Benchmarking Self-Supervised Learning on Diverse Pathology Datasets (Supplementary Materials)

Mingu Kang* Heon Song* Seonwook Park Donggeun Yoo Sérgio Pereira
Lunit Inc.
{jeffkang, heon.song, spark, dgyoo, sergio}@lunit.io

Overview. In this supplementary material, we describe the details of the downstream datasets adopted in the main paper and show some example images. This document also contains further implementation details regarding the pre-training and downstream training steps, including fine-tuning with limited labeled data. Last but not least, we provide further analyses, such as the effectiveness of pre-training for longer epochs and pre-training stability when using data from different magnifications.

Note that the corresponding or relevant sections from the main paper are referenced **in blue text** in the section titles.

A. Downstream Dataset Details (Section 4.2)

In this section, we describe the details of the datasets used in our analysis. We use BACH, CRC, PCam, and MHIST for the image classification task, and CoNSeP for the nuclei instance segmentation task. We sample a few training images from each dataset and present them in Fig. A.1 and Fig. A.2.

BACH. The goal of the Grand Challenge on BreAst Cancer Histology (BACH) [2] is to classify pathology images into four classes: normal, benign, in situ carcinoma, and invasive carcinoma. The dataset is composed of 400 training images and 100 test images. The test images are collected from a different set of patients from the training images. All images are collected from Hospital CUF Porto, Centro Hospitalar do Tâmega e Sousa, and Centro Hospitalar Cova da Beira.

CRC. This dataset [15] consists of 100,000 training images and 7,180 test images from H&E stained WSIs of human colorectal cancer (CRC) and normal tissue. The training and test images are extracted from 86 WSIs and 25 WSIs, respectively. The slides are collected from the NCT Tissue Bank and the University Medical Center Mannheim. The task is the identification of nine tissue classes: adipose tissue, background, debris, lymphocytes, mucus, smooth muscle, normal colon mucosa, cancer-associated stroma, and CRC epithelium. All images are color normalized with the Macenko method [18].

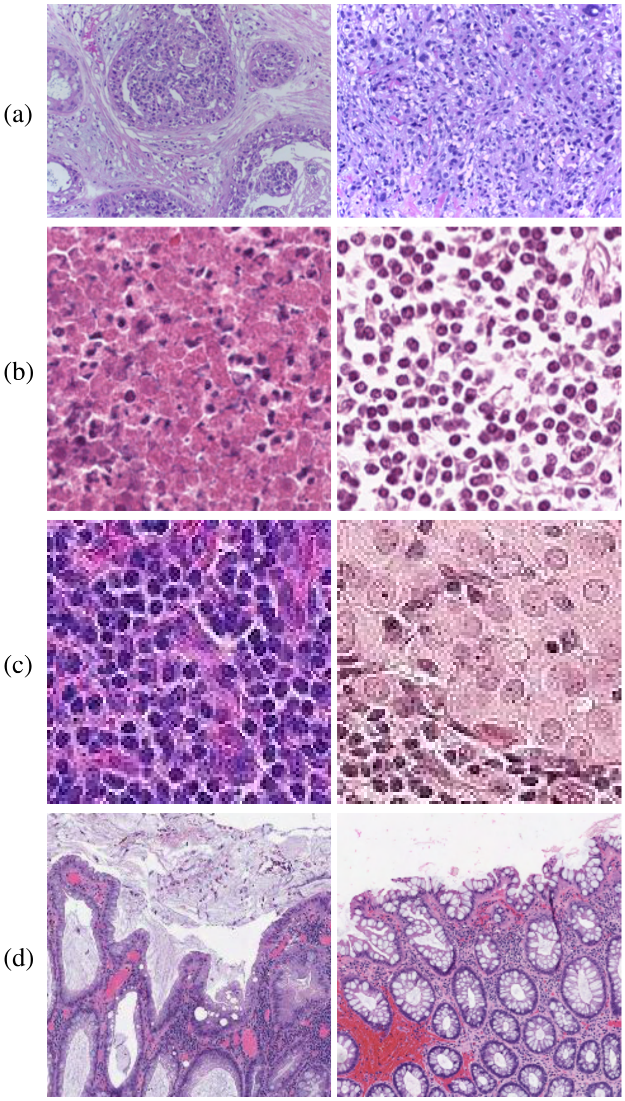

Figure A.1. **Example training images from the classification datasets**: (a) BACH, (b) CRC, (c) PCam, and (d) MHIST.

PCam. The PatchCamelyon (PCam) [21] dataset is derived from the Camelyon16 [3] dataset that contains 400 H&E stained WSIs from two hospitals: Radboud Univer-

[*] The first two authors contributed equally.

