A self-supervised framework for learning whole slide representations

Xinhai Hou[*] **Cheng Jiang**[*] **Akhil Kondepudi** **Yiwei Lyu**
Asadur Chowdury **Honglak Lee** **Todd Hollon**

University of Michigan [*]Equal Contribution

{xinhaih, chengjia, tocho}@umich.edu https://mlins.org/spt/

Abstract

Whole slide imaging is fundamental to biomedical microscopy and computational pathology. Previously, learning representations for gigapixel-sized whole slide images (WSIs) has relied on multiple instance learning with weak labels, which do not capture the diverse morphologic features and spatial heterogeneity of WSIs. A high-quality self-supervised learning method for WSIs would provide transferable visual representations for downstream computational pathology tasks, without the need for dense annotations. We present *Slide Pre-trained Transformers (SPT)*, a self-supervised learning framework for gigapixel-scale self-supervision of WSIs. Treating WSI patches as tokens, SPT combines data transformation strategies from language and vision modeling into a general and unified framework to generate views of WSIs for self-supervised pretraining. SPT leverages the inherent regional heterogeneity, histologic feature variability, and information redundancy within WSIs to learn high-quality whole slide representations. We benchmark SPT visual representations on five diagnostic tasks across three biomedical microscopy datasets. SPT significantly outperforms baselines for histopathologic diagnosis, cancer subtyping, and genetic mutation prediction. Finally, we demonstrate that SPT consistently improves whole slide representations when using off-the-shelf, in-domain, and foundational patch encoders for whole slide multiple instance learning.

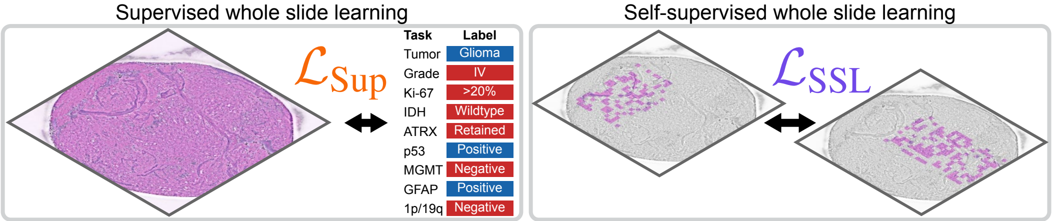

Supervised whole slide learning
Self-supervised whole slide learning
$\mathcal{L}_{\text{Sup}}$
Task
Label
Tumor
Glioma
Grade
IV
Ki-67
>20%
IDH
Wildtype
ATRX
Retained
p53
Positive
MGMT
Negative
GFAP
Positive
1p/19q
Negative
$\mathcal{L}_{\text{SSL}}$

Figure 1: **Self-supervised whole slide learning.** Previous work in computational pathology relies on multiple instance learning with weak supervision from slide or patient-level labels to learn whole slide representations [1, 2, 3, 4, 5, 6, 7]. We present a self-supervised framework for learning whole slide representations, called *Slide Pre-trained Transformers (SPT)*, by combining data transformations from vision and language modeling to generate high-quality paired views.

1 Introduction

Whole slide imaging is an integral part of tissue diagnosis and laboratory medicine. Computational pathology can provide rapid tissue analysis of complex WSIs, such as cancer detection, subtyping,

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

and grading. Developing scalable and efficient foundation models at the WSI level is crucial for advancing diagnosis, prognosis, and treatment planning in clinical settings. Traditional multiple instance learning (MIL) methods, including ABMIL [2], CLAM [3], DSMIL [8], TransMIL [4], can achieve good performance on the above diagnostic tasks. Unfortunately, these methods rely on slide annotations which are weak, sparse, incomplete, and expensive to obtain [9]. Moreover, WSIs are gigapixel-size and contain diverse morphologic and histopathologic features with extensive spatial heterogeneity. Weak slide labels may annotate only a small region within WSIs, demonstrating the limitation of relying on weak supervision to achieve high-quality and transferable whole slide representations.

To compensate for weak whole slide labels, self-supervised foundation models have been increasingly used in computational pathology. The majority of previous foundation models, such as UNI [7] and Virchow [10], have focused on the local patch level. However, few previous studies have investigated how to extend these foundation models to learn whole slide representations that captures tumor heterogeneity and global super-cellular structures [11, 5]. A general and unified framework for whole slide SSL would enable transferable whole slide feature learning and generalize to a wide range of downstream pathology tasks with minimal to no annotations required.

In this paper, we present Slide Pre-trained Transformers (SPT) for self-supervised whole slide representation learning. SPT treats gigapixel WSIs as a sequence of patch tokens and applies a domain-informed set of vision-language transformations, including splitting, cropping, and masking, to generate distinct views for self-supervised training. Over a range of patch encoders, including state-of-the-art foundation models, SPT learns high-quality patch feature aggregation and whole slide representations compared to the state-of-the-art baselines. The main contributions are:

- We introduce SPT, a general, flexible, and unified learning framework for WSIs at scale and benchmark on five computational pathology tasks.
- SPT outperforms previous state-of-the-art self-supervised and supervised methods for WSI representation learning.
- SPT offers a consistent performance boost across a wide range of patch encoders.

2 Related Work

2.1 Computational pathology

Computational pathology combines whole slide imaging and computer vision methods for the analysis of biomedical tissues. Cancer diagnosis, prognostication, and response-to-treatment are some of the most common computer vision tasks within computational pathology. WSIs pose a unique computer vision challenge due to image sizes ranging up to 150K × 150K pixels. Data annotations are often limited to whole slide or patient-level labels [9]. Moreover, WSIs have a unique data structure compared to natural images, such as being non-object-centric and containing regional heterogeneity and visual feature redundancy. Despite these challenges, significant progress has been made over the last decade due to more accessible slide scanners and modern deep-learning methods. Cancer diagnosis [9, 16, 3, 17], molecular classification [18, 19], and prognostication [20] with WSIs represent how computational pathology can be used to improve diagnostic medicine and patient care. Whole slide representation learning for gigapixel WSI search and public WSI-natural language supervision represent future directions in computational pathology [21, 22].

2.2 Multiple instance learning

Multiple instance learning (MIL) is a type of supervised learning such that labels are available only for bags of instances, rather than individual instances [23]. Learning to classify WSIs has generally been regarded as a MIL task, where each patch is an instance and the WSI is a bag of patches [2]. There are two main components in MIL for WSI: the patch encoder and aggregator. Early works directly train the patch encoder in a weakly supervised fashion with WSI labels and generate a whole slide representation with mean pooling [1, 24]. Many works have used ResNet or related architectures for patch encoders, often pre-trained on ImageNet [25, 26, 9, 27, 3, 4]. Attention-based multiple instance learning (ABMIL) was first introduced in 2018 for WSI classification [2]. Subsequently, clustering-constrained attention multiple instance learning (CLAM) was used for large-scale weakly

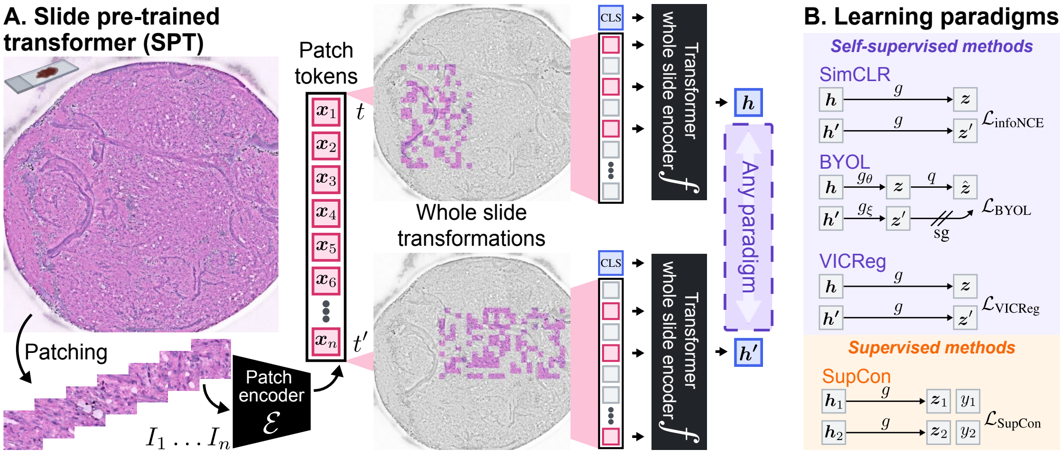

A. Slide pre-trained transformer (SPT)
Patching
$I_1 \ldots I_n$
Patch encoder $\mathcal{E}$
Patch tokens
x_1
x_2
x_3
x_4
x_5
x_6
x_n
t
t'
Whole slide transformations
CLS
CLS
Transformer whole slide encoder f
Transformer whole slide encoder f
h
h'
Any paradigm
B. Learning paradigms
Self-supervised methods
SimCLR
$h \xrightarrow{g} z$
$h' \xrightarrow{g} z'$
$\mathcal{L}_{\text{infoNCE}}$
BYOL
$h \xrightarrow{g_\theta} z \xrightarrow{q} \hat{z}$
$h' \xrightarrow{g_\xi} z'$
sg
$\mathcal{L}_{\text{BYOL}}$
VICReg
$h \xrightarrow{g} z$
$h' \xrightarrow{g} z'$
$\mathcal{L}_{\text{VICReg}}$
Supervised methods
SupCon
$h_1 \xrightarrow{g} z_1$ y_1
$h_2 \xrightarrow{g} z_2$ y_2
$\mathcal{L}_{\text{SupCon}}$

Figure 2: **SPT overview. A.** The SPT framework consists of a two-stage model architecture: 1) a pre-trained patch encoder $\mathcal{E}$ and 2) a transformer whole slide encoder f. WSIs are first divided into small patches, and the patch encoder extracts patch-level features. We then apply whole slide transformations to the patch tokens to create two views of the same WSI. The transformations combine splitting, cropping, and masking, which are informed by the structure and unique properties of WSIs. The transformed views are encoded by the transformer whole slide encoder, and the slide-level feature learning can use any paradigm. **B.** Example learning paradigms. In our experiments, we focus on three representative self-supervised paradigms, including SimCLR [12], BYOL [13], and VICReg [14], and supervised contrastive learning [15].

supervised computational pathology. Subsequent MIL work has used variations of attention and more powerful transformer-based aggregation models, such as TransMIL [27, 26, 8, 3, 4, 28, 29, 30].

2.3 Self-supervised learning and foundation models

Self-supervised learning (SSL) is a method of learning representations that does not require labels or annotations. SSL defines pretext learning tasks, such as instance discrimination or reconstruction, as training objectives and is typically evaluated by the performance of the learned representations on downstream tasks [31]. SSL has shown success in natural images [12, 13, 14], natural language processing [32, 33, 34, 35], and multimodal learning [22, 36]. SSL has been used for patch encoding in WSIs. DSMIL [8] applied SimCLR to learn patch representations. Most recently, PLIP [22] leverages histology images and comments posted to Twitter for a self-supervised vision-language foundation model. UNI [7] and Virchow[10] are general-purpose foundational H&E patch encoders for a wide range of tissue types and downstream tasks trained on a large-scale institutional dataset.

