Supplementary Material for

ScoreNet: Learning Non-Uniform Attention and Augmentation for Transformer-Based Histopathological Image Classification

Thomas Stegmüller[1] Behzad Bozorgtabar[1,2,3] Antoine Spahr[1] Jean-Philippe Thiran[1,2,3]
[1]EPFL, Switzerland [2]CHUV, Switzerland [3]CIBM, Switzerland
{firstname.lastname}@epfl.ch

A. Overview

In this supplementary draft, we provide additional ablation studies and experimental details. The remaining of this supplementary draft is organized as follows. The mathematical details of the patch selection operation are provided in Sec. B. In Sec. C we detail the architectural and training details, e.g., parameters choices. Additional ablations are detailed in Sec. D . A detailed derivation of the computational cost is presented in Sec. E. We discuss, in Sec. F, some properties of ScoreMix and present some examples of our proposed ScoreMix augmentation. Finally, the suitability of ScoreNet to learn from uncurated data is evaluated in Sec. G.

B. Differentiable Patch Selection

We now get down to the nuts and bolts of the differentiable patch selection module. For that purpose, let's assume that an input image, $x \in \mathbb{R}^{C\times H\times W}$, is tiled in a regular grid of N patches (of dimension $C \times P \times P$), conveniently stored in a tensor $\mathbf{P} \in \mathbb{R}^{N\times C\times P\times P}$. Provided an index matrix $\mathbf{Y} \in \{0,1\}^{N\times K}$ encoding the one-hot indices of K patches, the extraction operation, can be written as:

$$\mathbf{X} = \mathbf{Y}^T\mathbf{P} \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{K\times C\times P\times P}$ stores the selected patches. In a machine learning context it is desirable that the patch selection process is data driven, and hence based on a learnable criterion, well reflected by a score, $\mathbb{P}_{\text{patch}} \in \mathbb{R}^N$, which captures the relevance of each patch for the task at hand. We therefore seek a differentiable module, $\mathcal{T}$, which returns the indices, $\mathbf{Y}^*$, of the K patches that maximize this criterion when fed with $\mathbb{P}_{\text{patch}}$. The standard *Top-K* operator is not suitable for this purpose, as it is non-differentiable. Nonetheless, it has been demonstrated that it could be equivalently formulated as solving the linear program [7]:

$$\mathbf{Y}^* = \underset{\mathbf{Y}\in\mathcal{C}_1}{\arg\max} = \left\langle \mathbf{Y}, \tilde{\mathbb{P}} \right\rangle \tag{2}$$

where $\tilde{\mathbb{P}} \in \mathbb{R}^{N\times K}$ is obtained by broadcasting $\mathbb{P}_{\text{patch}}$ to match the dimension of $\mathbf{Y}$, and $\mathcal{C}_1$ is the convex hull:

$$\{\mathbf{Y} \in \mathbb{R}_+^{N\times K} : \sum_k \mathbf{Y}_{n,k} \leq 1 \,\forall n, \sum_n \mathbf{Y}_{n,k} = 1 \,\forall k\} \tag{3}$$

which falls under the set of problems to which the *perturbed optimizers* scheme can be applied to obtain a noisy, but differentiable solution [3]. In fact, for the *perturbed optimizers* framework to be applicable, the solution must be unique, which is not the case here, as any permutation of the columns of $\mathbf{Y}^*$ is still a valid solution. To that end, [7] proposes to use the *sorted Top-K* operator, whose solution is unique and can also be formulated as a linear program:

$$\mathbf{Y}^* = \underset{\mathbf{Y}\in\mathcal{C}}{\arg\max} = \left\langle \mathbf{Y}, \tilde{\mathbb{P}} \right\rangle \tag{4}$$

where $\mathcal{C} = \mathcal{C}_1 \cap \mathcal{C}_2$ and $\mathcal{C}_2$ is:

$$\{\mathbf{Y} \in \mathbb{R}_+^{N\times K} : \sum_i i\mathbf{Y}_{i,k} < \sum_j j\mathbf{Y}_{j,k'} \forall k < k'\} \tag{5}$$

As can be seen in Eq. 5, $\mathcal{C}_2$ ensures that the indices, encoded as one-hot columns of $\mathbf{Y}$, are sorted, namely the smallest index is stored in the first column and the largest in the last one.

Under that formalism, the perturbed *sorted Top-K* operator is defined as:

$$\mathbf{Y}_\sigma = \mathbb{E}_{\mathbf{N}}\left[\underset{\mathbf{Y}\in\mathcal{C}}{\arg\max} \left\langle \mathbf{Y}, \tilde{\mathbb{P}} + \sigma\mathbf{N} \right\rangle\right] \tag{6}$$

and $\sigma\mathbf{N} \in \mathbb{R}^{N\times K}$ is a centered Gaussian noise of variance σ^2. In principle, the *perturbed optimizers* scheme is not limited to Gaussian noises, but it has the nice property that the Jacobian of the perturbed maximizer $\mathbf{Y}_\sigma$ w.r.t. the learnable scores, $\tilde{\mathbb{P}}$, is well defined and can be efficiently computed:

$$J_{\tilde{\mathbb{P}}}\mathbf{Y}_\sigma = \mathbb{E}_{\mathbf{N}}\left[\underset{\mathbf{Y}\in\mathcal{C}}{\arg\max} \left\langle \mathbf{Y}, \tilde{\mathbb{P}} + \sigma\mathbf{N} \right\rangle \mathbf{N}^T/\sigma\right] \tag{7}$$

we refer the reader to [1] (Lemma 1.5) for a detailed derivation of the above Jacobian.

C. Experimental Setup & Datasets

C.1. Networks Architectures

ScoreNet. The proposed ScoreNet architecture comprises two stages: the recommendation and aggregation stages. The former leverages a modified ViT-Tiny to produces the semantic distribution. Similarly, the latter relies on an identical ViT-Tiny to independently embed the selected high-resolution patches (*local fine-grained attention*) and on a transformer encoder to mix the embedded patches (*global coarse-grained attention*). The following parameters of the two identical ViT-Tiny were modified to be tailored for the task:

- `embed_dim=96`.
- `depth=8`.
- `num_heads=4`.
- `mlp_ratio=2`.