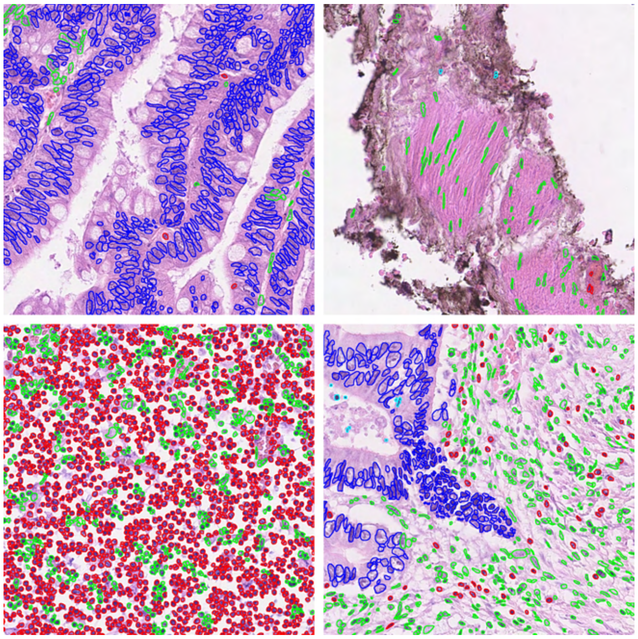

Figure A.2. **Example training images from the CoNSeP dataset.** The dataset provides annotated nuclei masks along with cell type labels. Following the original HoVer-Net paper [11], we use the following nuclei types for training and evaluation: 🟦 epithelial, 🟥 inflammatory, 🟩 spindle-shaped, and 🟦 miscellaneous.

sity Medical Center (RUMC), and University Medical Center Utrecht (UMCU). The PCam dataset includes 262,144 training images, 32,768 validation images, and 32,768 test images. Each image is annotated with a binary label for determining the presence of metastases.

MHIST. The minimalist histopathology image analysis (MHIST) [22] dataset is comprised of 2,175 training images and 977 test images. The images are extracted from 328 H&E stained Formalin Fixed Paraffin-Embedded (FFPE) WSIs of colorectal polyps from Dartmouth-Hitchcock Medical Center. The task is the binary classification between hyperplastic polyps (HPs) and sessile serrated adenomas (SSAs), where HPs are benign and SSAs are precancerous lesions.

CoNSeP. The Colorectal Nuclear Segmentation and Phenotypes (CoNSeP) dataset [11] consists of 41 H&E images and is split into 27 images and 14 images for training and test sets, respectively. The data comes from University Hospitals Coventry and Warwickshire, UK (UHCW). The annotation contains segmentation masks of each nucleus along with its class (See Fig. A.2). Note that the healthy epithelial and dysplastic/malignant epithelial are considered general epithelial types. Fibroblast, muscle, and endothelial are matched into a spindle-shaped nuclei type. In total, 24,319 unique nuclei masks along with 4 major types out of 7 cell types are used during training.

B. Implementation Details

In the interest of improving the reproducibility of our study, we provide further details regarding our pre-training data, setup, as well as details on how we conducted our downstream evaluations. Furthermore, we discuss the details of the limited labeled data experiments.

B.1. Preparation of Pre-training Data (Section 4.1)

In selecting image patches to compose the TCGA dataset, we first use an internal model with a DeepLab v3+ architecture [6] to segment the foreground regions of WSI. From the candidate patches that are located in areas predicted as foreground, we select up to 500 patches per magnification, per slide, with equal spacing between them. To ensure that we have informative image patches in our pre-training dataset, we filter out patches that are too white (mean saturation value below 5) or too smooth (mean squared Laplacian below 15). For TULIP, we do not apply such filtering logic due to the relatively smaller foreground area (too many patches are lost otherwise).

B.2. Calculating Statistics of the Pre-training Data (Section 4.1)

For the purpose of input image standardization during SSL pre-training, we collect the per-channel mean and standard deviation of intensities in RGB space, using 10% of the full unlabeled image data. This subsampling is done per WSI, to maintain diversity and reduce computational cost.

In a similar manner, we compute the per-channel means and variances in 3 color spaces (HSV, Lab, HED) for use with the RandStainNA method, using 10% of the full image data. Specifically, we compute per color space, and per channel, the mean and standard deviation of per-image mean intensity, as well as the mean and standard deviation of the per-image standard deviation of intensity. Please refer to [20] for further details.

For RandStainNA$_{GMM}$, we similarly compute per color space, and per channel, the per-image mean intensity and its standard deviation. However, instead of simply finding the mean and standard deviation of those values independently (fitting individual unimodal Gaussian distributions 18 times as in RandStainNA), we fit a 10-component Gaussian Mixture Model (GMM) for each color space, yielding 3 models. This is done to fit the covariance between the input variables (6 variables exist for each color space) and respect their multi-modal nature.

B.3. Augmentation Details (Section 3.2)

Unless otherwise stated, in our experiments, we pre-train by applying the following changes to the default method-specific augmentation scheme:

- Random vertical flip `(p=0.5)`.