While there have been attempts at learning whole slide representations in an end-to-end manner with supervision [37], or enhancing them with a supervised contrastive loss [38], few works have focused on self-supervision beyond patches. HiDisc [39] uses a data hierarchy to learn patch representations that are invariant to slide and patient membership. HIPT [5] uses a self-supervised patch encoder and an encoder for local regions, but weak supervision is used for learning WSI representations. Giga-SSL [11] and SS-MIL [40] attempted to learn whole slide representations using self-supervision by masking patch tokens. Giga-SSL also emulates slide augmentations using pre-computed augmented patch representations. However, this strategy is memory intensive, does not span all combinations of possible random augmentations, and is infeasible for large-scale datasets.

3 Methods

3.1 The SPT framework

WSIs are partitioned into smaller non-overlapping fields-of-view or patches. Let $\boldsymbol{I} = \{I_i\}_{i=1}^n$ be a WSI, where n is the number of patches in a WSI, and each $I_i \in \mathbb{R}^{3\times H\times W}$, with coordinate $p_i \in \mathbb{Z}^2$. $\boldsymbol{p} = \{p_i\}_{i=1}^n$ are the coordinates of all patches in the WSI. SPT is a two-stage learning framework to learn WSI representations: 1) a pre-trained patch encoder $\mathcal{E}$; and 2) a transformer whole slide encoder f. The pre-trained patch encoder $\mathcal{E}$ can have the architecture of any visual feature extractor. The overall model architecture of SPT is illustrated in Figure 2.

In the SPT framework, the patch encoder $\mathcal{E}$ encodes each I_i into a patch token, $x_i = \mathcal{E}(I_i)$. We represent all patch tokens in a WSI as $\boldsymbol{x} = \{x_i\}_{i=1}^n = \mathcal{E}(\boldsymbol{I})$. The whole-slide encoder f serves as an aggregation function that learns a whole-slide representation using patch tokens and their corresponding coordinates: $h = f(\boldsymbol{x}, \boldsymbol{p}) = f(\mathcal{E}(\boldsymbol{I}), \boldsymbol{p})$. f can be any learned aggregation architecture, such as in ABMIL [2] or a transformer [4, 41]. Due to the scale of WSIs, it is infeasible to both train the patch encoder $\mathcal{E}$ and whole-slide encoder f jointly in an end-to-end manner. Thus, we freeze the patch encoder $\mathcal{E}$ to allow for large mini-batch training of the whole-slide encoder f. *The aim of the SPT framework is to learn high-quality whole slide representations with f using self-supervision.*

Self-supervised methods share a common strategy: apply random transformations to a single data example to generate distinct views, called positive pairs. We designed our SPT framework to be compatible with existing SSL objectives, where a WSI undergoes transformations into different views. With a two-view SSL paradigm, $t(\boldsymbol{I}) = [I_i \dots I_k]$, $t(\boldsymbol{p}) = [p_i \dots p_k]$, $t'(\boldsymbol{I}) = [I_j \dots I_\ell]$, and $t'(\boldsymbol{p}) = [p_j \dots p_\ell]$, where $t, t' \in \mathcal{T}$ are randomly drawn from a set of transformations described in section 3.2. The patches and their corresponding coordinates from these transformed views are processed through $\mathcal{E}$ and f to obtain the whole slide representations:

$$h = f\left(\mathcal{E}(t(\boldsymbol{I})), t(\boldsymbol{p})\right), \quad h' = f\left(\mathcal{E}(t'(\boldsymbol{I})), t'(\boldsymbol{p})\right).$$

An SSL loss is used to self-supervise whole slide training:

$$\mathcal{L}_{\text{SPT}}\left(h, h'\right).$$

In our experiments, we selected representative methods from different SSL families [42], including SimCLR [12] from contrastive learning, BYOL [13] from self-distillation, and VICReg [14] from canonical component analysis. We denote the self-supervised SPT as ***ssSPT***.

SPT with supervision. SPT can be adapted to fully supervised training using weak slide- or patient-level labels, by applying a supervised contrastive loss. Positive pairs for supervised contrastive learning are defined by class labels [15]. We denote the supervised variant of SPT as ***suSPT***.

3.2 SPT transformation

While previous work focused on pixel space transformations [11], we hypothesized that these augmentations at patch level are insufficient to generate high-quality views for whole slide SSL. A patch encoder trained with instance discrimination, for example, should be invariant to pixel-space augmentations. Thus, pixel space augmentations, such as color jittering, should have minimal effect on the representation space and resulting similar embeddings, as illustrated with SimCLR in Figure 3, because it is explicitly enforced as the pretext task. We opt to bypass patch augmentations altogether, thereby reducing SPT memory and compute burden, and focusing on transformations at WSI level.

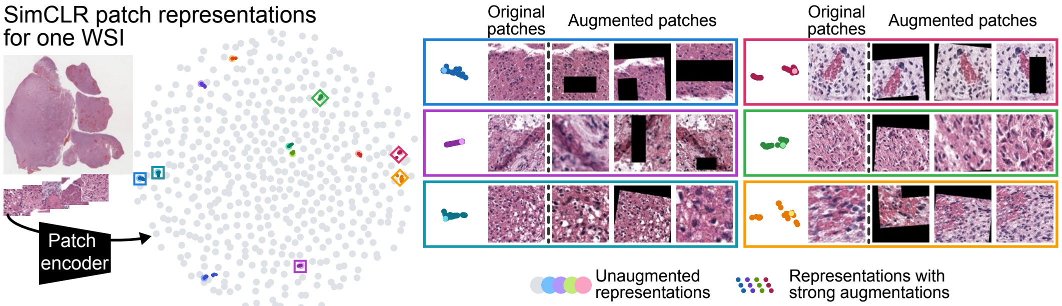


Figure 3: **Limited effect of pixel-level patch augmentations.** We qualitatively evaluate the effect of pixel-level augmentation on the patch representations by visualizing the *t*SNE plot of SimCLR pre-trained patch representations sampled from a single WSI. We observe that strong augmentations at the pixel level have a minimal effect on the patch embeddings. The invariant behavior of the patch encoder is explicitly enforced by the SimCLR pretext task.

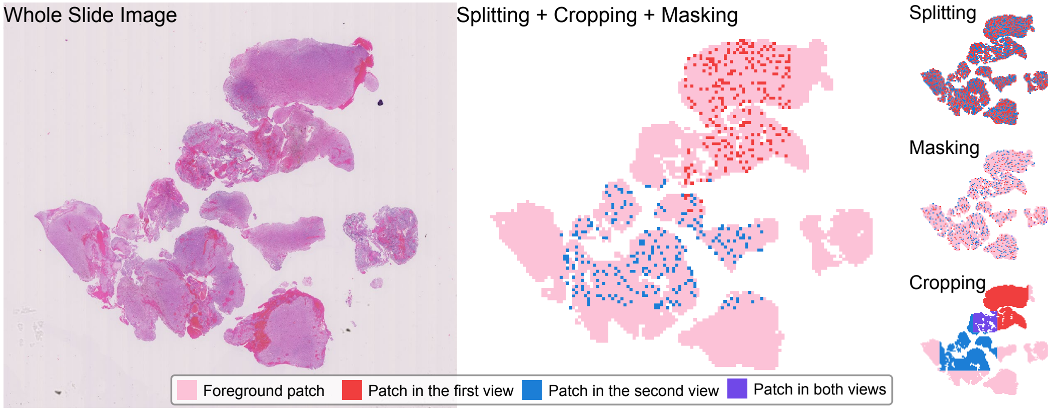


Figure 4: **SPT transformation strategy.** SPT combines splitting, cropping, and masking to generate views, and they are motivated by the size, region diversity, and information redundancy of WSIs. Splitting partitions patches into mutually exclusive sets decreases mutual information between views; cropping can generate spatially diverse views covering different regions on the WSI; masking reduces redundant visual features and improves training efficiency. The combination of these transformations can create optimal positive pairs for whole slide representation learning.

Transformation strategy. WSIs are divided into a sequence of patch tokens. The SPT transformation strategy, as shown in Figure 4, is inspired by both vision and language modeling, and was selected to address the domain-specific properties of WSI:

- **Splitting** *to decrease mutual information between views.* Reducing mutual information between SSL views, while keeping task-relevant information intact, improves downstream performance [43, 44]. The splitting transformation randomly partitions patch tokens into disjoint sets, ensuring mutually exclusive set membership for each token. Enforcing mutual exclusion reduces mutual information between SSL views.
- **Cropping** *to capture regional heterogeneity*. WSIs contain regional differences in histologic features. Similar to multi-cropping in visual SSL [12, 45, 46], cropping generates spatially diverse views of WSI with variable sizes and features.
- **Masking** *to reduce redundant visual features*. Due to their gigapixel size, WSIs can contain large regions of similar histologic features and tissue phenotypes, causing redundancy. Masking is an effective transformation for vision and language modeling [5, 33, 47]. For each view, the masking transformation samples a subset of patch tokens without replacement. Additionally, masking increases training efficiency by reducing sequence lengths.

While splitting, cropping and masking can be used individually, we apply them jointly to generate versatile and high-quality views for SPT training.

3.3 SPT Implementation

End-to-end whole slide learning with large batch sizes is infeasible due to the gigapixel scale. To enable efficient training, all unaugmented patch tokens were computed on a frozen patch encoder, and SPT transformations were then applied to patch tokens. Each whole slide was represented as an embedding $\boldsymbol{x} \in \mathbb{R}^{n \times d}$ and a coordinate $\boldsymbol{p} \in \mathbb{Z}^{n \times 2}$. Splitting and masking were implemented as row-wise partitioning and dropout on $\boldsymbol{x}$, $\boldsymbol{p}$, respectively. Cropping was implemented as coordinate filtering on $\boldsymbol{p}$. A detailed description of our implementation with pseudocode is in Appendix B.

4 Experiments

4.1 Benchmarks

We evaluated SPT visual representations on five benchmarks across three clinical tasks, including two different imaging modalities. Additional dataset information and breakdown are in Appendix A.

SRH CNS benchmark. Stimulated Raman Histology (SRH) is a novel optical microscopy method that enables fast imaging of unprocessed tissues [48, 49]. The benchmark includes six central nervous system (CNS) tumors and normal brains, with 2035 and 925 WSIs for training and evaluation, respectively. These data were collected at [institution redacted for double-blind review], following the imaging protocol in [50], and were labeled by board-certified pathologists. The study has been approved by the Institutional Review Board, with informed consent from each patient.

H&E glioma molecular classification benchmark. We also evaluated SPT using publicly available diffuse glioma H&E stained WSIs from the Cancer Genome Atlas (TCGA) and Digital Brain Tumour Atlas (DBTA) [51]. We focused on the classification of three molecular subgroups, as defined by the World Health Organization [52]. Molecular classification is a challenging computer vision task because the diagnoses are made via molecular testing (such as genetic sequencing), and are not possible for expert pathologists using H&E images alone. Our glioma dataset is comprised of 2309 training slides and 341 slides from the TCGA dataset set for evaluation.

TCGA BRCA, TCGA NSCLC, and TCGA RCC benchmarks. We further evaluated SPT using three widely used TCGA H&E classification benchmarks: 1) invasive breast carcinoma (BRCA) subtyping, 2) non-small cell lung carcinoma (NSCLC) subtyping, and 3) renal cell carcinoma (RCC) subtyping. For these benchmarks, we followed the well-established study design of [5].

