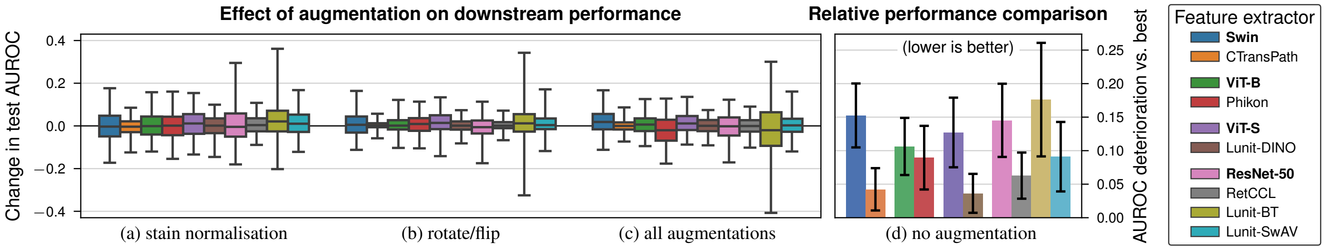
A Good Feature Extractor Is All You Need for Weakly Supervised Learning in Histopathology

Georg Wölflein[1,2,*] Dyke Ferber[2,3] Asier R. Meneghetti[2] Omar S. M. El Nahhas[2] Daniel Truhn[4]
Zunamys I. Carrero[2] David J. Harrison[1,5] Ognjen Arandjelović[1] Jakob N. Kather[2,3,6]

[1]University of St Andrews [2]EKFZ for Digital Health, TU Dresden [3]University of Heidelberg
[4]University Hospital Aachen [5]Lothian NHS University Hospitals [6]University Hospital Dresden



(a) stain normalisation (b) rotate/flip (c) all augmentations (d) no augmentation

Figure 1. **Stain normalisation and image augmentations do not impact downstream performance.** We empirically evaluate ten feature extractors across nine weakly supervised pathology tasks, observing no benefit in employing stain normalisation (a) or augmentations (b, c) before feature extraction. The best models (d), Lunit-DINO and CTransPath, are particularly robust, unlike ImageNet baselines (**bold**).

Abstract

Deep learning is revolutionising pathology, offering novel opportunities in disease prognosis and personalised treatment. Historically, stain normalisation has been a crucial preprocessing step in computational pathology pipelines, and persists into the deep learning era. Yet, with the emergence of feature extractors trained using self-supervised learning (SSL) on diverse pathology datasets, we call this practice into question. In an empirical evaluation of publicly available feature extractors, we find that omitting stain normalisation and image augmentations does not compromise downstream performance, while incurring substantial savings in memory and compute. Further, we show that the top-performing feature extractors are remarkably robust to variations in stain and augmentations like rotation in their latent space. Contrary to previous patch-level benchmarking studies, our approach emphasises clinical relevance by focusing on slide-level prediction tasks in a weakly supervised setting with external validation cohorts. This work represents the most comprehensive robustness evaluation of public pathology SSL feature extractors to date, involving more than 6,000 training runs across nine tasks, five datasets, three downstream architectures, and various preprocessing setups. Our findings stand to streamline digital pathology workflows by minimising preprocessing needs and informing the selection of feature extractors.

1. Introduction

There has been a recent surge in studies using deep learning in oncology to predict clinical variables such as genetic alterations and survival directly from routinely available histopathology whole slide images (WSIs) [23, 30, 32, 47, 54, 57, 62, 89, 97]. Due to their immense size, reaching billions of pixels at 20× magnification, these images are first divided into small, non-overlapping patches. What follows can be broken down into two stages: (i) feature extraction, where a feature vector is obtained separately for each patch and (ii) feature aggregation, where the extracted feature vectors are combined to form the slide-level prediction [7, 74]. Both steps are parametrised using neural networks; usually, the feature extractor is a deep backbone architecture whose parameters are frozen[1], while the aggregator is shallower, but trainable. In the past, convolutional neural networks (CNNs) such as ResNet-50 [37] pretrained on ImageNet [26] were used to perform feature extraction.

Recent advances in SSL make it possible to train powerful feature extractors without labels, a development that is gaining traction in the field of computational pathology, where large quantities of images are available but annotated data is sparse. As such, the last few years have witnessed the emergence of several SSL models trained on large-scale pathology datasets [2, 15, 16, 31, 44, 58, 87, 91–93]. These models produce better representations for downstream tasks than their ImageNet-pretrained counterparts [8, 14, 22, 25, 44, 75], and are quickly establishing themselves as the leading choice for feature extraction [30, 36, 62, 68, 89, 95, 96].

In computational pathology, stain normalisation has tra-

[*] georgwoelflein.de

[1] Employing a frozen feature extractor has significant computational benefits because the feature vectors can be pre-computed before training.

ditionally been a standard preprocessing step which was crucial in order to account for variations in scanners and haematoxylin and eosin (H&E) stains by adjusting WSIs to match a reference image [59, 64, 83]. Yet, with the shift from ImageNet CNNs to SSL models trained on vast and varied data from multiple centres, it is worth reconsidering its need. Beyond stain normalisation, image augmentations are a broad category of image-to-image transformations that may be applied during training, such as random flips, rotations, and colour transformations. Some augmentations, like rotation, are particularly well-suited for pathology due to the rotational invariance of micrographs [69]. SSL feature extractors that have been trained on a wide variety of images from multiple international sites might therefore extract diagnostically/prognostically relevant features irrespective of site- or scanner-specific traits. This leads to our primary research question: *with SSL feature extractors trained on rich datasets, is there still a need for image augmentations and stain normalisation to improve the generalisability of weakly supervised learning models?* Our study approaches this question in two ways:

1. We assess the latent space similarity between original patches and their stain-normalised/augmented counterparts in Sec. 3. Our analysis reveals that many augmentations induce only minor perturbations in the extracted features, especially compared to ImageNet backbones.
2. In the most comprehensive robustness evaluation of publicly available pathology SSL feature extractors to date, we compare over 6,000 trained models, both with and without normalisation/augmentation, across multiple externally validated slide-level tasks to determine whether the increased preprocessing effort holds merit in terms of downstream performance (Fig. 1 and Sec. 4).

Our analysis has implications for computational pathology practitioners and researchers alike, given the overhead incurred by image augmentations and stain normalisation in feature extraction pipelines. Code is available publicly[2].

1.1. Problem formulation

In a slide classification task, we have a dataset of labelled WSIs. Each WSI $X \in \mathbb{R}^{W \times H \times 3}$ is a RGB image of width W and height H, but dimensions vary between slides. It is associated with a ground truth label $y \in \mathcal{Y} = \mathbb{R}^c$ for a c-way classification problem. Due to their large size, we usually consider each WSI as a bag of patches, framing the WSI classification problem as a weakly supervised learning task. More specifically, we split each WSI X into a set of n non-overlapping patches $\{x_1, x_2, \ldots, x_n\}$ where each $x_i \in \mathcal{X} = \mathbb{R}^{P \times P \times 3}$ for fixed patch size P. Here, n varies depending on the particular slide's dimensions (usually between 1,000 and 10,000 at 10× magnification with $P = 224$). The task is to find a model $M : \mathcal{X}^n \to \mathcal{Y}$ that

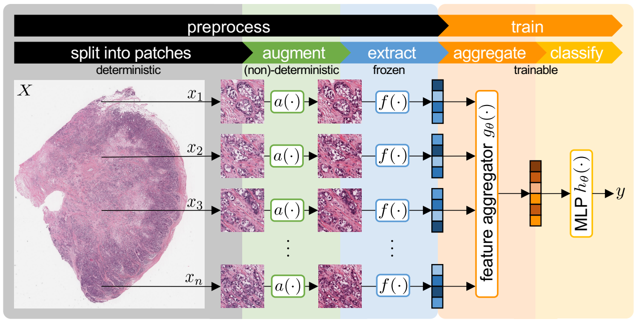


Figure 2. Common setup for weakly supervised learning on WSIs. In the preprocessing stage, the input image is split into patches that undergo independent image augmentations a before feature extraction. The feature aggregator and classifier are trained jointly as a single neural network $g_\theta \circ h_\theta$, which, given the feature vectors as inputs, predicts the output y. For stain normalisation (shown here), the same $a(\cdot)$ is applied every time, though in general, the augmentation function may vary between patches and epochs.

predicts the label given a bag of patches representing a WSI.

It is computationally infeasible to parametrise M using a single deep neural network trained end-to-end. Instead, the common approach in the literature is a two-step process consisting of preprocessing (feature extraction) and training (aggregation and classification), outlined in Fig. 2. The preprocessing stage often entails stain normalisation, and may optionally include image augmentations as well.

We first consider the simple case with a predetermined augmentation function $a : \mathcal{X} \to \mathcal{X}$ that is applied independently to each patch x_i to obtain the augmented patches $\hat{x}_i = a(x_i)$ for $i = 1, 2, \ldots, n$. Then, we apply the feature extractor $f : \mathcal{X} \to \mathbb{R}^{d_x}$, which for each patch $\hat{x}_i$ outputs a d_x-dimensional feature vector $z_i = f(\hat{x}_i)$. Now, we have n feature vectors, $z_1, z_2, \ldots, z_n$, which are aggregated into a single vector $\bar{z} \in \mathbb{R}^{d_z}$ (usually $d_x = d_z$) via an aggregation function $g_\theta : \mathbb{R}^{n \times d_x} \to \mathbb{R}^{d_z}$ with learnable parameters θ. Finally, the aggregated feature vector $\bar{z}$ passes through a classifier $h_\theta : \mathbb{R}^{d_z} \to \mathcal{Y}$, to obtain the final prediction. In summary, we can express the process $M : \mathcal{X}^n \to \mathcal{Y}$ of obtaining a prediction y from a bag of patches $\{x_i\}_{i=1}^n$ as

$$M(\{x_i\}_{i=1}^n) = \underbrace{(h_\theta \circ g_\theta)\,(\overbrace{\{(f \circ a)(x_i)\}_{i=1}^n}^{\text{preprocessing}})}_{\text{training}}, \tag{1}$$

where $\circ$ denotes function composition. Notice that $f \circ a$ is independent of the learnable parameters θ and thus can be pre-computed for all patches x_i before training.

In the general case, we define a set of augmentation functions $\mathcal{A} \in \mathcal{X}^\mathcal{X}$ before training ($\mathcal{X}^\mathcal{X}$ is the set of functions from $\mathcal{X}$ to $\mathcal{X}$). During training, for every patch x_i, we uniformly sample[3] an augmentation $a_i \sim \mathcal{A}$. Then, the augmented feature vector is $\hat{x}_i = a_i(x_i)$, so Eq. (1) becomes

$$M(\{x_i\}_{i=1}^n) = (h_\theta \circ g_\theta)\,(\{(f \circ a_i)(x_i)\}_{i=1}^n)\,. \tag{2}$$

[2] https://github.com/georg-wolflein/histaug

[3] The augmentation is resampled for every patch at every epoch.

While just a small modification in terms of notation, this change incurs a significant increase in time and memory complexity of the preprocessing task by a factor of $|\mathcal{A}|$, since augmentation and feature extraction must be performed for all possible augmentations $a_i \in \mathcal{A}$ for every patch[4]. As a result of this overhead, practitioners must carefully choose which augmentations to apply, if any. We address this problem by assessing the performance benefit obtained by different augmentations on our benchmark tasks.

2. Related work

Weakly supervised WSI classification Early work on WSI classification with slide-level labels employed CNNs such as ResNet [37] which were pretrained on ImageNet [26] and then fine-tuned on the classification task using slide-level labels as patch-level supervision [23, 47]. Recognising that this approach introduces excessive noise in the patch-level supervision to the detriment of the training process, later work [42, 90] reframed this task as an embedding-based multiple instance learning (MIL) problem [29]. In this line of work, a feature vector is extracted for every patch using a CNN (f in Fig. 2), and these feature vectors are aggregated and classified via a learnable pooling function and classifier ($h_\theta \circ g_\theta$ in Fig. 2). Initially, the entire network, including feature extraction, was trained end-to-end [42]. However, end-to-end training becomes intractable as MIL approaches scale to larger datasets, so more recent models operate on frozen features extracted using ImageNet pretrained models [57]. The frozen feature approach is now widely adopted for weakly supervised learning on WSIs, albeit with better feature extractors trained using SSL.

SSL in pathology The goal of SSL is to learn useful representations for downstream tasks from unlabelled data. Unlike supervised learning, SSL leverages structures inherent in data through pretext tasks, without needing explicit labels. The development of SSL models is an active area of research, from which a variety of algorithms like contrastive learning [17, 20, 38], non-contrastive learning [35, 99] and clustering-based methods [10, 11] have emerged in recent years, each with unique advantages and challenges. These models have quickly found adoption in the pathology field, which is well-situated to benefit from SSL due to the availability of large datasets that lack patch-level labels. Indeed, SSL feature extractors pretrained on pathology data have been shown to outperform ImageNet pretrained models on downstream pathology tasks [8, 14, 44, 72, 75]. It is also not surprising that obtaining more diverse data (*e.g.* from multiple centres) improves generalisability [72].

In the last three years, a number of SSL models have been developed [2, 15, 16, 31, 44, 51, 58, 87, 91–93] that were pretrained on large multi-centre pathology datasets, such as The Cancer Genome Atlas (TCGA) [94]. Wang *et al.* [91, 92] proposed CTransPath, a Swin Transformer [53] feature extractor trained using semantically-relevant contrastive learning (SRCL), a novel SSL technique based on MoCo v3 [20] specifically tailored to pathology. Previously, they had put forth RetCCL [93], a ResNet-50 model trained using a SSL technique they termed clustering-guided contrastive learning (CCL) based on MoCo [38]. Owkin [31] evaluated different ViT variants [49] using the iBOT framework [101], and terming their best ViT-B variant "Phikon"[5]. Lunit [44] benchmarked various SSL techniques including Barlow Twins [99], SwAV [11], MoCo v2 [19], and DINO [12] for pathology by training them on TCGA. All of the aforementioned models are available publicly, and – with one exception[6] – form the basis of our study. We refer the reader to Appendix B for a more detailed overview.

This year, a number of pathology foundation models have emerged [2, 8, 16, 58, 87] that were trained on considerably larger datasets. Unfortunately, we could not include these in our study since their weights remain proprietary, yet provide a more detailed account of these in Appendix B.1.

Stain normalisation Different medical sites employ different microscopes, scanners, protocols, and dyes, resulting in variations in the appearance of WSIs. For over 20 years [59, 64, 67], stain normalisation has been commonplace in digital pathology workflows to account for these factors by adjusting colours to match a reference image. Classical techniques [59, 64, 83] achieve this by performing colour deconvolution, standardising stain intensity, and then transforming the colour space of the input images to that of a reference image. More recently, GAN-based approaches have been proposed to this end as well [61, 88, 98]. Boschman *et al.* [5] compared eight classical and GAN-based stain normalisation techniques, concluding that stain normalisation, especially the methods of Vahadane *et al.* [83] and Macenko *et al.* [59], can indeed bolster slide-level classification performance when validating on external datasets. However, their approach aggregated patch-level predictions through a simplistic majority vote and did not integrate SSL feature extractors. In contrast, we contend that with SSL feature extractors, stain normalisation becomes obsolete. To show this, we focus our analysis on Macenko normalisation [59], the technique most widely adopted in the literature [21, 30, 32, 71].

Image augmentations As a common regularisation technique for neural network training in general [24], image augmentationsand have unsurprisingly found widespread adoption in histopathology as well [69]. In this field, the

[4] If the number of augmentations $|\mathcal{A}|$ is smaller than the number of training epochs, it is cheaper to pre-compute all augmentations before training. Otherwise, it is better to sample the augmentations for every patch and epoch before training, and just pre-compute for those combinations.

[5] In this paper, we employ the authors' initially recommended 'student' model, but note that this recommendation has since changed (Appendix B).

[6] To save computational resources, we excluded Lunit's MoCo model because both CTransPath and RetCCL already employ MoCo.

most popular augmentations include flipping, scaling, rotating, and colour alterations due to the nature of pathology slides [69], though a recent line of research introduces "stain augmentation" as a combination of stain normalisation and image augmentations to increase data diversity as well [60, 73, 80]. In this work, we study 26 image augmentations, focusing our analysis on those popular in pathology.

Robustness of feature extractors in pathology Assessing the robustness and generalisation ability of deep learning pathology models in the face of domain shift and out of distribution (OOD) data is an active area of research [33, 43, 70, 100] and an important undertaking, considering the stakes may be human life. Our work builds upon Lunit's aforementioned SSL benchmarking initiative [44], which involves training and evaluating four pathology-oriented SSL feature extractors; we have integrated three of these into our study[6]. Lunit's evaluation, however, is confined to patch classification and nuclei segmentation. While such tile-based tasks are scientifically interesting and the predominant means of evaluation in the literature [44, 78, 81], it has been suggested [8] that for evaluations to have greater clinical relevance, they should instead focus on slide-level tasks – predicting patient variables such as prognostic outcomes and biomarkers – and validate results on independent external cohorts. In response to this, we evaluate a total of six SSL feature extractors across nine slide-level classification targets (whose clinical utility we detail in Appendix A), and use external cohorts that were unseen during training (including both SSL pretraining and downstream training).

Similar to our work, Tellez *et al*. [81] explore the influence of stain normalisation and image augmentations on the generalisability of pathology models. However, their 2019 study predates SSL models trained on expansive pathology datasets akin to those employed in our evaluation; their analysis is limited to CNNs trained from scratch on narrow patch classification tasks. Springenberg *et al*. [78] empirically assess the robustness of CNNs and ViTs in pathology with and without self-supervised pretraining (CTransPath [92] and RetCCL [93]), but their evaluation, again, is confined to patch classification. Sikaroudi *et al*. [75] compare the OOD generalisability of pathology pretrained models (focusing on supervised and self-supervised models trained on natural images as well as a non-SSL pathology-specific model [65], the latter achieving the best results), but also only consider patch classification.

3. Effect on latent space

An ideal feature extractor for pathology extracts meaningful features from a patch. More specifically, it should:

1. be invariant to factors we deem unimportant, *e.g*. stain, orientation, etc.; and
2. vary with properties we are interested in, *e.g*. tissue type, cell type, and many other factors not known *a priori*.

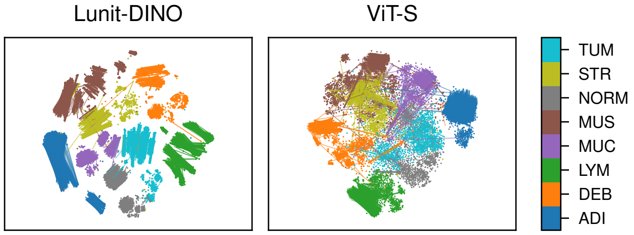


Figure 3. Latent space visualisations (t-SNE [84]) of features extracted with Lunit-DINO (left) *vs*. ImageNet baseline (right). Colours represent tissue classes [45]. Both feature extractors use the same architecture (ViT-S), but the left was trained on pathology images using SSL. Each dot represents a feature vector in latent space extracted from an unaltered image patch, and we draw a line from that dot to the corresponding stain-normalised version.

For example, a good feature extractor will produce a similar embedding for a particular patch and its stain-normalised version (as we want the feature extractor to be invariant to this factor), but yield very different embeddings for two patches of different tissue classes (*i.e*. normal *vs*. tumour).

In this section, we study the effect of various augmentations on the latent space, beginning with stain normalisation. We employ the NCT-CRC-HE-100K dataset [45, 46], comprising 100,000 patches extracted from H&E images of colorectal cancer (CRC) without stain normalisation. This dataset includes patch-level labels of the tissue type which enables more fine-grained analysis and visualisation.

3.1. Stain normalisation

How similar are feature vectors extracted from image patches to those derived from their stain-normalised counterparts? We contend that simply looking at the average distance between original embeddings and their stain-normalised versions does not provide enough information to make claims about the quality of a feature extractor. To obtain a more nuanced view of how stain normalisation affects embeddings, we present a dimensionality-reduced latent space visualisation of Lunit's DINO feature extractor in Fig. 3. This feature extractor is highlighted due to its superior downstream performance (see Fig. 1d and analysis in Sec. 4.1). In our visualisation, each point corresponds to a feature vector, with a line connecting each original feature vector to its stain-normalised version. Notably, Lunit-DINO clusters tissue types in latent space and the displacement of the feature vectors induced by stain normalisation is largely confined to these clusters. In contrast, a baseline extractor using the same ViT-S architecture [53] but trained via supervised learning on ImageNet, demonstrates less effective clustering and exhibits a different pattern: some features move hardly at all while others make large jumps between clusters, as indicated by the longer inter-cluster lines in Fig. 3, right. In fact, this pattern is consistent across various feature extractors: those pretrained on pathology data are less prone to "jump" between tissue type clusters compared to their ImageNet-pretrained counterparts when undergoing

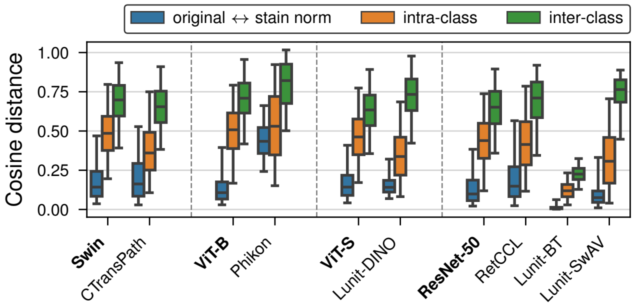


Figure 4. Boxplot of cosine distances between patch embeddings and their stain-normalised versions, as well as between embeddings of randomly chosen patches of the same or of differing tissue types. Feature extractors are grouped by architecture (ImageNet baselines are **bold**). Whiskers represent 95% of the distances.

stain normalisation, further detailed in Appendix D.

In Fig. 4, we compare the cosine distances of the embedding displacement caused by stain normalisation across all ten feature extractors. Despite the important difference in terms of intra-cluster *vs*. inter-cluster jumps identified in the latent space visualisation above, Lunit-DINO and ViT-S exhibit similar averages (*cf*. their medians in Fig. 4, blue). This observation highlights the importance of examining the distribution of distances, not merely their averages: the boxplot in Fig. 4 reflects this difference by the increased range of the whiskers of ViT-S compared to Lunit-DINO.

We note that an analysis that considers embedding displacement only from the perspective of stain normalisation is insufficient to make meaningful claims about feature extractor utility. For example, an extractor that maps all images to a single point in latent space would negate any embedding displacement induced by stain normalisation and prevent inter-cluster jumps, yet its features would be wholly useless to the downstream model. This observation leads us to also consider the second key criterion outlined at the beginning of this section: the ability of feature extractors to vary embeddings according to characteristics critical for downstream tasks. We select tissue type a surrogate marker to investigate this second criterion. However, it is important to recognise as a limitation of this analysis that there are numerous other potentially significant characteristics that remain unidentified at this stage, and for which specific labels are unavailable. Nonetheless, we posit that feature vectors from similar tissue types (indicated in blue in Fig. 4) should be closer in latent space compared to those from different tissue types (shown in green). Upon examining the disparity between these distance measures, we find that the ImageNet baselines tend to lump all features more closely together, regardless of tissue type. In contrast, the SSL models show better differentiation, as indicated by a greater separation between the blue and green boxes in the boxplot. Furthermore, the extent to which patches of different tissue types are distanced in the latent space (green) also provides a useful scale for contextualising the original *vs*. stain-normalised distances (blue). These findings suggest that the choice of pretraining data influences the stability of feature vectors in the context of stain normalisation. More specifically, feature extractors that have seen diverse stains as part of their SSL pretraining can learn to become more robust to variations in stain, while still preserving variations in aspects relevant to downstream tasks, *i.e*. tissue type.