- Color dropping `(p=0.2)`: the color of images are converted randomly to grayscale.
- Weak color jittering `(p=0.8)`: the brightness, contrast, saturation, and hue of images are randomly adjusted with a strength of `0.2, 0.2, 0.2, 0.1`, respectively.
- RandStainNA$_{GMM}$ `(p=0.8)`: per image, a color space is randomly selected (from HSV, Lab, or HED), then channel-wise mean and standard deviation values are sampled from a GMM which is fitted on statistics from part of the pre-training data (10%). The input image is re-normalized based on these values, using Reinhard's method [19].

B.4. SSL Methods (Section 4.3)

We provide implementation details of each SSL method used in our analysis. We use the VISSL [10] library to pre-train the the 4 studied SSL methods, and follow the same configurations as originally proposed in [4, 5, 9, 24]. All representations are trained for 200 ImageNet epochs, distributed over 64 V100 16GB GPUs. A linear warmup schedule is applied for the first 10 epochs and a cosine learning rate decay is applied subsequently. Each method was originally proposed with its specific augmentation schemes, and we follow those original data augmentation pipelines while adding our proposed techniques on top. Regarding the RandStainNA augmentation, it requires the statistics of 3 color spaces (HSV, Lab, HED) to produce augmented images. To compute the statistics, we randomly sample 10% of the unlabeled image patches from the corresponding pre-training dataset.

MoCo v2. We use the SGD optimizer with an initial learning rate of 0.3. The learning rate is linearly scaled up based on $lr = lr * batchsize/256$, where *batchsize* is 4,096. The memory bank size is fixed to 65,536, and a momentum coefficient m of 0.999 is used. Weight decay of 10^{-4} is utilized for regularization.

SwAV. We use the SGD optimizer with an initial learning rate of 0.3. The learning rate is linearly scaled up based on $lr = lr * batchsize/256$, where *batchsize* is 2,048. The number of prototypes is 3,000 to avoid intractable computational costs from the Sinkhorn algorithm. 2×224 + 6×96 multi-crop augmentation is employed as done in the original paper.

Barlow Twins. The LARS optimizer [23] is adopted for Barlow Twins pre-training. Note that, as in the original work [24], we apply different learning rates for weights and biases, 0.2 and 0.0048, respectively. The biases and batch normalization layers are excluded from LARS optimization to follow the original implementation. The learning rates of weights and biases are linearly scaled up based on $lr = lr * batchsize/256$, where *batchsize* is 2,048. The dimension of the embeddings is 8,192, and training is conducted with a coefficient of off-diagonal term $\lambda = 5 \cdot 10^{-3}$ and a weight decay of $1.5 \cdot 10^{-6}$.

DINO. We train the model with the AdamW [16] optimizer. The learning rate of 0.0005 is used for stability during pre-training. The learning rate is linearly scaled up based on $lr = lr * batchsize/256$, where *batchsize* is 1,024. Similar to the learning rate decay, the weight decay also follows a cosine schedule from 0.04 to 0.4. For DINO$_{p=16}$, 2× 224 + 8× 96 multi-crop augmentation is employed, while 2× 224 + 6× 96 multi-crop augmentation is used for DINO$_{p=8}$.

B.5. Downstream Training Details (Section 4.4)

Image Classification. We split each downstream dataset into training, validation, and test sets. The learning rate and weight decay values are optimized using training and validation, only. In the BACH dataset, the labels for the test set are not provided. Hence, we split the training set by a 6:1:3 ratio (training, validation, test). For the CRC and MHIST datasets, the test set is provided with labels, and the training set is split by a 7:3 ratio (training, validation). For the PCam dataset, we follow the original data split. When splitting the data, we do it randomly but in a class-balanced manner. Based on the performance measured on the validation sets, we perform a grid search of learning rates from $\{1, 0.1, 0.01, 0.001\}$ and weight decay values from $\{0.1, 0.01, 0.001, 0\}$.

As data augmentation for ResNet-50, the input image is randomly flipped both horizontally and vertically, at training time. For the BACH dataset, we apply random cropping and resizing to 1024 × 768 at training time; at test time, we resize the images to 1024 × 768. For ViT-S, the same augmentation is used but all images are resized to 224 × 224. We train the models with the SGD optimizer with a momentum of 0.9 and a cosine learning rate decay. The ResNet-50 based models are trained for 200, 20, 20, and 90 epochs on the BACH, CRC, PCam, and MHIST datasets, respectively. The Transformer-based models are trained for 30 epochs on the CRC and PCam datasets and for 200 and 90 epochs on the BACH and MHIST datasets, respectively. During fine-tuning, the backbone layers (i.e., ResNet-50 and ViT-S) are trained with a learning rate 100 times lower than that of the last classification layer.