These modifications were brought to allow for a self-supervised pre-training with a sufficiently large batch size ($bs \geq 128$), which was reported to be of significant importance to reach good performance [5]. The parameters of the transformer encoder implementing the *global coarse-grained attention* mechanism are:

- `embed_dim=96`.
- `depth=4`.
- `num_heads=4`.
- `mlp_ratio=2`.

Overall ScoreNet's model totals approximately 1.79M parameters.

SwinTransformer. SwinTransformers [12] relies on hierarchical architecture attention mechanism, namely intra- and inter-window attentions. The patch-merging operation reduces the time, and memory cost of SwinTransformers [12] significantly, which decreases the total number of tokens by 4, while increasing the embedding by 2. The architecture is modified to accept non-square windows, allowing SwinTransformers to process non-square images images. The resulting parameters are:

- `patch_size=16`.
- `input_embed_dim_size=24`.
- `output_embed_dim_size=192`.
- `depths=[2, 2, 6, 2]`.
- `num_heads=[3, 6, 12, 24]`.
- `window_size=(6, 8)`.
- `mlp_ratio=4`.

Overall the SwinTransformer model totals approximately 1.77M parameters.

TransPath. As described in [18], TransPath's architecture leverages a CNN encoder to jointly reduce the input image's size, extract relevant features, and tile the image in pre-embedded patches. Subsequently, a transformer encoder processes the CNN encoder's features to capture global interactions. The CNN encoder's architecture is as follows:

- `n_convolutions=4`.
- `n_filters=[8, 32, 128, 512]`.
- `kernel_sizes=[(3, 3), (3, 3), (3, 3), (3, 3)]`.
- `pooling_kernel_sizes=[(4, 4), (2, 2), (4, 4), (4, 4)]`.
- `activation=ReLU` [10].

A projection convolution is used to match the desired embedding dimension of the transformer encoder. Its parameters are:

- `n_filters=192`.
- `kernel_sizes=(1, 1)`.

The parameters of the transformer encoder are:

- `embed_dim=192`.
- `depth=4`.
- `num_heads=4`.
- `mlp_ratio=2`.

Each transformer block rely on TransPath's customized token-aggregating and excitation multi-head self-attention (MHSA-TAE) [18]. Overall, TransPath's model totals approximately 1.93M parameters.

TransMIL. We adopt the original implementation as provided by the authors [16]. It relies on a ResNet-50 [11] pre-trained on ImageNet [8] to embed the individual 256×256 patches. Overall, TransMIL's model totals approximately 3.19M parameters (not counting the parameters of the ResNet-50).

CLAM. The implementation of CLAM follows the code provided by the authors [13]. It relies on a ResNet-50 [11] pre-trained on ImageNet [8] to embed the individual 256×256 patches. Overall, the variations of CLAM-(SB/MB)/(S/B) total from 1.32M to 1.46M parameters (not counting the parameters of the ResNet-50).

C.2. Self-Supervised Pre-training

Modular Pre-training. Our modular architecture allows for independent self-supervised pre-trainings of the recommendation stage's ViT and that of the local fine-grained attention mechanism. A two steps pre-training can be beneficial, as it provides the possibility to validate each part

independently. Similarly, one of the modules, typically the one implementing the fine-grained local attention, can be trained on an auxiliary annotated dataset or be replaced by a standard pre-trained model.

The self-supervised pre-training follows the guidelines of [5]. Apart from the differences in architectures described in Sec. C.1, minor modifications were made in the projection head to account for the reduced heterogeneity in our datasets compared to that in ImageNet [9]. The modifications are:

- `hidden_dim=1024`.
- `bottleneck_dim=128`.
- `out_dim=1024`.

These modifications are in line with the interpretation of [4] which considers the last linear layer as a projection on a set of learnable centroids and that their number should reflect the level of diversity present in the dataset. For this interpretation to hold, it is required that both the last layer's input and its weights are normalized, which is the case in our setup. The remaining parameters, aside from the position encoding which is discussed in Sec. D, are set to the default values (see [5] for details).

End-to-end Pre-training. In some cases, an end-to-end pre-training of ScoreNet is preferable. For that purpose, we experimented with two approaches: DINO and a variant of it for that purpose. The former uses the default values for all parameters but those of the projection head described above. On the contrary, the latter benefits from different augmentations and another pretext task and thereby avoid a potential pitfall of DINO: encouraging contextual bias [17], which occurs when the similarity between the representations of views depicting distinct tissue types is enforced.

In this regard, the set of admissible augmentations are constrained to those that change the pixels' values, but not their locations. Consequently, a given image's different views are bounded to bear identical semantic content. A key aspect of DINO's strong performance is due to enforcing the local-to-global correspondence between the student's local crops and that of the teacher's global crop. To mimic that knowledge distillation mechanism, we encourage the student, which only processes the most discriminative high-resolution patches, to match the teacher's representation, which on the contrary, is based on all high-resolution patches. One can observe that this pretext task enforces local-to-global correspondence while providing a strong supervisory signal to the student's scorer, which has to highlight the most relevant regions for the task to be successfully accomplished.

In that setting, ScoreNet's representation is obtained by the concatenation of the `[CLS]` tokens of the *global coarse-grained attention* module's last two transformer blocks. This representation benefits from global contextual information through the teacher, which processes the whole high-resolution image. The projection head's parameters are identified as described above.

C.3. Datasets

In addition to the annotated TRoIs from two datasets, namely BRACS [14] and BACH [2], additional sets of unlabeled of images are used to pre-train the models and for various ablations. The sets of unlabeled images are detailed here.