4.2 Implementation details

We trained in-domain ResNet-34 [53] patch encoders to extract features for all patches in each slide. These patch encoders were trained with SimCLR [12] and HiDisc [39]. Additional off-the-shelf and foundational patch encoders are described in Figure 5a. SPT slide encoders are six-layer transformers [41] with four heads, and learnable Fourier positional embeddings [54] (visualized in Appendix B.2). We used a two-layer projection head for SimCLR and VICReg, and one-layer projection and prediction heads for BYOL. We used AdamW optimizers and cosine decay schedulers after warm-up in the initial 10% of the iterations. The learning rate was adjusted between 10^{-3} and 10^{-7} to accommodate the training dynamics of different SSL methods and tasks. With our SPT slide transformation strategy, we adjusted cropping and masking sizes for each experiment and utilized up to 64 patches per slide for each view. We trained ssSPT and suSPT experiments up to 800 and 100 epochs, respectively, with an effective batch size of up to 1024 WSIs. All models were trained with mixed-precision on a NVIDIA A40 GPU, taking up to 8 hours.

4.3 Evaluation Protocol

We benchmarked SPT using standard linear evaluation protocols. Since linear classifiers are sensitive to hyperparameters [46], we also employed k nearest neighbor (kNN) for direct evaluation. H&E glioma molecular evaluation used only WSI from TCGA. For SRH and H&E Glioma, experiments were repeated with three random seeds, and for TCGA BRCA, NSCLC, and RCC, we used 10-fold cross-validation for error bars, following the protocols in previous work [5, 11]. For the baselines, original embeddings or logits were used when available. We used mean class accuracy (MCA), F1 scores, and area under the receiver operating characteristic (AUC) to evaluate all benchmarks.

5 Results

We first benchmarked ssSPT performance with self-supervised WSI learning strategies in section 5.1. We compared ssSPT and suSPT with supervised WSI MIL strategies in section 5.2. We then evaluated the ability of SPT to generalize across a wide range of patch encoders in section 5.3. Next, we showed that SPT improves MIL results with SOTA foundation model patch encoders in section 5.4. Finally, we visualized SPT self-attention heatmaps in section 5.5. Full results with error bars, ablation studies on the **SPT transformations and parameters,** and additional visualizations are in Appendix C.

5.1 ssSPT learns high-quality whole slide representations

We benchmarked ssSPT with self-supervised baselines using in-domain patch encoders. WSI features were evaluated directly using a kNN classifier. As shown in Table 1, ssSPT surpasses all baselines across all self-supervised objectives in all metrics, except for AUC on the highly imbalanced BRCA benchmark. ssSPT outperforms all existing self-supervised methods by a large margin, outperforming the previous best self-supervised method with a 10 and 5 points increase in MCA on the SRH CNS and H&E glioma benchmarks, respectively.

	SRH CNS			H&E Glioma			TCGA BRCA			TCGA NSCLC			TCGA RCC		
	MCA	F1	AUC	MCA	F1	AUC	MCA	F1	AUC	MCA	F1	AUC	MCA	F1	AUC
Self-supervised methods															
Pooling	72.5	73.2	94.8	68.3	67.9	87.9	58.1	27.4	74.7	75.1	76.5	85.3	84.2	86.3	96.6
HIPT	-	-	-	-	-	-	63.2	39.9	77.5	80.6	80.1	88.9	88.4	89.2	97.4
Giga-SSL	71.2	72.7	94.2	71.6	71.5	89.1	70.4	53.9	**85.4**	84.3	84.9	**92.3**	90.0	89.3	97.5
ssSPT (Ours)	**82.3**	**82.3**	**94.7**	**76.5**	**76.1**	**90.9**	**72.7**	**58.3**	80.4	**86.1**	**86.1**	**92.3**	**91.7**	**90.5**	**98.3**

Table 1: **Self-supervised benchmarks.** We use kNN classifier to evaluate ssSPT and baselines. We report the best performing SSL objective for ssSPT, with additional SSL objectives in Appendix C Table 11. Mean values are reported here and standard deviations are in Appendix C Table 7.

5.2 SPT outperforms previous fully supervised methods

We benchmarked ssSPT and suSPT with linear evaluation using in-domain patch encoders. As shown in Table 2, SPT outperforms previous fully supervised methods across all tasks. Remarkably, ssSPT outperforms or matches previous *fully supervised* methods on MCA in SRH CNS (+1.0), H&E glioma (match), TCGA BRCA (+2.2), and TCGA RCC (+1.4) benchmarks. suSPT outperforms existing MIL methods across all five benchmarks on nearly all metrics. In comparison with the best-performing baselines on MCA of these five benchmarks, suSPT achieves a performance increase of 1.2, 0.7, 3.4, 2.1, and 0.3 points, respectively. Thus, SPT provides a performance increase for both self-supervised and supervised learning, generalizing to different tissue types and diagnostic tasks.

	SRH CNS			H&E Glioma			TCGA BRCA			TCGA NSCLC			TCGA RCC		
	MCA	F1	AUC	MCA	F1	AUC	MCA	F1	AUC	MCA	F1	AUC	MCA	F1	AUC
Self-supervised methods															
Giga-SSL	78.6	75.6	96.1	76.5	73.9	90.7	81.7	62.6	**91.2**	87.5	87.9	93.7	89.9	87.8	97.7
ssSPT (Ours)	**85.4**	**83.2**	**97.6**	**80.0**	**78.0**	**91.9**	**83.0**	**67.4**	89.4	**88.0**	**88.0**	**94.8**	**93.4**	**91.1**	**98.7**
Supervised methods															
ABMIL	79.8	78.9	96.0	75.0	74.5	90.2	78.6	62.8	86.1	86.8	86.7	93.7	89.9	89.4	97.8
CLAM	83.5	82.9	96.7	78.9	77.7	88.5	76.8	59.3	85.8	84.3	83.6	92.8	85.6	83.6	97.3
DSMIL	-	-	-	-	-	-	77.0	58.0	83.8	84.9	84.0	92.0	90.1	87.1	97.1
TransMIL	84.4	84.0	84.0	80.0	79.4	91.9	80.8	69.4	90.0	87.4	87.5	95.3	87.4	87.5	95.3
HIPT	-	-	-	-	-	-	75.8	60.2	87.4	88.2	88.2	95.2	92.0	**92.1**	98.0
suSPT (Ours)	**85.6**	**84.4**	**97.2**	**80.7**	**80.3**	**93.7**	**84.2**	**70.4**	**90.5**	**90.3**	**90.4**	**95.7**	**92.3**	91.0	**98.6**

Table 2: **SPT benchmarks.** We report linear evaluation results on five histology benchmarks. We report the best performing SSL objective for ssSPT, with additional SSL objectives in Appendix C Table 11. Mean values are reported here and standard deviations are in Appendix C Table 8.

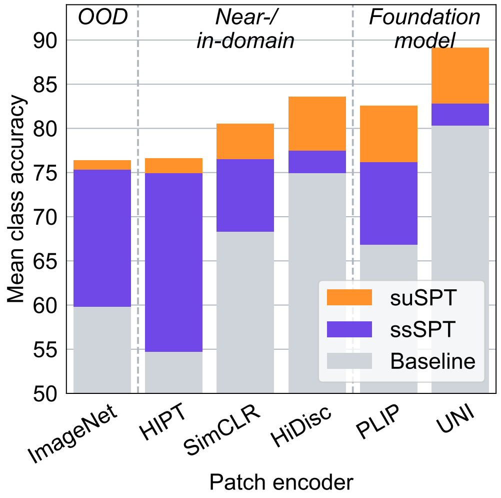
5.3 SPT offers performance boost across a range of patch encoders

We explored the generalizability of SPT to different patch encoders using the more challenging H&E Glioma benchmark. As described in Table 5a, we tested ImageNet, HIPT [5], PLIP [22], and UNI [7], in addition to in-domain patch encoders trained with SimCLR and HiDisc. The ImageNet patch encoder is off-the-shelf and out-of-distribution (OOD). HIPT [5] is near-domain since it was trained with H&E WSIs from TCGA, including multiple organ systems and institutions. PLIP [22] and UNI [7] are visual foundational models for histology, trained with OpenPath and large-scale institutional datasets, respectively. SPT performance is shown in Figure 5b, with additional results for SRH CNS benchmark in Appendix C Figure 8, and extended metrics with error bars are in Table 9.

For all patch encoders, SPT training significantly improves whole slide representations over pooling baselines. This improvement is the most significant for ImageNet and HIPT patch encoders. As expected, the performance boost is smaller for in-domain patch encoders, especially HiDisc, where slide discrimination was learned during patch training. SPT also achieves a large performance boost using PLIP patch encoders, as it bridges the domain gap between our benchmark and the pre-training dataset OpenPath which is collected on Twitter. As for the state-of-the-art UNI foundational patch encoder, we still observed a large performance boost achieving MCA near 90 points for suSPT. Overall, these results demonstrate that SPT can enhance whole-slide representation learning using encoders from different domains, thereby reducing the reliance on a specialized in-domain patch encoder. *SPT training time ranges from 6-8 GPU hours, while the patch encoder is computationally intensive and usually took over 80 GPU hours to train.*

Patch encoder	Architecture	Training dataset
ImageNet [53]	ResNet34	ImageNet
HIPT [5]	ViT-S/16	All of TCGA
SimCLR [12]	ResNet34	In-domain (see 4.2)
HiDisc [39]	ResNet34	In-domain (see 4.2)
PLIP [22]	ViT-B/32	OpenPath
UNI [7]	ViT-L/16	Mass-100K + GTEx

(a) **Patch encoders.** ImageNet is an off-the-shelf model trained on OOD data. HIPT is near-domain for H&E glioma. SimCLR and HiDisc patch encoders are trained with in-domain data. PLIP and UNI are SOTA foundational patch encoders.



(b) **SPT results on H&E Glioma.**

Figure 5: **SPT benchmarks with different patch encoders.** ssSPT and suSPT offer performance boosts with a wide range of patch encoders. ssSPT approaches supervised performance upperbound. Additional metrics with error bars are in Appendix C Table 9.

5.4 SPT improves the state-of-the-art foundation model

Most recently, foundation models have been developed for computational pathology, promising a general approach for WSI diagnostic tasks. UNI is a foundational patch encoder that achieves state-of-the-art results [7]. The UNI authors found that "ABMIL with UNI features outperforms many sophisticated MIL architectures" [7]. We benchmarked suSPT using UNI patch features, in comparison to ABMIL in Table 3. suSPT outperforms ABMIL in all benchmarks with UNI features, improving upon the previous state-of-the-art performance on these tasks. This showcases that SPT is complementary to the innovative medical foundation models for learning patch representations.

	SRH CNS			H&E Glioma			TCGA BRCA			TCGA NSCLC			TCGA RCC		
	MCA	F1	AUC	MCA	F1	AUC	MCA	F1	AUC	MCA	F1	AUC	MCA	F1	AUC
UNI + ABMIL	86.3	86.3	96.3	86.8	85.2	95.5	88.6	79.1	88.8	93.6	90.6	93.4	95.2	93.3	98.5
UNI + suSPT	**86.4**	**86.8**	**98.4**	**89.1**	**88.7**	**97.3**	**89.6**	**82.2**	**94.7**	**95.5**	**95.6**	**97.6**	**96.2**	**94.9**	**98.9**

Table 3: **suSPT improves state-of-the-art MIL results with UNI patch features.** Mean values are reported here, and standard deviations are in Appendix C Table 10.