Nuclei Instance Segmentation. We follow the standard pipeline of HoVer-Net [11], as provided in its open-source implementation[1], including data augmentation and patch extraction. Hover-Net defines a two-stage training procedure. At the first stage, only the decoders are trained while freezing the backbone layers. With the trained decoders,

[1] https://github.com/vqdang/hover_net

Arch.	Method	BACH		CRC		PCam		MHIST	
		Linear	Fine-tune	Linear	Fine-tune	Linear	Fine-tune	Linear	Fine-tune
ResNet-50	*Random*	51.67	61.67	68.91	89.99	76.52	75.71	63.15	75.54
	Supervised	80.83	86.67	90.93	92.09	80.79	80.63	76.25	78.92
	Epoch 200								
	MoCo v2	77.50	90.83	93.52	**96.21**	86.78	**87.62**	77.07	**85.88**
	SwAV	83.33	82.50	95.78	93.31	85.28	87.60	71.14	77.99
	BT	**87.50**	85.00	94.60	93.23	**88.15**	86.92	**78.81**	81.27
	Epoch 800								
	MoCo v2	79.17	**91.67**	95.01	95.45	87.84	86.90	72.77	84.95
	SwAV	82.50	85.83	**96.46**	92.74	86.16	87.05	75.54	85.47
	BT	86.67	**91.67**	94.48	94.99	86.26	86.75	78.20	80.25
ViT-S	*Random*$_{p=16}$	45.00	57.50	69.90	86.10	74.43	75.42	63.46	62.13
	Supervised$_{p=16}$	75.83	85.83	91.56	95.81	80.96	88.30	**78.51**	**81.68**
	Epoch 200								
	DINO$_{p=16}$	85.83	87.50	94.19	95.81	**88.78**	90.40	76.15	79.43
	Epoch 400								
	DINO$_{p=16}$	**86.67**	**88.33**	**95.13**	**96.48**	88.60	89.50	75.44	81.06

Table C.1. **Downstream evaluation of image classification tasks under a different number of pre-training epochs.** We report Top-1 accuracy for both linear and fine-tuning experiment protocols trained using the TCGA data source. Note that p represents the patch size used in ViT. We compare results column-wise and mark the best results in **bold** and the second-best results in underline for ResNet-50 based methods and ViT-S methods separately.

all layers are then fine-tuned at the second stage. Technically, Preact-ResNet-50 [14] is employed in the original implementation, but we replace it with the standard ResNet-50 [13] and reproduce the results for a fair comparison. This is done to perform SSL pre-training in a standard manner while permitting this nuclei instance segmentation downstream task. Since we change the backbone, we perform Grid Search to find a proper learning rate. We use $5 \cdot 10^{-4}$ learning rate for both stages of HoVer-Net. Moreover, based on the open-source implementation, the authors of HoVer-Net fine-tune the first convolutional layer of ResNet at the first stage, but we keep them frozen.

Similar to the architecture of FPN-based instance segmentation, HoVer-Net requires features from multiple scales in the encoder. However, the outputs of the ViT-based encoder are not compatible with the existing decoders of Hover-Net without further modifications. In order to provide multi-scale features to the decoder, we refer to the protocol from [1] where the feature scales are interpolated using several transposed convolution layers with kernel size $k = 2$ and stride $s = 2$. More specifically, features from the 4^{th}, 6^{th}, 8^{th}, and 12^{th} layers are extracted from the ViT-S architecture, which consists of 12 layers in total. With this design, the decoders remain unchanged. For the sake of a fair comparison, we also perform Grid Search on the ViT-S architecture. The learning rate of $5 \cdot 10^{-4}$ is used for both stages of HoVer-Net.

B.6. Fine-tuning with Limited Labeled Data (Section 5.4)

Image Classification. Following prior works [8, 12, 24], we randomly sample 1% and 10% of the CRC training set by balancing classes. Based on the fine-tuning procedure, we train the models for 60 and 90 epochs for 1% and 10% labeled data, respectively.

Nuclei Instance Segmentation using CoNSeP. In our limited labeled data experiments using CoNSeP, we control the number of H&E images instead of the number of extracted patches to mimic the real-world setting where one H&E image corresponds to one unique patient. Since assuming 1% of training data is unreasonable in the current setting (i.e, 0.27 H&E image), instead, we define the ratio of 10% and 30% for nuclei instance segmentation. Note that the reported values in the experiments are the averaged number from 3 repetitive experiments with different seed values for image/patient selection. This is necessitated by the smaller dataset size (compared to CRC) and is done for a fair comparison between methods.

