature methods* **11**(9), 951–958 (2014)
2. 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. *Nature methods* **22**(3), 579–591 (2025)
3. Attarpour, A., Osmann, J., Rinaldi, A., Qi, T., Lal, N., Patel, S., Rozak, M., Yu, F., Cho, N., Squair, J., et al.: A deep learning pipeline for three-dimensional brain-wide mapping of local neuronal ensembles in teravoxel light-sheet microscopy. *Nature methods* **22**(3), 600–611 (2025)
4. Attarpour, A., Raffiee, M., Xu, T., Osmann, J., Patel, S., Yu, F., Au, B., Clappison, A., Biparva, M., Zhu, R., et al.: Quantitative profiling of whole-brain connectomes at single-axon resolution using deep learning and high-resolution light sheet microscopy. *bioRxiv* pp. 2025–11 (2025)
5. Berg, S., Kutra, D., Kroeger, T., Straehle, C.N., Kausler, B.X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., et al.: Ilastik: interactive machine learning for (bio) image analysis. *Nature methods* **16**(12), 1226–1232 (2019)
6. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems* **32** (2019)
7. Chen, R.J., Ding, T., Lu, M.Y., Williamson, D.F., Jaume, G., Song, A.H., Chen, B., Zhang, A., Shao, D., Shaban, M., et al.: Towards a general-purpose foundation model for computational pathology. *Nature medicine* **30**(3), 850–862 (2024)
8. Chen, Y., Al-Maskari, R., Horvath, I., Ali, M., Hoher, L., Yang, K., Lin, Z., Zhai, Z., Shen, M., Xun, D., et al.: Selma3d challenge: Self-supervised learning for 3d light-sheet microscopy image segmentation. *arXiv preprint arXiv:2501.03880* (2025)
9. Ciga, O., Xu, T., Martel, A.L.: Self supervised contrastive learning for digital histopathology. *Machine learning with applications* **7**, 100198 (2022)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020)
11. Forouzesh, M., Thiran, P.: Differences between hard and noisy-labeled samples: An empirical study. In: *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. pp. 91–99. SIAM (2024)
12. Gui, J., Chen, T., Zhang, J., Cao, Q., Sun, Z., Luo, H., Tao, D.: A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **46**(12), 9052–9071 (2024)

13. Gupta, A., Wefers, Z., Kahnert, K., Hansen, J.N., Misra, M.K., Leineweber, W., Cesnik, A., Lu, D., Axelsson, U., Ballllosera, F., et al.: Subcell: Proteome-aware vision foundation models for microscopy capture single-cell biology. *bioRxiv* pp. 2024–12 (2025)
14. Haberl, M.G., Churas, C., Tindall, L., Boassa, D., Phan, S., Bushong, E.A., Madany, M., Akay, R., Deerinck, T.J., Peltier, S.T., et al.: Cdeep3m—plug-and-play cloud-based deep learning for image segmentation. *Nature methods* **15**(9), 677–680 (2018)
15. Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., Roth, H.R., Xu, D.: Unetr: Transformers for 3d medical image segmentation. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. pp. 574–584 (2022)
16. Karim, N., Rizve, M.N., Rahnavard, N., Mian, A., Shah, M.: Unicon: Combating label noise through uniform selection and contrastive learning. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 9676–9686 (2022)
17. Kenney, M., Vasylieva, I., Hood, G., Cao-Berg, I., Tuite, L., Laghaei, R., Smith, M.C., Watson, A.M., Ropelewski, A.J.: The brain image library: A community-contributed microscopy resource for neuroscientists. *Scientific Data* **11**(1), 1212 (2024)
18. Kim, Y., Venkataraju, K.U., Pradhan, K., Mende, C., Taranda, J., Turaga, S.C., Arganda-Carreras, I., Ng, L., Hawrylycz, M.J., Rockland, K.S., et al.: Mapping social behavior-induced brain activation at cellular resolution in the mouse. *Cell reports* **10**(2), 292–305 (2015)
19. Klinghoffer, T., Morales, P., Park, Y.G., Evans, N., Chung, K., Brattain, L.J.: Self-supervised feature extraction for 3d axon segmentation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. pp. 978–979 (2020)
20. Kousaka, J., Iwane, A.H., Togashi, Y.: Automated cell structure extraction for 3d electron microscopy by deep learning. *Scientific Reports* **15**(1), 17481 (2025)
21. Li, J., Socher, R., Hoi, S.C.: Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394* (2020)
22. Li, P., Purkait, P., Ajanthan, T., Abdolshah, M., Garg, R., Husain, H., Xu, C., Gould, S., Ouyang, W., Van Den Hengel, A.: Semi-supervised semantic segmentation under label noise via diverse learning groups. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 1229–1238 (2023)
23. Li, R., Kudryashev, M., Yakimovich, A.: A weak-labelling and deep learning approach for in-focus object segmentation in 3d widefield microscopy. *Scientific reports* **13**(1), 12275 (2023)
24. Liu, E.Z., Haghgoo, B., Chen, A.S., Raghunathan, A., Koh, P.W., Sagawa, S., Liang, P., Finn, C.: Just train twice: Improving group robustness without training group information. In: *International Conference on Machine Learning*. pp. 6781–6792. PMLR (2021)
25. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017)
26. Lu, M.Y., Chen, B., Williamson, D.F., Chen, R.J., Liang, I., Ding, T., Jaume, G., Odintsov, I., Le, L.P., Gerber, G., et al.: A visual-language foundation model for computational pathology. *Nature medicine* **30**(3), 863–874 (2024)
27. McDole, K., Guignard, L., Amat, F., Berger, A., Malandain, G., Royer, L.A., Turaga, S.C., Branson, K., Keller, P.J.: In toto imaging and reconstruction of