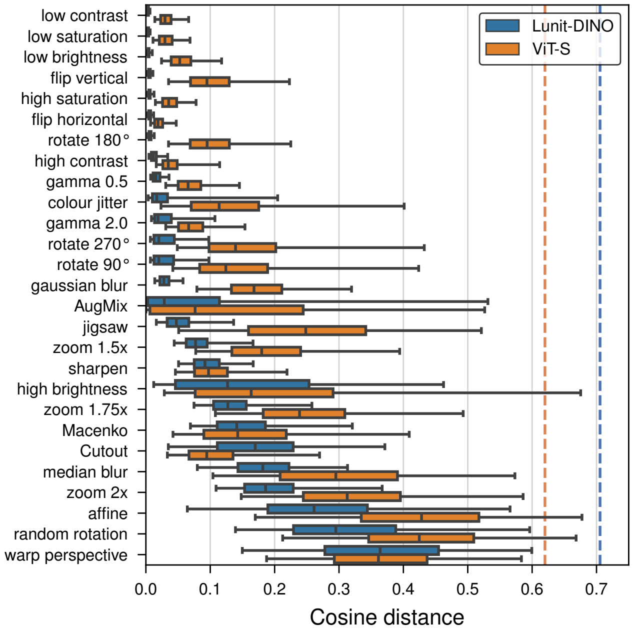


Figure 5. Boxplot of embedding displacement induced by image augmentations for Lunit-DINO and ViT-S. Dashed lines represent the average distance between randomly selected patches (without augmentation), indicating how 'dispersed' the latent spaces are.

3.2. Image augmentations

In principle, the methodology presented above is suitable to study how *any* transformation of the input patches, not just stain normalisation,manifests itself in latent space. Here, we consider 26 common image augmentations, for which we provide representative examples in Appendix D. For Lunit-DINO and the ViT-S baseline, we compare the magnitudes of the embedding displacement across augmentations in Fig. 5. We observe that Lunit-DINO's embeddings are more robust to image augmentations compared to the ImageNet baseline: for all augmentations except 'Cutout' [27] and 'warp perspective', the cosine distances tend to be smaller in Lunit-DINO. That is even though Lunit-DINO's embeddings are spread out more in latent space, *i.e*. the average distance between any two randomly selected non-augmented patches is greater, indicated by the dashed lines in Fig. 5. When normalising the distances by this average, Cutout remains as the only (minor) exception.

We observe that Lunit-DINO excels in terms of robustness to right-angle rotations and flips – a much desired property considering that WSIs, unlike natural images, lack a canonical orientation. In fact, in selecting augmentations

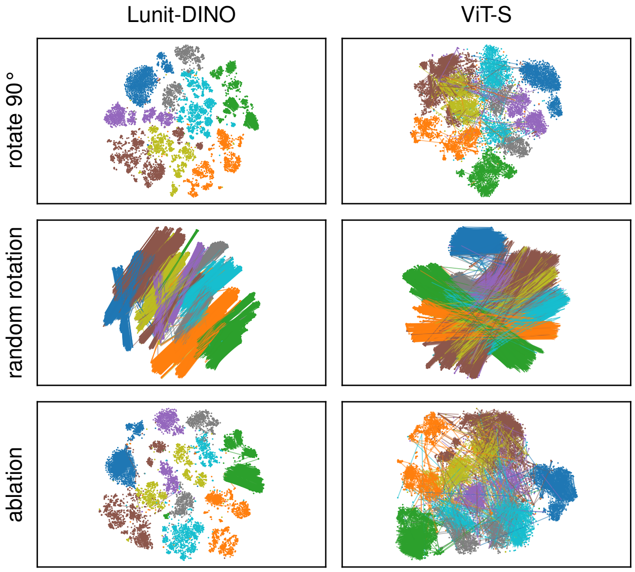


Figure 6. Visualisations of latent space transformations caused by rotation-based augmentations (rows) in Lunit-DINO (left) and ViT-S (right). Colours and lines are as explained in Fig. 3. Top row: 90° rotation. Middle: rotating by a random angle. Bottom (ablation): each line represents the transformation from (a) the embedding of the $1.5\times$ zoomed version of a patch to (b) the embedding obtained by randomly rotating before the $1.5\times$ zoom.

for generating positive pairs for SSL pretraining of the Lunit feature extractors, Kang *et al.* [44] employed the aforementioned augmentations for this precise reason, incentivising rotated/flipped embeddings to be close in latent space. On the other hand, the ImageNet baseline is significantly less robust to such augmentations. Interestingly, it is more robust to horizontal flips than vertical flips, which may be explained by the fact that it was trained on natural images.

Although Lunit-DINO is remarkably robust to rotating by angles that are multiples of 90°, non-right angles cause the greatest displacement in latent space aside from perspective warp (penultimate row in Fig. 5). To investigate this further, we visualise the latent space in Fig. 6. As expected, for 90° rotation (top row), Lunit-DINO's latent space remains largely unchanged, as opposed to ViT-S. However, for random angle rotations (middle row), we observe a high degree of chaotic jumps in both feature extractors, indicating neither is robust to this augmentation. We hypothesise that this is caused by the loss of pixels at the edges of the patches in off-angle rotations, and design an ablation study to investigate this phenomenon. To eliminate the black pixel problem, we perform a centercrop on the original and augmented patches in a manner that ensures there are no black pixels in any rotation. The corresponding latent space visualisations at the bottom of Fig. 6 confirm our assumption: Lunit-DINO's latent space remains unchanged whereas ViT-S's embeddings move significantly. Similar reasoning may explain the poor robustness regarding 'random affine', 'warp perspective', and 'Cutout' [27].

4. Impact on downstream performance

Motivated by the findings above – that some augmentations have larger effects than others on the latent representations, we investigate in the remainder of this paper how stain normalisation and augmentations affect downstream performance. To do so, we train weakly supervised models on nine downstream tasks using publicly available datasets.

Models We compare three parametrisations of the downstream aggregation model $g_\theta(\cdot)$ in Eq. (2): (1) AttMIL [42], the most common approach in the literature, (2) a two-layer decoder-only transformer [86], which is gaining popularity in recent works [89, 95], and (3) a simple baseline performing mean average pooling across features ($g_\theta(\{x_i\}_{i=1}^n) = \frac{1}{n}\sum_{i=1}^n x_i$ where each x_i is a feature vector). In our experiments, we parametrise $h_\theta(\cdot)$ as a linear layer with softmax activation over the number of classes for the particular task.

Tasks and datasets In selecting downstream tasks, we prioritise those with clinical utility and whose underlying variables are also available in adequately sized public datasets. Training on TCGA-BRCA [94] and testing on CPTAC-BRCA [50], we predict four breast cancer (🎗) targets: subtype as well as the CDH1, TP53, and PIK3CA genetic mutations. Furthermore, we predict 🎗-lymph node status in the CAMELYON17 breast cancer dataset [4] (which contains data from five centres – we used one of the centres for testing and the others for training). Finally, we predict four markers in colorectal cancer (): MSI status as well as BRAF, KRAS, and SMAD4 genetic mutations (training on TCGA-CRC [94] and testing on CPTAC-COAD [85]). We elaborate on these variables, their clinical relevancy, and the underlying datasets in Appendix A.

The aforementioned choice of tasks and datasets uses external cohorts for testing, so that we can assess generalisability to unseen datasets. We were also diligent in ensuring no data leakage occurred between the SSL pretraining and downstream test datasets. Notably, given that all evaluated pathology feature extractors included TCGA in their pretraining, we deliberately chose other datasets for testing.

Training details We train each model using the AdamW optimiser [56] with an initial learning rate of 0.001 which is decayed using cosine annealing [55] for up to 30 epochs, though training typically ends sooner due to our use of early stopping (when the validation loss fails to improve for ten consecutive epochs). For this, we allocate 80% of the training set for model training and 20% for validation. We conduct training with five distinct random seeds for the cartesian product of the ten feature extractors, nine tasks, three downstream models, and six preprocessing/augmentation setups (slidewise or patchwise stain normalisation, rotate/flip, all augmentations, or none), resulting in over 6,000 trained models. The training and validation splits are kept fixed per-task across the seeds for all tasks except for lymph node status classification. This latter

Feature extractor	🎗-subtype	🎗-CDH1	🎗-TP53	🎗-PIK3CA	🎗-LN status	-MSI	-KRAS	-BRAF	-SMAD4	Average
Swin [53]	0.07 ± 0.02	0.17 ± 0.03	0.28 ± 0.02	0.07 ± 0.04	0.17 ± 0.08	0.18 ± 0.04	0.14 ± 0.04	0.14 ± 0.07	0.16 ± 0.05	0.15 ± 0.05
CTransPath [92]	**0.00 ± 0.00**	**0.01 ± 0.01**	**0.01 ± 0.01**	0.04 ± 0.03	**0.06 ± 0.07**	0.08 ± 0.03	0.06 ± 0.03	0.06 ± 0.03	0.06 ± 0.03	**0.04 ± 0.03**
ViT-B [49]	0.08 ± 0.04	0.11 ± 0.02	0.15 ± 0.03	0.07 ± 0.03	0.17 ± 0.06	0.15 ± 0.03	0.03 ± 0.04	0.18 ± 0.07	**0.01 ± 0.01**	0.11 ± 0.04
Phikon [31]	0.09 ± 0.02	0.09 ± 0.02	0.09 ± 0.03	0.09 ± 0.03	0.07 ± 0.06	0.06 ± 0.04	0.07 ± 0.04	0.07 ± 0.06	0.17 ± 0.08	0.09 ± 0.05
ViT-S [49]	0.13 ± 0.03	0.08 ± 0.03	0.14 ± 0.05	0.08 ± 0.05	0.19 ± 0.09	0.18 ± 0.04	0.06 ± 0.03	0.19 ± 0.04	0.08 ± 0.08	0.13 ± 0.05
Lunit-DINO [44]	0.08 ± 0.03	0.03 ± 0.03	0.03 ± 0.02	0.02 ± 0.03	0.07 ± 0.04	**0.00 ± 0.00**	0.06 ± 0.04	**0.02 ± 0.02**	0.02 ± 0.02	**0.04 ± 0.03**
ResNet-50 [37]	0.15 ± 0.03	0.09 ± 0.04	0.11 ± 0.03	**0.01 ± 0.02**	0.18 ± 0.08	0.22 ± 0.04	0.11 ± 0.03	0.23 ± 0.07	0.21 ± 0.09	0.15 ± 0.05
RetCCL [93]	0.07 ± 0.03	0.04 ± 0.02	0.04 ± 0.03	0.05 ± 0.03	0.07 ± 0.06	0.08 ± 0.03	**0.03 ± 0.02**	0.14 ± 0.03	0.06 ± 0.03	0.06 ± 0.03
Lunit-BT [44]	0.13 ± 0.03	0.06 ± 0.04	0.02 ± 0.01	0.13 ± 0.04	0.34 ± 0.15	0.28 ± 0.13	0.03 ± 0.04	0.35 ± 0.13	0.25 ± 0.03	0.18 ± 0.08
Lunit-SwAV [44]	0.06 ± 0.02	0.06 ± 0.03	0.06 ± 0.02	0.13 ± 0.06	0.07 ± 0.05	0.10 ± 0.03	0.13 ± 0.06	0.07 ± 0.07	0.14 ± 0.08	0.09 ± 0.05

Table 1. Comparative evaluation of feature extractors. This table presents the normalised differential AUROC scores (lower is better) for all feature extractors, across the evaluated targets using the AttMIL [42] aggregation model. The scores reflect the expected decrease in test AUROC when selecting a given feature extractor relative to the best-performing one for each task-model combination (see Sec. 4.1).

task uses the CAMELYON17 dataset, allowing us to perform leave-one-hospital-out cross-validation with a different random seed for each of the five hospitals. For the experiments involving augmentations, we apply these augmentations only on the images of the training datasets, never the test datasets (except for the stain normalisation experiments, where we ensure the same normalisation is applied to training and test datasets). We perform feature extraction once before training, caching for every patch in every dataset its original feature vector as well as the feature vectors of all 27 augmented versions of that patch, including stain normalisation. More details are provided in Appendix F.2.

4.1. Lunit-DINO and CTransPath extract the most useful features

Having trained a large number of downstream models based on ten feature extractors across a diverse set of tasks, we are in a position to identify the most effective feature extractor overall. We present these findings first and focus our later discussion on these feature extractors in particular.

First, let us consider how to determine the best feature extractor for a given task and downstream aggregator (such as predicting 🎗-CDH1 with AttMIL aggregation). For any such task-model pair, we trained 50 models – spanning the ten feature extractors across five random seeds. We define a 'trial' as one particular configuration pairing each feature extractor with a random seed, leading to 5^{10} ($\approx$ 10 million) unique trials. Within each trial, we evaluate the feature extractors based on the difference between their test area under the receiver operating characteristic curve (AUROC) and the highest test AUROC observed, thus assigning a score of zero to the top performer. By calculating the mean across all 5^{10} trials, we derive the 'normalised differential AUROC score' – a measure that captures the relative efficacy of the feature extractors and allows fair comparisons across tasks of varying difficulty. The outcomes of this analysis, when considering the downstream AttMIL model and no augmentations, are detailed individually per task in Tab. 1 and averaged across tasks in Fig. 1d. Notably, Lunit-DINO and CTransPath are tied in achieving best task-averaged performance. Indeed, they consistently perform best, regardless of downstream aggregation model and type of input augmentation, as we show in the extended data table in Appendix E. Moreover, we find the ImageNet baselines perform worse than the pathology models (with the exception of Lunit-BT which performs very poorly indeed), which is in line with previous work [8, 14, 22, 25, 44, 75].

4.2. Stain normalisation does not impact downstream performance

We quantify the effect of stain normalisation on downstream model performance by determining the expected difference in test AUROC between models trained with stain normalisation *vs.* without. Given a feature extractor and downstream aggregation model, *e.g.* Lunit-DINO with AttMIL, we must compare 45 models trained with stain normalisation (nine tasks times five random seeds) with another 45 models trained without stain normalisation. To estimate the difference in AUROC, we perform bootstrapping. For each of the 45 task-seed pairs, we generate 25 random resamples of the respective test datasets with replacement, totalling $45 \times 25 = 1{,}125$ bootstraps. Since each bootstrap is associated with a particular task-seed combination, it corresponds to two trained models: one trained with stain normalisation and one without. We deploy both models on the given bootstrap, computing the difference in AUROC. Repeating for all bootstraps, we obtain a distribution of 1,125 AUROC differences which we present as a boxplot in Fig. 1a, with a separate box for every feature extractor (we focus on the AttMIL [42] aggregation model because it is the most widely used, but provide analogous plots for the other two in Appendix E). We find no clear AUROC difference between the two groups, for any feature extractor: all 95% confidence intervals (and interquartile ranges) include zero. Surprisingly, this holds even for ImageNet extractors, whose latent spaces we previously identified more susceptible to larger displacements due to stain normalisation.

Slidewise versus patchwise normalisation While in Fig. 1a, we perform stain normalisation on a per-slide basis, a more computationally efficient[7] alternative is normalising

[7] Macenko normalisation [59] requires an eigenvalue decomposition across all pixels in the image, scaling cubically in the number of pixels. For a slide with n patches of k pixels each, the complexity is in $\mathcal{O}(n^3k^3)$ for slidewise normalisation, but $\mathcal{O}(nk^3)$ for patchwise normalisation. More-

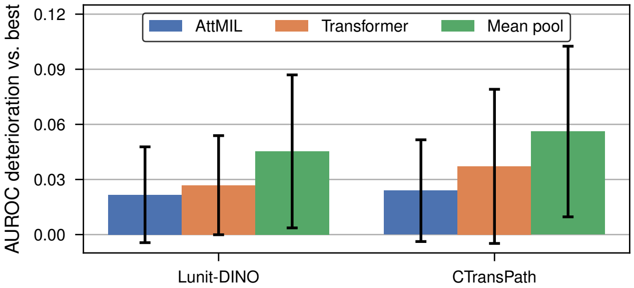


Figure 7. Performance impact of choosing a particular downstream aggregation model (lower is better).

each patch individually. However, in this approach, adjacent patches might experience different colour transformations, potentially affecting consistency across the slide. We perform an ablation study, detailed in Appendix C, where we employ the bootstrapping procedure from above with patchwise instead of slidewise normalisation, but find no consistent performance differences between both methods. Therefore, we recommend the patchwise approach for practitioners still seeking to employ stain normalisation in their preprocessing pipelines, due to its computational benefit.

4.3. Augmentations do not impact performance

Having emphasised rotation and reflection as augmentations of particular relevance to pathology in our investigation of the latent space, we now study their downstream impact on performance. To this end, we trained a batch of models where at each epoch, each patch is randomly flipped (horizontally or vertically) or rotated by a right angle before feature extraction. Analogous to our analysis of stain normalisation, we perform a bootstrapped quantification of the difference in performance incurred by employing the augmented versus non-augmented features in Fig. 1b. Again, we observe no consistent benefit in employing this type of augmentation. Interestingly, we find the variance in the differences to be even smaller in Lunit-DINO and CTransPath compared to stain normalisation in Fig. 1a. Furthermore, expanding the set of augmentations to all 27 studied transformations (with each being equally likely to be selected for every patch at every epoch) yields similar results (Fig. 1c). While Fig. 1 employs AttMIL [42], we come to the same conclusion for the other downstream aggregation architectures, for which we present extended results in Appendix E.

4.4. Downstream aggregation models

In Sec. 4.1, we identified Lunit-DINO and CTransPath as the best feature extractors in terms of achieving the lowest normalised differential AUROC scores averaged across all tasks, and found this to be the case for all three downstream models. Yet, it remains to be seen *which* downstream model achieves the best results. To answer this question, we employ the technique from Sec. 4.1, but instead of keeping fixed the downstream model and determining the best feature extractor, we choose a feature extractor and vary the downstream aggregation model. As shown in Fig. 7, AttMIL performs best, closely followed by the two-layer transformer, and finally mean average pooling, but we note the differences are small and exhibit high variance.

5. Discussion

We dedicate this section to answer key questions that may arise among computational pathology researchers about employing SSL feature extractors for slide-level prediction.

What is the best feature extractor for pathology? We recommend Lunit-DINO and CTransPath for feature extraction, since they consistently achieve the best task-averaged downstream performance (Fig. 1d), independent of the employed augmentations and downstream aggregation model. In general, we find pathology-specific extractors to outperform their ImageNet baselines, adding to the body of evidence suggesting that SSL models pretrained on pathology data produce more useful features [8, 14, 44, 72, 75].

Which aggregation model should we use? For Lunit-DINO and CTransPath, we notice slight benefits in employing the AttMIL aggregation model downstream, though the differences are small and exhibit high variance (Fig. 7). The primary factor remains the choice feature extractor.

Should we perform stain normalisation and augmentations? Our data does not support the necessity of either – they do not markedly improve outcomes, yet introduce significant preprocessing overhead. For those still interested in stain normalisation, we recommend the patchwise approach due to its lower computational cost. Image augmentations, on the other hand, should be avoided because in addition to the preprocessing cost, they considerably increase training time (Appendix F.2), and the top extractors resist pathology-relevant augmentations in their latent space (Sec. 3.2).

6. Conclusion and future work

In this work, we perform the most comprehensive robustness evaluation of publicly available pathology feature extractors for slide-level prediction to date, spanning ten feature extractors, three aggregation models, stain normalisation, and numerous image augmentations, on nine downstream weakly supervised learning tasks with external validation cohorts. Among these factors, we identify the choice of feature extractor as most consequential for downstream performance, and observe no benefit in employing stain normalisation or image augmentations.

Our latent space analysis reveals a remarkable robustness to stain variations and image augmentations in the top-performing feature extractors, Lunit-DINO [44] and CTransPath [92], which employ domain-specific knowledge in their SSL training regimes. This underlines a key

over, the latter is embarrassingly parallel across the n dimension.

direction for future research into the development of pathology feature extractors and foundation models [2, 8, 16, 58, 87]: the importance of not only scaling size and diversity of pretraining datasets, but also tailoring SSL methods to the pathology domain, in order to effectively leverage this data.

Looking ahead, we aim to investigate whether the ineffectiveness of augmentations persists in limited-data scenarios, and how slide magnification impacts extracted features.