BRACS. The BRACS dataset encompasses both the annotated TRoIs and the 547 whole-slide images from which they were extracted. We use the WSIs to create an unlabeled pre-training set. More precisely, two types of auxiliary datasets are extracted from BRACS's WSIs: tiles set at $40\times$ and low-resolution thumbnails set at $\frac{40}{s}\times$, where s is the down-scaling ratio. The former set is used to pre-train the *local fine-grained attention* module, whereas the latter serves to pre-train the recommendation stag's scorer. We experimented with two variants of these paired sets. The first variant is designed for a version of ScoreNet, where the dimension of the finely attended regions is $P_h = 224$, the recommendation stage processes low-resolution patches of dimension $P_l = 16$ and consequently a down-scaling ratio $s = 14$. The second variant is designed for a version of ScoreNet, where the dimension of the finely attended regions is $P_h = 128$, the recommendation stage processes low-resolution patches of dimension $P_l = 16$ and consequently a down-scaling ratio $s = 8$. The resulting sets contain approximately 150k images (for a fair comparison of the two versions, see Sec. D).

The last images are extracted from BRACS to conduct TransPath's self-supervised pre-training. From the WSIs, an unlabeled set of approximately 100k images at $40\times$ are extracted. The images have dimensions 1536×1536, which is approximately the median dimensions of the annotated TRoIs.

BACH. Similarly, the BACH dataset comprises an annotated set of TRoIs and the accompanying 40 whole-slide images. From the WSIs, an unlabeled pre-training set of approximately 11k images at $20\times$ are extracted. The images have the exact dimensions as the annotated TRoIs, 1536×2048.

CAMELYON16. Finally, additional tiles set is extracted from CAMELYON16, which is, to our knowledge, the only one with patch-level annotations. This set is used to evaluate the pre-training of the fine-grained attention module. The latter is composed of 10k images at $40\times$, of dimensions 128×128 or 224×224. It is class-balanced, and any patch which contains tumorous tissue is considered tumour positive. This set is also used to measure the effectiveness

of the position encoding on the fine-grained attention module in Sec. D.

D. Additional Ablations

Down-Scaling Ratio & Dimensions of the Attended Regions. A key component of the proposed pipeline is to determine the down-scaling ratio, s, and the dimension of the square patches in low-resolution, $P_l \times P_l$, and in high-resolution, $P_h \times P_h$. Considering the well-studied nature of the ViTs scorers, we use the standard patch dimension $P_l = 16$ for the patches in low-resolution. It has been shown that smaller patches ($P_l = 8$ or $P_l = 5$) improve the quality of the learned representations [5], nonetheless the incurred increase in computational and memory cost is unsuitable for our application. For the high-resolution patches, we experiment with two standard patch dimensions: $P_h = 128$ and $P_h = 224$. As the self-attention of the recommendation stage is used as a learnable distribution of the semantic content, there should exist a 1-to-1 mapping between the low-resolution patches and the high-resolution regions that can be extracted. As a consequence, the down-scaling ratio is fully determined by the dimensions of the patches: $s = P_h/P_l$. In our case, it translates to down-scaling ratio of either $s = 8$, or $s = 14$.

To find out which of these two setups is the most suitable for our application, we compare the models obtained by each of them via a weighted k Nearest Neighbours classifier, which has the advantage of being fast and not requiring any finetuning. In Table 2, we compare the classification results on the low-resolution ($\frac{40}{s}\times$) BRACS dataset. We report the results of both the teacher and the student models as well as those obtained by a CNN with comparable capacity and identical pre-training. We do not observe significant differences between the two scales. On the other hand, these differences are much more emphasized when evaluating the same models on the low-resolution ($\frac{20}{s}\times$) BACH dataset (see Table 1). These promising results on the BACH dataset, despite the mismatched scales, are to be credited to the local to global views pre-training method [5].

The quality of the fine-grained attention module is assessed with the aforementioned method on the tile CAMELYON16 dataset introduced in Sec. C.3, and the hereby obtained results are reported in Table 3. In conclusion, we observe that the difference is either marginal (Table 2 & 3) or significantly in favor of the setup where $s = 8$ (Table 1) and therefore we choose this setup for the remaining experiments and architectures. As a side note, the CNN architecture performs substantially worth, but it is most likely due to the fact that the DINO [5] method is biased towards ViT architectures. **Positional Encoding.** Without position encoding (PE), a ViT processes tokens as a set and hence completely discards the global shape information; therefore, position encoding is essential. The typical approach is to learn a single matrix of absolute and additive position encoding jointly during the training phase. This approach suffers from two drawbacks: *i)* the absolute encoding of each token's position implies that a pattern is different at every location it occurs, which reduces the sample efficiency [15], and *ii)* as a consequence of the storage of the position encoding in a single matrix, the model treats the input tokens as a 1D sequence and thus mislays the multi-dimensionality of the inputs. The latter is not an issue as long as the input images have the same aspect ratio, as is the case with the local/global crops strategy of DINO [5]. Nonetheless, and as depicted in Fig. 1, this approach fails when the model is fed an image of a different aspect ratio than those used to train the position encoding. As illustrated in Fig. 1, the 2D sine-cos position encoding does not introduce any artifacts when used with images of different resolutions. On the other hand, any absolute position encoding is not a translation equivariant operation, an undesired property for planar images. For these reasons, we experiment with Conditional Position encoding Vision Transformer (CPVT) [6]. This PE is input-dependent and convolution-based; consequently, it is suitable for any input resolution and patch-wise translation-equivariant. Fig. 2 reveals that the PE of border tokens is slightly different due to the needed zero-padding. This finding suggests that the absolute position encoding can be inferred from zero-paddings [6]. We argue that CPVT is well suited to be used conjointly with ScoreMix as the local processing of the token is convenient for detecting local discontinuity caused by the pasting operation, which is needed to incorporate the added content to the global representation (see Sec. F). In Table 4 and Table 5, we evaluate the discriminability of the features obtained by a pre-training under the DINO framework and with various position encoding methods. Table 5, which reports results on the tile CAMELYON16 dataset (see Sec. C.3), does not provide substantial shreds of evidence in favor of one PE or the other; we postulate that this lack of significant differences is due to the lessened importance of position encoding for the tile dataset. Indeed, at $40\times$ and with tiles of dimension 128×128, the available features are mostly texture-based, and the relative organization of the patches is less relevant. This claim is well supported by the substantial differences in performance obtained by distinct PE when evaluated on the low-resolution BACH and BRACS datasets (see Table 5). These differences are further exacerbated by the fact that images on which performance is evaluated are either of varied size (BRACS) or at least of a different dimension than those used during the pre-training (BACH). Notably, there seems to be a significant performance discrepancy between the models using a [CLS] token (CPVT) and those based on a global average pooling (CPVT-GAP). Based on these results, we select the CPVT-GAP approach for the remaining experiments. Note that we

Table 1: A weighted k Nearest Neighbors classifier assesses the learned features' discriminability (weighted F1-score) on the low-resolution BACH dataset. The performances of CNN and ViT-based architectures are compared, and similarly for two down-scaling ratios ($s = 8$ or $s = 14$). We use a 4-fold scheme with $75\%/25\%$ train/test splits.