C. Pre-training for more epochs (Section 5)

Typically, increasing the number of pre-training epochs has shown to be effective in improving the learned representations in various SSL methods. To investigate the effective-

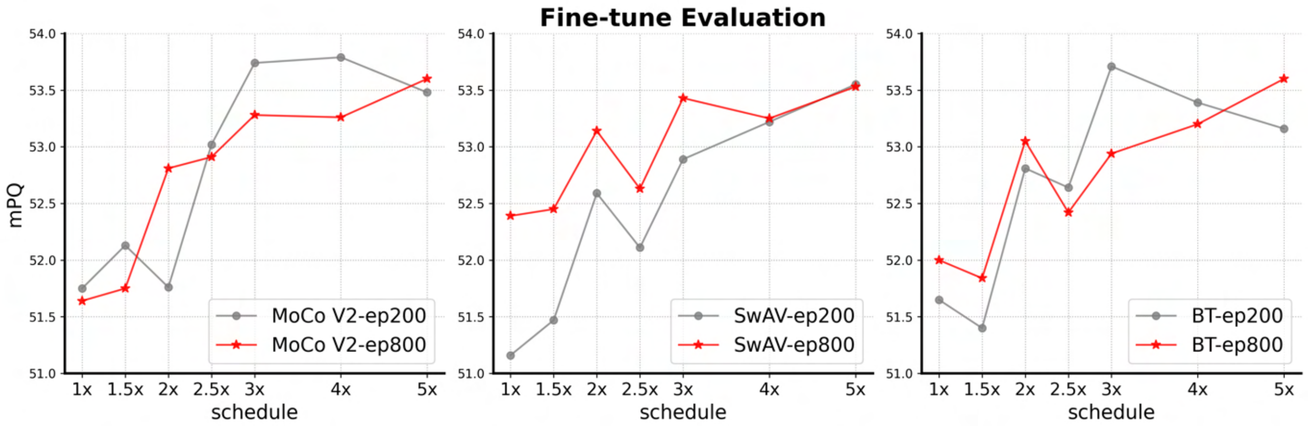

Figure C.1. **The effectiveness of longer pre-training according to learning schedules.** We present fine-tuning evaluation results for the nuclei instance segmentation task using the CoNSeP dataset. We see that there are few differences between the 200 epoch models and 800 epoch models, except that SwAV benefits from longer pre-training when the downstream task is fine-tuned with a limited learning schedule.

Arch.	**Method**	**CoNSeP**	
		Linear	Fine-tune
ResNet-50	*Random*	22.29	46.72
	Supervised	34.25	49.60
	Epoch 200		
	MoCo v2	39.85	51.75
	SwAV	40.45	51.16
	BT	40.79	51.61
	Epoch 800		
	MoCo v2	**40.93**	51.64
	SwAV	40.59	**52.39**
	BT	40.65	52.00
ViT-S	*Random*$_{p=16}$	20.55	27.19
	Supervised$_{p=16}$	21.43	36.70
	Epoch 200		
	DINO$_{p=16}$	32.54	38.43
	Epoch 400		
	DINO$_{p=16}$	**32.93**	**39.03**

Table C.2. **Downstream evaluation for the nuclei instance segmentation task under a different number of pre-training epochs**. We report the mPQ score for both linear and fine-tuning experiment protocols for models trained using the TCGA data source. We compare results column-wise and mark the best results in **bold** and the second-best results in underline for ResNet-50 based methods and ViT-S methods separately.

ness of the longer pre-training in the pathology domain, we pre-train the model for 800 ImageNet epochs using MoCo v2, SwAV, and Barlow Twins. Note that, due to computational costs, we report the results from DINO$_{p=16}$ trained for 400 ImageNet epochs.

Tab. C.1 and Tab. C.2 present the performance of image classification and nuclei instance segmentation, respectively. Compared to the results from 200 ImageNet epochs, SwAV is the only method that benefits from the longer pre-training in the fine-tuning protocol, especially in BACH, MHIST, and CoNSeP datasets. In contrast, the other methods show marginal improvements or are on par with the 200 ImageNet epoch counterparts. DINO$_{p=16}$ shows a slightly improved performance on image classification, while nuclei instance segmentation remains on par. Even in the different learning schedules illustrated in Fig. C.1, we observe that no clear benefit of the longer pre-training stands out in MoCo v2 and Barlow Twins, yet SwAV consistently maintains the benefit of the longer pre-training.

Across all experiments, we confirm that certain SSL methods (e.g., SwAV) may require more pre-training iterations, but generally increasing the number of pre-training epochs shows marginal improvements on both image classification and nuclei instance segmentation tasks. In other words, pre-training for 200 ImageNet epochs can be sufficient to achieve satisfactory downstream performance, especially for MoCo v2, Barlow Twins, and DINO. We therefore suggest that using 200 ImageNet epochs would be adequate to study the potential of SSL pre-training in the pathology domain.