References

[1] Fabrice André, Eva Ciruelos, Gabor Rubovszky, Mario Campone, Sibylle Loibl, Hope S Rugo, Hiroji Iwata, Pierfranco Conte, Ingrid A Mayer, Bella Kaufman, Toshinari Yamashita, Yen-Shen Lu, Kenichi Inoue, Masato Takahashi, Zsuzsanna Pápai, Anne-Sophie Longin, David Mills, Celine Wilke, Samit Hirawat, and Dejan Juric. Alpelisib for PIK3CA-Mutated, hormone Receptor–Positive advanced breast cancer. *N. Engl. J. Med.*, 380(20):1929–1940, 2019. 1

[2] Shekoofeh Azizi, Laura Culp, Jan Freyberg, Basil Mustafa, Sebastien Baur, Simon Kornblith, Ting Chen, Nenad Tomasev, Jovana Mitrović, Patricia Strachan, S Sara Mahdavi, Ellery Wulczyn, Boris Babenko, Megan Walker, Aaron Loh, Po-Hsuan Cameron Chen, Yuan Liu, Pinal Bavishi, Scott Mayer McKinney, Jim Winkens, Abhijit Guha Roy, Zach Beaver, Fiona Ryan, Justin Krogue, Mozziyar Etemadi, Umesh Telang, Yun Liu, Lily Peng, Greg S Corrado, Dale R Webster, David Fleet, Geoffrey Hinton, Neil Houlsby, Alan Karthikesalingam, Mohammad Norouzi, and Vivek Natarajan. Robust and data-efficient generalization of self-supervised machine learning for diagnostic imaging. *Nat Biomed Eng*, 7(6):756–779, 2023. 1, 3, 9, 2

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[4] Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018. 6, 1

[5] Jeffrey Boschman, Hossein Farahani, Amirali Darbandsari, Pouya Ahmadvand, Ashley Van Spankeren, David Farnell, Adrian B Levine, Julia R Naso, Andrew Churg, Steven Jm Jones, Stephen Yip, Martin Köbel, David G Huntsman, C Blake Gilks, and Ali Bashashati. The utility of color normalization for AI-based diagnosis of hematoxylin and eosin-stained pathology images. *J. Pathol.*, 256(1):15–24, 2022. 3

[6] Bruno Buecher, Wulfran Cacheux, Etienne Rouleau, Barbara Dieumegard, Emmanuel Mitry, and Astrid Lièvre. Role of microsatellite instability in the management of colorectal cancers. *Dig. Liver Dis.*, 45(6):441–449, 2013. 1

[7] Gabriele Campanella, Matthew G Hanna, Luke Geneslaw, Allen Miraflor, Vitor Werneck Krauss Silva, Klaus J Busam, Edi Brogi, Victor E Reuter, David S Klimstra, and Thomas J Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nat. Med.*, 25(8):1301–1309, 2019. 1

[8] Gabriele Campanella, Ricky Kwan, Eugene Fluder, Jennifer Zeng, Aryeh Stock, Brandon Veremis, Alexandros D Polydorides, Cyrus Hedvat, Adam Schoenfeld, Chad Vanderbilt, Patricia Kovatch, Carlos Cordon-Cardo, and Thomas J Fuchs. Computational pathology at health system scale – Self-Supervised foundation models from three billion images. 2023. 1, 3, 4, 7, 8, 9, 2

[9] F Cardoso, S Kyriakides, S Ohno, F Penault-Llorca, P Poortmans, I T Rubio, S Zackrisson, E Senkus, and ESMO Guidelines Committee. Electronic address: clinicalguidelines@esmo.org. Early breast cancer: ESMO clinical practice guidelines for diagnosis, treatment and follow-up†. *Ann. Oncol.*, 30(8):1194–1220, 2019. 1

[10] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018. 3

[11] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Adv. Neural Inf. Process. Syst.*, 33:9912–9924, 2020. 3, 2

[12] Mathilde Caron, Hugo Touvron, Ishan Misra, Herve Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2021. 3, 2

[13] M Chalabi, Y L Verschoor, J Van den Berg, and others. LBA7 neoadjuvant immune checkpoint inhibition in locally advanced MMR-deficient colon cancer: The NICHE-2 study. *Annals of*, 2022. 1

[14] Richard J Chen and Rahul G Krishnan. Self-Supervised vision transformers learn visual concepts in histopathology. *Learning Meaningful Representations of Life, NeurIPS 2021*, 2021. 1, 3, 7, 8

[15] 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 hierarchical self-supervised learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16144–16155. IEEE, 2022. 1, 3

[16] Richard J Chen, Tong Ding, Ming Y Lu, Drew F K Williamson, Guillaume Jaume, Bowen Chen, Andrew Zhang, Daniel Shao, Andrew H Song, Muhammad Shaban, Mane Williams, Anurag Vaidya, Sharifa Sahai, Lukas Oldenburg, Luca L Weishaupt, Judy J Wang, Walt Williams, Long Phi Le, Georg Gerber, and Faisal Mahmood. A General-Purpose Self-Supervised model for computational pathology. 2023. 1, 3, 9, 2

[17] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020. 3, 2

[18] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big Self-Supervised

models are strong Semi-Supervised learners. In *Advances in Neural Information Processing Systems*, pages 22243–22255. Curran Associates, Inc., 2020. 2

[19] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. 2020. 3

[20] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2021. 3, 2

[21] Philip Chikontwe, Hyun Jung Sung, Jaehoon Jeong, Meejeong Kim, Heounjeong Go, Soo Jeong Nam, and Sang Hyun Park. Weakly supervised segmentation on neural compressed histopathology with self-equivariant regularization. *Med. Image Anal.*, 80:102482, 2022. 3

[22] Ozan Ciga, Tony Xu, and Anne Louise Martel. Self supervised contrastive learning for digital histopathology. *Machine Learning with Applications*, 7:100198, 2022. 1, 7

[23] Nicolas Coudray, Paolo Santiago Ocampo, Theodore Sakellaropoulos, Navneet Narula, Matija Snuderl, David Fenyö, Andre L Moreira, Narges Razavian, and Aristotelis Tsirigos. Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. *Nat. Med.*, 24(10):1559–1567, 2018. 1, 3

[24] Ekin Dogus Cubuk, Ethan S Dyer, Rapha Gontijo Lopes, and Sylvia Smullin. Tradeoffs in data augmentation: An empirical study. In *ICLR*, 2021. 3

[25] Olivier Dehaene, Axel Camara, Olivier Moindrot, Axel de Lavergne, and Pierre Courtiol. Self-Supervision closes the gap between weak and strong supervision in histology. 2020. 1, 7

[26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 1, 3

[27] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. 2017. 5, 6, 2

[28] R Dienstmann, M J Mason, F A Sinicrope, A I Phipps, S Tejpar, A Nesbakken, S A Danielsen, A Sveen, D D Buchanan, M Clendenning, C Rosty, B Bot, S R Alberts, J Milburn Jessup, R A Lothe, M Delorenzi, P A Newcomb, D Sargent, and J Guinney. Prediction of overall survival in stage II and III colon cancer beyond TNM system: a retrospective, pooled biomarker study. *Ann. Oncol.*, 28(5): 1023–1031, 2017. 1

[29] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89(1):31–71, 1997. 3

[30] Omar S M El Nahhas, Chiara M L Loeffler, Zunamys I Carrero, Marko van Treeck, Fiona R Kolbinger, Katherine J Hewitt, Hannah S Muti, Mara Graziani, Qinghe Zeng, Julien Calderaro, Nadina Ortiz-Brüchle, Tanwei Yuan, Michael Hoffmeister, Hermann Brenner, Alexander Brobeil, Jorge S Reis-Filho, and Jakob Nikolas Kather. Regression-based Deep-Learning predicts molecular biomarkers from pathology slides. *arXiv preprint arXiv*, 2023. 1, 3

[31] Alexandre Filiot, Ridouane Ghermi, Antoine Olivier, Paul Jacob, Lucas Fidon, Alice Mac Kain, Charlie Saillard, and Jean-Baptiste Schiratti. Scaling self-supervised learning for histopathology with masked image modeling. 2023. 1, 3, 7, 2

[32] Narmin Ghaffari Laleh, Hannah Sophie Muti, Chiara Maria Lavinia Loeffler, Amelie Echle, Oliver Lester Saldanha, Faisal Mahmood, Ming Y Lu, Christian Trautwein, Rupert Langer, Bastian Dislich, Roman D Buelow, Heike Irmgard Grabsch, Hermann Brenner, Jenny Chang-Claude, Elizabeth Alwers, Titus J Brinker, Firas Khader, Daniel Truhn, Nadine T Gaisa, Peter Boor, Michael Hoffmeister, Volkmar Schulz, and Jakob Nikolas Kather. Benchmarking weakly-supervised deep learning pipelines for whole slide classification in computational pathology. *Med. Image Anal.*, 79:102474, 2022. 1, 3

[33] Narmin Ghaffari Laleh, Daniel Truhn, Gregory Patrick Veldhuizen, Tianyu Han, Marko van Treeck, Roman D Buelow, Rupert Langer, Bastian Dislich, Peter Boor, Volkmar Schulz, and Jakob Nikolas Kather. Adversarial attacks and adversarial robustness in computational pathology. *Nat. Commun.*, 13(1):5711, 2022. 4

[34] A Goldhirsch, E P Winer, A S Coates, R D Gelber, M Piccart-Gebhart, B Thürlimann, H-J Senn, and Panel members. Personalizing the treatment of women with early breast cancer: highlights of the st gallen international expert consensus on the primary therapy of early breast cancer 2013. *Ann. Oncol.*, 24(9):2206–2223, 2013. 1

[35] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to Self-Supervised learning. In *Advances in Neural Information Processing Systems*, pages 21271–21284. Curran Associates, Inc., 2020. 3

[36] Yonghang Guan, Jun Zhang, Kuan Tian, Sen Yang, Pei Dong, Jinxi Xiang, Wei Yang, Junzhou Huang, Yuyao Zhang, and Xiao Han. Node-aligned graph convolutional network for whole-slide image representation and classification. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18813–18823. IEEE, 2022. 1

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 1, 3, 7, 2

[38] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020. 3, 2

[39] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15979–15988. IEEE, 2022. 2

[40] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 5

[41] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 2

[42] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2127–2136. PMLR, 2018. 3, 6, 7, 8, 4, 5

[43] Mostafa Jahanifar, Manahil Raza, Kesi Xu, Trinh Vuong, Rob Jewsbury, Adam Shephard, Neda Zamanitajeddin, Jin Tae Kwak, Shan E Ahmed Raza, Fayyaz Minhas, and Nasir Rajpoot. Domain generalization in computational pathology: Survey and guidelines. 2023. 4

[44] Mingu Kang, Heon Song, Seonwook Park, Donggeun Yoo, and Sérgio Pereira. Benchmarking self-supervised learning on diverse pathology datasets. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3344–3354. IEEE, 2023. 1, 3, 4, 6, 7, 8, 2

[45] Jakob Nikolas Kather, Niels Halama, and Alexander Marx. 100,000 histological images of human colorectal cancer and healthy tissue, 2018. 4

[46] Jakob Nikolas Kather, Johannes Krisam, Pornpimol Charoentong, Tom Luedde, Esther Herpel, Cleo-Aron Weis, Timo Gaiser, Alexander Marx, Nektarios A Valous, Dyke Ferber, Lina Jansen, Constantino Carlos Reyes-Aldasoro, Inka Zörnig, Dirk Jäger, Hermann Brenner, Jenny Chang-Claude, Michael Hoffmeister, and Niels Halama. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS Med.*, 16(1):e1002730, 2019. 4

[47] Jakob Nikolas Kather, Alexander T Pearson, Niels Halama, Dirk Jäger, Jeremias Krause, Sven H Loosen, Alexander Marx, Peter Boor, Frank Tacke, Ulf Peter Neumann, Heike I Grabsch, Takaki Yoshikawa, Hermann Brenner, Jenny Chang-Claude, Michael Hoffmeister, Christian Trautwein, and Tom Luedde. Deep learning can predict microsatellite instability directly from histology in gastrointestinal cancer. *Nat. Med.*, 25(7):1054–1056, 2019. 1, 3

[48] Yoo Jung Kim, Hyungjoon Jang, Kyoungbun Lee, Seongkeun Park, Sung-Gyu Min, Choyeon Hong, Jeong Hwan Park, Kanggeun Lee, Jisoo Kim, Wonjae Hong, Hyun Jung, Yanling Liu, Haran Rajkumar, Mahendra Khened, Ganapathy Krishnamurthi, Sen Yang, Xiyue Wang, Chang Hee Han, Jin Tae Kwak, Jianqiang Ma, Zhe Tang, Bahram Marami, Jack Zeineh, Zixu Zhao, Pheng-Ann Heng, Rüdiger Schmitz, Frederic Madesta, Thomas Rösch, Rene Werner, Jie Tian, Elodie Puybareau, Matteo Bovio, Xiufeng Zhang, Yifeng Zhu, Se Young Chun, Won-Ki Jeong, Peom Park, and Jinwook Choi. PAIP 2019: Liver cancer segmentation challenge. *Med. Image Anal.*, 67:101854, 2021. 2

[49] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 3, 7, 2

[50] Karsten Krug, Eric J Jaehnig, Shankha Satpathy, Lili Blumenberg, Alla Karpova, Meenakshi Anurag, George Miles, Philipp Mertins, Yifat Geffen, Lauren C Tang, et al. Proteogenomic landscape of breast cancer tumorigenesis and targeted therapy. *Cell*, 183(5):1436–1456, 2020. 6, 1

[51] Tristan Lazard, Marvin Lerousseau, Etienne Decencière, and Thomas Walter. Giga-ssl: Self-supervised learning for gigapixel images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4304–4313, 2023. 3

[52] Yang Liu, Nilay S Sethi, Toshinori Hinoue, Barbara G Schneider, Andrew D Cherniack, Francisco Sanchez-Vega, Jose A Seoane, Farshad Farshidfar, Reanne Bowlby, Mirazul Islam, et al. Comparative molecular analysis of gastrointestinal adenocarcinomas. *Cancer cell*, 33(4):721–735, 2018. 1

[53] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2021. 3, 4, 7, 2

[54] Chiara Maria Lavinia Loeffler, Omar S M El Nahhas, Hannah Sophie Muti, Tobias Seibel, Didem Cifci, Marko van Treeck, Marco Gustav, Zunamys I Carrero, Nadine T Gaisa, Kjong-Van Lehmann, Alexandra Leary, Pier Selenica, Jorge S Reis-Filho, Nadina Ortiz Bruechle, and Jakob Nikolas Kather. Direct prediction of homologous recombination deficiency from routine histology in ten different tumor types with attention-based multiple instance learning: a development and validation study. *medRxiv*, 2023. 1

[55] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 6, 5

[56] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. 6, 5

[57] Ming Y Lu, Drew F K Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nat Biomed Eng*, 5(6):555–570, 2021. 1, 3

[58] Ming Y Lu, Bowen Chen, Drew F K Williamson, Richard J Chen, Ivy Liang, Tong Ding, Guillaume Jaume, Igor Odintsov, Andrew Zhang, Long Phi Le, Georg Gerber, Anil V Parwani, and Faisal Mahmood. Towards a Visual-Language foundation model for computational pathology. 2023. 1, 3, 9, 2

[59] Marc Macenko, Marc Niethammer, J 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 *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1107–1110, 2009. 2, 3, 7, 8, 10

[60] Niccolò Marini, Sebastian Otalora, Marek Wodzinski, Selene Tomassini, Aldo Franco Dragoni, Stephane Marchand-Maillet, Juan Pedro Dominguez Morales, Lourdes Duran-Lopez, Simona Vatrano, Henning Müller, and Manfredo Atzori. Data-driven color augmentation for H&E stained images in computational pathology. *J. Pathol. Inform.*, 14: 100183, 2023. 4

[61] Haseeb Nazki, Ognjen Arandjelovic, In Hwa Um, and David Harrison. MultiPathGAN: Structure preserving stain normalization using unsupervised multi-domain adversarial network with perception loss. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, pages 1197–1204, New York, NY, USA, 2023. Association for Computing Machinery. 3

[62] Jan Moritz Niehues, Philip Quirke, Nicholas P West, Heike I Grabsch, Marko van Treeck, Yoni Schirris, Gregory P Veldhuizen, Gordon G A Hutchins, Susan D Richman, Sebastian Foersch, Titus J Brinker, Junya Fukuoka, Andrey Bychkov, Wataru Uegami, Daniel Truhn, Hermann Brenner, Alexander Brobeil, Michael Hoffmeister, and Jakob Nikolas Kather. Generalizable biomarker prediction from cancer pathology slides with self-supervised deep learning: A retrospective multi-centric study. *Cell Rep Med*, 4(4):100980, 2023. 1

[63] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. 2023. 2

[64] E Reinhard, M Adhikhmin, B Gooch, and P Shirley. Color transfer between images. *IEEE Comput. Graph. Appl.*, 21 (5):34–41, 2001. 2, 3

[65] Abtin Riasatian, Morteza Babaie, Danial Maleki, Shivam Kalra, Mojtaba Valipour, Sobhan Hemati, Manit Zaveri, Amir Safarpoor, Sobhan Shafiei, Mehdi Afshari, Maral Rasoolijaberi, Milad Sikaroudi, Mohd Adnan, Sultaan Shah, Charles Choi, Savvas Damaskinos, Clinton Jv Campbell, Phedias Diamandis, Liron Pantanowitz, Hany Kashani, Ali Ghodsi, and H R Tizhoosh. Fine-Tuning and training of densenet for histopathology image representation using TCGA diagnostic slides. *Med. Image Anal.*, 70:102032, 2021. 4

[66] Arnaud D Roth, Sabine Tejpar, Mauro Delorenzi, Pu Yan, Roberto Fiocca, Dirk Klingbiel, Daniel Dietrich, Bart Biesmans, György Bodoky, Carlo Barone, Enrique Aranda, Bernard Nordlinger, Laura Cisar, Roberto Labianca, David Cunningham, Eric Van Cutsem, and Fred Bosman. Prognostic role of KRAS and BRAF in stage II and III resected colon cancer: results of the translational study on the PETACC-3, EORTC 40993, SAKK 60-00 trial. *J. Clin. Oncol.*, 28(3):466–474, 2010. 1

[67] A C Ruifrok and D A Johnston. Quantification of histochemical staining by color deconvolution. *Anal. Quant. Cytol. Histol.*, 23(4):291–299, 2001. 3

[68] Oliver Lester Saldanha, Chiara M L Loeffler, Jan Moritz Niehues, Marko van Treeck, Tobias P Seraphin, Katherine Jane Hewitt, Didem Cifci, Gregory Patrick Veldhuizen, Siddhi Ramesh, Alexander T Pearson, and Jakob Nikolas Kather. Self-supervised attention-based deep learning for pan-cancer mutation prediction from histopathology. *NPJ Precis Oncol*, 7(1):35, 2023. 1

[69] Massimo Salvi, U Rajendra Acharya, Filippo Molinari, and Kristen M Meiburger. The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis. *Comput. Biol. Med.*, 128:104129, 2021. 2, 3, 4

[70] Birgid Schömig-Markiefka, Alexey Pryalukhin, Wolfgang Hulla, Andrey Bychkov, Junya Fukuoka, Anant Madabhushi, Viktor Achter, Lech Nieroda, Reinhard Büttner, Alexander Quaas, and Yuri Tolkach. Quality control stress test for deep learning-based diagnostic model in digital pathology. *Mod. Pathol.*, 34(12):2098–2108, 2021. 4

[71] Peter Leonard Schrammen, Narmin Ghaffari Laleh, Amelie Echle, Daniel Truhn, Volkmar Schulz, Titus J Brinker, Hermann Brenner, Jenny Chang-Claude, Elizabeth Alwers, Alexander Brobeil, Matthias Kloor, Lara R Heij, Dirk Jäger, Christian Trautwein, Heike I Grabsch, Philip Quirke, Nicholas P West, Michael Hoffmeister, and Jakob Nikolas Kather. Weakly supervised annotation-free cancer detection and prediction of genotype in routine histopathology. *J. Pathol.*, 256(1):50–60, 2022. 3

[72] Zhuchen Shao, Liuxi Dai, Jitendra Jonnagaddala, Yang Chen, Yifeng Wang, Zijie Fang, and Yongbing Zhang. Generalizability of Self-Supervised training models for digital pathology: A multicountry comparison in colorectal cancer. *JCO Clin Cancer Inform*, 7:e2200178, 2023. 3, 8

[73] Yiqing Shen, Yulin Luo, Dinggang Shen, and Jing Ke. RandStainNA: Learning Stain-Agnostic features from histology slides by bridging stain augmentation and normalization. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, pages 212–221. Springer Nature Switzerland, 2022. 4

[74] Artem Shmatko, Narmin Ghaffari Laleh, Moritz Gerstung, and Jakob Nikolas Kather. Artificial intelligence in histopathology: enhancing cancer research and clinical oncology. *Nat Cancer*, 3(9):1026–1038, 2022. 1

[75] Milad Sikaroudi, Maryam Hosseini, Ricardo Gonzalez, Shahryar Rahnamayan, and H R Tizhoosh. Generalization of vision pre-trained models for histopathology. *Sci. Rep.*, 13(1):6065, 2023. 1, 3, 4, 7, 8

[76] T C Smyrk, P Watson, K Kaul, and H T Lynch. Tumor-infiltrating lymphocytes are a marker for microsatellite instability in colorectal carcinoma. *Cancer*, 91(12):2417–2422, 2001. 1

[77] T Sørlie, C M Perou, R Tibshirani, T Aas, S Geisler, H Johnsen, T Hastie, M B Eisen, M van de Rijn, S S Jeffrey, T Thorsen, H Quist, J C Matese, P O Brown, D Botstein, P E Lønning, and A L Børresen-Dale. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc. Natl. Acad. Sci. U. S. A.*, 98(19):10869–10874, 2001. 1

[78] Maximilian Springenberg, Annika Frommholz, Markus Wenzel, Eva Weicken, Jackie Ma, and Nils Strodthoff.

From modern CNNs to vision transformers: Assessing the performance, robustness, and classification strategies of deep learning models in histopathology. *Med. Image Anal.*, 87:102809, 2023. 4

[79] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? data, augmentation, and regularization in vision transformers. 2021. 2, 3

[80] David Tellez, Maschenka Balkenhol, Nico Karssemeijer, Geert Litjens, Jeroen van der Laak, and Francesco Ciompi. H and E stain augmentation improves generalization of convolutional networks for histopathological mitosis detection. In *Medical Imaging 2018: Digital Pathology*, pages 264–270. SPIE, 2018. 4

[81] David Tellez, Geert Litjens, Péter Bándi, Wouter Bulten, John-Melle Bokhorst, Francesco Ciompi, and Jeroen van der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Med. Image Anal.*, 58: 101544, 2019. 4

[82] United States Food and Drug Administration. FDA grants accelerated approval to pembrolizumab for first tissue/site agnostic indication. 2017. 1

[83] Abhishek Vahadane, Tingying Peng, Amit Sethi, Shadi Albarqouni, Lichao Wang, Maximilian Baust, Katja Steiger, Anna Melissa Schlitter, Irene Esposito, and Nassir Navab. Structure-Preserving color normalization and sparse stain separation for histological images. *IEEE Trans. Med. Imaging*, 35(8):1962–1971, 2016. 2, 3

[84] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9(11), 2008. 4, 3

[85] Suhas Vasaikar, Chen Huang, Xiaojing Wang, Vladislav A Petyuk, Sara R Savage, Bo Wen, Yongchao Dou, Yun Zhang, Zhiao Shi, Osama A Arshad, et al. Proteogenomic analysis of human colon cancer reveals new therapeutic opportunities. *Cell*, 177(4):1035–1049, 2019. 6, 1

[86] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 6, 5

[87] Eugene Vorontsov, Alican Bozkurt, Adam Casson, George Shaikovski, Michal Zelechowski, Siqi Liu, Philippe Mathieu, Alexander van Eck, Donghun Lee, Julian Viret, Eric Robert, Yi Kan Wang, Jeremy D Kunz, Matthew C H Lee, Jan Bernhard, Ran A Godrich, Gerard Oakley, Ewan Millar, Matthew Hanna, Juan Retamero, William A Moye, Razik Yousfi, Christopher Kanan, David Klimstra, Brandon Rothrock, and Thomas J Fuchs. Virchow: A Million-Slide digital pathology foundation model. 2023. 1, 3, 9, 2

[88] Sophia J Wagner, Nadieh Khalili, Raghav Sharma, Melanie Boxberg, Carsten Marr, Walter de Back, and Tingying Peng. Structure-Preserving multi-domain stain color augmentation using Style-Transfer with disentangled representations. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 257–266. Springer International Publishing, 2021. 3

[89] Sophia J Wagner, Daniel Reisenbüchler, Nicholas P West, Jan Moritz Niehues, Jiefu Zhu, Sebastian Foersch, Gregory Patrick Veldhuizen, Philip Quirke, Heike I Grabsch, Piet A van den Brandt, Gordon G A Hutchins, Susan D Richman, Tanwei Yuan, Rupert Langer, Josien C A Jenniskens, Kelly Offermans, Wolfram Mueller, Richard Gray, Stephen B Gruber, Joel K Greenson, Gad Rennert, Joseph D Bonner, Daniel Schmolze, Jitendra Jonnagaddala, Nicholas J Hawkins, Robyn L Ward, Dion Morton, Matthew Seymour, Laura Magill, Marta Nowak, Jennifer Hay, Viktor H Koelzer, David N Church, TransSCOT consortium, Christian Matek, Carol Geppert, Chaolong Peng, Cheng Zhi, Xiaoming Ouyang, Jacqueline A James, Maurice B Loughrey, Manuel Salto-Tellez, Hermann Brenner, Michael Hoffmeister, Daniel Truhn, Julia A Schnabel, Melanie Boxberg, Tingying Peng, and Jakob Nikolas Kather. Transformer-based biomarker prediction from colorectal cancer histology: A large-scale multicentric study. *Cancer Cell*, 2023. 1, 6, 3, 4, 5