- post-implantation mouse development at the single-cell level. *Cell* **175**(3), 859–876 (2018)
28. Ntiri, E.E., Xu, T., Rozak, M., Attarpour, A., Dorr, A., Stefanovic, B., Goubran, M.: A self-supervised deep learning pipeline for segmentation in two-photon fluorescence microscopy. *bioRxiv* pp. 2025–01 (2025)
29. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023)
30. Renier, N., Adams, E.L., Kirst, C., Wu, Z., Azevedo, R., Kohl, J., Autry, A.E., Kadiri, L., Venkataraju, K.U., Zhou, Y., et al.: Mapping of brain activity by automated volume analysis of immediate early genes. *Cell* **165**(7), 1789–1802 (2016)
31. Reynolds, D.A., et al.: Gaussian mixture models. *Encyclopedia of biometrics* **741**(659-663), 3 (2009)
32. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical image computing and computer-assisted intervention*. pp. 234–241. Springer (2015)
33. Rozak, M.W., Mester, J.R., Attarpour, A., Dorr, A., Patel, S., Koletar, M., Hill, M.E., McLaurin, J., Goubran, M., Stefanovic, B.: A deep learning pipeline for mapping in situ network-level neurovascular coupling in multi-photon fluorescence microscopy. *Elife* **13**, RP95525 (2026)
34. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 761–769 (2016)
35. Stringer, C., Wang, T., Michaelos, M., Pachitariu, M.: Cellpose: a generalist algorithm for cellular segmentation. *Nature methods* **18**(1), 100–106 (2021)
36. Taleb, A., Loetzsch, W., Danz, N., Severin, J., Gaertner, T., Bergner, B., Lippert, C.: 3d self-supervised methods for medical imaging. *Advances in neural information processing systems* **33**, 18158–18172 (2020)
37. Tang, Y., Yang, D., Li, W., Roth, H.R., Landman, B., Xu, D., Nath, V., Hatamizadeh, A.: Self-supervised pre-training of swin transformers for 3d medical image analysis. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 20730–20740 (2022)
38. Tyson, A.L., Rousseau, C.V., Niedworok, C.J., Keshavarzi, S., Tsitoura, C., Cossell, L., Strom, M., Margrie, T.W.: A deep learning algorithm for 3d cell detection in whole mouse brain image datasets. *PLoS computational biology* **17**(5), e1009074 (2021)
39. Vo, H.V., Khalidov, V., Darcet, T., Moutakanni, T., Smetanin, N., Szafraniec, M., Touvron, H., Couprie, C., Oquab, M., Joulin, A., et al.: Automatic data curation for self-supervised learning: A clustering-based approach. *arXiv preprint arXiv:2405.15613* (2024)
40. Vorontsov, E., Bozkurt, A., Casson, A., Shaikovski, G., Zelechowski, M., Liu, S., Severson, K., Zimmermann, E., Hall, J., Tenenholtz, N., et al.: Virchow: A million-slide digital pathology foundation model. *arXiv preprint arXiv:2309.07778* (2023)
41. Wang, X., Yang, S., Zhang, J., Wang, M., Zhang, J., Yang, W., Huang, J., Han, X.: Transformer-based unsupervised contrastive learning for histopathological image classification. *Medical image analysis* **81**, 102559 (2022)
42. Xu, A.S., Shamsi, N.I., Gjestebj, L.A., Brattain, L.J.: Self-supervised edge detection reconstruction for topology-informed 3d axon segmentation and centerline detection. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. pp. 7831–7839 (2024)

43. Xu, T., Hosseini, S., Anderson, C., Rinaldi, A., Krishnan, R.G., Martel, A.L., Goubran, M.: A generalizable 3d framework and model for self-supervised learning in medical imaging. *npj Digital Medicine* **8**(1), 639 (2025)
44. Xu, T., Rozak, M., Ntiri, E., Dorr, A., Mester, J., Stefanovic, B., Martel, A., Goubran, M.: Masked image modeling for label-efficient segmentation in two-photon excitation microscopy. In: *Workshop on Medical Image Learning with Limited and Noisy Data*. pp. 117–127. Springer (2023)
45. Zhang, J., Menon, A., Veit, A., Bhojanapalli, S., Kumar, S., Sra, S.: Coping with label shift via distributionally robust optimisation. *arXiv preprint arXiv:2010.12230* (2020)
46. Zhao, H., Meng, H., Yang, D., Xie, X., Wu, X., Li, Q., Niu, J.: Guidednet: semi-supervised multi-organ segmentation via labeled data guide unlabeled data. In: *Proceedings of the 32nd ACM international conference on multimedia*. pp. 886–895 (2024)
47. Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A.M., Litany, O.: Contrast to divide: Self-supervised pre-training for learning with noisy labels. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. pp. 1657–1667 (2022)
48. Zheng, W., Peng, C., Hou, Z., Lyu, B., Wang, M., Mi, X., Qiao, S., Wan, Y., Yu, G.: Nis3d: a completely annotated benchmark for dense 3d nuclei image segmentation. *Advances in neural information processing systems* **36**, 4741–4752 (2023)
49. Zhou, Y., Li, X., Liu, F., Wei, Q., Chen, X., Yu, L., Xie, C., Lungren, M.P., Xing, L.: L2b: Learning to bootstrap robust models for combating label noise. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 23523–23533 (2024)
50. Zhu, Z., Dong, Z., Liu, Y.: Detecting corrupted labels without training a model to predict. In: *International conference on machine learning*. pp. 27412–27427. PMLR (2022)
51. Zhu, Z., Song, Y., Liu, Y.: Clusterability as an alternative to anchor points when learning with noisy labels. In: *International Conference on Machine Learning*. pp. 12912–12923. PMLR (2021)