D. Pre-training Stability with Different Magnifications (Section 5.6)

In the main paper, we show that it is beneficial to train on image data from a combination of 20× and 40× objective magnifications. Here, we show that pre-training stability is also affected by the choice of magnification. In Fig. D.1, we present the loss trajectory during the pre-training stage using Barlow Twins. As shown in the graph, using a single

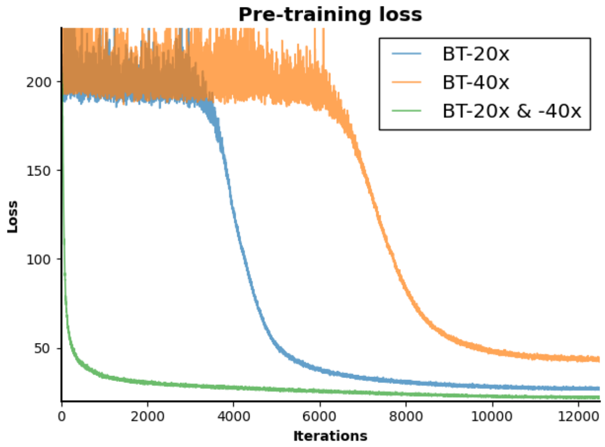

Figure D.1. **Loss progression while pre-training Barlow Twins on different magnifications.** Training on a combination of 20× and 40× results in quick convergence and stable pre-training.

magnification produces unstable losses and the loss begins to converge after approximately 4,000 and 7,000 iterations for magnifications of 20× and 40×, respectively. The loss values at the end of the pre-training stage are also higher in the case of using a single magnification. In contrast, using multiple magnifications results in stable pre-training and fast convergence, in addition to improved downstream task performance.

E. Larger Inputs for ViT (Section 5.2)

The implementation of the standard HoVer-Net [11] method involves the fine-tuning of a pre-trained ResNet, using images with a resolution of 270 × 270. However, by design, ViT expects input images of 224 × 224 resolution. Given the potential advantages that larger input resolutions can bring to the task of nuclei instance segmentation, we adopt a positional embedding interpolation technique to increase the input image size to 272 × 272, which is divisible by both 16 and 8. Through this technique, we aim to maintain consistent input resolutions across the ResNet and ViT backbones being evaluated. Tab. E.1 presents the result according to the input size. We observe that the larger input size improves performance for DINO$_{p=16}$, while the performance of DINO$_{p=8}$ reduces.

F. Further Data Augmentation Ablation Study (Section 5.6)

To provide a compelling demonstration of the effectiveness of the proposed techniques, we opted for the most practical, yet challenging fine-tuning setting of nuclei instance segmentation. Through the application of the linear evaluation protocol, we further validate the effectiveness of our techniques by showcasing improvements across all datasets. Notably, our set of techniques consistently and significantly improves the performance compared to the baseline approach that relies on augmentations designed for natural images. The improvement presented in Tab. F.1 serves as a clear signal of the effectiveness of our proposed techniques, which were carefully designed with the aid of domain-specific knowledge.

Arch.	**Method**	**CoNSeP**	
		Linear	Fine-tune
ViT-S	**224 input**		
	Supervised$_{p=16}$	21.43	36.70
	DINO$_{p=16}$	32.54	38.43
	DINO$_{p=8}$	**42.71**	**46.70**
	272 input		
	Supervised$_{p=16}$	28.60	34.50
	DINO$_{p=16}$	35.81	41.13
	DINO$_{p=8}$	**40.08**	**44.24**

Table E.1. **Downstream evaluation for the nuclei instance segmentation task under a different input resolution.** We report the mPQ score for both linear and fine-tuning experiment protocols for models trained using the TCGA data source. We compare results column-wise and mark the best results in **bold** and the second-best results in underline.

	BACH	CRC	PCam	MHIST	CoNSeP
BT trained on TCGA	84.2	94.2	84.5	78.0	40.9
+ our aug. techniques	**87.5**	**94.7**	**87.6**	**79.5**	**41.3**

Table F.1. **Benefit of our augmentation techniques.** Linear evaluation results show that our proposed augmentation techniques consistently and significantly improve performance.

G. Intriguing Properties of Self-supervised ViT (Section 5.2)

As part of an effort to explore the potential of domain-aligned pre-training, we visualize the attention maps of self-supervised ViT and supervised ViT pre-trained on ImageNet. Our results, as illustrated in Fig. G.1, demonstrate that SSL ViT interestingly identifies and locates cells while also recognizing morphological phenotypes, which is aligned with recent observations [7]. Specifically, attention heads 1 ∼ 4 attend to epithelial and inflammatory cells, whereas heads 5 ∼ 6 focus on fibroblast cells. In contrast, supervised ViT pre-trained on ImageNet fails to generate interpretable signals due to the domain gap, highlighting the effectiveness of domain-aligned pre-training in generating informative signals for downstream tasks. We believe that this intriguing property can be leveraged to enable future potentials in the field of histopathology.