[90] Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, and Wenyu Liu. Revisiting multiple instance neural networks. *Pattern Recognit.*, 74:15–24, 2018. 3

[91] Xiyue Wang, Sen Yang, Jun Zhang, Minghui Wang, Jing Zhang, Junzhou Huang, Wei Yang, and Xiao Han. TransPath: Transformer-Based self-supervised learning for histopathological image classification. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 186–195. Springer International Publishing, 2021. 1, 3, 2

[92] Xiyue Wang, Sen Yang, Jun Zhang, Minghui Wang, Jing Zhang, Wei Yang, Junzhou Huang, and Xiao Han. Transformer-based unsupervised contrastive learning for histopathological image classification. *Med. Image Anal.*, 81:102559, 2022. 3, 4, 7, 8, 2

[93] Xiyue Wang, Yuexi Du, Sen Yang, Jun Zhang, Minghui Wang, Jing Zhang, Wei Yang, Junzhou Huang, and Xiao Han. RetCCL: Clustering-guided contrastive learning for whole-slide image retrieval. *Med. Image Anal.*, 83:102645, 2023. 1, 3, 4, 7, 2

[94] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013. 3, 6, 1, 2

[95] Georg Wölflein, Lucie Charlotte Magister, Pietro Liò, David J Harrison, and Ognjen Arandjelović. Deep multiple instance learning with Distance-Aware Self-Attention. 2023. 1, 6

[96] Jinxi Xiang, Xiyue Wang, Xinran Wang, Jun Zhang, Sen Yang, Wei Yang, Xiao Han, and Yueping Liu. Automatic diagnosis and grading of prostate cancer with weakly supervised learning on whole slide images. *Comput. Biol. Med.*, 152:106340, 2023. 1

[97] Zhongyi Yang, Xiyue Wang, Jinxi Xiang, Jun Zhang, Sen Yang, Xinran Wang, Wei Yang, Zhongyu Li, Xiao Han, and Yueping Liu. The devil is in the details: a small-lesion sensitive weakly supervised learning framework for prostate cancer detection and grading. *Virchows Arch.*, 482(3):525–538, 2023. 1

[98] Farhad Ghazvinian Zanjani, Svitlana Zinger, Babak Ehteshami Bejnordi, Jeroen A W M van der Laak, and Pe-

ter H N de With. Stain normalization of histopathology images using generative adversarial networks. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 573–577. IEEE, 2018. 3

[99] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow twins: Self-Supervised learning via redundancy reduction. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 3, 2

[100] Yunlong Zhang, Yuxuan Sun, Honglin Li, Sunyi Zheng, Chenglu Zhu, and Lin Yang. Benchmarking the robustness of deep neural networks to common corruptions in digital pathology. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, pages 242–252. Springer Nature Switzerland, 2022. 4

[101] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022. 3

A Good Feature Extractor Is All You Need for Weakly Supervised Learning in Histopathology
Supplementary Material

A. Downstream tasks and their clinical relevance

Targets We extensively evaluated the models on nine downstream tasks, summarised in Tab. 2. All of the targets were treated as binary variables, except for breast cancer subtype, which is a five-way classification target, determined by immunohistochemistry: Luminal A (HR+/HER2-/low Ki-67), Luminal B (HR+/HER2+/high Ki-67), HER2 overexpressed (HR-), Basal (which is a subgroup of triple-negative breast cancer), or Normal breast-like (a subtype for which the clinical and molecular characteristics remain largely undefined throughout the existing scientific literature) [77]. This molecular subtyping of early-stage invasive breast cancer has become an essential procedure in clinical management due to its implications in treatment recommendations and providing valuable prognostic insights for a patient's survival [9, 34]. In addition, our investigation also included analysis for prevalent mutations in CDH1 and TP53 as well as PIK3CA, the latter of which opens new possibilities for targeted therapies in advanced disease stages [1]. Microsatellite instability (MSI) status is a key marker in colorectal cancer owing to its profound implications in shaping a patient's prognosis and responsiveness to immunotherapies [13, 82]. It is driven by either spontaneous or germline (hence hereditary) mutations in DNA-repair related genes [6] and leads to phenotypic changes in the tumour tissue [76]. Therefore, the performance of various AI models is commonly evaluated based on their ability to predict MSI from routine histopathology [47], often performed in conjunction with other prevalent genetic markers such as KRAS and BRAF: these are key-driver mutations in colorectal cancer, that shape a patient's survival chances and hold strong influence over the selection of targeted therapies best suited for each individual patient [28, 66]. Given the high clinical relevance and availability of robust ground truth data, we have strategically selected these particular tasks for our analysis.

Data Here, we provide additional details about where we obtained data for the downstream tasks, further to what is mentioned in Sec. 4.

We predict the 🎗-LN status using the CAMELYON17 dataset [4], which contains data from five centres. For this dataset, we perform centre-wise cross-validation, where we use one of the centres for testing and the others for training (each of the five random seeds uses a different centre for testing). The training and validation sets are an 80%/20% split of the other four centres. We treat 🎗-LN status as a binary classification task, where the positive class corresponds to the presence of metastatic cancer cells in the lymph nodes. Each slide in the dataset is of a lymph node tissue section, and we treat each slide as a single sample, *i.e.* a separate patient. This is slightly different to the original CAMELYON17 challenge [4], where groups of five slides were arranged into "virtual patients" (though the slides themselves may be from different actual patients), and the task was to predict a virtual patient-level label based on a specific rule for aggregating the slide-level predictions. We do not use the virtual patient labels, but instead use the slide-level labels provided in the dataset.

Target	Training and validation	Test dataset
🎗-Subtype 🎗-CDH1 mutation 🎗-TP53 mutation 🎗-PIK3CA mutation	TCGA-BRCA [94] (833 train, 208 val samples)	CPTAC-BRCA [50] (120 samples)
🎗-LN status	CAMELYON17 [4] (centre-wise cross-validation; 320 train, 80 val samples)	CAMELYON17 [4] (centre-wise cross-validation; 100 samples)
∩-MSI status ∩-RAS mutation ∩-RAF mutation ∩-MAD4 mutation	TCGA-CRC [94] (558 samples)	CPTAC-COAD [85] (110 samples)

Table 2. Overview of the evaluated downstream tasks. Dataset sizes are shown in parentheses. The 🎗 targets are related to breast cancer, while the ∩ targets are related to colorectal cancer.

For all other targets, we use either TCGA-BRCA [94] or TCGA-CRC [94] for training, and respectively either CPTAC-BRCA [50] or CPTAC-COAD [85] for testing. We obtain the patient-level labels from the respective studies via `cbioportal.org`. The only exception is ∩-MSI status which is not available for TCGA-CRC in `cbioportal.org`, but is provided in the supplementary material of Liu *et al.* [52].

B. Feature extractors

In Tab. 3, we provide an overview of the SSL feature extractors evaluated in this study. We use the weights from the respective authors' GitHub repositories. The feature extractor called Lunit-DINO in our paper corresponds to Kang *et al.*'s DINO$_{p=16}$ model [44]. For the Phikon extractor [31], we employ the 'student' model as recommended by the authors in their GitHub repository at the time of writing this paper. However, the reader should note that this recommendation has since been updated to prefer the 'teacher' model instead.

Name	Architecture	SSL method	SSL dataset, magnification	Embedding size (d_x)
CTransPath [91, 92]	Swin Transformer [53]	semantically-relevant contrastive learning [92] based on MoCo v3 [20]	TCGA [94] and PAIP [48] (20×)	768
Phikon [31]	ViT-B [49]	semantically-relevant contrastive learning [92] based on MoCo v3 [20]	TCGA [94] (20×)	768
Lunit-DINO [44]	ViT-S [49]	DINO [12]	TCGA [94] and non-public TULIP [44] (20×, 40×)	384
RetCCL [93]	ResNet-50 [37]	clustering-guided contrastive learning [93] based on MoCo [38]	TCGA [94] and PAIP [48] (20×)	2048
Lunit-BT [44]	ResNet-50 [37]	Barlow Twins [99]	TCGA [94] and non-public TULIP [44] (20×, 40×)	2048
Lunit-SwAV [44]	ResNet-50 [37]	SwAV [11]	TCGA [94] and non-public TULIP [44] (20×, 40×)	2048

Table 3. Overview of SSL feature extractors evaluated in this study, their architecture, SSL method, pretraining dataset, and embedding size. As baselines, we additionally compare against the respective ImageNet pretrained backbones: Swin Transformer [53], Vit-B [49], ViT-S [49, 79] and ResNet-50 [37].

B.1. Foundation models

This year, a number of foundation models have emerged for pathology that were trained on datasets of unprecedented size. Unfortunately, we could not include these in our study since their weights remain proprietary. Notably, UNI [16] has been trained on a dataset exceeding 100,000 slides, while Virchow [87] utilises an even larger corpus of 1.5 million slides, both employing the DINOv2 framework [63]. Moreover, Campanella *et al.* [8] trained two foundation models on over 400,000 slides using DINO [12] and MAE [39]. On the other hand, Azizi *et al.* [2] integrate both medical and non-medical images to train their foundation model, REMEDIS, using SimCLR/BiT [17, 18]. Furthermore, Lu *et al.* [58] made use of 1.17 million image-caption pairs to develop a vision-language foundation model named CONCH. In stark contrast, the publicly available models employ orders of magnitude fewer WSIs, as TCGA contains around 30,000 diagnostic and tissue slides in total [94].

C. Stain normalisation

In Fig. 8, we show the effect of stain normalisation on the latent space of the remaining eight feature extractors that were not depicted in Fig. 3.

Patchwise versus slidewise stain normalisation In Sec. 4.2, we state that there is no consistent improvement obtained by employing stain normalisation, regardless of whether it is performed on a per-patch or per-slide basis. Further to the results in Fig. 1a showing only slidewise stain normalisation, we perform an ablation study where we normalise each patch individually. We provide an analogous boxplot for both types of stain normalisation in Fig. 9, which shows that the conclusion holds for both types of stain normalisation.

D. Augmentations

Including patchwise stain normalisation, we study 27 image augmentations in this work. We provide representative examples of these augmentations in Fig. 10, and describe them below:

- **macenko**: Macenko stain normalisation [59] (patchwise)
- **rotate {90°, 180°, 270°}**: rotate by the specified angle
- **random rotation**: rotate by an angle β sampled uniformly such that $(\beta \mod 90) \in [10, 80]$, *i.e.* forcing an off-axis rotation
- **flip {horizontal, vertical}**: flip along the specified axis
- **zoom {1.5×, 1.75×, 2×}**: enlarge the patch by the specified factor and crop the centre
- **affine**: random affine transformation with a maximum rotation of 10°, maximum translation of 20% of the patch size, maximum scaling of 20%, and maximum shear of 10°
- **warp perspective**: random perspective transformation with a maximum distortion of 0.2
- **jigsaw**: cut the patch into a 4×4 grid and randomly permute the tiles
- **Cutout**: randomly erase a rectangle that covers between 2% and 25% of the total area [27]
- **AugMix**: see Hendrycks *et al.* [41]
- **{low, high} brightness**: reduce the brightness by a factor of 0.7 or increase it by a factor of 1.5
- **{low, high} contrast**: reduce the contrast by a factor of 0.7 or increase it by a factor of 1.5
- **{low, high} saturation**: reduce the saturation by a factor

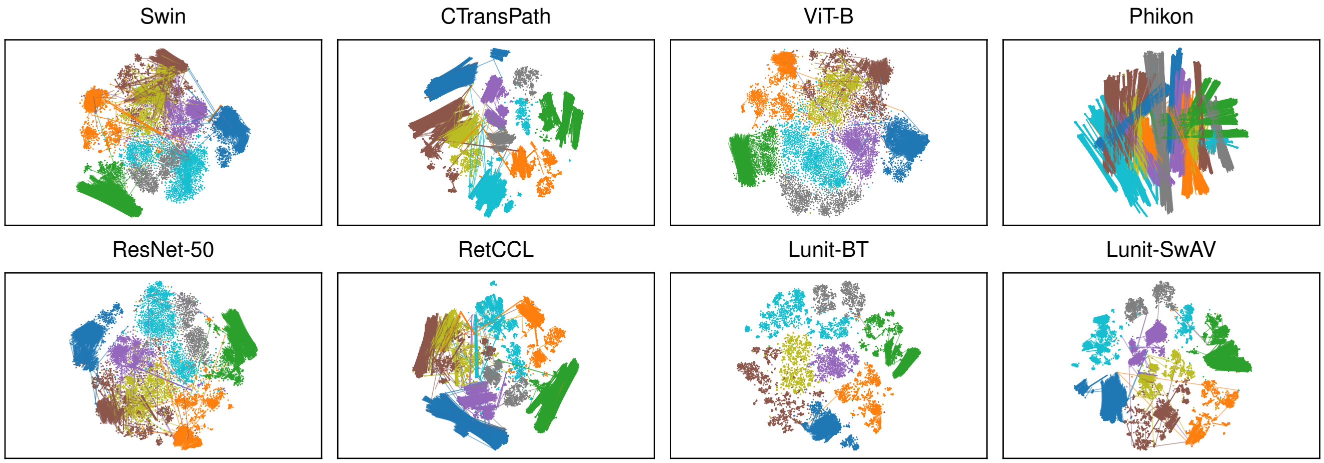


Figure 8. Latent space visualisations (t-SNE [84]), showing the effect of stain normalisation [59]. This figure extends Fig. 3, which depicts only two feature extractors, Lunit-DINO [44] and its ViT-S [49, 79] ImageNet baseline; here, we show the other eight. Colours are as in Fig. 3.

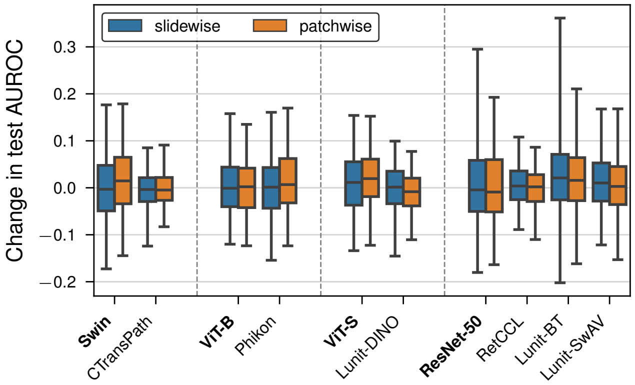


Figure 9. Improvement obtained by employing slidewise (blue, boxes are as in Fig. 1a) or patchwise (orange) stain normalisation compared to no normalisation. There is no clear benefit or detriment in applying either type of stain normalisation (all confidence intervals cross zero). While this figure reports results only for the downstream AttMIL model, the conclusion holds for the other models as well, as reported in **??**.

of 0.7 or increase it by a factor of 1.5

- **colour jitter**: randomly adjust the brightness, contrast, saturation, and hue by maximum factors of 0.4, 0.4, 0.4, and 0.1, respectively
- **gamma {0.5, 2.0}**: apply a gamma correction with the specified exponent
- **sharpen**: sharpen the image by a factor of 5
- **Gaussian blur**: apply a Gaussian blur with a kernel size of 5 and a standard deviation of 2.0
- **median blur**: apply a median blur with a kernel size of 5

Augmentation groups In Sec. 4, we study the effect of various *groups* of augmentations on downstream performance. These groups are defined as follows:

- **none**: no augmentations, *i.e.* the original patches are used
- **Macenko (patchwise)**: Macenko stain normalisation [59] is applied on a per-patch basis
- **Macenko (slidewise)**: Macenko stain normalisation [59] is applied on a per-slide basis
- **rotation/flipping**: each patch is randomly rotated by a right angle or flipped along the horizontal or vertical axis, with equal probability
- **all**: any of the 27 augmentations, or no augmentation, is applied to each patch with equal probability

We apply no augmentations to the test set (except when applying slidewise or patchwise stain normalisation, in which case we normalise the test set in the same way as the training set).

E. Extended data tables and figures

In much of our discussion in Secs. 4 and 5, we focus on particular augmentations, models or feature extractors. Here, we produce extended versions of figures and tables from the main text providing more results for different choices of the above.

Figure 11 summarises the main results for all three downstream aggregation models: AttMIL [42], the two-layer transformer as employed by Wagner *et al.* [89], and the mean average pooling baseline.

Normalised differential AUROC scores In Tabs. 4 to 8, we present the normalised differential AUROC scores for all tasks, feature extractors, and downstream models, and augmentation groups (one table per augmentation group). This extends Tab. 1 from the main text, which only shows the results for the AttMIL [42] aggregation

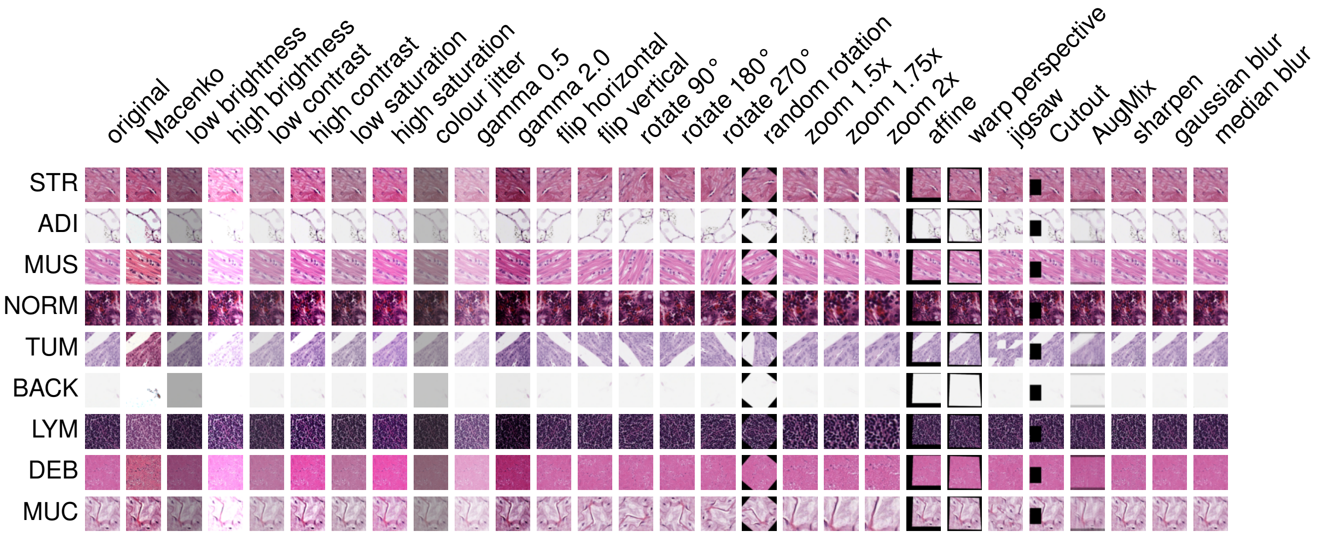


Figure 10. Examples of original and augmented patches (columns) from the NCT-CRC-HE-100K dataset [45, 46]. Each row corresponds to a representative patch from a different patch class.

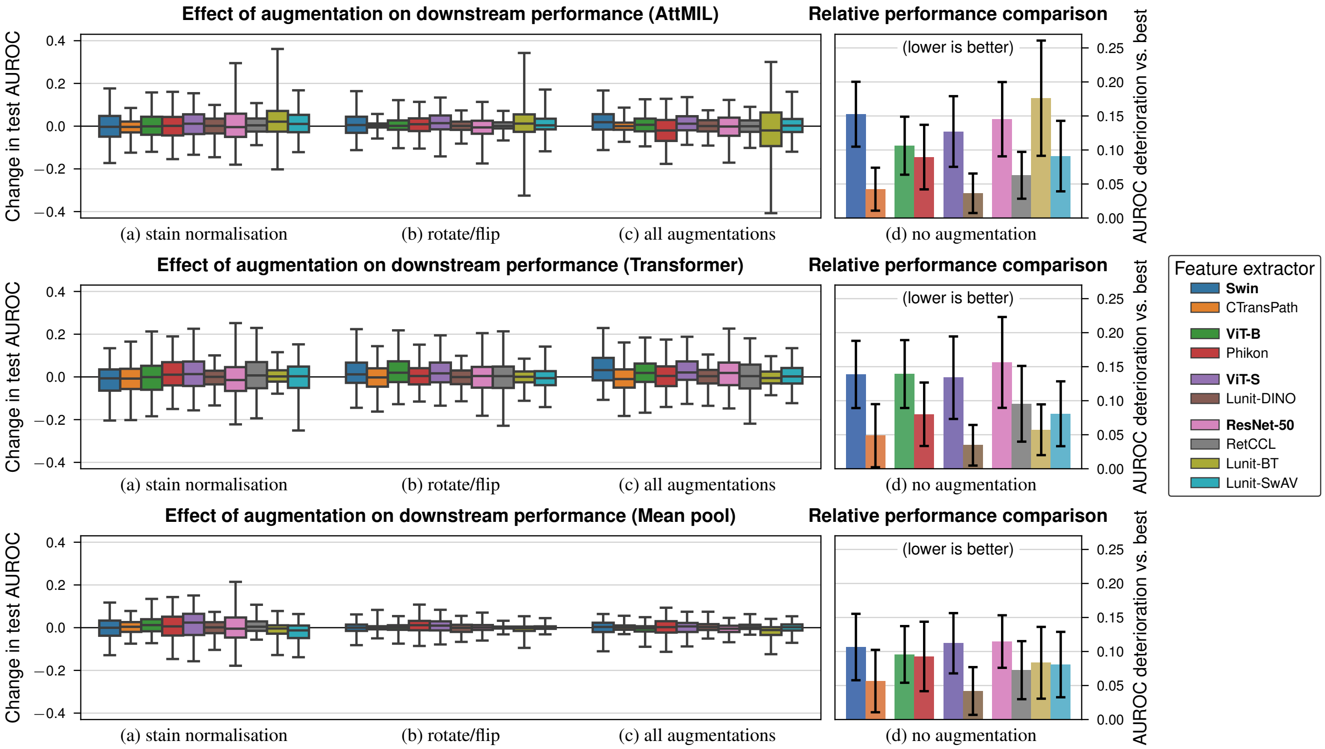


Figure 11. Extended version of Fig. 1 showing the main results for all three downstream models: AttMIL [42] (top, same as Fig. 1), a two-layer transformer [89] (middle), and mean average pooling (bottom).

model without augmentations (corresponding to the first ten rows in Tab. 4). We observe that Lunit-DINO [44] and CTransPath [92] consistently achieve the best task-averaged results, independent of the choice of downstream aggregation model and augmentation group.

Absolute AUROC scores While the normalised differential AUROC score provides a relative performance measure to facilitate a fair comparison between feature extractors, we also provide the seed-averaged absolute test AUROC scores for all tasks, feature extractors, and downstream models, and augmentation groups in Tabs. 4 to 8 (one table per augmentation group). Looking at these absolute scores, we find that the predicting the 🎗-PIK3CA target is the most difficult task across the board for all feature extractors and downstream models, while the 🎗-LN status and -MSI status targets are the easiest. However, we emphasise that the normalised differential AUROC score is the more meaningful metric for comparing feature extractors, since it is independent of the task difficulty and accounts for the variance across seeds (see Sec. 4.1).