	ViT				CNN			
	Teacher		Student		Teacher		Student	
k	$s = 14$	$s = 8$	$s = 14$	$s = 8$	$s = 14$	$s = 8$	$s = 14$	$s = 8$
1	71.7 ± 6.4	$\mathbf{78.5 \pm 6.4}$	73.6 ± 5.1	77.4 ± 5.1	63.6 ± 5.1	64.4 ± 1.9	63.8 ± 3.2	63.9 ± 2.2
5	71.5 ± 1.7	$\mathbf{81.7 \pm 3.2}$	72.8 ± 1.9	81.0 ± 4.0	65.1 ± 3.3	64.7 ± 2.1	64.1 ± 4.6	65.4 ± 2.7
10	71.9 ± 2.4	77.8 ± 2.8	72.5 ± 2.5	$\mathbf{77.9 \pm 3.4}$	62.0 ± 3.8	58.9 ± 2.9	61.5 ± 5.8	61.1 ± 2.5
20	71.3 ± 4.0	76.3 ± 3.0	72.5 ± 3.0	$\mathbf{76.5 \pm 4.0}$	64.0 ± 6.7	55.5 ± 1.8	61.0 ± 9.2	55.4 ± 2.4
50	71.2 ± 4.0	$\mathbf{74.7 \pm 4.7}$	70.9 ± 3.3	74.3 ± 5.7	59.3 ± 5.3	56.1 ± 3.2	58.1 ± 6.2	54.6 ± 3.9
100	71.7 ± 4.1	$\mathbf{74.0 \pm 5.5}$	71.4 ± 3.8	73.6 ± 5.9	57.4 ± 3.4	50.6 ± 5.1	56.2 ± 3.0	48.7 ± 4.7

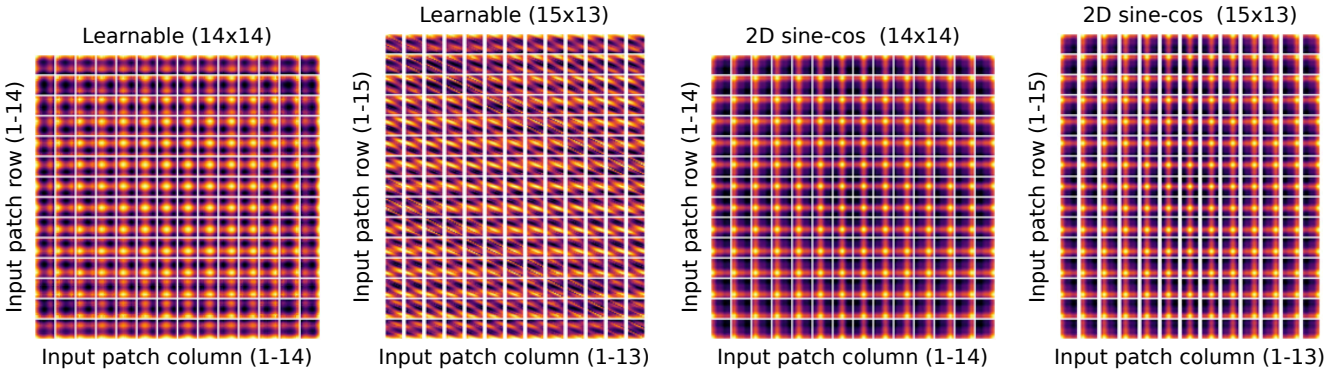

Figure 1: The cosine similarity of a learnable and 2D sine-cos positional encoding is compared. The learnable positional encoding introduces undesirable artifacts when the aspect ratio changes (*Learnable (15×13)*).

Table 2: A weighted k Nearest Neighbors classifier assesses the learned features' discriminability (weighted F1-score) on the low-resolution BRACS dataset. The performances of CNN and ViT-based architectures are compared, and similarly for two down-scaling ratios ($s = 8$ or $s = 14$). The k-NN classifier is trained on the merged train/valid set and evaluated on the test set (see [14]), hence the high performances.

	ViT				CNN			
	Teacher		Student		Teacher		Student	
k	$s = 14$	$s = 8$	$s = 14$	$s = 8$	$s = 14$	$s = 8$	$s = 14$	$s = 8$
1	52.5	54.3	51.6	**55.0**	45.2	45.5	45.4	44.7
5	55.2	**56.1**	55.4	55.8	47.1	47.6	46.6	46.2
10	57.2	56.4	**57.5**	56.7	49.3	46.5	50.5	45.8
20	56.9	58.0	**58.1**	57.6	47.1	47.6	45.9	47.0
50	56.2	**57.5**	55.7	56.9	41.2	44.9	40.6	44.9
100	53.9	54.0	**54.3**	53.7	40.3	43.5	40.1	44.2

referred to [CLS] token throughout this text when referring to a GAP token. Additionally, we have slightly modified the method to be able to extract one self-attention map per transformer head: instead of performing the GAP operation after the very last layer of the transformer encoder, we do it after the $(L-1)^{th}$ layer and concatenate the resulting token to the sequence, thereby producing a pseudo [CLS] token. Similarly, when m pseudo [CLS] tokens are used, this operation is performed after the $(L-m)^{th}$ layer.

Selecting the Number of Finely Attended Regions. The effect of the number of selected regions is depicted in Table 6. One can observe that it does not appear as the most determining factor, particularly that the results are not monotonically increasing, which is unexpected. There are two potential explanations for this behavior. The first is due to the heterogeneity of the BRACS dataset. More precisely, it encompasses images containing less than 50 patches, which implies that the image must first be resized, potentially harming the predictions. The second explanation is that the model used for this ablation is a $\mathrm{ScoreNet}/4/1$ variant, which by design relies less on the high-resolution images than its $\mathrm{ScoreNet}/4/3$ counterpart. The respective properties of these two variants are subject to Sec. D.1.