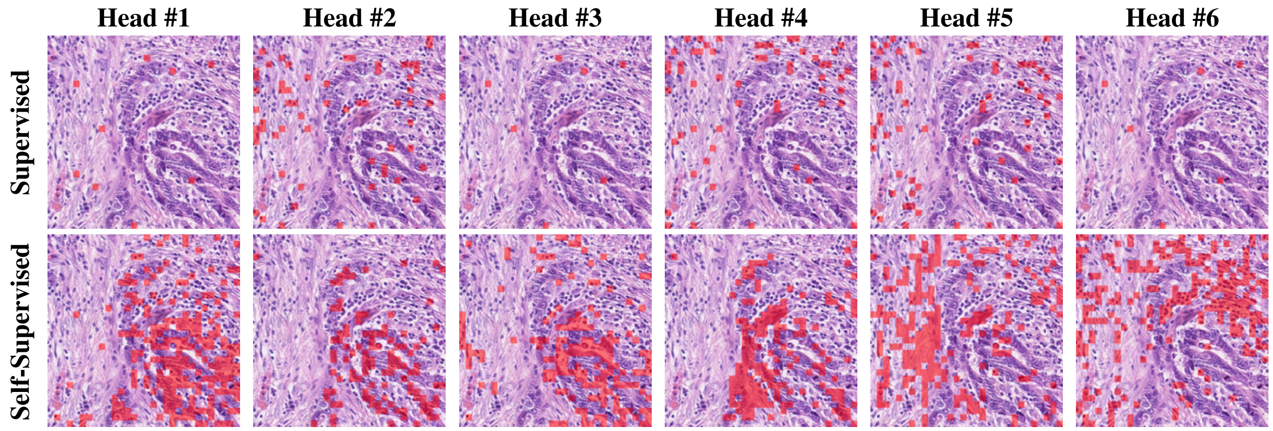

Figure G.1. **Visualizing multi-head self-attentions of ViT**. We visualize the attention map of several pre-trained ViT-S. Specifically, ViT-S has 6 attention heads. We visualize each head from the last layer of ViT. Our visualizations are presented in rows, with each row displaying attention maps alongside their corresponding overlayed image. The first two rows showcase the qualitative result of the supervised ViT pre-trained on ImageNet, while the next two rows display the qualitative result of the self-supervised ViT (DINO$_{p=16}$) pre-trained on TCGA. Note that, the input image is resized to 480 × 480 resolution, and overlaid in "red" are visual tokens whose attention weight > 0.5 and span 16 × 16 pixels.

H. Qualitative Results of Nuclei Instance Segmentation (Section 5.2)

In order to perform a qualitative assessment of the effect of domain-aligned pre-training on nuclei instance segmentation, we compare the predictions of models using supervised ImageNet pre-training and self-supervised TCGA pre-training, adapted under the linear evaluation protocol. The result presented in Fig. H.1 shows that domain-aligned pre-training can offer the benefit on downstream tasks effectively, resulting in capturing foreground cells and accurately classifying them, in contrast to the model trained using ImageNet pre-trained weight.

I. Slide-level Evaluation

The slide-level classification task is outside of the scope of our work. Nonetheless, we conduct a preliminary experiment to demonstrate the usefulness of the features learned through SSL for this task, too. We train and test models for the classification of breast cancer metastases in WSIs, using the same configuration as CLAM [17] but on the Camelyon16 [3] dataset. To extract features from the WSIs, we use two pre-trained weights: “Supervised (IN)” and “MoCo v2 (TC+TU)”. We find that models achieve an AUROC of 0.986 when using “MoCo v2 (TC+TU)” pre-trained weights, while models achieve an AUROC of 0.927 when using “Supervised (IN)” pre-trained weights. This result indicates that domain-aligned pre-training also can be beneficial to the slide-level task.

References

[1] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *NeurIPS*, 34:20014–20027, 2021. 4

[2] Guilherme Aresta, Teresa Araújo, Scotty Kwok, Sai Saketh Chennamsetty, Mohammed Safwan, Varghese Alex, Bahram Marami, Marcel Prastawa, Monica Chan, Michael Donovan, et al. Bach: Grand challenge on breast cancer histology images. *Medical Image Analysis*, 56:122–139, 2019. 1

[3] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017. 1, 7

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 33:9912–9924, 2020. 3

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 3

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 2

[7] Richard J Chen, Chengkuan Chen, Yicong Li, Tiffany Y Chen, Andrew D Trister, Rahul G Krishnan, and Faisal Mahmood. Scaling vision transformers to gigapixel images via hi-