F. Training and implementation details

Training For downstream model training, we use the AdamW [56] optimiser with an initial learning rate of 10^{-3}, weight decay of 10^{-2}, and a batch size of one. The learning rate is decayed using a cosine annealing learning schedule [55] over 30 epochs, but we halt training when the validation loss does not improve for ten epochs.

In MIL terminology, we refer to the *patient* as the *bag*, and the *patches* as the *instances*. Note that some datasets have multiple WSIs per patient; in these cases, we simply mix the patches from all WSIs into a single bag. An epoch represents a full pass over all patients in the training set. At every step, we sample a maximum of 8,192 patches per patient, though most patients have fewer patches. We found it beneficial to employ a batch size of one: not only does this reduce GPU memory requirements, it also accelerates training. Indeed, we found that padding the bags to the maximum number of patches per patient (8,192) slows down training considerably, but with a batch size of one, we can use a variable number of patches per bag. Nonetheless, we accumulate gradients over four steps before performing a weight update, which effectively increases the batch size to four.

F.1. Downstream aggregation models

We describe the three downstream aggregation models in more detail below. In essence, these are different parametrisations of the g_θ function in Eq. (2) that aggregate the patch embeddings into a single slide-level embedding. All three models first pass the patch embeddings through a linear layer with 512 output units and ReLU activation, *i.e.*

$$g_\theta(\{x_i\}_{i=1}^n) = \bar{g_\theta}(\{\max(0, \bar{W_\theta} x_i + \bar{b_\theta})\}_{i=1}^n) \tag{3}$$

with learnable parameters $\bar{W_\theta} \in \mathbb{R}^{512 \times d_x}$ and $\bar{b_\theta} \in \mathbb{R}^{512}$. However, the three models differ in how they aggregate the resulting patch embeddings in the $\bar{g_\theta}$ function, which we describe below.

In any case, the classifier h_θ in Eq. (2) is a linear layer with softmax activation over the number of classes, to which we apply a cross-entropy loss. Note that we employ dropout with a probability of 0.5 to the slide-level embedding before passing it to the linear layer.

Mean average pooling As a baseline model, we compute the slide-level embedding as the mean of the patch embeddings, *i.e.*

$$\bar{g_\theta}(\{x_i\}_{i=1}^n) = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{4}$$

AttMIL [42] This model takes a weighted average of the patch embeddings, where the weights are computed independently for each embedding. More formally, the slide-level embedding is given by

$$\bar{g_\theta}(\{x_i\}_{i=1}^n) = \sum_{i=1}^{n} \alpha_i x_i, \tag{5}$$

where the attention[8] weights $\alpha_i \in \mathbb{R}$ are obtained via a two-layer network with 256 tanh-activated hidden units that is applied to each patch embedding x_i independently and then normalised across all patches using a softmax function, *i.e.*

$$e_i = W_2 \tanh(W_1 x_i + b_1) + b_2, \tag{6}$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)}. \tag{7}$$

Here, $W_1 \in \mathbb{R}^{256 \times 512}$, $b_1 \in \mathbb{R}^{256}$, $W_2 \in \mathbb{R}^{1 \times 256}$, and $b_2 \in \mathbb{R}$ are learnable parameters (captured within the set of learnable parameters θ).

Two-layer transformer We also employ a two-layer transformer [86], closely aligned with the configuration presented by Wagner *et al.* [89]. This setup differs from the classical transformer architecture [86] in that there is just one branch, *i.e.* just the decoder (or encoder, depending on perspective). Both layers have 512 hidden units and 8 attention heads, employ a dropout rate of 0.1, use GELU activation [40] in the feedforward layers, and use layer normalisation [3] before the attention layers. We employ no masking. The input tokens are the patch embeddings, and the output tokens are averaged like in Eq. (4) to obtain the slide-level embedding.

F.2. Overhead and caching

Feature extraction Prior to training, we extract features from all patches in the training and validation sets, and store them on disk. We do this for each of the ten feature extractors. For the training sets, we additionally perform feature extraction for all 27 augmented versions of each patch, and store these on disk as well. For both the training and test sets, we also extract features for the stain-normalised

[8] Ilse *et al.* [42]'s use of the term "attention" should not be confused with the scaled dot product attention in the transformer architecture [86]. Here, the attention weight for a particular token is computed solely based on that token alone.

versions of the patches. This way, we effectively have a cache of the $a_i \circ f$ function in Eq. (2) for all inputs (*i.e.* patches), all augmentations a_i, and all feature extractors f. During training, we only need to load the features from disk (d_x floating point values per patch, *e.g.* in the case of CTransPath $d_x = 768$), as opposed to loading the patches directly (224 × 224 × 3 byte values) and having to perform augmentation and feature extraction on the fly (very expensive).

Training with augmentation Even though our training runs employed already extracted features, they took 30× longer with all augmentations, or 5× longer with just the rotation augmentations as compared to employing no augmentations. This approximately linear scaling in the number of augmentations a is the result of slower data loading, as random reads are performed over a times as many features compared to the no-augmentation case. We alleviated some of this bottleneck by implementing additional caches, but even this solution only bore fruit because we ran many experiments with similar dataset configurations. Thus, we emphasise again that augmentations are too expensive to be viable in computational pathology pipelines, due their significant preprocessing *and* training overhead which does not even yield a consistent improvement in downstream performance.

Total training time In total, we trained 6,750 models across the cartesian product of:

- 10 feature extractors,
- 5 augmentation groups,
- 3 downstream aggregation models,
- 9 downstream tasks, and
- 5 random seeds.

We trained these models on NVIDIA Tesla V100 GPUs (one training run per GPU at a time), which cumulatively took 4,648 GPU hours (193.7 days).

G. Acknowledgements

GW is supported by Lothian NHS. This project received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 101017453 as part of the KATY project. This work is supported in part by the Industrial Centre for AI Research in Digital Diagnostics (iCAIRD) which is funded by Innovate UK on behalf of UK Research and Innovation (UKRI) (project number 104690).

Model	Target Feature extractor	-Subtype	-CDH1	-TP53	-PIK3CA	-LN status	-MSI	-KRAS	-BRAF	-SMAD4	Average
AttMIL	Swin	0.07 ± 0.02	0.17 ± 0.03	0.28 ± 0.02	0.07 ± 0.04	0.17 ± 0.08	0.18 ± 0.04	0.14 ± 0.04	0.14 ± 0.07	0.16 ± 0.05	0.15 ± 0.05
	CTransPath	**0.00 ± 0.00**	**0.01 ± 0.01**	**0.01 ± 0.01**	0.04 ± 0.03	**0.06 ± 0.07**	0.08 ± 0.03	0.06 ± 0.03	0.06 ± 0.03	0.06 ± 0.03	0.04 ± 0.03
	ViT-B	0.08 ± 0.04	0.11 ± 0.02	0.15 ± 0.03	0.07 ± 0.03	0.17 ± 0.06	0.15 ± 0.03	0.03 ± 0.04	0.18 ± 0.07	**0.01 ± 0.01**	0.11 ± 0.04
	Phikon	0.09 ± 0.02	0.09 ± 0.02	0.09 ± 0.03	0.09 ± 0.03	0.07 ± 0.06	0.06 ± 0.04	0.07 ± 0.04	0.07 ± 0.06	0.17 ± 0.08	0.09 ± 0.05
	ViT-S	0.13 ± 0.03	0.08 ± 0.03	0.14 ± 0.05	0.08 ± 0.05	0.19 ± 0.09	0.18 ± 0.04	0.06 ± 0.03	0.19 ± 0.04	0.08 ± 0.08	0.13 ± 0.05
	Lunit-DINO	0.08 ± 0.03	0.03 ± 0.03	0.03 ± 0.02	0.02 ± 0.03	0.07 ± 0.04	**0.00 ± 0.00**	0.06 ± 0.04	**0.02 ± 0.02**	0.02 ± 0.02	**0.04 ± 0.03**
	ResNet-50	0.15 ± 0.03	0.09 ± 0.04	0.11 ± 0.03	**0.01 ± 0.02**	0.18 ± 0.08	0.22 ± 0.04	0.11 ± 0.03	0.23 ± 0.07	0.21 ± 0.09	0.15 ± 0.05
	RetCCL	0.07 ± 0.03	0.04 ± 0.02	0.04 ± 0.03	0.05 ± 0.03	0.07 ± 0.06	0.08 ± 0.03	**0.03 ± 0.02**	0.14 ± 0.03	0.06 ± 0.03	0.06 ± 0.03
	Lunit-BT	0.13 ± 0.03	0.06 ± 0.04	0.02 ± 0.01	0.13 ± 0.04	0.34 ± 0.15	0.28 ± 0.13	0.03 ± 0.04	0.35 ± 0.13	0.25 ± 0.03	0.18 ± 0.08
	Lunit-SwAV	0.06 ± 0.02	0.06 ± 0.03	0.06 ± 0.02	0.13 ± 0.06	0.07 ± 0.05	0.10 ± 0.03	0.13 ± 0.06	0.07 ± 0.07	0.14 ± 0.08	0.09 ± 0.05
Transformer	Swin	0.09 ± 0.04	0.11 ± 0.03	0.21 ± 0.04	0.09 ± 0.03	0.16 ± 0.08	0.19 ± 0.07	0.09 ± 0.04	0.17 ± 0.05	0.14 ± 0.05	0.14 ± 0.05
	CTransPath	**0.01 ± 0.02**	**0.01 ± 0.02**	0.03 ± 0.03	0.08 ± 0.07	0.07 ± 0.07	0.02 ± 0.02	0.04 ± 0.04	0.08 ± 0.06	0.09 ± 0.05	0.05 ± 0.05
	ViT-B	0.08 ± 0.03	0.10 ± 0.02	0.17 ± 0.04	0.11 ± 0.02	0.21 ± 0.07	0.18 ± 0.05	0.13 ± 0.05	0.20 ± 0.08	0.06 ± 0.05	0.14 ± 0.05
	Phikon	0.13 ± 0.04	0.08 ± 0.05	0.08 ± 0.03	0.05 ± 0.03	0.07 ± 0.05	0.05 ± 0.04	0.05 ± 0.04	0.11 ± 0.07	0.12 ± 0.06	0.08 ± 0.05
	ViT-S	0.10 ± 0.02	0.07 ± 0.03	0.22 ± 0.07	0.11 ± 0.06	0.21 ± 0.09	0.16 ± 0.06	0.08 ± 0.04	0.23 ± 0.09	0.03 ± 0.02	0.13 ± 0.06
	Lunit-DINO	0.04 ± 0.03	0.06 ± 0.03	0.03 ± 0.02	**0.02 ± 0.02**	**0.05 ± 0.04**	**0.01 ± 0.01**	0.06 ± 0.03	**0.02 ± 0.04**	**0.02 ± 0.03**	**0.03 ± 0.03**
	ResNet-50	0.13 ± 0.04	0.10 ± 0.07	0.15 ± 0.03	0.04 ± 0.07	0.19 ± 0.08	0.19 ± 0.07	0.11 ± 0.04	0.19 ± 0.06	0.30 ± 0.11	0.16 ± 0.07
	RetCCL	0.09 ± 0.04	0.04 ± 0.04	0.02 ± 0.02	0.09 ± 0.06	0.07 ± 0.06	0.15 ± 0.03	0.12 ± 0.05	0.22 ± 0.11	0.06 ± 0.04	0.10 ± 0.06
	Lunit-BT	0.04 ± 0.03	0.05 ± 0.03	**0.02 ± 0.02**	0.10 ± 0.04	0.07 ± 0.07	0.02 ± 0.02	**0.02 ± 0.02**	0.13 ± 0.05	0.07 ± 0.02	0.06 ± 0.04
	Lunit-SwAV	0.08 ± 0.04	0.04 ± 0.05	0.05 ± 0.03	0.11 ± 0.05	0.07 ± 0.06	0.06 ± 0.03	0.08 ± 0.03	0.07 ± 0.05	0.17 ± 0.07	0.08 ± 0.05
Mean pool	Swin	0.08 ± 0.01	0.10 ± 0.04	0.13 ± 0.05	0.05 ± 0.02	0.17 ± 0.12	0.17 ± 0.02	**0.02 ± 0.02**	0.13 ± 0.03	0.10 ± 0.02	0.11 ± 0.05
	CTransPath	**0.00 ± 0.00**	0.04 ± 0.02	0.02 ± 0.02	**0.00 ± 0.01**	0.15 ± 0.11	0.03 ± 0.02	0.11 ± 0.05	0.06 ± 0.03	0.09 ± 0.02	0.06 ± 0.05
	ViT-B	0.07 ± 0.01	0.08 ± 0.01	0.07 ± 0.02	0.09 ± 0.02	0.15 ± 0.11	0.15 ± 0.02	0.07 ± 0.04	0.18 ± 0.04	0.02 ± 0.02	0.10 ± 0.04
	Phikon	0.11 ± 0.02	0.04 ± 0.03	0.13 ± 0.03	0.06 ± 0.03	0.11 ± 0.11	0.07 ± 0.03	0.12 ± 0.03	0.09 ± 0.07	0.11 ± 0.05	0.09 ± 0.05
	ViT-S	0.11 ± 0.01	0.03 ± 0.03	0.13 ± 0.02	0.07 ± 0.03	0.15 ± 0.11	0.19 ± 0.03	0.03 ± 0.02	0.21 ± 0.04	0.07 ± 0.03	0.11 ± 0.04
	Lunit-DINO	0.08 ± 0.01	0.04 ± 0.02	**0.01 ± 0.02**	0.05 ± 0.03	**0.09 ± 0.09**	**0.00 ± 0.00**	0.09 ± 0.02	**0.00 ± 0.00**	**0.01 ± 0.02**	**0.04 ± 0.04**
	ResNet-50	0.08 ± 0.01	**0.00 ± 0.01**	0.09 ± 0.02	0.03 ± 0.02	0.21 ± 0.09	0.22 ± 0.03	0.03 ± 0.04	0.24 ± 0.02	0.13 ± 0.05	0.11 ± 0.04
	RetCCL	0.01 ± 0.00	0.03 ± 0.01	0.06 ± 0.02	0.06 ± 0.02	0.15 ± 0.11	0.10 ± 0.04	0.03 ± 0.03	0.15 ± 0.01	0.06 ± 0.02	0.07 ± 0.04
	Lunit-BT	0.06 ± 0.03	0.04 ± 0.01	0.06 ± 0.04	0.07 ± 0.02	0.18 ± 0.11	0.08 ± 0.02	0.03 ± 0.03	0.21 ± 0.09	0.03 ± 0.02	0.08 ± 0.05
	Lunit-SwAV	0.07 ± 0.00	0.03 ± 0.02	0.04 ± 0.02	0.11 ± 0.02	0.13 ± 0.13	0.05 ± 0.02	0.13 ± 0.03	0.03 ± 0.01	0.13 ± 0.04	0.08 ± 0.05

Table 4. Normalised differential AUROC scores for all tasks, feature extractors, downstream models, when employing **no augmentations**.

Model	Target Feature extractor	-Subtype	-CDH1	-TP53	-PIK3CA	-LN status	-MSI	-KRAS	-BRAF	-SMAD4	Average
AttMIL	Swin	0.08 ± 0.04	0.23 ± 0.03	0.27 ± 0.03	0.07 ± 0.05	0.18 ± 0.08	0.19 ± 0.05	0.15 ± 0.02	0.11 ± 0.07	0.16 ± 0.05	0.16 ± 0.05
	CTransPath	**0.00 ± 0.00**	0.04 ± 0.04	0.04 ± 0.03	0.03 ± 0.02	0.06 ± 0.08	0.08 ± 0.03	0.08 ± 0.04	0.07 ± 0.06	0.08 ± 0.03	0.05 ± 0.04
	ViT-B	0.08 ± 0.04	0.16 ± 0.03	0.12 ± 0.02	0.04 ± 0.03	0.16 ± 0.07	0.15 ± 0.04	0.10 ± 0.04	0.10 ± 0.05	**0.01 ± 0.02**	0.10 ± 0.04
	Phikon	0.09 ± 0.04	0.08 ± 0.04	0.08 ± 0.04	0.10 ± 0.03	0.09 ± 0.08	0.06 ± 0.03	0.13 ± 0.04	0.12 ± 0.05	0.07 ± 0.05	0.09 ± 0.05
	ViT-S	0.12 ± 0.05	0.13 ± 0.04	0.11 ± 0.04	0.04 ± 0.03	0.17 ± 0.09	0.20 ± 0.06	0.09 ± 0.04	0.15 ± 0.05	0.10 ± 0.11	0.12 ± 0.06
	Lunit-DINO	0.04 ± 0.04	0.03 ± 0.02	0.04 ± 0.03	**0.02 ± 0.02**	**0.06 ± 0.07**	**0.00 ± 0.01**	0.05 ± 0.04	**0.01 ± 0.02**	0.09 ± 0.06	**0.04 ± 0.04**
	ResNet-50	0.13 ± 0.04	0.20 ± 0.04	0.16 ± 0.03	0.03 ± 0.02	0.16 ± 0.08	0.22 ± 0.04	0.17 ± 0.05	0.15 ± 0.05	0.11 ± 0.07	0.15 ± 0.05
	RetCCL	0.07 ± 0.04	0.02 ± 0.02	0.03 ± 0.03	0.05 ± 0.03	0.09 ± 0.06	0.06 ± 0.03	**0.01 ± 0.02**	0.13 ± 0.05	0.08 ± 0.02	0.06 ± 0.04
	Lunit-BT	0.13 ± 0.06	0.04 ± 0.03	0.06 ± 0.08	0.12 ± 0.03	0.27 ± 0.17	0.17 ± 0.15	0.06 ± 0.06	0.34 ± 0.07	0.23 ± 0.07	0.16 ± 0.09
	Lunit-SwAV	0.07 ± 0.04	**0.02 ± 0.02**	**0.02 ± 0.02**	0.05 ± 0.04	0.07 ± 0.06	0.13 ± 0.04	0.12 ± 0.04	0.05 ± 0.05	0.13 ± 0.06	0.07 ± 0.04
Transformer	Swin	0.09 ± 0.03	0.15 ± 0.04	0.20 ± 0.03	0.04 ± 0.03	0.17 ± 0.09	0.21 ± 0.09	0.12 ± 0.06	0.21 ± 0.05	0.16 ± 0.08	0.15 ± 0.06
	CTransPath	0.04 ± 0.03	**0.01 ± 0.02**	0.05 ± 0.05	0.08 ± 0.04	**0.05 ± 0.07**	0.02 ± 0.02	0.06 ± 0.03	**0.03 ± 0.03**	0.17 ± 0.09	0.06 ± 0.05
	ViT-B	0.12 ± 0.04	0.14 ± 0.03	0.17 ± 0.03	0.02 ± 0.02	0.20 ± 0.08	0.22 ± 0.06	0.11 ± 0.04	0.23 ± 0.11	0.04 ± 0.03	0.14 ± 0.06
	Phikon	0.11 ± 0.02	0.08 ± 0.02	0.09 ± 0.04	0.03 ± 0.02	0.09 ± 0.08	0.04 ± 0.03	0.06 ± 0.06	0.09 ± 0.08	**0.03 ± 0.03**	0.07 ± 0.05
	ViT-S	0.09 ± 0.02	0.15 ± 0.04	0.15 ± 0.05	0.05 ± 0.03	0.15 ± 0.09	0.22 ± 0.08	0.10 ± 0.03	0.15 ± 0.04	0.04 ± 0.03	0.12 ± 0.05
	Lunit-DINO	**0.02 ± 0.03**	0.06 ± 0.04	**0.02 ± 0.03**	**0.02 ± 0.02**	0.06 ± 0.05	**0.01 ± 0.02**	0.10 ± 0.05	0.04 ± 0.05	0.07 ± 0.07	**0.04 ± 0.04**
	ResNet-50	0.15 ± 0.03	0.20 ± 0.07	0.16 ± 0.04	0.03 ± 0.02	0.22 ± 0.07	0.21 ± 0.04	0.13 ± 0.03	0.13 ± 0.07	0.20 ± 0.13	0.16 ± 0.06
	RetCCL	0.07 ± 0.05	0.06 ± 0.03	0.03 ± 0.02	0.06 ± 0.04	0.10 ± 0.04	0.09 ± 0.03	0.11 ± 0.04	0.21 ± 0.09	0.08 ± 0.04	0.09 ± 0.05
	Lunit-BT	0.03 ± 0.02	0.03 ± 0.02	0.02 ± 0.03	0.05 ± 0.02	0.06 ± 0.06	0.04 ± 0.03	**0.02 ± 0.02**	0.15 ± 0.07	0.05 ± 0.03	0.05 ± 0.04
	Lunit-SwAV	0.07 ± 0.03	0.02 ± 0.02	0.04 ± 0.04	0.06 ± 0.05	0.08 ± 0.09	0.13 ± 0.06	0.14 ± 0.05	0.15 ± 0.10	0.18 ± 0.08	0.10 ± 0.06
Mean pool	Swin	0.07 ± 0.01	0.14 ± 0.02	0.15 ± 0.04	0.03 ± 0.01	0.20 ± 0.09	0.18 ± 0.03	0.05 ± 0.05	0.13 ± 0.06	0.08 ± 0.03	0.11 ± 0.05
	CTransPath	**0.00 ± 0.00**	0.02 ± 0.01	0.03 ± 0.03	**0.00 ± 0.00**	0.14 ± 0.10	0.03 ± 0.02	0.07 ± 0.05	0.06 ± 0.03	0.08 ± 0.02	0.05 ± 0.04
	ViT-B	0.05 ± 0.01	0.08 ± 0.01	0.08 ± 0.03	0.04 ± 0.01	0.14 ± 0.11	0.16 ± 0.02	0.08 ± 0.03	0.13 ± 0.07	**0.00 ± 0.01**	0.08 ± 0.05
	Phikon	0.13 ± 0.01	0.03 ± 0.02	0.11 ± 0.04	0.06 ± 0.02	0.12 ± 0.11	**0.01 ± 0.01**	**0.02 ± 0.02**	0.11 ± 0.05	0.07 ± 0.03	0.07 ± 0.05
	ViT-S	0.08 ± 0.01	0.09 ± 0.02	0.12 ± 0.04	0.05 ± 0.03	0.18 ± 0.11	0.17 ± 0.06	0.02 ± 0.02	0.21 ± 0.04	0.03 ± 0.03	0.11 ± 0.05
	Lunit-DINO	0.06 ± 0.01	0.02 ± 0.02	0.05 ± 0.04	0.05 ± 0.02	**0.07 ± 0.08**	0.02 ± 0.01	0.05 ± 0.03	0.03 ± 0.04	0.05 ± 0.02	**0.04 ± 0.04**
	ResNet-50	0.08 ± 0.01	0.11 ± 0.04	0.15 ± 0.03	0.03 ± 0.01	0.21 ± 0.10	0.18 ± 0.04	0.07 ± 0.04	0.15 ± 0.03	0.06 ± 0.06	0.12 ± 0.05
	RetCCL	0.02 ± 0.00	**0.01 ± 0.01**	0.07 ± 0.04	0.05 ± 0.01	0.12 ± 0.10	0.05 ± 0.02	0.02 ± 0.02	0.13 ± 0.03	0.05 ± 0.01	0.06 ± 0.04
	Lunit-BT	0.09 ± 0.03	0.01 ± 0.00	0.04 ± 0.03	0.07 ± 0.01	0.21 ± 0.10	0.16 ± 0.04	0.06 ± 0.05	0.20 ± 0.08	0.02 ± 0.01	0.10 ± 0.05
	Lunit-SwAV	0.08 ± 0.01	0.01 ± 0.01	**0.02 ± 0.03**	0.15 ± 0.02	0.13 ± 0.11	0.15 ± 0.01	0.17 ± 0.02	**0.01 ± 0.02**	0.13 ± 0.03	0.10 ± 0.04

Table 5. Normalised differential AUROC scores for all tasks, feature extractors, downstream models, when employing **slidewise stain normalisation** [59].