D.1. ScoreNet Under the Magnifying Glass

Just a Glorified Low-resolution ViT? We explore the usage of high-resolution images for predictions. For that

Table 3: A weighted k Nearest Neighbors classifier assesses the discriminability (weighted F1-score) of the learned features on the tile CAMELYON16 dataset (see Sec. C.3). The performances of CNN and ViT-based architectures is compared, and similarly for two tile dimensions (128×128 and 224×224) corresponding to down-scaling ratios of $s = 8$ and $s = 14$, respectively. A 4-fold approach with $75\%/25\%$ train/test splits is used.

	ViT				CNN			
	Teacher		Student		Teacher		Student	
k	$s = 14$	$s = 8$	$s = 14$	$s = 8$	$s = 14$	$s = 8$	$s = 14$	$s = 8$
1	89.7 ± 0.6	89.1 ± 0.4	$\mathbf{89.6 \pm 0.6}$	89.1 ± 0.4	87.0 ± 0.8	85.8 ± 1.1	87.2 ± 0.9	85.8 ± 0.8
5	$\mathbf{91.7 \pm 0.4}$	91.1 ± 0.5	91.6 ± 0.3	91.2 ± 0.6	89.9 ± 1.7	88.8 ± 1.8	89.8 ± 1.7	88.7 ± 1.7
10	$\mathbf{91.9 \pm 0.5}$	91.4 ± 0.5	$\mathbf{91.9 \pm 0.5}$	91.5 ± 0.4	90.3 ± 1.0	89.0 ± 0.5	90.2 ± 1.1	89.0 ± 0.6
20	$\mathbf{91.6 \pm 0.6}$	91.2 ± 0.3	$\mathbf{91.6 \pm 0.4}$	91.2 ± 0.4	90.0 ± 1.1	89.0 ± 0.5	89.8 ± 1.1	88.9 ± 0.6
50	$\mathbf{91.4 \pm 0.9}$	90.7 ± 0.6	91.3 ± 1.0	90.7 ± 0.5	88.8 ± 1.1	88.6 ± 0.8	88.9 ± 1.1	88.5 ± 0.8
100	$\mathbf{90.9 \pm 1.1}$	90.1 ± 0.6	$\mathbf{90.9 \pm 1.1}$	90.0 ± 0.5	88.2 ± 1.0	87.6 ± 1.0	88.1 ± 1.0	87.6 ± 0.9

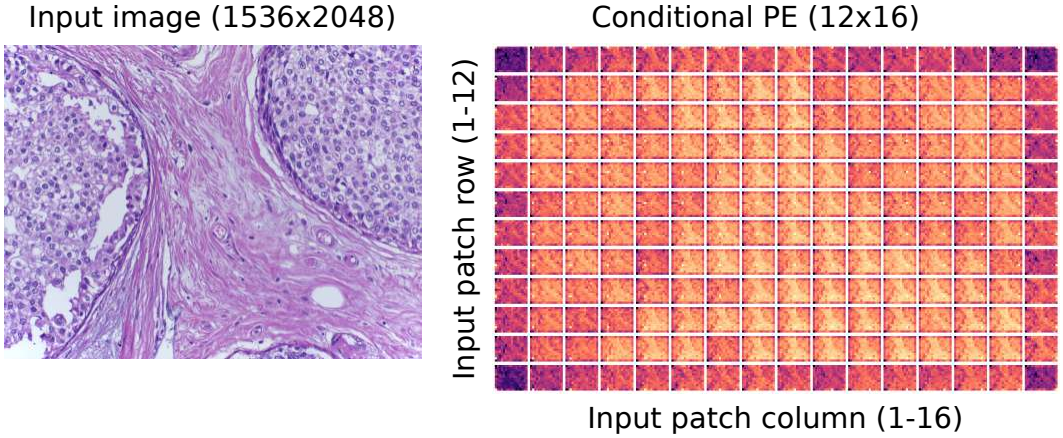

Figure 2: The conditional position encoding [6] of a non-squared input image is represented. The PE is image-dependent and captures the local interactions between tokens.

purpose, at test time, we mask 75% of the selected high-resolution regions and report the obtained results in Table 7. As expected, we observe that the $\mathrm{ScoreNet}/4/3$ variant uses the high-resolution content more. Furthermore, these results shed light on how the high-resolution information is not equally relevant for each class. An interesting observation is that for each variant of $\mathrm{ScoreNet}$, the higher the performance of a given model is, the more it is affected by the removal of the high-resolution information (see Table 8).

Table 8: The performance drop incurred by the high-resolution masking operation of individual models is monitored. The models that rely the most on the high-resolution content are the ones that perform the best.

$\mathrm{ScoreNet}/4/1$			$\mathrm{ScoreNet}/4/3$		
63.3	$\xrightarrow{-0.6}$	62.7 ± 0.2	63.3	$\xrightarrow{-2.8}$	60.5 ± 0.1
63.8	$\xrightarrow{-2.2}$	61.6 ± 0.1	64.8	$\xrightarrow{-5.2}$	59.6 ± 0.3
64.9	$\xrightarrow{-2.2}$	62.7 ± 0.3	65.0	$\xrightarrow{-6.4}$	58.6 ± 0.3

Despite that, we expected a more considerable drop in performance from this masking operation, which raises the question; *is* $\mathrm{ScoreNet}$ *nothing but a glorified low-resolution* ViT*?* To answer that question, we train the same ViT as the one used in the recommendation stage and the same setting, but basing the predictions on the scorer's [CLS] tokens and hence without the feedback from the high-resolution stage. Table 9 clearly shows a gap of almost 10% compared to

Table 4: A weighted k Nearest Neighbors classifier assesses the learned features' discriminability (weighted F1-score) on the low-resolution BACH and BRACS datasets. A fixed and absolute PE (2D sine-cos)'s performances are compared to a learnable and conditional PE (CPVT and CPVT-GAP). The k-NN classifier is trained on the merged train/valid set and evaluated on the test set (BRACS), and a 4-fold approach with $75\%/25\%$ train/test splits is used for BACH dataset.