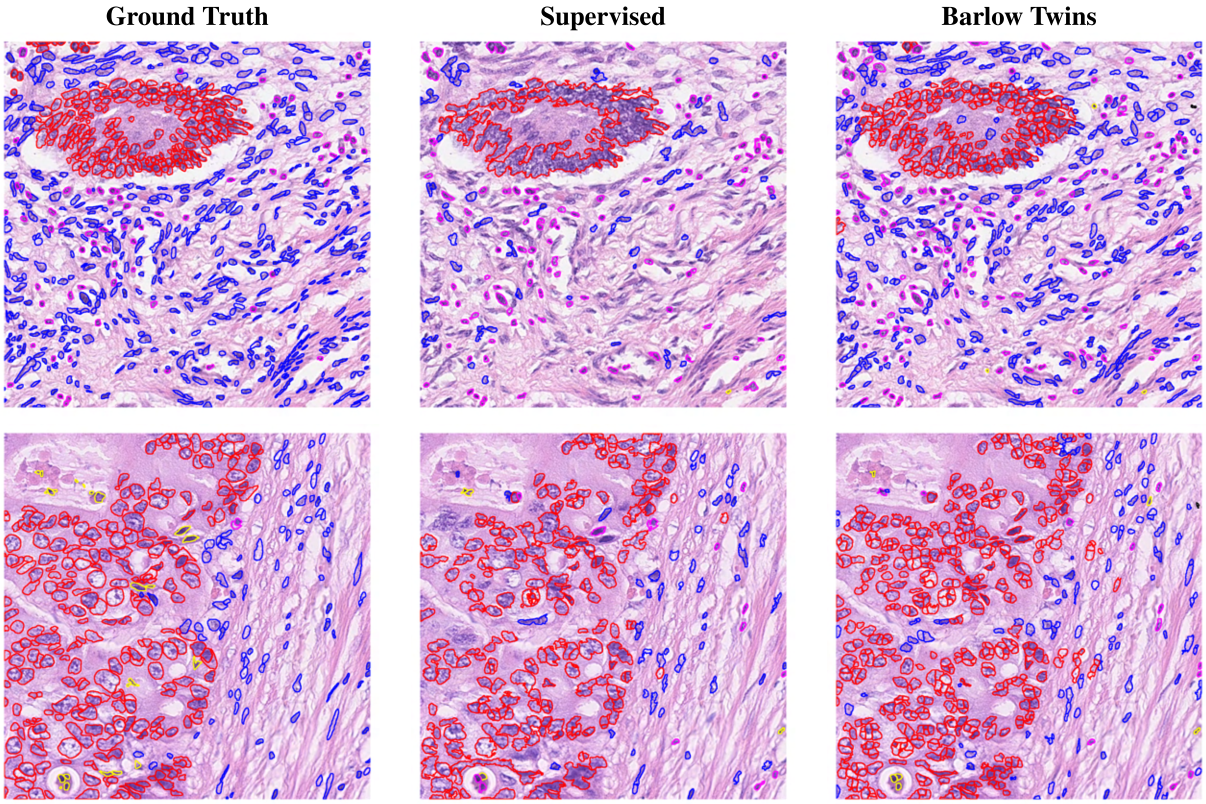

Figure H.1. **Visualizing predictions of models.** We visualize the overlay predictions of different models on CoNSeP. A linear evaluation protocol is adopted to more directly assess the quality of the representations learned during pre-training. We selected the best-performing pre-trained model, Barlow Twins, based on the results of Table 4., obtained from the linear evaluation protocol. We find that predictions from Barlow Twins are similar to the ground-truth, whereas the "Supervised" alternative produces poor nuclei boundaries and merges cells incorrectly.

erarchical self-supervised learning. In *CVPR*, pages 16144–16155, 2022. 6

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 4

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 3

[10] Priya Goyal, Quentin Duval, Jeremy Reizenstein, Matthew Leavitt, Min Xu, Benjamin Lefaudeux, Mannat Singh, Vinicius Reis, Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Ishan Misra. Vissl. https://github.com/facebookresearch/vissl, 2021. 3

[11] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Medical Image Analysis*, 58:101563, 2019. 2, 3, 6

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33:21271–21284, 2020. 4

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer, 2016. 4

[15] Jakob Nikolas Kather, Niels Halama, and Alexander Marx. 100,000 histological images of human colorectal cancer and healthy tissue, Apr. 2018. 1

[16] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 3

[17] Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering*, 5(6):555–570, 2021. 7

[18] Marc Macenko, Marc Niethammer, James S Marron, David Borland, John T Woosley, Xiaojun Guan, Charles Schmitt, and Nancy E Thomas. A method for normalizing histology

slides for quantitative analysis. In *ISBI*, pages 1107–1110. IEEE, 2009. 1

[19] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001. 3

[20] Yiqing Shen, Yulin Luo, Dinggang Shen, and Jing Ke. Randstainna: Learning stain-agnostic features from histology slides by bridging stain augmentation and normalization. In *MICCAI*, pages 212–221. Springer, 2022. 2

[21] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *MICCAI*, pages 210–218. Springer, 2018. 1

[22] Jerry Wei, Arief Suriawinata, Bing Ren, Xiaoying Liu, Mikhail Lisovsky, Louis Vaickus, Charles Brown, Michael Baker, Naofumi Tomita, Lorenzo Torresani, et al. A petri dish for histopathology image analysis. In *International Conference on Artificial Intelligence in Medicine*, pages 11–24. Springer, 2021. 2

[23] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 3

[24] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, pages 12310–12320. PMLR, 2021. 3, 4