	Target	🎗-Subtype	🎗-CDH1	🎗-TP53	🎗-PIK3CA	🎗-LN status	-MSI	-KRAS	-BRAF	-SMAD4	Average
Model	Feature extractor										
AttMIL	Swin	0.08 ± 0.03	0.20 ± 0.04	0.24 ± 0.03	0.05 ± 0.03	0.16 ± 0.07	0.14 ± 0.03	0.11 ± 0.04	0.11 ± 0.07	0.20 ± 0.03	0.14 ± 0.04
	CTransPath	**0.00 ± 0.00**	0.02 ± 0.03	0.03 ± 0.02	0.04 ± 0.01	**0.04 ± 0.05**	0.07 ± 0.06	0.07 ± 0.03	0.06 ± 0.04	0.06 ± 0.03	0.04 ± 0.03
	ViT-B	0.11 ± 0.04	0.12 ± 0.03	0.12 ± 0.02	0.04 ± 0.04	0.16 ± 0.12	0.14 ± 0.04	0.10 ± 0.04	0.13 ± 0.06	**0.02 ± 0.02**	0.10 ± 0.05
	Phikon	0.11 ± 0.03	0.04 ± 0.01	0.09 ± 0.04	0.06 ± 0.02	0.09 ± 0.09	0.03 ± 0.03	0.05 ± 0.05	0.09 ± 0.05	0.06 ± 0.06	0.07 ± 0.05
	ViT-S	0.10 ± 0.03	0.10 ± 0.03	0.10 ± 0.04	**0.01 ± 0.02**	0.19 ± 0.07	0.16 ± 0.04	0.06 ± 0.05	0.18 ± 0.07	0.07 ± 0.03	0.11 ± 0.05
	Lunit-DINO	0.04 ± 0.03	**0.01 ± 0.01**	0.04 ± 0.03	0.02 ± 0.02	0.06 ± 0.06	**0.01 ± 0.02**	0.07 ± 0.04	**0.02 ± 0.04**	0.05 ± 0.03	**0.04 ± 0.03**
	ResNet-50	0.17 ± 0.04	0.18 ± 0.04	0.17 ± 0.05	0.02 ± 0.01	0.17 ± 0.07	0.18 ± 0.03	0.13 ± 0.03	0.17 ± 0.07	0.15 ± 0.07	0.15 ± 0.05
	RetCCL	0.09 ± 0.04	0.03 ± 0.02	0.03 ± 0.04	0.03 ± 0.02	0.10 ± 0.09	0.07 ± 0.03	**0.02 ± 0.03**	0.14 ± 0.04	0.07 ± 0.03	0.06 ± 0.04
	Lunit-BT	0.11 ± 0.04	0.04 ± 0.01	**0.02 ± 0.02**	0.13 ± 0.02	0.25 ± 0.13	0.33 ± 0.07	0.08 ± 0.06	0.28 ± 0.10	0.15 ± 0.08	0.16 ± 0.07
	Lunit-SwAV	0.05 ± 0.03	0.03 ± 0.03	0.04 ± 0.02	0.05 ± 0.03	0.08 ± 0.07	0.12 ± 0.04	0.12 ± 0.08	0.07 ± 0.05	0.11 ± 0.05	0.08 ± 0.05
Transformer	Swin	0.11 ± 0.04	0.19 ± 0.05	0.20 ± 0.05	0.09 ± 0.04	0.19 ± 0.08	0.19 ± 0.04	0.15 ± 0.04	0.22 ± 0.07	0.09 ± 0.06	0.16 ± 0.05
	CTransPath	**0.01 ± 0.02**	0.05 ± 0.04	**0.02 ± 0.02**	0.06 ± 0.04	0.06 ± 0.07	0.04 ± 0.04	0.08 ± 0.05	0.07 ± 0.07	0.08 ± 0.05	0.05 ± 0.05
	ViT-B	0.11 ± 0.03	0.13 ± 0.03	0.18 ± 0.03	0.08 ± 0.03	0.16 ± 0.10	0.21 ± 0.07	0.13 ± 0.07	0.21 ± 0.04	0.09 ± 0.03	0.14 ± 0.05
	Phikon	0.08 ± 0.03	0.09 ± 0.03	0.07 ± 0.03	0.07 ± 0.04	0.07 ± 0.06	0.04 ± 0.02	0.05 ± 0.06	0.08 ± 0.04	**0.04 ± 0.03**	0.06 ± 0.04
	ViT-S	0.11 ± 0.03	0.11 ± 0.06	0.18 ± 0.03	0.07 ± 0.05	0.16 ± 0.09	0.16 ± 0.02	**0.04 ± 0.05**	0.19 ± 0.04	0.05 ± 0.06	0.12 ± 0.05
	Lunit-DINO	0.04 ± 0.03	0.05 ± 0.04	0.02 ± 0.02	**0.03 ± 0.03**	**0.04 ± 0.05**	**0.02 ± 0.03**	0.10 ± 0.04	0.09 ± 0.07	0.06 ± 0.06	**0.05 ± 0.04**
	ResNet-50	0.16 ± 0.05	0.18 ± 0.10	0.23 ± 0.04	0.04 ± 0.05	0.14 ± 0.08	0.21 ± 0.06	0.13 ± 0.05	0.16 ± 0.05	0.30 ± 0.11	0.17 ± 0.07
	RetCCL	0.06 ± 0.03	0.06 ± 0.04	0.04 ± 0.04	0.06 ± 0.02	0.08 ± 0.06	0.08 ± 0.04	0.09 ± 0.06	0.15 ± 0.08	0.07 ± 0.04	0.08 ± 0.05
	Lunit-BT	0.03 ± 0.03	0.04 ± 0.03	0.05 ± 0.03	0.07 ± 0.04	0.05 ± 0.06	0.05 ± 0.03	0.09 ± 0.06	0.15 ± 0.04	0.07 ± 0.06	0.07 ± 0.04
	Lunit-SwAV	0.06 ± 0.03	**0.01 ± 0.01**	0.03 ± 0.02	0.08 ± 0.04	0.07 ± 0.06	0.08 ± 0.04	0.15 ± 0.05	**0.07 ± 0.10**	0.12 ± 0.02	0.08 ± 0.05
Mean pool	Swin	0.06 ± 0.01	0.12 ± 0.02	0.11 ± 0.04	0.01 ± 0.01	0.20 ± 0.11	0.11 ± 0.03	0.04 ± 0.03	0.15 ± 0.04	0.04 ± 0.01	0.09 ± 0.04
	CTransPath	**0.00 ± 0.00**	**0.01 ± 0.01**	**0.02 ± 0.02**	**0.01 ± 0.01**	0.18 ± 0.10	0.03 ± 0.03	0.09 ± 0.05	0.07 ± 0.04	0.05 ± 0.02	0.05 ± 0.04
	ViT-B	0.03 ± 0.00	0.09 ± 0.01	0.07 ± 0.03	0.03 ± 0.01	0.17 ± 0.10	0.17 ± 0.04	0.10 ± 0.05	0.16 ± 0.06	0.02 ± 0.02	0.09 ± 0.05
	Phikon	0.11 ± 0.01	0.01 ± 0.01	0.11 ± 0.03	0.08 ± 0.04	0.16 ± 0.15	**0.02 ± 0.03**	0.05 ± 0.03	0.09 ± 0.03	0.07 ± 0.06	0.08 ± 0.06
	ViT-S	0.06 ± 0.01	0.05 ± 0.04	0.09 ± 0.05	0.02 ± 0.02	0.17 ± 0.12	0.17 ± 0.03	**0.02 ± 0.01**	0.22 ± 0.06	0.07 ± 0.04	0.10 ± 0.05
	Lunit-DINO	0.05 ± 0.01	0.02 ± 0.01	0.04 ± 0.04	0.04 ± 0.02	**0.11 ± 0.12**	0.04 ± 0.04	0.07 ± 0.04	**0.00 ± 0.00**	0.03 ± 0.02	**0.04 ± 0.05**
	ResNet-50	0.08 ± 0.00	0.11 ± 0.04	0.07 ± 0.03	0.03 ± 0.01	0.22 ± 0.11	0.15 ± 0.05	0.03 ± 0.03	0.21 ± 0.04	0.11 ± 0.10	0.11 ± 0.06
	RetCCL	0.01 ± 0.00	0.02 ± 0.01	0.05 ± 0.03	0.03 ± 0.01	0.14 ± 0.10	0.04 ± 0.03	0.05 ± 0.05	0.14 ± 0.05	0.03 ± 0.01	0.06 ± 0.04
	Lunit-BT	0.06 ± 0.03	0.02 ± 0.01	0.03 ± 0.03	0.05 ± 0.01	0.18 ± 0.12	0.11 ± 0.04	0.02 ± 0.03	0.18 ± 0.03	**0.00 ± 0.01**	0.07 ± 0.05
	Lunit-SwAV	0.06 ± 0.00	0.02 ± 0.01	0.04 ± 0.02	0.12 ± 0.01	0.12 ± 0.11	0.12 ± 0.03	0.15 ± 0.02	0.04 ± 0.03	0.09 ± 0.02	0.08 ± 0.04

Table 6. Normalised differential AUROC scores for all tasks, feature extractors, downstream models, when employing **patchwise stain normalisation** [59].

	Target	🎗-Subtype	🎗-CDH1	🎗-TP53	🎗-PIK3CA	🎗-LN status	-MSI	-KRAS	-BRAF	-SMAD4	Average
Model	Feature extractor										
AttMIL	Swin	0.05 ± 0.03	0.16 ± 0.05	0.28 ± 0.03	0.07 ± 0.02	0.14 ± 0.08	0.11 ± 0.03	0.13 ± 0.03	0.10 ± 0.04	0.20 ± 0.03	0.14 ± 0.04
	CTransPath	**0.00 ± 0.00**	**0.02 ± 0.03**	0.03 ± 0.02	**0.01 ± 0.01**	**0.05 ± 0.05**	0.04 ± 0.03	0.07 ± 0.05	0.07 ± 0.02	0.06 ± 0.03	0.04 ± 0.03
	ViT-B	0.07 ± 0.05	0.10 ± 0.02	0.15 ± 0.03	0.08 ± 0.03	0.16 ± 0.06	0.13 ± 0.03	0.09 ± 0.08	0.13 ± 0.04	**0.01 ± 0.02**	0.10 ± 0.04
	Phikon	0.07 ± 0.03	0.07 ± 0.03	0.06 ± 0.06	0.11 ± 0.03	0.07 ± 0.06	0.04 ± 0.03	0.07 ± 0.04	0.09 ± 0.08	0.19 ± 0.09	0.08 ± 0.06
	ViT-S	0.06 ± 0.03	0.04 ± 0.02	0.14 ± 0.04	0.06 ± 0.04	0.21 ± 0.10	0.19 ± 0.06	0.05 ± 0.04	0.19 ± 0.05	0.07 ± 0.08	0.11 ± 0.05
	Lunit-DINO	0.06 ± 0.03	0.04 ± 0.03	**0.02 ± 0.02**	0.01 ± 0.02	0.05 ± 0.06	**0.00 ± 0.00**	0.06 ± 0.02	**0.01 ± 0.02**	0.04 ± 0.03	**0.03 ± 0.03**
	ResNet-50	0.13 ± 0.03	0.10 ± 0.04	0.13 ± 0.04	0.03 ± 0.03	0.15 ± 0.10	0.22 ± 0.05	0.14 ± 0.05	0.22 ± 0.06	0.29 ± 0.08	0.16 ± 0.06
	RetCCL	0.05 ± 0.03	0.04 ± 0.03	0.03 ± 0.03	0.04 ± 0.03	0.07 ± 0.07	0.06 ± 0.03	**0.03 ± 0.04**	0.16 ± 0.03	0.06 ± 0.03	0.06 ± 0.04
	Lunit-BT	0.08 ± 0.04	0.03 ± 0.03	0.04 ± 0.05	0.12 ± 0.03	0.29 ± 0.20	0.25 ± 0.12	0.08 ± 0.08	0.34 ± 0.11	0.21 ± 0.05	0.16 ± 0.09
	Lunit-SwAV	0.06 ± 0.03	0.06 ± 0.03	0.07 ± 0.04	0.10 ± 0.05	0.07 ± 0.06	0.06 ± 0.03	0.08 ± 0.05	0.05 ± 0.05	0.11 ± 0.04	0.07 ± 0.04
Transformer	Swin	0.06 ± 0.03	0.10 ± 0.04	0.24 ± 0.04	0.04 ± 0.03	0.15 ± 0.09	0.10 ± 0.03	0.05 ± 0.05	0.19 ± 0.08	0.13 ± 0.06	0.12 ± 0.05
	CTransPath	**0.01 ± 0.01**	**0.02 ± 0.02**	0.04 ± 0.04	0.08 ± 0.03	0.05 ± 0.05	0.08 ± 0.05	0.07 ± 0.05	0.13 ± 0.08	0.06 ± 0.03	0.06 ± 0.05
	ViT-B	0.07 ± 0.03	0.08 ± 0.04	0.15 ± 0.04	0.08 ± 0.04	0.17 ± 0.06	0.20 ± 0.03	0.11 ± 0.02	0.14 ± 0.06	0.05 ± 0.04	0.12 ± 0.04
	Phikon	0.06 ± 0.04	0.09 ± 0.03	0.08 ± 0.02	0.08 ± 0.03	0.06 ± 0.04	0.05 ± 0.04	0.04 ± 0.03	0.05 ± 0.04	0.15 ± 0.05	0.07 ± 0.04
	ViT-S	0.06 ± 0.03	0.04 ± 0.03	0.17 ± 0.04	0.10 ± 0.06	0.18 ± 0.07	0.19 ± 0.02	0.10 ± 0.04	0.16 ± 0.05	**0.04 ± 0.04**	0.12 ± 0.04
	Lunit-DINO	0.03 ± 0.03	0.08 ± 0.03	0.03 ± 0.02	**0.02 ± 0.02**	**0.04 ± 0.04**	**0.00 ± 0.01**	0.07 ± 0.03	**0.02 ± 0.02**	0.06 ± 0.06	**0.04 ± 0.03**
	ResNet-50	0.09 ± 0.02	0.09 ± 0.05	0.18 ± 0.04	0.04 ± 0.05	0.18 ± 0.06	0.24 ± 0.05	0.09 ± 0.04	0.17 ± 0.07	0.32 ± 0.05	0.16 ± 0.05
	RetCCL	0.07 ± 0.05	0.06 ± 0.04	**0.02 ± 0.02**	0.09 ± 0.04	0.06 ± 0.05	0.19 ± 0.05	0.12 ± 0.07	0.16 ± 0.06	0.11 ± 0.08	0.10 ± 0.06
	Lunit-BT	0.02 ± 0.02	0.05 ± 0.04	0.05 ± 0.04	0.06 ± 0.03	0.07 ± 0.06	0.04 ± 0.04	**0.02 ± 0.03**	0.12 ± 0.04	0.05 ± 0.02	0.05 ± 0.04
	Lunit-SwAV	0.06 ± 0.04	0.04 ± 0.03	0.05 ± 0.02	0.12 ± 0.04	0.07 ± 0.05	0.08 ± 0.05	0.10 ± 0.03	0.05 ± 0.06	0.18 ± 0.06	0.08 ± 0.05
Mean pool	Swin	0.08 ± 0.01	0.10 ± 0.03	0.17 ± 0.04	0.06 ± 0.03	0.15 ± 0.11	0.14 ± 0.02	0.05 ± 0.05	0.13 ± 0.02	0.13 ± 0.03	0.11 ± 0.05
	CTransPath	**0.00 ± 0.00**	0.04 ± 0.02	0.04 ± 0.03	**0.01 ± 0.02**	0.16 ± 0.11	0.03 ± 0.02	0.10 ± 0.03	0.04 ± 0.02	0.06 ± 0.03	0.05 ± 0.04
	ViT-B	0.07 ± 0.01	0.08 ± 0.01	0.10 ± 0.02	0.09 ± 0.02	0.17 ± 0.09	0.13 ± 0.03	0.09 ± 0.05	0.16 ± 0.03	**0.01 ± 0.01**	0.10 ± 0.04
	Phikon	0.08 ± 0.01	0.05 ± 0.02	0.13 ± 0.03	0.03 ± 0.03	0.12 ± 0.12	**0.01 ± 0.01**	0.13 ± 0.04	0.08 ± 0.08	0.09 ± 0.02	0.08 ± 0.05
	ViT-S	0.08 ± 0.01	0.03 ± 0.02	0.15 ± 0.03	0.09 ± 0.02	0.14 ± 0.08	0.15 ± 0.02	**0.02 ± 0.02**	0.21 ± 0.05	0.07 ± 0.03	0.10 ± 0.03
	Lunit-DINO	0.09 ± 0.01	0.04 ± 0.02	**0.00 ± 0.01**	0.05 ± 0.03	**0.09 ± 0.09**	0.01 ± 0.02	0.10 ± 0.04	**0.00 ± 0.01**	0.01 ± 0.02	**0.04 ± 0.03**
	ResNet-50	0.08 ± 0.01	**0.00 ± 0.01**	0.12 ± 0.02	0.04 ± 0.03	0.19 ± 0.10	0.21 ± 0.05	0.04 ± 0.03	0.23 ± 0.04	0.12 ± 0.04	0.12 ± 0.04
	RetCCL	0.01 ± 0.00	0.04 ± 0.01	0.08 ± 0.03	0.07 ± 0.03	0.14 ± 0.12	0.11 ± 0.04	0.07 ± 0.05	0.14 ± 0.01	0.05 ± 0.01	0.08 ± 0.05
	Lunit-BT	0.06 ± 0.02	0.04 ± 0.01	0.06 ± 0.04	0.08 ± 0.02	0.22 ± 0.08	0.09 ± 0.05	0.02 ± 0.02	0.16 ± 0.01	0.02 ± 0.01	0.08 ± 0.04
	Lunit-SwAV	0.07 ± 0.00	0.04 ± 0.02	0.08 ± 0.04	0.11 ± 0.03	0.13 ± 0.12	0.05 ± 0.02	0.13 ± 0.03	0.03 ± 0.02	0.11 ± 0.04	0.08 ± 0.05

Table 7. Normalised differential AUROC scores for all tasks, feature extractors, downstream models, when employing **rotation/flipping augmentations**.