	BACH						BRACS					
	2D sine-cos		CPVT		CPVT-GAP		2D sine-cos		CPVT		CPVT-GAP	
k	Teacher	Student	Teacher	Student	Teacher	Student	Teacher	Student	Teacher	Student	Teacher	Student
1	76.0 ± 3.4	75.0 ± 4.0	76.6 ± 2.9	77.7 ± 2.0	$\mathbf{78.5 \pm 6.4}$	77.4 ± 5.1	42.2	42.3	49.6	49.2	54.3	**55.0**
5	74.6 ± 4.2	75.6 ± 4.2	76.8 ± 3.0	76.3 ± 3.7	$\mathbf{81.7 \pm 3.2}$	81.0 ± 4.0	45.3	45.7	53.3	53.2	**56.1**	55.8
10	76.3 ± 4.1	75.6 ± 4.6	76.3 ± 5.0	76.0 ± 5.2	77.8 ± 2.8	$\mathbf{77.9 \pm 3.4}$	47.2	46.3	54.3	54.5	56.4	**56.7**
20	73.9 ± 3.5	73.9 ± 3.5	75.7 ± 5.3	72.9 ± 5.8	76.3 ± 3.0	$\mathbf{76.5 \pm 4.0}$	48.2	47.6	53.3	51.5	**58.0**	57.6
50	73.5 ± 4.3	73.0 ± 4.1	74.2 ± 5.1	73.4 ± 6.5	$\mathbf{74.7 \pm 4.7}$	74.3 ± 5.7	47.0	47.3	50.8	49.7	**57.5**	56.9
100	72.8 ± 3.7	73.0 ± 3.1	73.6 ± 5.8	71.4 ± 7.4	$\mathbf{74.0 \pm 5.5}$	73.6 ± 5.9	45.5	45.0	48.4	48.1	**54.0**	53.7

Table 5: A weighted k Nearest Neighbors classifier assesses the discriminability (weighted F1-score) of the learned features on the tile CAMELYON16 dataset (see Sec. C.3). A fixed and absolute PE (2D sine-cos)'s performances are compared to a learnable and conditional PE (CPVT and CPVT-GAP). A 4-fold approach with $75\%/25\%$ train/test splits is used.

	2D sine-cos		CPVT		CPVT-GAP	
k	Teacher	Student	Teacher	Student	Teacher	Student
1	88.2 ± 0.9	88.4 ± 0.6	88.8 ± 0.4	88.8 ± 0.4	$\mathbf{89.1 \pm 0.4}$	89.1 ± 0.4
5	91.1 ± 0.9	91.0 ± 1.0	90.9 ± 0.5	90.9 ± 0.6	91.1 ± 0.5	$\mathbf{91.2 \pm 0.6}$
10	91.2 ± 0.7	91.2 ± 0.5	91.1 ± 0.4	91.1 ± 0.3	91.4 ± 0.5	$\mathbf{91.5 \pm 0.4}$
20	91.1 ± 0.7	91.1 ± 0.6	91.3 ± 0.5	$\mathbf{91.4 \pm 0.5}$	91.2 ± 0.3	91.2 ± 0.4
50	90.5 ± 0.7	90.6 ± 0.7	$\mathbf{90.8 \pm 0.6}$	$\mathbf{90.8 \pm 0.5}$	90.7 ± 0.6	90.7 ± 0.5
100	$\mathbf{90.2 \pm 1.8}$	$\mathbf{90.2 \pm 0.8}$	$\mathbf{90.2 \pm 0.6}$	$\mathbf{90.2 \pm 0.6}$	90.1 ± 0.6	90.0 ± 0.5

ScoreNet's results and, more interestingly, a gap of more than 5% when compared to the same ViT, but trained with the high-resolution feedback. The above results indicate that **high-resolution information distillation occurs during the training of ScoreNet**.

E. Computational Cost

Vision transformers heavily rely on the attention mechanism to learn a high-level representation from low-level regions. The underlying assumption is that the different sub-regions of the image are not equally important for the overall representation. Despite this key observation, the computation cost dedicated to a sub-region is independent of its contribution to the high-level representation, which is inefficient and undesirable. Our ScoreNet attention mechanism overcomes this drawback by learning to attribute more computational resources to regions of high interest. Let us consider a high-resolution input image $x_h \in \mathbb{R}^{C \times H \times W}$, a low-resolution version of the image $x_l \in \mathbb{R}^{C \times h \times w}$ is obtained by applying a down-scaling factor s, as $h = H/s$ and $w = W/s$. The low-resolution image is fed to a scorer model (recommendation stage), which recommends the regions where to apply fine-grained attention. If this operation is implemented by a ViT, its computational cost is $\mathcal{O}\left(\left(\frac{h}{P_l} \cdot \frac{w}{P_l}\right)^2\right)$ with P_l is the dimension of the patches in low-resolution. Using a ViT as the scorer model, there is a one-to-one mapping between the low-resolution patches and the regions the model can process with fine-grained attention; as a consequence, the dimension of the regions is $P_h = s \cdot P_l$. Attending to such regions with a patch size, P_a, has a computational cost of $\mathcal{O}\left(\left(\frac{P_h}{P_a} \cdot \frac{P_h}{P_a}\right)^2\right)$ and the model processes k of them, hence $\mathcal{O}\left(k \cdot \left(\frac{P_h}{P_a} \cdot \frac{P_h}{P_a}\right)^2\right)$. Finally, a coarse attention mechanism is applied to endow the locally attended regions with contextual information. This final step costs $\mathcal{O}\left(k^2\right)$. On the other hand, a vanilla ViT would attend uniformly across the whole image with a cost of $\mathcal{O}\left(\left(\frac{H}{P_a} \cdot \frac{W}{P_a}\right)^2\right)$. Importantly, we observe that only the recommendation stage's cost depends on the input size; consequently, if this step is implemented as a ViT and with a down-scaling ratio $s \in [8, 14]$, the asymptotic cost is reduced by approximately two orders of magnitude, as we typically used $P_a = P_l$ in practice. At last, one can observe that the asymptotic cost can be made linear w.r.t. the input dimension by adopting a convolution-based architecture for the recommendation stage.