5.5 Self-attention visualizations reveal tissue phenotypes in full gigapixel WSIs

Finally, we adopted the strategy in [46] and evaluated ssSPT qualitatively by visualizing self-attention on the full WSI. As illustrated in Figure 6, self-attention maps on H&E WSI can distinguish different tissue phenotypes such as blood, dense tumor, and necrosis. The CLS token attends to dense tumor regions instead of non-diagnostic regions such as blood and necrosis for slide representations. This observation is consistent with existing literature on self-supervised vision transformers, where attention maps can serve as unsupervised segmentation [46, 5]. To our knowledge, *our work is the first to generate self-supervised transformer-based attention maps on full gigapixel WSIs, instead of localized regions,* showing the strong representation capacity, flexibility, and efficiency of ssSPT. More H&E and SRH attention map visualizations are in Appendix C.6.

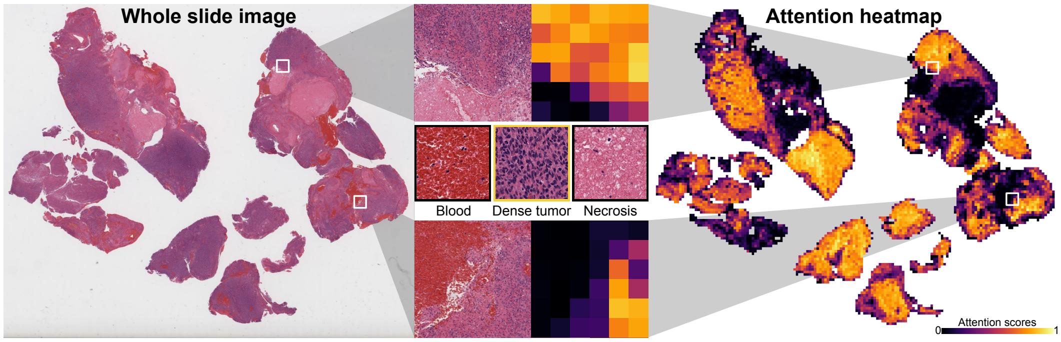


Figure 6: **Attention heatmap visualization.** Self-attention visualization demonstrates ssSPT's ability to distinguish between different tissue phenotypes such as blood, dense tumor, and necrosis. To our knowledge, these are the first self-supervised transformer-based attention maps on full gigapixel WSIs, instead of localized regions.

6 Conclusion

We present *Slide Pre-trained Transformers (SPT),* a self-supervised approach for whole slide representation learning, in a general, versatile, and lightweight framework. SPT complements existing foundation patch encoders by enhancing their capability to learn whole slide representation. SPT uses domain-informed, vision-language transformations for high-quality WSI view generation and self-supervised learning. SPT achieves superior performance on five computational pathology benchmarks across three clinical tasks, including histopathologic diagnosis, cancer subtyping, and molecular genetic prediction. Our work has shown SPT to be a suitable self-supervised strategy for training scalable and efficient foundation models at the whole slide level.

Limitations. Our experimentation was limited to two biomedical microscopy modalities and three major self-supervised learning paradigms. Future work includes validating SPT on additional datasets, including WSIs from different microscopy modalities, such as fluorescent microscopy, as well as different SSL paradigms such as generative SSL. Another limitation of SPT is hyperparameter tuning for different configurations of SPT transformations, which are domain and dataset-specific. An important future direction of our work is to train a largescale whole-slide foundation model using publicly available and institutional datasets.

Broader impact. This paper aims to advance the field of whole slide representation learning and computational pathology. Our work builds toward an automated diagnostic support system for biomedical microscopy and histopathology, paving the way for more accurate diagnoses and personalized treatment recommendations. Moreover, we strive to develop self-supervised learning frameworks in order to promote and maintain data privacy and ethical use of patient data within healthcare AI systems. Additionally, SSL is essential for developing medical foundation models. We hope that SPT can serve as a potential foundation model training strategy within computational pathology. While our experiments are in medical applications, there is a potential for impact on other similar gigapixel image modalities, such as geospatial and astronomical images.

References

[1] Xinliang Zhu, Jiawen Yao, Feiyun Zhu, and Junzhou Huang. Wsisa: Making survival prediction from whole slide histopathological images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7234–7242, 2017.

[2] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018.

[3] 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.

[4] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in neural information processing systems*, 34:2136–2147, 2021.

[5] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16144–16155, 2022.

[6] Syed Ashar Javed, Dinkar Juyal, Harshith Padigela, Amaro Taylor-Weiner, Limin Yu, and Aaditya Prakash. Additive mil: intrinsically interpretable multiple instance learning for pathology. *Advances in Neural Information Processing Systems*, 35:20689–20702, 2022.

[7] Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Andrew H Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 30(3):850–862, 2024.

[8] Bin Li, Yin Li, and Kevin W Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14318–14328, 2021.

[9] 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. *Nature medicine*, 25(8):1301–1309, 2019.

[10] Eugene Vorontsov, Alican Bozkurt, Adam Casson, George Shaikovski, Michal Zelechowski, Kristen Severson, Eric Zimmermann, James Hall, Neil Tenenholtz, Nicolo Fusi, et al. A foundation model for clinical-grade computational pathology and rare cancers detection. *Nature Medicine*, pages 1–12, 2024.

[11] 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 Workshops*, pages 4305–4314, 2023.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

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

[14] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2021.

[15] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[16] Ming Y Lu, Tiffany Y Chen, Drew FK Williamson, Melissa Zhao, Maha Shady, Jana Lipkova, and Faisal Mahmood. Ai-based pathology predicts origins for cancers of unknown primary. *Nature*, 594(7861):106–110, 2021.

[17] Todd C Hollon, Balaji Pandian, Arjun R Adapa, Esteban Urias, Akshay V Save, Siri Sahib S Khalsa, Daniel G Eichberg, Randy S D'Amico, Zia U Farooq, Spencer Lewis, et al. Near real-time intraoperative brain tumor diagnosis using stimulated raman histology and deep neural networks. *Nature medicine*, 26(1):52–58, 2020.

[18] 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. *Nature medicine*, 24(10):1559–1567, 2018.

[19] Todd Hollon, Cheng Jiang, Asadur Chowdury, Mustafa Nasir-Moin, Akhil Kondepudi, Alexander Aabedi, Arjun Adapa, Wajd Al-Holou, Jason Heth, Oren Sagher, et al. Artificial-intelligence-based molecular classification of diffuse gliomas using rapid, label-free optical imaging. *Nature Medicine*, 29(4):828–832, 2023.

[20] Richard J Chen, Ming Y Lu, Wei-Hung Weng, Tiffany Y Chen, Drew FK Williamson, Trevor Manz, Maha Shady, and Faisal Mahmood. Multimodal co-attention transformer for survival prediction in gigapixel whole slide images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4025, 2021.

[21] Chengkuan Chen, Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Andrew J Schaumberg, and Faisal Mahmood. Fast and scalable search of whole-slide images via self-supervised deep learning. *Nature Biomedical Engineering*, 6(12):1420–1434, 2022.

[22] Zhi Huang, Federico Bianchi, Mert Yuksekgonul, Thomas J Montine, and James Zou. A visual–language foundation model for pathology image analysis using medical twitter. *Nature medicine*, 29(9):2307–2316, 2023.

[23] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.

[24] Le Hou, Dimitris Samaras, Tahsin M Kurc, Yi Gao, James E Davis, and Joel H Saltz. Patch-based convolutional neural network for whole slide tissue image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2424–2433, 2016.

[25] Jiawen Yao, Xinliang Zhu, and Junzhou Huang. Deep multi-instance learning for survival prediction from whole slide images. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part I 22*, pages 496–504. Springer, 2019.

[26] Jiawen Yao, Xinliang Zhu, Jitendra Jonnagaddala, Nicholas Hawkins, and Junzhou Huang. Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks. *Medical Image Analysis*, 65:101789, 2020.

[27] Weijian Li, Viet-Duy Nguyen, Haofu Liao, Matt Wilder, Ke Cheng, and Jiebo Luo. Patch transformer for multi-tagging whole slide histopathology images. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part I 22*, pages 532–540. Springer, 2019.

[28] 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. *Medical image analysis*, 81:102559, 2022.

[29] Jiahui Yu, Tianyu Ma, Yu Fu, Hang Chen, Maode Lai, Cheng Zhuo, and Yingke Xu. Local-to-global spatial learning for whole-slide image representation and classification. *Computerized Medical Imaging and Graphics*, 107:102230, 2023.

[30] Tiancheng Lin, Zhimiao Yu, Hongyu Hu, Yi Xu, and Chang-Wen Chen. Interventional bag multi-instance learning on whole-slide pathological images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19830–19839, 2023.

[31] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[34] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[35] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[36] Ming Y Lu, Bowen Chen, Andrew Zhang, Drew FK Williamson, Richard J Chen, Tong Ding, Long Phi Le, Yung-Sung Chuang, and Faisal Mahmood. Visual language pretrained multiple instance zero-shot transfer for histopathology images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19764–19775, 2023.

[37] Sobhan Hemati, Shivam Kalra, Cameron Meaney, Morteza Babaie, Ali Ghodsi, and Hamid Tizhoosh. Cnn and deep sets for end-to-end whole slide image representation learning. In *Medical Imaging with Deep Learning*, pages 301–311. PMLR, 2021.

[38] Xiyue Wang, Jinxi Xiang, Jun Zhang, Sen Yang, Zhongyi Yang, Ming-Hui Wang, Jing Zhang, Wei Yang, Junzhou Huang, and Xiao Han. Scl-wc: Cross-slide contrastive learning for weakly-supervised whole-slide image classification. *Advances in neural information processing systems*, 35:18009–18021, 2022.

[39] Cheng Jiang, Xinhai Hou, Akhil Kondepudi, Asadur Chowdury, Christian W Freudiger, Daniel A Orringer, Honglak Lee, and Todd C Hollon. Hierarchical discriminative learning improves visual representations of biomedical microscopy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19798–19808, 2023.

[40] Thomas E Tavolara, Metin N Gurcan, and M Khalid Khan Niazi. Contrastive multiple instance learning: An unsupervised framework for learning slide-level representations of whole slide histopathology images without labels. *Cancers*, 14(23):5778, 2022.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[42] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

[43] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020.

[44] Ravid Shwartz Ziv and Yann LeCun. To compress or not to Compress-Self-Supervised learning and information theory: A review. *Entropy*, 26(3), March 2024.

[45] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[46] 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 IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[47] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[48] Christian W. Freudiger, Wei Min, Brian G. Saar, Sijia Lu, Gary R. Holtom, Chengwei He, Jason C. Tsai, Jing X. Kang, and X. Sunney Xie. Label-free biomedical imaging with high sensitivity by stimulated raman scattering microscopy. *Science*, 322(5909):1857–1861, 2008.

[49] Daniel A Orringer, Balaji Pandian, Yashar S Niknafs, Todd C Hollon, Julianne Boyle, Spencer Lewis, Mia Garrard, Shawn L Hervey-Jumper, Hugh JL Garton, Cormac O Maher, et al. Rapid intraoperative histology of unprocessed surgical specimens via fibre-laser-based stimulated raman scattering microscopy. *Nature biomedical engineering*, 1(2):0027, 2017.