Model	Target Feature extractor	🎗-Subtype	🎗-CDH1	🎗-TP53	🎗-PIK3CA	🎗-LN status	-MSI	-KRAS	-BRAF	-SMAD4	Average
AttMIL	Swin	0.04 ± 0.03	0.14 ± 0.02	0.21 ± 0.02	0.07 ± 0.04	0.13 ± 0.08	0.15 ± 0.04	0.10 ± 0.05	0.16 ± 0.08	0.17 ± 0.05	0.13 ± 0.05
	CTransPath	**0.00 ± 0.00**	**0.01 ± 0.02**	**0.00 ± 0.01**	0.04 ± 0.03	**0.03 ± 0.03**	0.10 ± 0.04	0.06 ± 0.03	0.09 ± 0.07	0.07 ± 0.03	0.04 ± 0.03
	ViT-B	0.04 ± 0.03	0.10 ± 0.04	0.12 ± 0.03	0.08 ± 0.04	0.14 ± 0.06	0.13 ± 0.04	0.06 ± 0.03	0.16 ± 0.05	**0.02 ± 0.02**	0.09 ± 0.04
	Phikon	0.13 ± 0.05	0.09 ± 0.03	0.10 ± 0.05	0.12 ± 0.05	0.07 ± 0.07	0.06 ± 0.03	0.10 ± 0.05	0.18 ± 0.08	0.13 ± 0.06	0.11 ± 0.05
	ViT-S	0.08 ± 0.03	0.07 ± 0.02	0.14 ± 0.02	0.08 ± 0.04	0.17 ± 0.09	0.15 ± 0.03	0.04 ± 0.03	0.19 ± 0.06	0.06 ± 0.05	0.11 ± 0.05
	Lunit-DINO	0.05 ± 0.04	0.03 ± 0.03	0.04 ± 0.02	0.04 ± 0.04	0.06 ± 0.06	**0.00 ± 0.01**	0.07 ± 0.04	**0.01 ± 0.02**	0.05 ± 0.05	**0.04 ± 0.04**
	ResNet-50	0.09 ± 0.03	0.07 ± 0.03	0.14 ± 0.03	**0.01 ± 0.01**	0.16 ± 0.08	0.24 ± 0.05	0.14 ± 0.03	0.24 ± 0.08	0.30 ± 0.11	0.15 ± 0.06
	RetCCL	0.06 ± 0.04	0.03 ± 0.03	0.03 ± 0.02	0.07 ± 0.03	0.06 ± 0.05	0.11 ± 0.06	**0.04 ± 0.05**	0.18 ± 0.05	0.06 ± 0.02	0.07 ± 0.04
	Lunit-BT	0.17 ± 0.05	0.11 ± 0.07	0.20 ± 0.20	0.17 ± 0.03	0.40 ± 0.07	0.21 ± 0.10	0.12 ± 0.05	0.24 ± 0.09	0.20 ± 0.05	0.20 ± 0.09
	Lunit-SwAV	0.07 ± 0.03	0.03 ± 0.02	0.07 ± 0.04	0.10 ± 0.04	0.07 ± 0.06	0.08 ± 0.03	0.07 ± 0.05	0.13 ± 0.07	0.11 ± 0.05	0.08 ± 0.04
Transformer	Swin	0.07 ± 0.02	0.13 ± 0.06	0.21 ± 0.03	0.03 ± 0.03	0.13 ± 0.09	0.13 ± 0.03	0.06 ± 0.06	0.07 ± 0.04	0.11 ± 0.03	0.10 ± 0.05
	CTransPath	**0.02 ± 0.02**	0.06 ± 0.02	0.03 ± 0.02	0.04 ± 0.03	**0.04 ± 0.04**	0.06 ± 0.04	0.08 ± 0.03	0.07 ± 0.08	0.13 ± 0.06	0.06 ± 0.04
	ViT-B	0.04 ± 0.03	0.11 ± 0.04	0.15 ± 0.03	0.09 ± 0.02	0.18 ± 0.13	0.15 ± 0.02	0.16 ± 0.05	0.23 ± 0.07	0.03 ± 0.03	0.13 ± 0.06
	Phikon	0.12 ± 0.03	0.10 ± 0.04	0.06 ± 0.03	0.11 ± 0.03	0.08 ± 0.05	0.05 ± 0.04	0.04 ± 0.03	**0.01 ± 0.02**	0.15 ± 0.05	0.08 ± 0.04
	ViT-S	0.06 ± 0.03	0.06 ± 0.03	0.14 ± 0.05	0.08 ± 0.03	0.19 ± 0.05	0.17 ± 0.05	0.06 ± 0.04	0.20 ± 0.04	**0.02 ± 0.02**	0.11 ± 0.04
	Lunit-DINO	0.04 ± 0.03	0.05 ± 0.03	0.02 ± 0.01	0.04 ± 0.03	0.06 ± 0.06	**0.01 ± 0.01**	0.09 ± 0.05	0.06 ± 0.04	0.02 ± 0.02	**0.04 ± 0.04**
	ResNet-50	0.09 ± 0.03	0.12 ± 0.04	0.10 ± 0.03	**0.01 ± 0.02**	0.18 ± 0.08	0.18 ± 0.03	0.04 ± 0.03	0.18 ± 0.05	0.27 ± 0.07	0.13 ± 0.05
	RetCCL	0.03 ± 0.03	0.06 ± 0.04	**0.01 ± 0.01**	0.09 ± 0.04	0.08 ± 0.07	0.12 ± 0.07	0.13 ± 0.05	0.24 ± 0.08	0.13 ± 0.07	0.10 ± 0.06
	Lunit-BT	0.03 ± 0.03	0.03 ± 0.03	0.04 ± 0.03	0.10 ± 0.03	0.09 ± 0.08	0.07 ± 0.06	**0.03 ± 0.02**	0.13 ± 0.04	0.05 ± 0.02	0.06 ± 0.04
	Lunit-SwAV	0.08 ± 0.03	**0.02 ± 0.03**	0.03 ± 0.03	0.10 ± 0.03	0.07 ± 0.06	0.10 ± 0.04	0.10 ± 0.04	0.06 ± 0.04	0.16 ± 0.06	0.08 ± 0.04
Mean pool	Swin	0.06 ± 0.01	0.10 ± 0.03	0.16 ± 0.03	0.04 ± 0.01	0.19 ± 0.12	0.15 ± 0.02	0.03 ± 0.04	0.18 ± 0.05	0.13 ± 0.04	0.12 ± 0.05
	CTransPath	**0.00 ± 0.00**	0.03 ± 0.02	0.04 ± 0.02	**0.00 ± 0.00**	0.15 ± 0.11	0.04 ± 0.03	0.08 ± 0.03	0.04 ± 0.02	0.09 ± 0.03	0.05 ± 0.04
	ViT-B	0.07 ± 0.01	0.08 ± 0.01	0.10 ± 0.02	0.08 ± 0.01	0.18 ± 0.08	0.17 ± 0.02	0.11 ± 0.05	0.20 ± 0.02	0.02 ± 0.02	0.11 ± 0.04
	Phikon	0.11 ± 0.01	0.02 ± 0.02	0.13 ± 0.03	0.07 ± 0.04	0.12 ± 0.11	0.02 ± 0.02	0.11 ± 0.05	0.09 ± 0.07	0.12 ± 0.03	0.09 ± 0.05
	ViT-S	0.11 ± 0.01	0.03 ± 0.02	0.16 ± 0.02	0.06 ± 0.01	0.16 ± 0.11	0.20 ± 0.03	0.04 ± 0.02	0.23 ± 0.03	0.06 ± 0.03	0.12 ± 0.04
	Lunit-DINO	0.09 ± 0.01	0.02 ± 0.02	**0.01 ± 0.02**	0.04 ± 0.03	**0.08 ± 0.09**	**0.01 ± 0.02**	0.09 ± 0.02	**0.00 ± 0.00**	**0.00 ± 0.01**	**0.04 ± 0.03**
	ResNet-50	0.08 ± 0.01	**0.01 ± 0.01**	0.11 ± 0.02	0.02 ± 0.01	0.23 ± 0.10	0.22 ± 0.03	**0.01 ± 0.01**	0.27 ± 0.05	0.15 ± 0.06	0.12 ± 0.04
	RetCCL	0.01 ± 0.01	0.03 ± 0.01	0.07 ± 0.02	0.06 ± 0.01	0.14 ± 0.11	0.10 ± 0.05	0.08 ± 0.07	0.16 ± 0.03	0.06 ± 0.02	0.08 ± 0.05
	Lunit-BT	0.08 ± 0.04	0.04 ± 0.01	0.10 ± 0.05	0.09 ± 0.02	0.29 ± 0.09	0.12 ± 0.07	0.03 ± 0.02	0.19 ± 0.02	0.09 ± 0.14	0.11 ± 0.07
	Lunit-SwAV	0.07 ± 0.00	0.02 ± 0.01	0.03 ± 0.02	0.10 ± 0.02	0.15 ± 0.13	0.05 ± 0.02	0.13 ± 0.04	0.11 ± 0.05	0.13 ± 0.05	0.09 ± 0.05

Table 8. Normalised differential AUROC scores for all tasks, feature extractors, downstream models, when employing **all augmentations**.

Model	Target Feature extractor	🎗-Subtype	🎗-CDH1	🎗-TP53	🎗-PIK3CA	🎗-LN status	-MSI	-KRAS	-BRAF	-SMAD4
AttMIL	Swin	0.75 ± 0.01	0.65 ± 0.02	0.54 ± 0.02	0.60 ± 0.02	0.74 ± 0.09	0.72 ± 0.04	0.51 ± 0.05	0.63 ± 0.07	0.55 ± 0.05
	CTransPath	**0.82 ± 0.02**	**0.81 ± 0.02**	**0.80 ± 0.02**	0.62 ± 0.01	**0.86 ± 0.08**	0.82 ± 0.03	0.60 ± 0.03	0.71 ± 0.01	0.65 ± 0.02
	ViT-B	0.74 ± 0.04	0.70 ± 0.01	0.66 ± 0.03	0.59 ± 0.01	0.74 ± 0.06	0.75 ± 0.03	0.62 ± 0.05	0.59 ± 0.08	**0.70 ± 0.03**
	Phikon	0.73 ± 0.01	0.73 ± 0.02	0.72 ± 0.03	0.57 ± 0.02	0.85 ± 0.08	0.84 ± 0.05	0.59 ± 0.05	0.70 ± 0.06	0.54 ± 0.08
	ViT-S	0.69 ± 0.02	0.73 ± 0.02	0.68 ± 0.06	0.58 ± 0.04	0.73 ± 0.10	0.72 ± 0.04	0.59 ± 0.03	0.58 ± 0.03	0.63 ± 0.08
	Lunit-DINO	0.74 ± 0.02	0.78 ± 0.04	0.79 ± 0.03	0.64 ± 0.02	0.85 ± 0.03	**0.90 ± 0.02**	0.59 ± 0.04	**0.76 ± 0.04**	0.69 ± 0.02
	ResNet-50	0.67 ± 0.02	0.73 ± 0.04	0.70 ± 0.03	**0.65 ± 0.04**	0.74 ± 0.09	0.68 ± 0.04	0.54 ± 0.04	0.55 ± 0.07	0.50 ± 0.10
	RetCCL	0.76 ± 0.03	0.78 ± 0.01	0.78 ± 0.03	0.62 ± 0.01	0.85 ± 0.07	0.82 ± 0.03	**0.63 ± 0.03**	0.63 ± 0.02	0.66 ± 0.02
	Lunit-BT	0.69 ± 0.03	0.75 ± 0.04	0.80 ± 0.00	0.54 ± 0.03	0.58 ± 0.17	0.62 ± 0.15	0.62 ± 0.05	0.43 ± 0.15	0.46 ± 0.03
	Lunit-SwAV	0.76 ± 0.01	0.75 ± 0.03	0.76 ± 0.02	0.54 ± 0.06	0.84 ± 0.06	0.80 ± 0.03	0.53 ± 0.06	0.70 ± 0.08	0.58 ± 0.09
Transformer	Swin	0.74 ± 0.04	0.70 ± 0.02	0.61 ± 0.03	0.54 ± 0.04	0.76 ± 0.09	0.69 ± 0.08	0.56 ± 0.03	0.60 ± 0.04	0.57 ± 0.05
	CTransPath	**0.81 ± 0.03**	**0.80 ± 0.01**	0.80 ± 0.03	0.55 ± 0.08	0.85 ± 0.09	0.86 ± 0.02	0.60 ± 0.04	0.68 ± 0.07	0.62 ± 0.05
	ViT-B	0.74 ± 0.03	0.71 ± 0.02	0.65 ± 0.03	0.52 ± 0.01	0.71 ± 0.07	0.70 ± 0.06	0.51 ± 0.05	0.56 ± 0.08	0.65 ± 0.06
	Phikon	0.69 ± 0.04	0.73 ± 0.05	0.75 ± 0.02	0.59 ± 0.03	0.85 ± 0.06	0.83 ± 0.04	0.60 ± 0.04	0.65 ± 0.07	0.59 ± 0.06
	ViT-S	0.72 ± 0.01	0.74 ± 0.03	0.60 ± 0.08	0.52 ± 0.07	0.71 ± 0.10	0.72 ± 0.07	0.57 ± 0.04	0.53 ± 0.10	0.68 ± 0.03
	Lunit-DINO	0.78 ± 0.04	0.75 ± 0.03	0.79 ± 0.01	**0.62 ± 0.02**	**0.87 ± 0.05**	**0.87 ± 0.02**	0.59 ± 0.02	**0.74 ± 0.05**	**0.69 ± 0.03**
	ResNet-50	0.69 ± 0.04	0.71 ± 0.08	0.67 ± 0.02	0.59 ± 0.08	0.73 ± 0.09	0.69 ± 0.07	0.54 ± 0.03	0.57 ± 0.06	0.41 ± 0.12
	RetCCL	0.73 ± 0.03	0.77 ± 0.05	0.80 ± 0.04	0.55 ± 0.06	0.85 ± 0.07	0.73 ± 0.03	0.53 ± 0.05	0.55 ± 0.11	0.65 ± 0.06
	Lunit-BT	0.78 ± 0.03	0.76 ± 0.03	**0.80 ± 0.01**	0.53 ± 0.05	0.85 ± 0.08	0.86 ± 0.02	**0.63 ± 0.03**	0.63 ± 0.04	0.65 ± 0.02
	Lunit-SwAV	0.74 ± 0.05	0.77 ± 0.06	0.77 ± 0.02	0.53 ± 0.06	0.85 ± 0.06	0.82 ± 0.03	0.57 ± 0.03	0.69 ± 0.05	0.54 ± 0.07
Mean pool	Swin	0.73 ± 0.01	0.68 ± 0.04	0.62 ± 0.05	0.59 ± 0.02	0.67 ± 0.13	0.72 ± 0.02	**0.66 ± 0.02**	0.67 ± 0.03	0.61 ± 0.02
	CTransPath	**0.82 ± 0.00**	0.74 ± 0.02	0.72 ± 0.02	**0.64 ± 0.02**	0.69 ± 0.12	0.86 ± 0.02	0.58 ± 0.06	0.73 ± 0.04	0.62 ± 0.02
	ViT-B	0.75 ± 0.01	0.71 ± 0.01	0.68 ± 0.02	0.56 ± 0.01	0.69 ± 0.11	0.74 ± 0.02	0.61 ± 0.04	0.61 ± 0.04	0.69 ± 0.02
	Phikon	0.71 ± 0.02	0.74 ± 0.03	0.61 ± 0.03	0.59 ± 0.03	0.73 ± 0.12	0.82 ± 0.04	0.57 ± 0.03	0.70 ± 0.07	0.60 ± 0.05
	ViT-S	0.71 ± 0.01	0.76 ± 0.04	0.61 ± 0.01	0.57 ± 0.02	0.69 ± 0.11	0.70 ± 0.04	0.65 ± 0.03	0.58 ± 0.05	0.64 ± 0.02
	Lunit-DINO	0.74 ± 0.01	0.74 ± 0.02	**0.73 ± 0.02**	0.60 ± 0.03	**0.75 ± 0.12**	**0.89 ± 0.02**	0.60 ± 0.01	**0.79 ± 0.01**	**0.70 ± 0.03**
	ResNet-50	0.74 ± 0.01	**0.78 ± 0.01**	0.65 ± 0.01	0.61 ± 0.01	0.63 ± 0.09	0.67 ± 0.03	0.66 ± 0.04	0.56 ± 0.03	0.58 ± 0.05
	RetCCL	0.81 ± 0.00	0.75 ± 0.01	0.68 ± 0.02	0.58 ± 0.01	0.69 ± 0.12	0.79 ± 0.05	0.66 ± 0.03	0.64 ± 0.01	0.65 ± 0.00
	Lunit-BT	0.76 ± 0.03	0.75 ± 0.00	0.69 ± 0.05	0.57 ± 0.01	0.66 ± 0.12	0.81 ± 0.02	0.66 ± 0.03	0.58 ± 0.10	0.68 ± 0.01
	Lunit-SwAV	0.75 ± 0.00	0.75 ± 0.02	0.70 ± 0.02	0.53 ± 0.01	0.71 ± 0.15	0.84 ± 0.01	0.56 ± 0.03	0.76 ± 0.01	0.58 ± 0.05

Table 9. Test AUROC scores (averaged across the five seeds) for all tasks, feature extractors, and downstream models, when employing **no augmentations**.

	Target	🎗-Subtype	🎗-CDH1	🎗-TP53	🎗-PIK3CA	🎗-LN status	⊓-MSI	⊓-KRAS	⊓-BRAF	⊓-SMAD4
Model	Feature extractor									
AttMIL	Swin	0.74 ± 0.02	0.58 ± 0.03	0.55 ± 0.03	0.58 ± 0.05	0.73 ± 0.09	0.72 ± 0.04	0.55 ± 0.01	0.66 ± 0.05	0.55 ± 0.05
	CTransPath	**0.82 ± 0.04**	0.77 ± 0.05	0.78 ± 0.03	0.62 ± 0.01	0.84 ± 0.10	0.84 ± 0.00	0.61 ± 0.04	0.69 ± 0.05	0.63 ± 0.03
	ViT-B	0.74 ± 0.02	0.65 ± 0.03	0.69 ± 0.02	0.61 ± 0.04	0.75 ± 0.08	0.77 ± 0.03	0.60 ± 0.03	0.67 ± 0.02	**0.70 ± 0.04**
	Phikon	0.73 ± 0.02	0.72 ± 0.04	0.73 ± 0.04	0.55 ± 0.03	0.82 ± 0.08	0.85 ± 0.03	0.57 ± 0.04	0.65 ± 0.01	0.65 ± 0.05
	ViT-S	0.70 ± 0.04	0.68 ± 0.04	0.70 ± 0.04	0.61 ± 0.03	0.74 ± 0.10	0.72 ± 0.06	0.61 ± 0.04	0.62 ± 0.03	0.62 ± 0.13
	Lunit-DINO	0.78 ± 0.02	0.77 ± 0.02	0.78 ± 0.03	**0.63 ± 0.01**	**0.84 ± 0.08**	**0.91 ± 0.04**	0.65 ± 0.04	**0.76 ± 0.06**	0.62 ± 0.07
	ResNet-50	0.69 ± 0.03	0.61 ± 0.05	0.66 ± 0.04	0.62 ± 0.02	0.74 ± 0.08	0.69 ± 0.03	0.53 ± 0.05	0.62 ± 0.02	0.60 ± 0.07
	RetCCL	0.76 ± 0.03	0.78 ± 0.03	0.78 ± 0.03	0.60 ± 0.03	0.82 ± 0.06	0.85 ± 0.03	**0.69 ± 0.01**	0.63 ± 0.02	0.64 ± 0.01
	Lunit-BT	0.69 ± 0.05	0.76 ± 0.04	0.75 ± 0.10	0.53 ± 0.02	0.64 ± 0.19	0.75 ± 0.17	0.63 ± 0.08	0.42 ± 0.07	0.49 ± 0.07
	Lunit-SwAV	0.75 ± 0.03	**0.78 ± 0.02**	**0.79 ± 0.02**	0.60 ± 0.05	0.83 ± 0.06	0.79 ± 0.04	0.58 ± 0.03	0.71 ± 0.04	0.58 ± 0.07
Transformer	Swin	0.73 ± 0.03	0.66 ± 0.04	0.61 ± 0.03	0.58 ± 0.03	0.74 ± 0.10	0.69 ± 0.10	0.57 ± 0.06	0.53 ± 0.04	0.55 ± 0.09
	CTransPath	0.79 ± 0.03	**0.79 ± 0.03**	0.76 ± 0.05	0.54 ± 0.05	**0.87 ± 0.08**	0.88 ± 0.02	0.63 ± 0.03	**0.71 ± 0.05**	0.54 ± 0.09
	ViT-B	0.70 ± 0.04	0.67 ± 0.03	0.64 ± 0.03	0.60 ± 0.03	0.71 ± 0.09	0.68 ± 0.07	0.58 ± 0.04	0.52 ± 0.11	0.67 ± 0.04
	Phikon	0.72 ± 0.01	0.73 ± 0.01	0.72 ± 0.04	0.59 ± 0.01	0.82 ± 0.09	0.86 ± 0.03	0.63 ± 0.07	0.66 ± 0.08	**0.68 ± 0.04**
	ViT-S	0.73 ± 0.01	0.65 ± 0.05	0.66 ± 0.06	0.57 ± 0.03	0.76 ± 0.10	0.68 ± 0.09	0.59 ± 0.03	0.60 ± 0.02	0.67 ± 0.03
	Lunit-DINO	**0.81 ± 0.03**	0.74 ± 0.04	**0.79 ± 0.03**	**0.60 ± 0.03**	0.86 ± 0.06	**0.89 ± 0.03**	0.59 ± 0.07	0.71 ± 0.06	0.64 ± 0.07
	ResNet-50	0.68 ± 0.03	0.61 ± 0.07	0.64 ± 0.04	0.59 ± 0.02	0.70 ± 0.08	0.69 ± 0.04	0.56 ± 0.03	0.62 ± 0.06	0.51 ± 0.14
	RetCCL	0.76 ± 0.05	0.75 ± 0.04	0.78 ± 0.02	0.56 ± 0.05	0.81 ± 0.04	0.81 ± 0.02	0.58 ± 0.04	0.54 ± 0.09	0.63 ± 0.03
	Lunit-BT	0.80 ± 0.03	0.78 ± 0.02	0.78 ± 0.03	0.57 ± 0.01	0.85 ± 0.06	0.86 ± 0.02	**0.67 ± 0.02**	0.60 ± 0.07	0.66 ± 0.01
	Lunit-SwAV	0.76 ± 0.03	0.79 ± 0.01	0.77 ± 0.04	0.56 ± 0.06	0.83 ± 0.10	0.78 ± 0.06	0.55 ± 0.05	0.59 ± 0.11	0.53 ± 0.09
Mean pool	Swin	0.76 ± 0.01	0.62 ± 0.02	0.60 ± 0.04	0.61 ± 0.01	0.62 ± 0.09	0.73 ± 0.03	0.63 ± 0.05	0.67 ± 0.07	0.63 ± 0.03
	CTransPath	**0.83 ± 0.00**	0.74 ± 0.00	0.71 ± 0.01	**0.64 ± 0.01**	0.67 ± 0.09	0.89 ± 0.01	0.60 ± 0.05	0.74 ± 0.03	0.62 ± 0.02
	ViT-B	0.78 ± 0.01	0.68 ± 0.01	0.67 ± 0.02	0.60 ± 0.01	0.67 ± 0.12	0.75 ± 0.02	0.60 ± 0.04	0.67 ± 0.07	**0.70 ± 0.01**
	Phikon	0.70 ± 0.01	0.73 ± 0.02	0.64 ± 0.03	0.58 ± 0.02	0.69 ± 0.12	**0.91 ± 0.02**	**0.65 ± 0.03**	0.69 ± 0.06	0.63 ± 0.03
	ViT-S	0.75 ± 0.01	0.68 ± 0.02	0.63 ± 0.03	0.59 ± 0.03	0.63 ± 0.11	0.74 ± 0.06	0.65 ± 0.03	0.59 ± 0.04	0.67 ± 0.03
	Lunit-DINO	0.76 ± 0.01	0.74 ± 0.03	0.70 ± 0.05	0.60 ± 0.01	**0.75 ± 0.12**	0.89 ± 0.01	0.63 ± 0.03	0.77 ± 0.05	0.65 ± 0.02
	ResNet-50	0.74 ± 0.01	0.65 ± 0.05	0.60 ± 0.02	0.61 ± 0.01	0.61 ± 0.10	0.73 ± 0.04	0.61 ± 0.04	0.65 ± 0.02	0.65 ± 0.06
	RetCCL	0.80 ± 0.00	**0.76 ± 0.01**	0.68 ± 0.03	0.59 ± 0.00	0.69 ± 0.10	0.86 ± 0.01	0.65 ± 0.02	0.67 ± 0.03	0.66 ± 0.00
	Lunit-BT	0.73 ± 0.03	0.75 ± 0.00	0.71 ± 0.04	0.57 ± 0.00	0.60 ± 0.10	0.76 ± 0.04	0.61 ± 0.05	0.60 ± 0.08	0.68 ± 0.01
	Lunit-SwAV	0.74 ± 0.01	0.75 ± 0.01	**0.72 ± 0.02**	0.49 ± 0.02	0.69 ± 0.11	0.76 ± 0.01	0.51 ± 0.02	**0.78 ± 0.02**	0.57 ± 0.04

Table 10. Test AUROC scores (averaged across the five seeds) for all tasks, feature extractors, and downstream models, when employing **slidewise stain normalisation** [59].