F. ScoreMix Investigation & Examples

The underlying assumption of the "cut-and-paste"-based augmentation methods is that the trained model can assimilate the pasted region to the representation of the image it is pasted in. In the case of ScoreNet, it translates to attending to the pasted area in a low or high-resolution image. Fig. 3 depicts an example of ScoreNet being able to detect and localize the pasted regions even when it is small and hard to distinguish. We further observe that a local change in the image directly affects the global representations as the representation of each token is adapted to accommodate the local change in information. This behavior would typically

Table 6: The number of finely attended regions is selected by independently training our pipeline 5 times on 10% of the BRACS dataset with a varying number of proposal regions. The number of training epochs is fixed and is the same for all experiments. The models are trained with standard data augmentation methods, i.e., none of ScoreMix, SaliencyMix, or CutMix.

# Regions	Normal	Benign	UDH	ADH	FEA	DCIS	Invasive	Weighted F1
$k = 5$	**53.7 ± 5.2**	**44.0 ± 5.1**	29.7 ± 5.3	28.8 ± 6.8	69.3 ± 4.2	56.9 ± 6.5	**86.9** ± 3.2	54.2 ± 1.8
$k = 10$	52.1 ± 6.2	**44.0 ± 3.9**	**31.0 ± 5.3**	28.6 ± 4.3	69.8 ± 3.6	56.4 ± 3.9	85.9 ± 1.4	54.0 ± 0.8
$k = 20$	52.2 ± 3.4	42.2 ± 5.6	29.6 ± 7.5	**31.9 ± 5.3**	**71.9 ± 2.3**	**57.5 ± 3.6**	**86.9 ± 2.5**	**54.7 ± 0.8**
$k = 50$	51.5 ± 5.4	42.8 ± 4.7	30.0 ± 6.8	25.9 ± 7.1	70.5 ± 4.0	55.8 ± 5.2	85.7 ± 0.9	53.3 ± 2.5

Table 7: At test time, 75% of the selected high-resolution regions are randomly masked. ScoreNet/4/1 and ScoreNet/4/3 do not equally rely on the high-resolution content.

Masking	Normal	Benign	UDH	ADH	FEA	DCIS	Invasive	Weighted F1
ScoreNet/4/1	64.6 ± 2.2	52.6 ± 2.8	48.4 ± 2.2	47.4 ± 2.4	77.9 ± 0.7	59.3 ± 1.1	90.6 ± 1.5	64.1 ± 0.7
Masked ScoreNet/4/1	61.1 ± 2.7	50.8 ± 1.4	45.9 ± 2.2	41.0 ± 3.5	78.8 ± 0.5	59.9 ± 3.3	90.6 ± 1.1	62.4 ± 0.6
ScoreNet/4/3	64.3 ± 1.5	54.0 ± 2.2	45.3 ± 3.4	46.7 ± 1.0	78.1 ± 2.8	62.9 ± 2.0	91.0 ± 1.4	64.4 ± 0.9
Masked ScoreNet/4/3	64.9 ± 2.4	51.7 ± 0.5	44.4 ± 4.0	22.0 ± 6.2	77.6 ± 1.0	60.8 ± 1.6	87.2 ± 1.3	59.6 ± 0.7

not be observed in a CNN-based architecture until the very last layers. Fig. 3 further highlights the ability of ScoreMix to treat images of different dimensions and aspect ratios.

G. Learning From Uncurated Data.

We gauge the ability of ScoreNet to learn from unlabeled data on the BACH dataset [2], which encompasses both a small set of 400 annotated TRoIs images, and the WSIs containing the aforementioned TRoIs. Our model is first pre-trained using DINO's self-supervised learning scheme [5] on an unlabeled set of $\approx$ 11k images extracted from WSIs and then is evaluated on the labeled image set using standard protocols, namely linear probing and k-NN (see Table 10). We also report the non-empty cluster's purity for the clusters learned by DINO. This metric indicates the quality of a cluster containing samples from a single class. Learning from large uncurated images is particularly challenging, as the increased receptive field allows for the representation of more complex tissue interactions. This further deviates from the discriminative pretext task's assumption that the images represent a single centered object. Since the DINO method enforces a local-to-global correspondence between large and smaller image crops, it may enforce similarity between different tissue types. For that purpose, we modify DINO's pretext task so that the student network only processes the highly discriminative patches to match the teacher's representation, allowing the processing of all the high-resolution patches. To ensure that the pretext task does not encourage contextual biases [17], we only employ augmentations that change the image pixels' values, but not their locations, such that the semantic content of the two augmented views is identical. As can be observed in Table 10, this proposed strategy yields significant improvements compared to other baselines.

Table 10: Comparison with the prior art for learning capabilities from uncurated data on the BACH dataset using DINO's pre-training. A comparison results between the effectiveness of DINO's standard pretext task (ScoreNet) and the proposed unbiased pretext task (ScoreNet†) are also reported.

	ScoreNet†	ScoreNet	TransPath [18]	SwinTransformer [12]
k-NN	**73.7 ± 1.7**	65.0 ± 3.7	65.2 ± 1.4	63.7 ± 4.1
Lin. eval	**73.0 ± 2.9**	66.0 ± 2.6	64.2 ± 4.0	62.5 ± 1.7
Purity	**78.3 ± 23.9**	76.4 ± 24.9	74.0 ± 23.3	71.8 ± 23.9

Table 9: The ViT network of recommendation stage is trained without receiving any feedback from the high-resolution-based predictions. Its features discriminability is significantly worth than that of the same model but trained jointly with the high-resolution stage.