[50] Cheng Jiang, Asadur Zaman Chowdury, Xinhai Hou, Akhil Kondepudi, Christian Freudiger, Kyle Stephen Conway, Sandra Camelo-Piragua, Daniel A Orringer, Honglak Lee, and Todd Hollon. OpenSRH: optimizing brain tumor surgery using intraoperative stimulated raman histology. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[51] Thomas Roetzer-Pejrimovsky, Anna-Christina Moser, Baran Atli, Clemens Christian Vogel, Petra A Mercea, Romana Prihoda, Ellen Gelpi, Christine Haberler, Romana Höftberger, Johannes A Hainfellner, et al. the digital brain tumour atlas, an open histopathology resource. *Scientific Data*, 9(1):55, 2022.

[52] David N Louis, Arie Perry, Pieter Wesseling, Daniel J Brat, Ian A Cree, Dominique Figarella-Branger, Cynthia Hawkins, HK Ng, Stefan M Pfister, Guido Reifenberger, et al. The 2021 who classification of tumors of the central nervous system: a summary. *Neuro-oncology*, 23(8):1231–1251, 2021.

[53] 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.

[54] Yang Li, Si Si, Gang Li, Cho-Jui Hsieh, and Samy Bengio. Learnable fourier features for multi-dimensional spatial positional encoding. *Advances in Neural Information Processing Systems*, 34:15816–15829, 2021.

[55] Marc Macenko, Marc Niethammer, James S Marron, David Borland, John T Woosley, Xiaojun Guan, Charles Schmitt, and Nancy E Thomas. A method for normalizing histology slides for quantitative analysis. In *2009 IEEE international symposium on biomedical imaging: from nano to macro*, pages 1107–1110. IEEE, 2009.

[56] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

A Data Details

The five benchmarks used in our experimentation are described in Table 4.

Benchmark	# clases	Tissue organ	Imaging modality	Task
SRH CNS	7	Brain	SRH	Histological classification
H&E Glioma	3	Brain	H&E stained	Molecular classification
TCGA BRCA	2	Breast	H&E stained	Tumor subtyping
TCGA NSCLC	2	Lung	H&E stained	Tumor subtyping
TCGA RCC	3	Kidney	H&E stained	Tumor subtyping

Table 4: **Benchmark overview.** Our benchmarks span four different organs and two different microscopy modalities.

A.1 SRH CNS benchmark

Our SRH benchmark includes common brain tumors and normal brain tissues. Our data collection and processing follows the protocol in [50], where each specimen is scanned with a commercially available stimulated Raman histology microscope manufactured by Invenio Imaging. Each specimen is diagnosed by a board-certified neuropathologist. The number of slides in each class is in Table 5.

Class	Training	Validation
HGG	407	132
LGG	210	107
Mening.	434	204
Metast.	236	114
Pit.	448	194
Schwan.	47	22
Normal	253	152
Unlabeled	2919	-

Table 5: **SRH dataset breakdown.** Number of slides in the training and validation set for supervised training and evaluation. HGG, high grade glioma, LGG, low grade glioma, mening., meningioma, metast., metastasis, pit., pituitary adenoma, schwan., schwannoma, normal, normal brain tissue.

A.2 H&E Glioma benchmark

Our H&E benchmark includes glioma specimens from the Cancer Genome Atlas Program (TCGA) and the Digital Brain Tumour Atlas (DBTA) [51]. These publicly available datasets feature H&E images collected in both the United States and Europe. Slides are divided into three classes based on molecular labels as defined by the World Health Organization [52], using IDH and 1p/19q co-deletion status included in both datasets. At training time, both DBTA and TCGA training set is used, and a separate, held-out validation set from the TCGA dataset is used for benchmarking. A detailed number of slides per class is in Table 6. Our H&E data processing pipeline follows [39]. Each WSI is divided into 300 × 300 patches, and blank / background regions are excluded. All patches used in training are stain-normalized using the Macenko algorithm [55].

Class	Molecular label	Training		Validation
		DBTA	TCGA	TCGA
Oligodendroglioma	IDH mutant, 1p/19q co-deleted	176	265	62
Astrocytoma	IDH mutant	157	360	88
Glioblastoma	IDH wildtype	619	732	191

Table 6: **H&E glioma dataset breakdown.** Number of slides from both datasets.

B Methods and Implementation

B.1 SPT transformation details

As described in section 3.3, we implement our SPT transformations with pre-computed embeddings and patch coordinates. Recall that each whole slide is represented as a $(\boldsymbol{x}, \boldsymbol{p})$ tuple, where $\boldsymbol{x} \in \mathbb{R}^{n \times d}$ is the embedding tensor and $\boldsymbol{p} \in \mathbb{Z}^{n \times 2}$ is a corresponding coordinate tensor for each patch in the WSI. Each transformation is implemented by indexing into the rows (corresponding to patches/tokens) of $\boldsymbol{x}$ and $\boldsymbol{p}$. A PyTorch style pseudocode is in Algorithm 1.

Splitting. Splitting has one hyperparameter: the ratio of tokens between two views. The ratio specifies the fraction of tokens in the first view and is used to compute the number of tokens in each view. Tokens are split randomly between views accordingly, and two disjoint views are returned.

Cropping. Cropping has two sets of hyperparameters: cropping area range, and cropping aspect ratio range. The cropping area and cropping aspect ratio are randomly chosen from their respective ranges. A random token in the view with coordinates (r_A, c_A) is selected to be an anchor as the crop center of the view. A coordinate range $(r_0, c_0), (r_1, c_1)$ is computed based on the area and aspect ratio:

$$\begin{aligned}
H &= \sqrt{\text{Area}/\text{aspect}} \\
W &= H \cdot \text{aspect} \\
r_0 &= r_A - H/2; & r_1 &= r_A + H/2 \\
c_0 &= c_A - W/2; & c_1 &= c_A + W/2,
\end{aligned}$$

where H, W are the height and width of the crop. Tokens in this coordinate range are included in the transformed view.

Masking. Masking has two hyperparameters: masking ratio range and max masking token limit. The masking ratio is randomly drawn from the masking ratio range, and it is used to compute the number of tokens m to keep in the augmented view. Max masking token limit is a cap of the number of tokens in each view, i.e. $m = \min(m, \text{max\_token\_lim})$. This parameter may be omitted (set to ∞) when training on datasets with smaller WSIs. Finally, we randomly select m tokens to remain in the augmented view.

Algorithm 1 SPT transformations in PyTorch style

```
# hyperparameters:
# - split_ratio: fraction of token in first view
# - (area_min, area_max): cropping area range
# - (aspect_min, aspect_max): cropping aspect ratio range
# - (mask_min, mask_max): masking ratio range
# - max_token_lim: max masking token limit


@torch.no_grad()
def transform(x, p):
    # inputs:
    # - x: patch embeddings (n d)
    # - p: patch coordinates (n 2)
    # output:
    # - (x1, p1): patch embs and coords for the first view
    # - (x2, p2): patch embs and coords for the second view

    (x1, p1), (x2, p2) = split(x, p)
    (x1, p1) = mask(crop(x1, p1))
    (x2, p2) = mask(crop(x2, p2))
    return (x1, p1), (x2, p2)


def split(x, p):
    # randomly permute index of all tokens
    rand_idx = torch.randperm(len(p))

    # determine index for each view
    view1_sz = int(split_ratio * len(p))
    idx1, idx2 = rand_idx[:view1_sz], rand_idx[view1_sz:]

    # filter tokens for each view
    return ((x[idx1,:], p[idx1,:]), (x[idx2,:], p[idx2,:]))


def crop(x, p):
    # randomly choose an anchor as center of the crop
    anchor_id = torch.randint(high=len(p))
    anchor = p[anchor_id, :]

    # randomly select crop area and aspect ratio
    crop_area = torch.randint(low=crop_min, high=crop_max)
    aspect = torch.FloatTensor(1).uniform_(aspect_min, aspect_max)

    # compute min and max coordinate of the crop
    height = torch.sqrt(areas / aspect)
    width = (height * aspect)
    r0, r1 = anchor[0] - height / 2, anchor[0] + height / 2
    c0, c1 = anchor[1] - width / 2, anchor[1] + width / 2

    # filter tokens
    idx = ((p[:,0] > r0) & (p[:,0] < r1) &
           (p[:,1] > c0) & (p[:,1] < c1))
    return x[idx,:], p[idx,:]


def mask(x, p):
    # randomly select a number of tokens to keep
    mask_ratio = torch.randint(low=mask_min, high=mask_max)
    size = mask_ratio * len(p)

    # optional: set a upper bound on number of tokens
    size = torch.minimum(size, max_token_lim)

    # randomly generate index and filter tokens
    idx = torch.randperm(len(p))[:sizes]
    return x[idx,:], p[idx,:]
```

B.2 Relative positional embedding

To work with WSIs of different sizes, it is non-trivial to use a fixed-size learnable position embedding with absolute distance [33]. We adopt the relative positional encoding introduced in [54], which utilizes the learnable Fourier feature, modulated with a multi-layer perceptron. This positional encoding uses the coordinates of the patch encoder as input and the output is added directly to the transformer with the same dimension. We visualize the positional embedding similarity in Figure 7.

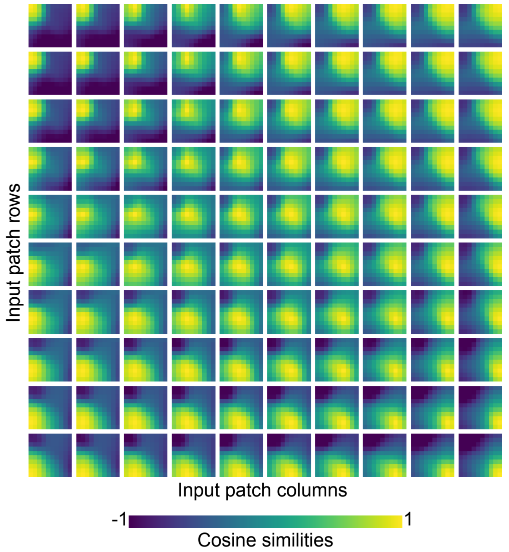


Figure 7: **Cosine similarity of learned positional embeddings.** Visualization is generated similarly as in [56]: each patch shows the cosine similarity between the position embedding of the token with the indicated row and column and the position embeddings of all other tokens.

C Extended Results

In this section, we present extended results presented in section 5.

C.1 Extended ssSPT evaluation with kNN classifier

We evaluate ssSPT and baselines using a kNN classifier in Table 1. Error bars from the table are reported in Table 7.