	Target	🎗-Subtype	🎗-CDH1	🎗-TP53	🎗-PIK3CA	🎗-LN status	⊓-MSI	⊓-KRAS	⊓-BRAF	⊓-SMAD4
Model	Feature extractor									
AttMIL	Swin	0.73 ± 0.02	0.61 ± 0.05	0.57 ± 0.03	0.60 ± 0.03	0.75 ± 0.08	0.76 ± 0.02	0.57 ± 0.04	0.65 ± 0.08	0.51 ± 0.02
	CTransPath	**0.81 ± 0.03**	0.78 ± 0.04	0.78 ± 0.02	0.60 ± 0.01	**0.88 ± 0.07**	0.83 ± 0.06	0.61 ± 0.03	0.70 ± 0.02	0.65 ± 0.02
	ViT-B	0.71 ± 0.03	0.69 ± 0.03	0.69 ± 0.01	0.60 ± 0.05	0.75 ± 0.13	0.76 ± 0.04	0.58 ± 0.04	0.63 ± 0.06	**0.69 ± 0.02**
	Phikon	0.70 ± 0.02	0.76 ± 0.01	0.72 ± 0.04	0.59 ± 0.02	0.82 ± 0.10	0.87 ± 0.03	0.62 ± 0.05	0.66 ± 0.03	0.65 ± 0.06
	ViT-S	0.72 ± 0.02	0.70 ± 0.03	0.71 ± 0.04	**0.63 ± 0.02**	0.72 ± 0.07	0.75 ± 0.04	0.62 ± 0.06	0.58 ± 0.07	0.64 ± 0.03
	Lunit-DINO	0.77 ± 0.02	**0.79 ± 0.01**	0.77 ± 0.03	0.62 ± 0.02	0.85 ± 0.07	**0.89 ± 0.03**	0.61 ± 0.04	**0.73 ± 0.07**	0.66 ± 0.03
	ResNet-50	0.64 ± 0.03	0.62 ± 0.04	0.64 ± 0.06	0.63 ± 0.01	0.75 ± 0.07	0.72 ± 0.02	0.55 ± 0.03	0.59 ± 0.07	0.57 ± 0.07
	RetCCL	0.73 ± 0.03	0.77 ± 0.02	0.78 ± 0.05	0.62 ± 0.02	0.82 ± 0.10	0.83 ± 0.03	**0.66 ± 0.04**	0.62 ± 0.02	0.64 ± 0.03
	Lunit-BT	0.70 ± 0.03	0.76 ± 0.01	**0.79 ± 0.03**	0.51 ± 0.02	0.66 ± 0.14	0.57 ± 0.08	0.60 ± 0.06	0.48 ± 0.10	0.56 ± 0.11
	Lunit-SwAV	0.76 ± 0.01	0.78 ± 0.03	0.77 ± 0.01	0.59 ± 0.03	0.83 ± 0.08	0.78 ± 0.04	0.55 ± 0.08	0.69 ± 0.05	0.60 ± 0.05
Transformer	Swin	0.71 ± 0.04	0.63 ± 0.05	0.61 ± 0.05	0.56 ± 0.03	0.72 ± 0.09	0.71 ± 0.04	0.53 ± 0.02	0.55 ± 0.07	0.61 ± 0.07
	CTransPath	**0.80 ± 0.02**	0.76 ± 0.04	**0.80 ± 0.02**	0.59 ± 0.04	0.85 ± 0.08	0.86 ± 0.05	0.60 ± 0.04	0.69 ± 0.08	0.62 ± 0.06
	ViT-B	0.70 ± 0.03	0.69 ± 0.02	0.64 ± 0.03	0.57 ± 0.02	0.75 ± 0.11	0.69 ± 0.08	0.54 ± 0.07	0.55 ± 0.03	0.61 ± 0.03
	Phikon	0.74 ± 0.03	0.73 ± 0.03	0.74 ± 0.03	0.58 ± 0.03	0.84 ± 0.07	0.86 ± 0.02	0.62 ± 0.06	0.69 ± 0.03	**0.67 ± 0.04**
	ViT-S	0.71 ± 0.03	0.70 ± 0.06	0.63 ± 0.02	0.59 ± 0.05	0.75 ± 0.10	0.74 ± 0.02	**0.63 ± 0.08**	0.57 ± 0.03	0.65 ± 0.07
	Lunit-DINO	0.78 ± 0.03	0.77 ± 0.04	0.79 ± 0.01	**0.62 ± 0.04**	**0.87 ± 0.06**	**0.88 ± 0.04**	0.58 ± 0.03	0.68 ± 0.09	0.64 ± 0.07
	ResNet-50	0.66 ± 0.05	0.64 ± 0.11	0.58 ± 0.04	0.61 ± 0.07	0.77 ± 0.09	0.69 ± 0.06	0.54 ± 0.04	0.61 ± 0.04	0.40 ± 0.12
	RetCCL	0.76 ± 0.03	0.76 ± 0.05	0.77 ± 0.04	0.59 ± 0.01	0.83 ± 0.07	0.82 ± 0.05	0.58 ± 0.05	0.62 ± 0.08	0.64 ± 0.05
	Lunit-BT	0.78 ± 0.03	0.77 ± 0.03	0.77 ± 0.03	0.58 ± 0.04	0.86 ± 0.07	0.85 ± 0.03	0.59 ± 0.06	0.62 ± 0.02	0.63 ± 0.07
	Lunit-SwAV	0.75 ± 0.03	**0.80 ± 0.02**	0.78 ± 0.04	0.57 ± 0.04	0.84 ± 0.06	0.82 ± 0.04	0.52 ± 0.04	**0.69 ± 0.13**	0.59 ± 0.01
Mean pool	Swin	0.74 ± 0.01	0.65 ± 0.02	0.61 ± 0.04	0.61 ± 0.01	0.65 ± 0.11	0.78 ± 0.02	0.64 ± 0.04	0.65 ± 0.03	0.64 ± 0.01
	CTransPath	**0.80 ± 0.00**	**0.77 ± 0.01**	**0.70 ± 0.02**	**0.62 ± 0.02**	0.67 ± 0.11	0.87 ± 0.02	0.59 ± 0.06	0.72 ± 0.03	0.64 ± 0.02
	ViT-B	0.77 ± 0.00	0.68 ± 0.01	0.65 ± 0.02	0.60 ± 0.01	0.68 ± 0.11	0.73 ± 0.03	0.58 ± 0.06	0.63 ± 0.06	0.66 ± 0.03
	Phikon	0.69 ± 0.01	0.76 ± 0.01	0.61 ± 0.02	0.55 ± 0.04	0.68 ± 0.16	**0.88 ± 0.05**	0.63 ± 0.03	0.70 ± 0.03	0.62 ± 0.07
	ViT-S	0.74 ± 0.01	0.72 ± 0.04	0.63 ± 0.05	0.61 ± 0.02	0.67 ± 0.13	0.73 ± 0.02	**0.67 ± 0.02**	0.58 ± 0.06	0.61 ± 0.04
	Lunit-DINO	0.76 ± 0.01	0.75 ± 0.02	0.68 ± 0.05	0.59 ± 0.02	**0.73 ± 0.15**	0.85 ± 0.03	0.61 ± 0.04	**0.79 ± 0.03**	0.65 ± 0.03
	ResNet-50	0.73 ± 0.00	0.66 ± 0.05	0.65 ± 0.01	0.60 ± 0.01	0.63 ± 0.11	0.75 ± 0.05	0.66 ± 0.03	0.58 ± 0.04	0.58 ± 0.11
	RetCCL	0.79 ± 0.00	0.75 ± 0.01	0.67 ± 0.02	0.60 ± 0.01	0.71 ± 0.10	0.85 ± 0.01	0.63 ± 0.05	0.66 ± 0.05	0.65 ± 0.01
	Lunit-BT	0.75 ± 0.04	0.75 ± 0.01	0.69 ± 0.05	0.57 ± 0.01	0.67 ± 0.12	0.79 ± 0.03	0.66 ± 0.03	0.61 ± 0.01	**0.68 ± 0.01**
	Lunit-SwAV	0.74 ± 0.00	0.75 ± 0.01	0.68 ± 0.01	0.51 ± 0.01	0.73 ± 0.14	0.78 ± 0.02	0.53 ± 0.01	0.75 ± 0.02	0.60 ± 0.02

Table 11. Test AUROC scores (averaged across the five seeds) for all tasks, feature extractors, and downstream models, when employing **patchwise stain normalisation** [59].

Model	Target Feature extractor	🎗-Subtype	🎗-CDH1	🎗-TP53	🎗-PIK3CA	🎗-LN status	-MSI	-KRAS	-BRAF	-SMAD4
AttMIL	Swin	0.76 ± 0.02	0.66 ± 0.06	0.54 ± 0.03	0.59 ± 0.01	0.77 ± 0.08	0.77 ± 0.03	0.53 ± 0.02	0.68 ± 0.04	0.52 ± 0.03
	CTransPath	**0.81 ± 0.04**	**0.79 ± 0.03**	0.80 ± 0.01	**0.65 ± 0.03**	**0.86 ± 0.06**	0.85 ± 0.03	0.59 ± 0.05	0.71 ± 0.01	0.65 ± 0.02
	ViT-B	0.74 ± 0.04	0.71 ± 0.01	0.67 ± 0.02	0.58 ± 0.03	0.76 ± 0.06	0.76 ± 0.03	0.57 ± 0.09	0.66 ± 0.03	**0.70 ± 0.04**
	Phikon	0.74 ± 0.02	0.74 ± 0.02	0.76 ± 0.08	0.55 ± 0.03	0.84 ± 0.07	0.85 ± 0.03	0.59 ± 0.04	0.70 ± 0.09	0.52 ± 0.10
	ViT-S	0.75 ± 0.01	0.77 ± 0.02	0.69 ± 0.04	0.59 ± 0.03	0.71 ± 0.11	0.70 ± 0.06	0.61 ± 0.04	0.60 ± 0.05	0.64 ± 0.09
	Lunit-DINO	0.76 ± 0.02	0.77 ± 0.03	**0.80 ± 0.01**	0.64 ± 0.01	0.86 ± 0.07	**0.88 ± 0.02**	0.59 ± 0.02	**0.77 ± 0.04**	0.68 ± 0.03
	ResNet-50	0.68 ± 0.02	0.71 ± 0.04	0.69 ± 0.04	0.63 ± 0.02	0.76 ± 0.11	0.66 ± 0.05	0.52 ± 0.06	0.57 ± 0.06	0.43 ± 0.08
	RetCCL	0.77 ± 0.02	0.77 ± 0.03	0.80 ± 0.02	0.61 ± 0.02	0.84 ± 0.08	0.82 ± 0.03	**0.62 ± 0.05**	0.63 ± 0.02	0.65 ± 0.01
	Lunit-BT	0.73 ± 0.02	0.78 ± 0.02	0.78 ± 0.05	0.53 ± 0.02	0.62 ± 0.22	0.64 ± 0.14	0.57 ± 0.10	0.44 ± 0.12	0.50 ± 0.05
	Lunit-SwAV	0.75 ± 0.01	0.75 ± 0.03	0.75 ± 0.04	0.56 ± 0.05	0.84 ± 0.07	0.82 ± 0.02	0.57 ± 0.06	0.73 ± 0.05	0.60 ± 0.04
Transformer	Swin	0.74 ± 0.04	0.70 ± 0.03	0.58 ± 0.03	0.60 ± 0.02	0.76 ± 0.09	0.79 ± 0.04	0.61 ± 0.06	0.56 ± 0.09	0.59 ± 0.07
	CTransPath	**0.79 ± 0.02**	**0.78 ± 0.02**	0.78 ± 0.05	0.57 ± 0.02	0.87 ± 0.06	0.82 ± 0.06	0.59 ± 0.06	0.62 ± 0.09	0.66 ± 0.01
	ViT-B	0.74 ± 0.04	0.72 ± 0.03	0.67 ± 0.04	0.57 ± 0.04	0.74 ± 0.06	0.70 ± 0.04	0.54 ± 0.01	0.61 ± 0.07	0.67 ± 0.05
	Phikon	0.75 ± 0.05	0.71 ± 0.02	0.74 ± 0.02	0.57 ± 0.02	0.86 ± 0.04	0.84 ± 0.04	0.61 ± 0.02	0.70 ± 0.05	0.57 ± 0.04
	ViT-S	0.75 ± 0.03	0.76 ± 0.01	0.65 ± 0.04	0.55 ± 0.06	0.74 ± 0.08	0.71 ± 0.01	0.55 ± 0.04	0.59 ± 0.05	**0.68 ± 0.04**
	Lunit-DINO	0.78 ± 0.03	0.72 ± 0.03	0.79 ± 0.02	**0.63 ± 0.03**	**0.87 ± 0.04**	**0.89 ± 0.02**	0.59 ± 0.03	**0.73 ± 0.03**	0.66 ± 0.07
	ResNet-50	0.72 ± 0.01	0.71 ± 0.05	0.64 ± 0.04	0.61 ± 0.07	0.74 ± 0.07	0.65 ± 0.05	0.57 ± 0.03	0.58 ± 0.07	0.39 ± 0.05
	RetCCL	0.74 ± 0.06	0.74 ± 0.04	**0.80 ± 0.04**	0.55 ± 0.04	0.86 ± 0.07	0.71 ± 0.06	0.54 ± 0.08	0.59 ± 0.06	0.61 ± 0.09
	Lunit-BT	0.79 ± 0.02	0.75 ± 0.04	0.77 ± 0.04	0.58 ± 0.02	0.84 ± 0.06	0.86 ± 0.04	**0.63 ± 0.04**	0.63 ± 0.03	0.67 ± 0.01
	Lunit-SwAV	0.74 ± 0.05	0.76 ± 0.05	0.77 ± 0.01	0.53 ± 0.04	0.84 ± 0.05	0.82 ± 0.05	0.56 ± 0.03	0.70 ± 0.08	0.54 ± 0.06
Mean pool	Swin	0.75 ± 0.01	0.69 ± 0.03	0.60 ± 0.04	0.59 ± 0.02	0.69 ± 0.12	0.74 ± 0.02	0.63 ± 0.06	0.65 ± 0.01	0.57 ± 0.03
	CTransPath	**0.82 ± 0.00**	0.75 ± 0.02	0.73 ± 0.02	**0.64 ± 0.03**	0.69 ± 0.12	0.85 ± 0.02	0.59 ± 0.03	0.75 ± 0.02	0.64 ± 0.03
	ViT-B	0.76 ± 0.01	0.71 ± 0.01	0.67 ± 0.01	0.56 ± 0.01	0.68 ± 0.09	0.75 ± 0.03	0.59 ± 0.06	0.63 ± 0.03	**0.69 ± 0.01**
	Phikon	0.74 ± 0.01	0.74 ± 0.02	0.64 ± 0.02	0.61 ± 0.03	0.73 ± 0.13	**0.87 ± 0.01**	0.56 ± 0.04	0.71 ± 0.09	0.61 ± 0.02
	ViT-S	0.74 ± 0.01	0.77 ± 0.03	0.62 ± 0.02	0.56 ± 0.01	0.70 ± 0.08	0.73 ± 0.01	**0.66 ± 0.03**	0.57 ± 0.05	0.63 ± 0.03
	Lunit-DINO	0.73 ± 0.01	0.75 ± 0.02	**0.77 ± 0.02**	0.60 ± 0.02	**0.76 ± 0.11**	0.87 ± 0.02	0.58 ± 0.04	**0.78 ± 0.02**	0.69 ± 0.02
	ResNet-50	0.74 ± 0.01	**0.79 ± 0.02**	0.65 ± 0.01	0.61 ± 0.03	0.66 ± 0.10	0.67 ± 0.05	0.64 ± 0.03	0.55 ± 0.04	0.58 ± 0.04
	RetCCL	0.81 ± 0.00	0.75 ± 0.00	0.69 ± 0.02	0.58 ± 0.02	0.70 ± 0.13	0.77 ± 0.04	0.61 ± 0.05	0.65 ± 0.01	0.65 ± 0.00
	Lunit-BT	0.76 ± 0.02	0.75 ± 0.00	0.71 ± 0.05	0.57 ± 0.01	0.63 ± 0.08	0.80 ± 0.05	0.66 ± 0.01	0.62 ± 0.00	0.68 ± 0.00
	Lunit-SwAV	0.75 ± 0.00	0.75 ± 0.01	0.69 ± 0.04	0.53 ± 0.01	0.71 ± 0.15	0.83 ± 0.02	0.55 ± 0.03	0.76 ± 0.02	0.59 ± 0.05

Table 12. Test AUROC scores (averaged across the five seeds) for all tasks, feature extractors, and downstream models, when employing **rotation/flipping augmentations**.

Model	Target Feature extractor	🎗-Subtype	🎗-CDH1	🎗-TP53	🎗-PIK3CA	🎗-LN status	-MSI	-KRAS	-BRAF	-SMAD4
AttMIL	Swin	0.77 ± 0.01	0.66 ± 0.02	0.61 ± 0.02	0.59 ± 0.03	0.79 ± 0.09	0.74 ± 0.04	0.56 ± 0.06	0.63 ± 0.06	0.54 ± 0.04
	CTransPath	**0.81 ± 0.03**	**0.79 ± 0.02**	**0.82 ± 0.01**	0.62 ± 0.02	**0.89 ± 0.05**	0.79 ± 0.03	0.60 ± 0.03	0.70 ± 0.05	0.65 ± 0.02
	ViT-B	0.77 ± 0.01	0.70 ± 0.05	0.70 ± 0.03	0.58 ± 0.03	0.78 ± 0.06	0.76 ± 0.04	0.60 ± 0.02	0.63 ± 0.02	**0.70 ± 0.04**
	Phikon	0.68 ± 0.05	0.71 ± 0.03	0.72 ± 0.06	0.54 ± 0.04	0.84 ± 0.07	0.84 ± 0.03	0.56 ± 0.08	0.61 ± 0.06	0.59 ± 0.07
	ViT-S	0.73 ± 0.02	0.73 ± 0.02	0.68 ± 0.02	0.58 ± 0.04	0.74 ± 0.10	0.75 ± 0.02	0.61 ± 0.03	0.60 ± 0.03	0.65 ± 0.06
	Lunit-DINO	0.76 ± 0.03	0.77 ± 0.03	0.78 ± 0.03	0.62 ± 0.03	0.86 ± 0.06	**0.89 ± 0.03**	0.59 ± 0.03	**0.78 ± 0.07**	0.67 ± 0.06
	ResNet-50	0.72 ± 0.01	0.74 ± 0.03	0.68 ± 0.03	**0.65 ± 0.04**	0.76 ± 0.09	0.65 ± 0.04	0.52 ± 0.02	0.55 ± 0.06	0.41 ± 0.13
	RetCCL	0.75 ± 0.03	0.77 ± 0.04	0.79 ± 0.03	0.59 ± 0.01	0.85 ± 0.05	0.79 ± 0.07	**0.62 ± 0.06**	0.61 ± 0.02	0.65 ± 0.01
	Lunit-BT	0.64 ± 0.05	0.69 ± 0.07	0.62 ± 0.22	0.49 ± 0.01	0.51 ± 0.07	0.68 ± 0.11	0.54 ± 0.05	0.55 ± 0.08	0.52 ± 0.06
	Lunit-SwAV	0.74 ± 0.01	0.77 ± 0.01	0.75 ± 0.04	0.56 ± 0.04	0.84 ± 0.06	0.82 ± 0.02	0.58 ± 0.05	0.66 ± 0.05	0.61 ± 0.05
Transformer	Swin	0.73 ± 0.01	0.67 ± 0.06	0.60 ± 0.03	0.62 ± 0.03	0.80 ± 0.10	0.76 ± 0.03	0.60 ± 0.08	0.69 ± 0.03	0.60 ± 0.03
	CTransPath	**0.79 ± 0.04**	0.74 ± 0.01	0.78 ± 0.03	0.61 ± 0.03	**0.89 ± 0.04**	0.83 ± 0.04	0.58 ± 0.03	0.69 ± 0.08	0.58 ± 0.07
	ViT-B	0.76 ± 0.02	0.70 ± 0.03	0.66 ± 0.03	0.56 ± 0.02	0.75 ± 0.14	0.74 ± 0.01	0.50 ± 0.06	0.53 ± 0.08	0.68 ± 0.04
	Phikon	0.69 ± 0.03	0.71 ± 0.03	0.75 ± 0.03	0.54 ± 0.03	0.85 ± 0.06	0.84 ± 0.05	0.63 ± 0.04	**0.75 ± 0.04**	0.56 ± 0.05
	ViT-S	0.75 ± 0.02	0.74 ± 0.02	0.66 ± 0.05	0.56 ± 0.03	0.74 ± 0.05	0.72 ± 0.05	0.60 ± 0.05	0.56 ± 0.03	**0.69 ± 0.01**
	Lunit-DINO	0.77 ± 0.02	0.75 ± 0.02	0.79 ± 0.01	0.61 ± 0.03	0.87 ± 0.07	**0.88 ± 0.02**	0.58 ± 0.05	0.71 ± 0.04	0.69 ± 0.04
	ResNet-50	0.71 ± 0.03	0.69 ± 0.04	0.71 ± 0.03	**0.64 ± 0.02**	0.75 ± 0.08	0.71 ± 0.02	0.63 ± 0.03	0.59 ± 0.05	0.44 ± 0.07
	RetCCL	0.78 ± 0.02	0.75 ± 0.06	**0.79 ± 0.01**	0.56 ± 0.04	0.85 ± 0.08	0.77 ± 0.08	0.54 ± 0.05	0.53 ± 0.09	0.58 ± 0.08
	Lunit-BT	0.77 ± 0.03	0.78 ± 0.04	0.76 ± 0.03	0.55 ± 0.03	0.84 ± 0.09	0.82 ± 0.06	**0.64 ± 0.03**	0.63 ± 0.03	0.65 ± 0.02
	Lunit-SwAV	0.72 ± 0.02	**0.78 ± 0.02**	0.77 ± 0.03	0.55 ± 0.03	0.86 ± 0.07	0.79 ± 0.05	0.57 ± 0.05	0.70 ± 0.03	0.55 ± 0.07
Mean pool	Swin	0.77 ± 0.01	0.68 ± 0.04	0.62 ± 0.03	0.60 ± 0.02	0.66 ± 0.12	0.75 ± 0.02	0.65 ± 0.04	0.61 ± 0.05	0.58 ± 0.04
	CTransPath	**0.83 ± 0.00**	0.75 ± 0.02	0.73 ± 0.01	**0.64 ± 0.01**	0.70 ± 0.12	0.86 ± 0.03	0.61 ± 0.03	0.75 ± 0.02	0.61 ± 0.02
	ViT-B	0.76 ± 0.01	0.70 ± 0.01	0.68 ± 0.01	0.56 ± 0.01	0.68 ± 0.08	0.72 ± 0.02	0.58 ± 0.05	0.59 ± 0.01	0.69 ± 0.01
	Phikon	0.71 ± 0.01	0.76 ± 0.03	0.65 ± 0.03	0.56 ± 0.04	0.73 ± 0.12	0.88 ± 0.02	0.57 ± 0.05	0.70 ± 0.07	0.59 ± 0.02
	ViT-S	0.72 ± 0.02	0.75 ± 0.02	0.62 ± 0.01	0.57 ± 0.00	0.69 ± 0.11	0.69 ± 0.03	0.65 ± 0.04	0.56 ± 0.03	0.65 ± 0.02
	Lunit-DINO	0.74 ± 0.01	0.76 ± 0.02	**0.77 ± 0.03**	0.59 ± 0.03	**0.77 ± 0.12**	**0.88 ± 0.03**	0.59 ± 0.02	**0.79 ± 0.01**	**0.70 ± 0.03**
	ResNet-50	0.75 ± 0.01	**0.77 ± 0.01**	0.67 ± 0.02	0.61 ± 0.01	0.62 ± 0.10	0.67 ± 0.03	**0.68 ± 0.01**	0.52 ± 0.05	0.55 ± 0.06
	RetCCL	0.82 ± 0.00	0.75 ± 0.01	0.71 ± 0.01	0.57 ± 0.01	0.71 ± 0.12	0.79 ± 0.05	0.61 ± 0.07	0.63 ± 0.03	0.65 ± 0.00
	Lunit-BT	0.74 ± 0.04	0.74 ± 0.00	0.68 ± 0.06	0.55 ± 0.02	0.57 ± 0.09	0.77 ± 0.07	0.66 ± 0.01	0.60 ± 0.01	0.61 ± 0.16
	Lunit-SwAV	0.76 ± 0.00	0.76 ± 0.02	0.75 ± 0.01	0.54 ± 0.02	0.70 ± 0.15	0.85 ± 0.01	0.55 ± 0.04	0.68 ± 0.05	0.58 ± 0.05

Table 13. Test AUROC scores (averaged across the five seeds) for all tasks, feature extractors, and downstream models, when employing **all augmentations**.