Model	Normal	Benign	UDH	ADH	FEA	DCIS	Invasive	Weighted F1
ViT	53.3 ± 2.8	42.8 ± 1.9	37.1 ± 2.9	32.4 ± 2.4	77.3 ± 0.2	51.2 ± 1.3	85.0 ± 1.8	55.5 ± 0.1
Lin. scorer's [CLS]	57.5 ± 4.2	48.8 ± 5.5	42.7 ± 3.5	42.7 ± 7.4	74.3 ± 5.2	60.5 ± 2.4	90.6 ± 0.2	60.9 ± 3.1
ScoreNet/4/1	$\mathbf{64.6 \pm 2.2}$	52.6 ± 2.8	$\mathbf{48.4 \pm 2.2}$	$\mathbf{47.4 \pm 2.4}$	77.9 ± 0.7	59.3 ± 1.1	90.6 ± 1.5	64.1 ± 0.7
ScoreNet/4/3	64.3 ± 1.5	$\mathbf{54.0 \pm 2.2}$	45.3 ± 3.4	46.7 ± 1.0	$\mathbf{78.1 \pm 2.8}$	$\mathbf{62.9 \pm 2.0}$	$\mathbf{91.0 \pm 1.4}$	$\mathbf{64.4 \pm 0.9}$

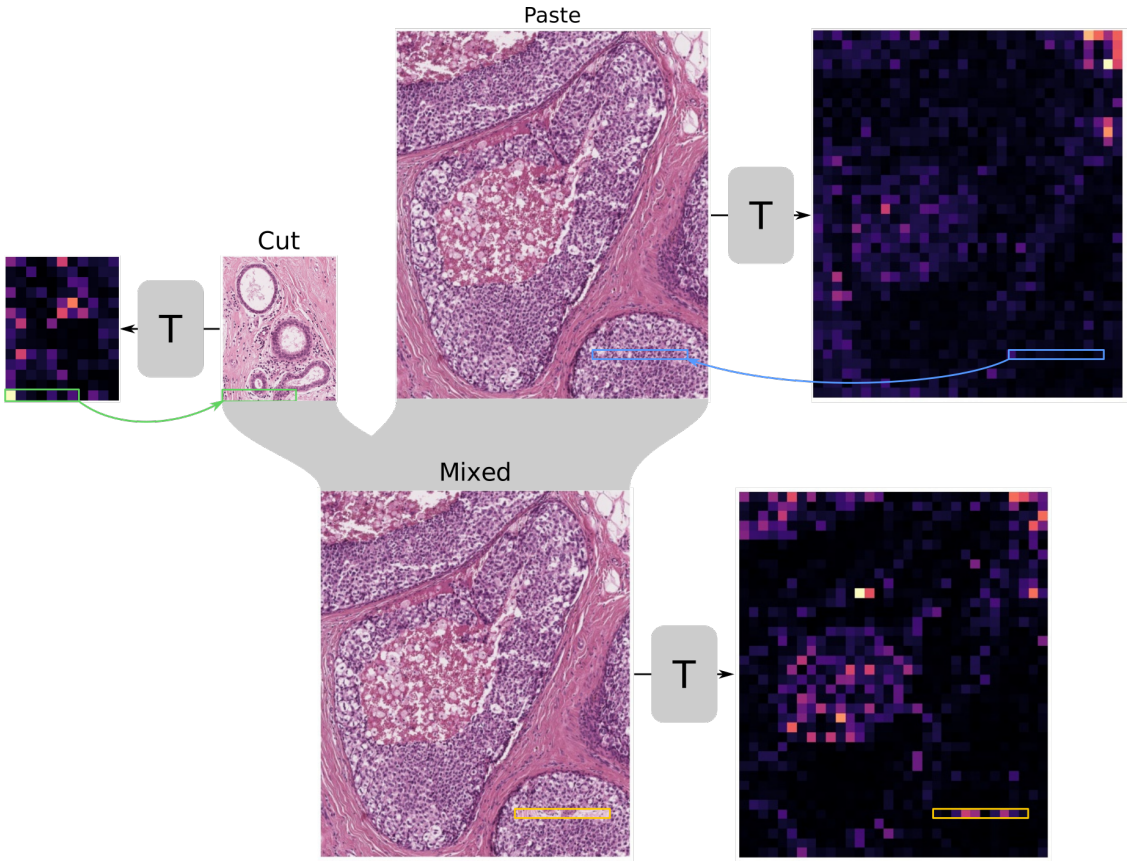

Figure 3: The learned semantic distribution can detect and localize the newly pasted content. The green box highlights the region pasted from the *cut* image to the *paste* image. The blue box represents the region where the new content is pasted. The yellow box highlights the modified region in the mixed image. T represents the scorer network of ScoreNet.

References

[1] Jacob Abernethy, Chansoo Lee, and Ambuj Tewari. Perturbation techniques in online learning and optimization. *Perturbations, Optimization, and Statistics*, 233, 2016.

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

[3] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. *arXiv preprint arXiv:2002.08676*, 2020.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[6] Xiangxiang Chu, Bo Zhang, Zhi Tian, Xiaolin Wei, and Huaxia Xia. Do we really need explicit position encodings for vision transformers? *CoRR*, abs/2102.10882, 2021.

[7] Jean-Baptiste Cordonnier, Aravindh Mahendran, Alexey Dosovitskiy, Dirk Weissenborn, Jakob Uszkoreit, and Thomas Unterthiner. Differentiable patch selection for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2351–2360, 2021.

[8] 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.

[9] 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. Ieee, 2009.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

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

[14] Pushpak Pati, Guillaume Jaume, Antonio Foncubierta, Florinda Feroce, Anna Maria Anniciello, Giosuè Scognamiglio, Nadia Brancati, Maryse Fiche, Estelle Dubruc, Daniel Riccio, et al. Hierarchical cell-to-tissue graph representations for breast cancer subtyping in digital pathology. *arXiv e-prints*, pages arXiv–2102, 2021.

[15] David W Romero and Jean-Baptiste Cordonnier. Group equivariant stand-alone self-attention for vision. In *International Conference on Learning Representations*, 2020.

[16] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, and Yongbing Zhang. Transmil: Transformer based correlated multiple instance learning for whole slide image classication. *arXiv preprint arXiv:2106.00908*, 2021.

[17] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: Learning to overcome contextual bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11070–11078, 2020.

[18] 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 *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 186–195. Springer, 2021.