Benchmark	Method	MCA	F1	AUC
SRH CNS	Mean pooling	72.5	73.2	94.8
	Max pooling	73.5	75.6	95.0
	Giga-SSL	71.2 (0.9)	72.7 (0.5)	94.2 (1.4)
	ssSPT (Ours)	**82.3 (0.5)**	**82.3 (0.3)**	**94.7 (0.1)**
H&E Glioma	Mean pooling	68.3	67.9	87.9
	Max pooling	68.2	69.2	87.0
	Giga-SSL	71.6 (1.3)	71.5 (1.1)	89.1 (0.2)
	ssSPT (Ours)	**76.5 (0.5)**	**76.1 (0.4)**	**90.9 (0.4)**
TCGA BRCA	Mean pooling	58.1 (4.0)	27.4 (11.3)	74.7 (7.9)
	Max pooling	58.3 (3.8)	29.1 (10.1)	77.6 (10.1)
	HIPT	63.2 (6.1)	39.9 (13.7)	77.5 (4.2)
	Giga-SSL	70.4 (9.7)	53.9 (17.5)	**85.4 (6.7)**
	ssSPT (Ours)	**72.7 (4.9)**	**58.3 (6.8)**	80.4 (7.8)
TCGA NSCLC	Mean pooling	75.1 (3.7)	76.5 (3.4)	85.3 (2.9)
	Max pooling	79.3 (3.9)	80.7 (3.2)	88.4 (1.7)
	HIPT	80.6 (2.9)	80.1 (3.3)	88.9 (2.7)
	Giga-SSL	84.3 (3.0)	84.9 (2.6)	**92.3 (2.7)**
	ssSPT (Ours)	**86.1 (2.3)**	**86.1 (2.6)**	**92.3 (1.7)**
TCGA RCC	Mean pooling	84.2 (3.3)	86.3 (2.8)	96.6 (1.0)
	Max pooling	79.4 (4.5)	82.8 (3.7)	96.6 (1.7)
	HIPT	88.4 (3.2)	89.2 (3.0)	97.4 (1.6)
	Giga-SSL	90.0 (2.5)	89.3 (2.9)	97.5 (1.0)
	ssSPT (Ours)	**91.7 (2.1)**	**90.5 (2.7)**	**98.3 (1.3)**

Table 7: **Self-supervised benchmarks.** We use kNN classifier to evaluate ssSPT and baselines. Extended Table 1 with standard deviations reported in (parentheses).

C.2 Extended ssSPT and suSPT evaluation with linear classifier

We evaluate ssSPT, suSPT, and baselines using linear evaluation in Table 2. KNN evaluation results and error bars from the table are reported in Table 8.

		*k*NN			Linear evaluation		
Benchmark	Method	MCA	F1	AUC	MCA	F1	AUC
SRH CNS	*Self-supervised methods*						
	Giga-SSL	71.2 (0.9)	72.7 (0.5)	94.2 (1.4)	78.6 (1.8)	75.6 (2.1)	96.1 (0.3)
	ssSPT (Ours)	**82.3 (0.5)**	**82.3 (0.3)**	**94.7 (0.1)**	**85.4 (0.3)**	**83.2 (0.3)**	**97.6 (0.3)**
	Supervised methods						
	ABMIL	77.0 (0.9)	76.3 (0.1)	93.1 (0.6)	79.8 (0.4)	78.9 (1.0)	96.0 (0.4)
	CLAM	-	-	-	83.5 (1.3)	82.9 (1.2)	96.7 (0.3)
	TransMIL	84.8 (0.8)	**84.7 (0.3)**	92.2 (0.1)	84.4 (1.2)	84.0 (0.7)	84.0 (0.7)
	suSPT (Ours)	**85.0 (0.5)**	84.2 (0.6)	**92.8 (0.7)**	**85.6 (0.5)**	**84.4 (0.5)**	**97.2 (0.3)**
H&E Glioma	*Self-supervised methods*						
	Giga-SSL	71.6 (1.3)	71.5 (1.1)	89.1 (0.2)	76.5 (0.6)	73.9 (0.4)	90.7 (0.3)
	ssSPT (Ours)	**76.5 (0.5)**	**76.1 (0.4)**	**90.9 (0.4)**	**80.0 (2.4)**	**78.0 (2.2)**	**91.9 (0.8)**
	Supervised methods						
	ABMIL	75.6 (1.8)	75.7 (1.8)	**88.4 (0.4)**	75.0 (0.6)	74.5 (0.6)	90.2 (0.3)
	CLAM	-	-	-	78.9 (2.2)	77.7 (1.8)	88.5 (1.5)
	TransMIL	79.1 (2.5)	78.8 (2.7)	86.5 (0.3)	80.0 (1.4)	79.4 (1.8)	91.9 (1.1)
	suSPT (Ours)	**80.5 (1.6)**	**79.9 (1.4)**	88.3 (0.5)	**80.7 (2.4)**	**80.3 (1.9)**	**93.7 (0.3)**
TCGA BRCA	*Self-supervised methods*						
	Giga-SSL	70.4 (9.7)	53.9 (17.5)	85.4 (6.7)	81.7 (6.5)	62.6 (9.2)	**91.2 (5.4)**
	ssSPT (Ours)	**72.7 (4.9)**	**58.3 (6.8)**	**80.4 (7.8)**	**83.0 (7.3)**	**67.4 (9.9)**	89.4 (6.3)
	Supervised methods						
	ABMIL	71.7 (9.7)	56.0 (17.5)	84.9 (9.1)	78.6 (9.1)	62.8 (14.3)	86.1 (7.5)
	CLAM	-	-	-	76.8 (8.9)	59.3 (13.8)	85.8 (6.3)
	DSMIL	-	-	-	77.0 (6.5)	58.0 (7.8)	83.8 (7.0)
	TransMIL	**79.0 (8.1)**	**67.7 (11.2)**	84.7 (6.8)	80.8 (8.1)	69.4 (11.3)	90.0 (5.5)
	HIPT	-	-	-	75.8 (8.2)	60.2 (11.7)	87.4 (5.7)
	suSPT (Ours)	78.2 (9.4)	66.1 (17.1)	**86.1 (6.1)**	**84.2 (5.3)**	**70.4 (8.2)**	**90.5 (4.8)**
TCGA NSCLC	*Self-supervised methods*						
	Giga-SSL	84.3 (3.0)	84.9 (2.6)	**92.3 (2.7)**	87.5 (2.3)	87.9 (2.2)	93.7 (2.2)
	ssSPT (Ours)	**86.1 (2.3)**	**86.1 (2.6)**	**92.3 (1.7)**	**88.0 (3.1)**	**88.0 (3.2)**	**94.8 (2.1)**
	Supervised methods						
	ABMIL	80.1 (4.0)	79.6 (4.5)	89.7 (2.1)	86.8 (2.4)	86.7 (2.3)	93.7 (1.2)
	CLAM	-	-	-	84.3 (3.4)	83.6 (4.1)	92.8 (2.0)
	DSMIL	-	-	-	84.9 (3.8)	84.0 (4.6)	92.0 (2.3)
	TransMIL	88.8 (2.4)	88.9 (2.5)	**94.2 (2.6)**	87.4 (2.8)	87.5 (2.8)	95.3 (1.5)
	HIPT	-	-	-	88.2 (2.5)	88.2 (2.5)	95.2 (2.0)
	suSPT (Ours)	**90.3 (1.8)**	**90.4 (2.0)**	94.1 (2.6)	**90.3 (2.1)**	**90.4 (2.2)**	**95.7 (2.4)**
TCGA RCC	*Self-supervised methods*						
	Giga-SSL	90.0 (2.5)	89.3 (2.9)	97.5 (1.0)	89.9 (2.7)	87.8 (2.6)	97.7 (0.9)
	ssSPT (Ours)	**91.7 (2.1)**	**90.5 (2.7)**	**98.3 (1.3)**	**93.4 (1.7)**	**91.1 (2.1)**	**98.7 (1.1)**
	Supervised methods						
	ABMIL	89.5 (6.0)	90.6 (5.2)	**97.5 (2.2)**	89.9 (2.6)	89.4 (3.2)	97.8 (1.1)
	CLAM	-	-	-	85.6 (8.0)	83.6 (7.0)	97.3 (1.7)
	DSMIL	-	-	-	90.1 (3.7)	87.1 (4.4)	97.1 (1.5)
	TransMIL	90.6 (2.3)	90.0 (2.5)	96.6 (1.3)	87.4 (2.8)	87.5 (2.8)	95.3 (1.5)
	HIPT	-	-	-	92.0 (2.0)	**92.1 (2.6)**	98.0 (1.2)
	suSPT (Ours)	**92.1 (3.0)**	**91.1 (3.5)**	96.3 (2.1)	**92.3 (2.9)**	91.0 (3.5)	**98.6 (0.7)**

Table 8: **SPT benchmarks.** Extended Table 2 with KNN evaluation results and standard deviations. Standard deviations are reported in (parentheses).

C.3 ssSPT results with different patch encoders

In Figure 5b, we compared the ssSPT's performance to the pooling baseline and suSPT performance upper bound with the H&E glioma benchmark. Here, we present the same experiment with the SRH CNS benchmark in Figure 8. In the SRH benchmark, WSI representations improved dramatically in OOD and foundation patch encoders, which are not trained on SRH data. After training with SPT, as demonstrated by a maximum 18.6% boost in MCA for ImageNet. We also observe the expected performance boost for in-domain patch encoder SimCLR and HiDisc. Similar to the H&E glioma benchmark, ssSPT performance approaches suSPT upperbound on all patch encoders, within 4 points in MCA for SRH CNS benchmarks.

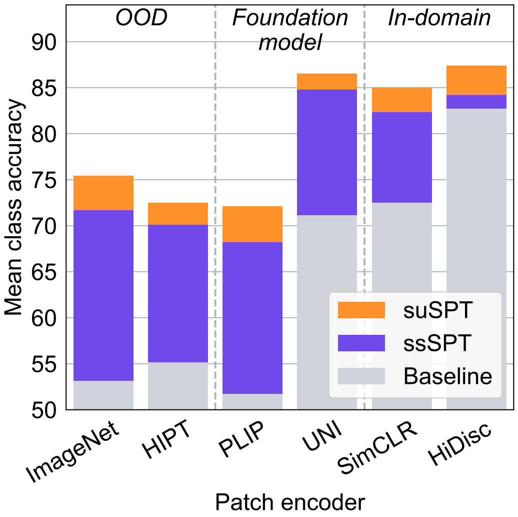


Figure 8: **SPT results on SRH CNS with a wide range of patch encoders.** Publicly available foundation patch encoders are trained using H&E data and are out-of-distribution (OOD) for SRH. In-domain patch encoders are trained using SRH data.

In addition, full metrics for each patch encoder for both SRH CNS and H&E glioma are in Table 9.

		SRH			H&E Gliomas		
Patch Encoder	Method	MCA	F1	AUC	MCA	F1	AUC
ImageNet	Pooling	53.1	53.0	88.1	59.8	74.2	81.7
	ssSPT	71.7 (0.7)	73.2 (0.4)	93.7 (0.3)	75.3 (2.5)	74.0 (2.5)	89.9 (0.9)
	suSPT	75.4 (1.4)	76.1 (0.8)	92.3 (0.5)	76.4 (1.1)	76.4 (1.2)	88.8 (0.6)
HiDisc	Pooling	82.7	82.9	95.9	74.9	74.2	90.4
	ssSPT	84.2 (1.4)	83.9 (1.2)	95.6 (0.4)	77.5 (1.3)	77.1 (1.3)	91.2 (0.3)
	suSPT	87.4 (0.8)	87.5 (0.7)	93.3 (0.3)	83.6 (0.3)	82.4 (0.2)	88.3 (0.3)
HIPT	Pooling	55.1	56.4	86.6	54.7	55.0	77.7
	ssSPT	70.1 (0.4)	71.9 (0.5)	92.2 (0.4)	74.9 (1.7)	73.9 (1.7)	89.2 (0.2)
	suSPT	72.5 (1.1)	73.7 (0.9)	90.9 (0.3)	76.6 (1.8)	76.1 (1.6)	90.6 (0.7)
PLIP	Pooling	51.7	51.9	86.8	66.8	66.3	88.0
	ssSPT	68.2 (0.4)	69.7 (0.4)	92.0 (0.4)	76.2 (2.0)	75.5 (2.2)	91.4 (0.4)
	suSPT	72.1 (1.4)	72.6 (1.2)	91.3 (0.4)	82.6 (0.7)	81.5 (0.9)	92.5 (0.2)
UNI	Pooling	71.1	73.4	93.6	80.3	78.3	93.5
	ssSPT	84.8 (1.7)	85.7 (1.5)	96.2 (0.3)	82.8 (2.1)	82.3 (2.2)	94.2 (0.5)
	suSPT	86.5 (0.9)	86.9 (0.6)	92.4 (0.5)	89.1 (0.9)	88.8 (0.7)	92.5 (0.5)

Table 9: **SPT results with different patch encoder.** Metrics reported for results in Figure 5b. Standard deviations are in (parentheses).

C.4 suSPT benchmarks with UNI patch features

We evaluated suSPT with UNI patch features in Table 3. Error bars are reported in Table 10.

		*k*NN			Linear evaluation		
Benchmark	Method	MCA	F1	AUC	MCA	F1	AUC
SRH CNS	ABMIL	84.1 (0.9)	84.7 (0.8)	**95.4 (0.2)**	86.3 (0.9)	86.3 (0.7)	96.3 (0.0)
	suSPT	**86.5 (0.9)**	**86.9 (0.6)**	92.4 (0.5)	**86.4 (0.8)**	**86.8 (0.5)**	**98.4 (0.3)**
H&E Glioma	ABMIL	86.4 (1.5)	85.6 (1.4)	**94.7 (1.1)**	86.8 (0.6)	85.2 (0.8)	95.5 (0.6)
	suSPT	**89.1 (0.9)**	**88.8 (0.7)**	92.5 (0.5)	**89.1 (1.0)**	**88.7 (0.7)**	**97.3 (0.3)**
TCGA BRCA	ABMIL	85.7 (6.8)	78.8 (9.5)	**91.7 (4.7)**	88.6 (6.3)	79.1 (6.5)	88.8 (7.6)
	suSPT	**90.1 (6.4)**	**84.2 (9.8)**	90.3 (5.8)	**89.6 (6.3)**	**82.2 (9.2)**	**94.7 (3.5)**
TCGA NSCLC	ABMIL	**95.6 (1.2)**	**95.7 (1.1)**	**97.3 (2.0)**	93.6 (1.7)	90.6 (2.0)	93.4 (2.9)
	suSPT	95.4 (2.4)	95.5 (2.3)	95.9 (1.8)	**95.5 (2.2)**	**95.6 (2.2)**	**97.6 (2.1)**
TCGA RCC	ABMIL	95.4 (1.3)	93.4 (2.4)	**98.3 (0.7)**	95.2 (1.1)	93.3 (2.0)	98.5 (0.9)
	suSPT	**96.3 (2.6)**	**94.9 (2.9)**	97.2 (1.7)	**96.2 (2.8)**	**94.9 (3.1)**	**98.9 (0.6)**

Table 10: **suSPT benchmark with UNI features.** Extended results for Table 3 with standard deviations reported in (parentheses).

C.5 Ablation stuides

In this section, we present ablation studies over SSL paradigms, SPT transformation choices, and SPT transformation parameters.

C.5.1 SSL paradigms

We examine the effect of different SSL paradigms on ssSPT in table 11. VICReg performs the best on the SRH CNS benchmark, BYOL performs the best on the TCGA RCC benchmark, while SimCLR performs the best on H&E glioma, TCGA BRCA, and TCGA NSCLC. These results demonstrate that the flexibility of SPT allows different algorithms to accommodate different training dynamics of different diagnostic tasks and image modalities.

		*k*NN			Linear evaluation		
Benchmark	Algorithm	MCA	F1	AUC	MCA	F1	AUC
SRH CNS	ssSPT-SimCLR	77.3 (0.6)	77.6 (0.3)	94.0 (0.4)	82.7 (0.7)	81.8 (0.5)	97.5 (0.3)
	ssSPT-VICReg	**82.3 (0.5)**	**82.3 (0.3)**	**94.7 (0.1)**	**85.4 (0.3)**	**83.2 (0.3)**	**97.6 (0.3)**
	ssSPT-BYOL	79.4 (0.4)	78.7 (1.2)	94.3 (0.4)	83.4 (0.5)	81.0 (1.3)	96.9 (0.0)
H&E Glioma	ssSPT-SimCLR	**76.5 (0.5)**	**76.1 (0.4)**	**90.9 (0.4)**	**80.0 (2.4)**	**78.0 (2.2)**	91.9 (0.8)
	ssSPT-VICReg	72.8 (0.7)	72.4 (0.9)	87.9 (0.7)	74.5 (0.8)	73.1 (0.8)	90.9 (0.5)
	ssSPT-BYOL	74.6 (0.9)	73.8 (0.9)	89.0 (0.5)	77.8 (0.6)	76.1 (0.8)	**92.2 (0.0)**
TCGA BRCA	ssSPT-SimCLR	**72.7 (4.9)**	**58.3 (6.8)**	80.4 (7.8)	83.0 (7.3)	**67.4 (9.9)**	89.4 (6.3)
	ssSPT-VICReg	69.2 (7.9)	51.0 (15.2)	77.4 (9.3)	78.6 (5.4)	59.3 (7.5)	85.3 (7.7)
	ssSPT-BYOL	70.0 (8.3)	52.6 (15.8)	**81.1 (9.0)**	**83.6 (5.4)**	66.7 (6.7)	**90.4 (6.6)**
TCGA NSCLC	ssSPT-SimCLR	**86.1 (2.3)**	**86.1 (2.6)**	**92.3 (1.7)**	**88.0 (3.1)**	**88.0 (3.2)**	**94.8 (2.1)**
	ssSPT-VICReg	80.2 (3.4)	80.7 (3.0)	86.0 (3.4)	87.6 (2.2)	87.9 (2.0)	93.8 (1.6)
	ssSPT-BYOL	81.4 (2.4)	81.8 (2.7)	87.2 (2.0)	86.0 (2.8)	86.2 (3.1)	92.7 (2.2)
TCGA RCC	ssSPT-SimCLR	90.7 (2.9)	90.4 (2.7)	**98.3 (1.0)**	91.7 (2.1)	89.4 (3.2)	98.2 (0.7)
	ssSPT-VICReg	87.2 (3.9)	87.2 (3.1)	96.0 (1.6)	88.6 (3.3)	86.6 (3.2)	96.9 (0.9)
	ssSPT-BYOL	**91.7 (2.1)**	**90.5 (2.7)**	**98.3 (1.3)**	**93.4 (1.7)**	**91.1 (2.1)**	**98.7 (1.1)**

Table 11: **SPT benchmark with different SSL paradigms.** We compare different SPT training algorithms. Standard deviations are reported in (parentheses).

C.5.2 SPT transformation choices

Table 12 reports the performance of individual and combinations of slide-level transformations. We leave out splitting alone since it does not reduce the total number of patch tokens for both views, making it prohibitive to train. For the masking and cropping alone, we show that cropping performs better than masking for both SRH CNS and H&E Glioma. Combining cropping and masking shows similar performance with cropping alone, but is more efficient to train due to the reduced number of tokens per view. Combining each transformation with splitting improves or maintains performance, showing the benefit of further decreasing mutual information (MI) between views. Overall, cropping is the most important transformation, because capturing regional heterogeneity is the most effective and challenging pre-text task.

			SRH CNS		H&E Glioma	
Splitting	Cropping	Masking	MCA	F1	MCA	F1
		X	75.4 (0.9)	76.0 (0.7)	69.9 (0.5)	69.7 (0.5)
	X		81.3 (1.5)	81.5 (1.5)	74.3 (0.2)	74.4 (0.2)
X		X	76.1 (0.3)	76.3 (0.8)	69.2 (1.5)	69.3 (1.5)
X	X		82.1 (1.4)	82.2 (1.2)	73.6 (0.6)	73.8 (0.4)
	X	X	81.2 (0.5)	81.8 (0.5)	75.5 (2.1)	75.4 (2.2)
X	X	X	**82.3 (0.5)**	**82.3 (0.3)**	**76.5 (0.5)**	**76.1 (0.4)**

Table 12: **Ablation over transformations.** X stands for transformations used. Standard deviations reported in (parentheses).

C.5.3 SPT transformation parameters

We examine the relationship between mutual information (MI) and model performance by combining and adjusting SPT transformation parameters in the H&E glioma benchmark.

Token limit. We studied the effect of the number of tokens, with splitting + cropping + masking and only masking. A higher token limit results in a higher likelihood of overlapping area coverage between views. The results are in Figure 9a. As expected, MI levels can be adjusted to optimize model performance. Previous studies have concluded that fewer patch tokens resulted in better model performance [11], but only in the case of using masking transformation alone.

Cropping size. We varied cropping size range in Figure 9b. Cropping also allows us to adjust the amount of MI between views, where a larger crop size leads to a higher probability of overlapping regions between views. Our ablation study shows that for SimCLR training on our H&E glioma benchmark, $[100, 400]$ is the optimal cropping size range, achieving a MCA of 76.5. These results correspond to the hypothesis in [43], where an optimal amount of MI between views achieves the best performance.

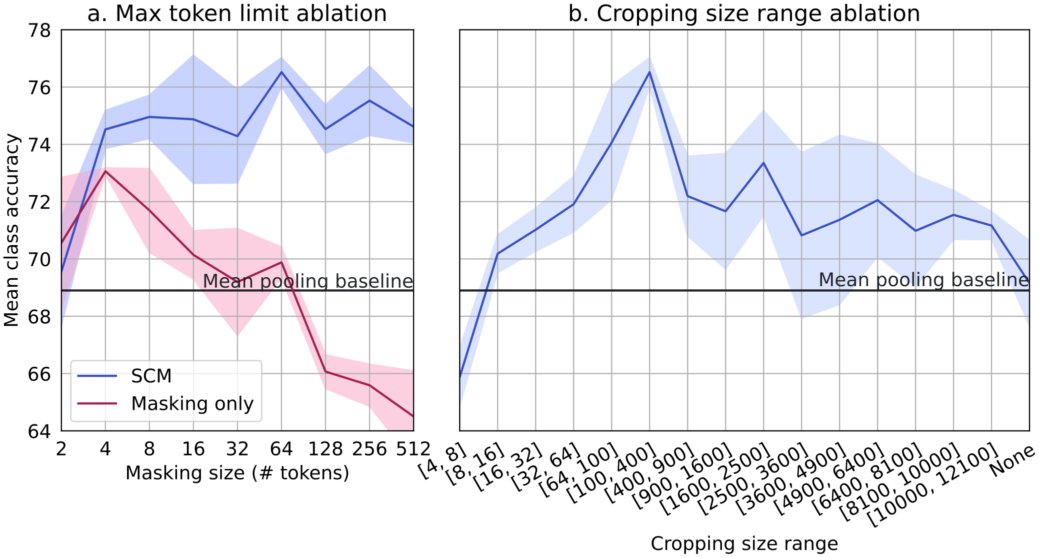


Figure 9: **SPT transformation parameter ablation for H&E glioma molecular classification. a) Max token limit ablation.** SCM, splitting-cropping-masking transformation. **b) Cropping size range ablation.** Cropping size range [min, max cropping size] is on the *x*-axis. None represent no cropping applied. The shaded area represents the standard deviation across three different random seeds.

C.5.4 SPT model size

We performed additional ablation studies with different SPT model sizes, and the results are reported in Table 13. The six- and two-layer transformers have the best performance for H&E glioma and SRH CNS, respectively. Smaller models performed better for SRH CNS, likely because of a smaller slide size and a relatively more uniform image. All models ranging from two to eight-layer transformers outperformed previous best self-supervised models.

		SRH			H&E Glioma		
# Layers	# Parameters	MCA	F1	AUC	MCA	F1	AUC
2	6.7 M	**83.5 (0.1)**	**83.8 (0.3)**	**95.4 (0.4)**	74.9 (0.9)	74.8 (1.0)	91.6 (0.5)
4	13 M	82.2 (0.9)	83.2 (1.6)	**95.4 (0.2)**	75.5 (0.8)	75.3 (0.5)	91.1 (0.4)
6	19.3 M	82.3 (0.5)	82.3 (0.3)	94.7 (0.1)	**76.5 (0.5)**	**76.1 (0.4)**	90.9 (0.4)
8	25.6 M	81.4 (0.2)	82.5 (0.3)	95.2 (0.2)	75.2 (1.2)	75.2 (1.1)	**91.7 (0.9)**

Table 13: **SPT ablation studies on model size.** Standard deviations are in (parentheses).

C.6 Extended attention heatmaps

We present extended attention heatmap visualization for SRH and H&E in Figures 10 and 11, respectively. Recall these attentions are the self-attention of the CLS token from the last layer of the whole slide encoder. On both SRH and H&E heatmaps, we can see that the whole slide encoder attends to clinically significant regions such as dense tumors, and regions with tumor infiltration, and avoids non-diagnostic regions such as blood, tissue processing artifacts, and blood vessels.

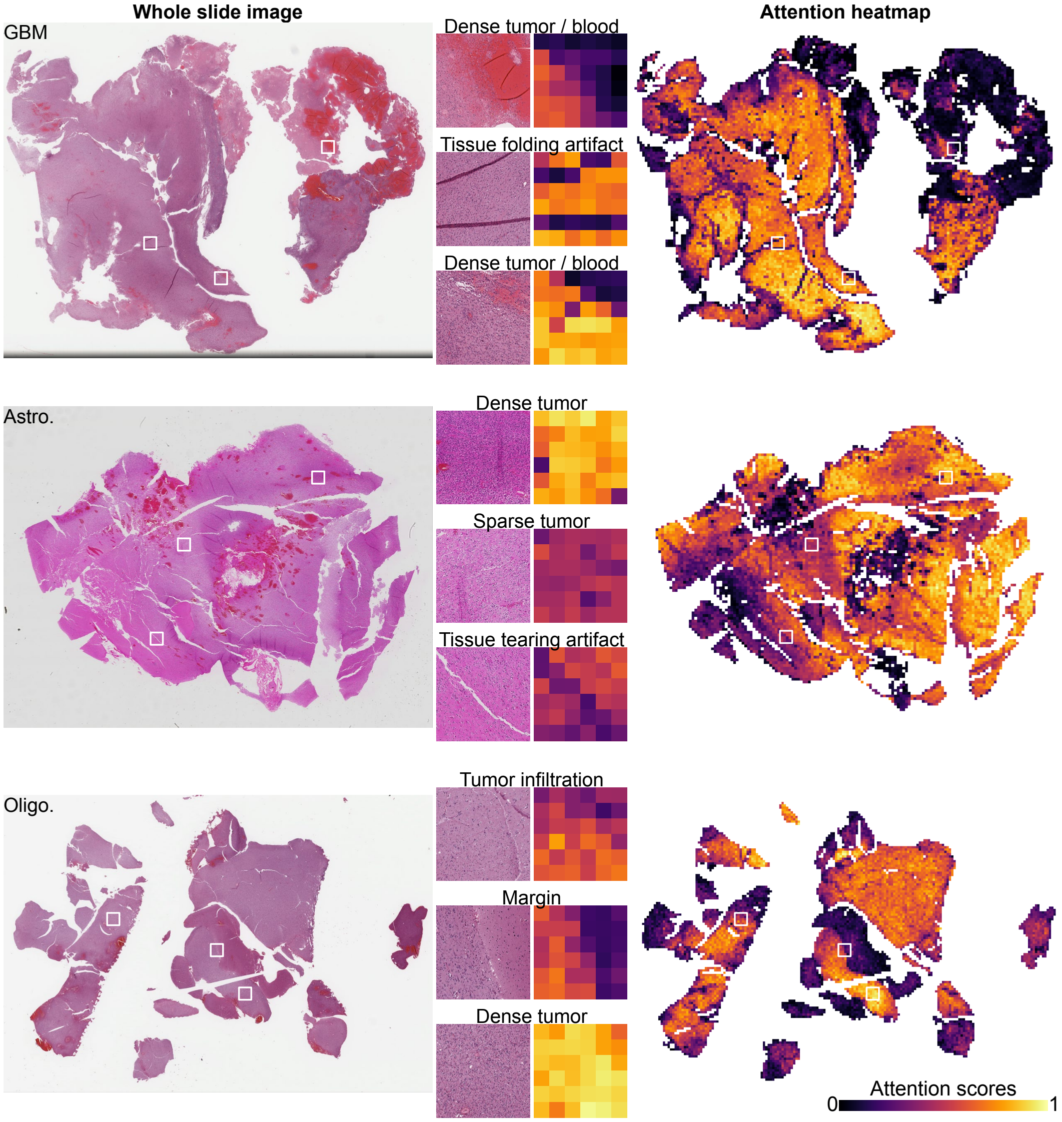


Figure 10: **Extended attention heatmap for H&E whole slides.** The attention map shows the ssSPT-trained whole slide transformer can differentiate between different morphologies like dense tumors, blood, and different artifacts on the gigapixel WSIs. Visualization shows high attention to varying degrees of tumor infiltration, and low attention to low cellularity, blood, necrotic regions, and tissue processing artifacts. Oligo, oligodendroglioma, Astro, astrocytoma, GBM, glioblastoma.

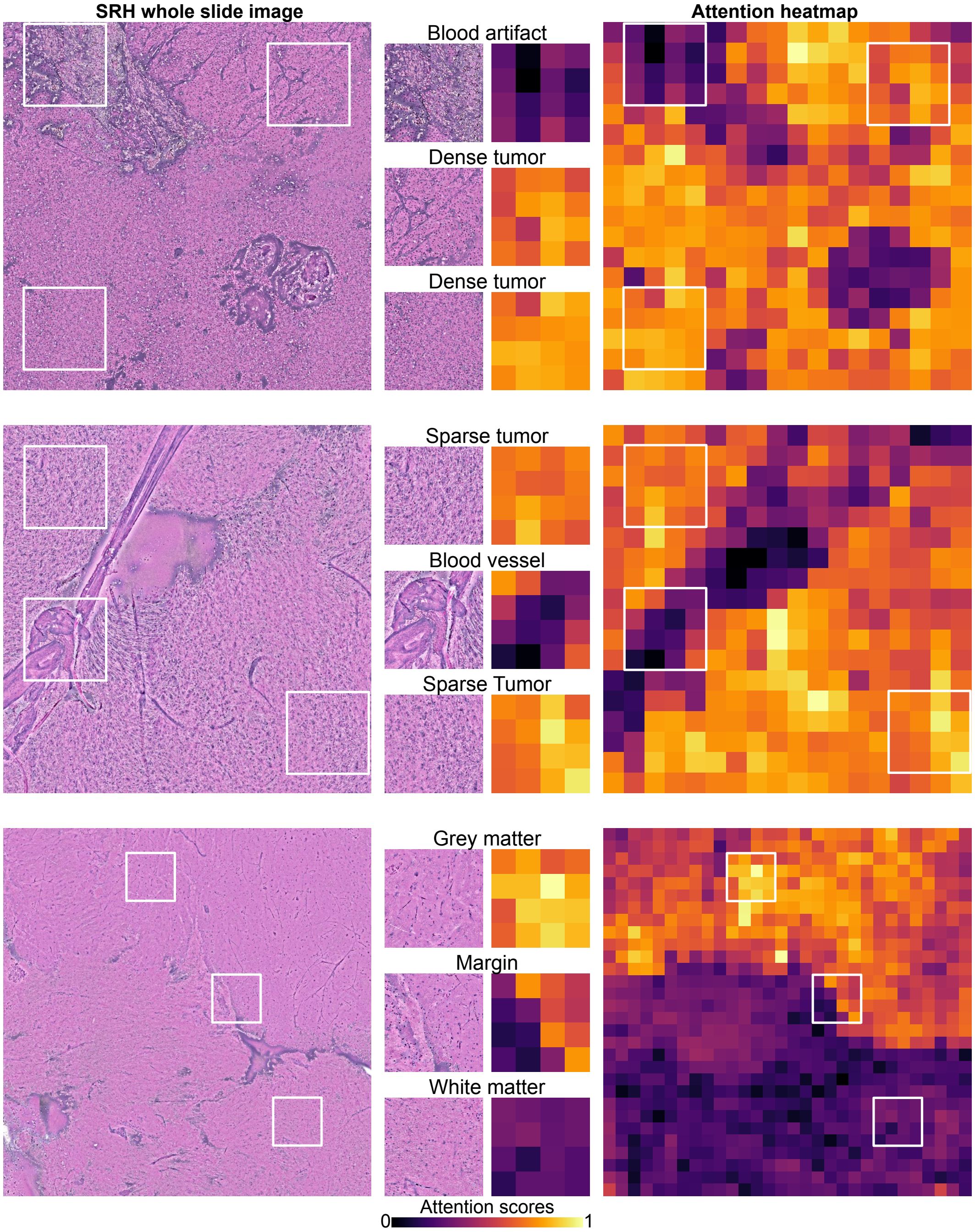


Figure 11: **Extended attention heatmap for SRH whole slides.** On this SRH WSI, relatively small compared to H&E, the model shows a strong capacity for unsupervised segmentation of histological features. Visualization shows high attention values to tumor regions, and low attention values to non-diagnostic regions such as blood vessels, laser noise, and empty space.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The scope is introduced in the abstract and Section 1 and the contributions are enumerated at the end of Section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are addressed in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our work does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimentation details are in Section 4, with training details are in Section 4.2, the evaluation protocol is in Section 4.3. Pseudocode for SPT transformations is provided in Algorithm 1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Open source code, model weights, and feature embeddings of publicly available benchmarks will be released on GitHub with camera-ready version.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The experimentation details are in Section 4, with dataset split details in 4.1, hyperparameter, model component details in Section 4.2,

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experiments are reported with error bars included in Appendix C. Detailed evaluation protocol describes error bars in Section 4.3. Error bars are computed over three random seeds for SRH CNS and H&E glioma benchmarks. Ten-fold cross validation is performed for TCGA BRCA, TCGA NSCLC, and TCGA RCC benchmarks, following previously established protocol in [5].

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computing resources used for our experimentation are described in Section 4.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have followed the NeurIPS Code of Ethics during this research.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impact section is described in Section 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This research does not involve data or models that have a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Existing code, data, and models are cited properly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Open source code, data, and models will be documented properly upon release in the camera-ready version.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The study does not involve crowdsourcing. The study used only use previously collected images and data from human tissues.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: This study has been approved by the Institutional Review Board (IRB # [redacted for double-blind review]). Informed consents were collected prior to imaging. Details are described in section 4.1.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.