Affinitention Nets: Kernel Perspective on Attention Architectures for Set Classification with Applications to Medical Text and Images

David Dov
david.dov@duke.edu
Duke University
Durham, NC, USA

Serge Assaad
serge.assaad@duke.edu
Duke University
Durham, NC, USA

Shijing Si
shijing.si@duke.edu
Duke University
Durham, NC, USA

Rui Wang
rui.wang16@duke.edu
Duke University
Durham, NC, USA

Hongteng Xu
hongtengxu@ruc.edu.cn
Gaoling School of Artificial
Intelligence, Renmin University of
China
Beijing Key Laboratory of Big Data
Management and Analysis Methods
Beijing, China

Shahar Ziv Kovalsky
shaharko@unc.edu
University of North Carolina
Chapel Hill, NC, USA

Jonathan Bell
jonathan.bell@duke.edu
Duke University Medical Center
Durham, NC, USA

Danielle Elliott Range
danielle.range@duke.edu
Duke University Medical Center
Durham, NC, USA

Jonathan Cohen
jonathanco@clalit.org.il
Kaplan Medical Center
Rehovot, Israel

Ricardo Henao
ricardo.henao@duke.edu
Duke University
Durham, NC, USA

Lawrence Carin
lcarin@duke.edu
Duke University
Durham, NC, USA

ABSTRACT

Set classification is the task of predicting a single label from a set comprising multiple instances. The examples we consider are pathology slides represented by sets of patches and medical text data represented by sets of word embeddings. State-of-the-art methods, such as the transformer network, typically use attention mechanisms to learn representations of set data, by modeling interactions between instances of the set. These methods, however, have complex heuristic architectures comprising multiple heads and layers. The complexity of attention architectures hampers their training when only a small number of labeled sets is available, as is often the case in medical applications. To address this problem, we present a kernel-based representation learning framework that links learning affinity kernels to learning representations from attention architectures. We show that learning a combination of the sum and the product of kernels is equivalent to learning representations from

multi-head multi-layer attention architectures. From our framework, we devise a simplified attention architecture which we term *affinitention* (affinity-attention) nets. We demonstrate the application of affinitention nets to the classification of the Set-Cifar10 dataset, thyroid malignancy prediction from pathology slides, as well as patient text-message triage. We show that affinitention nets provide competitive results compared to heuristic attention architectures and outperform other competing methods.

CCS CONCEPTS

• **Applied computing** → **Health informatics**; • **Computing methodologies** → **Machine learning algorithms**.

KEYWORDS

attention, transformer, set classification, multiple instance learning, whole slide images, medical text

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM CHIL '21, April 08–10, 2021,

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8359-2/21/04...\$15.00

<https://doi.org/10.1145/3450439.3451856>

ACM Reference Format:

David Dov, Serge Assaad, Shijing Si, Rui Wang, Hongteng Xu, Shahar Ziv Kovalsky, Jonathan Bell, Danielle Elliott Range, Jonathan Cohen, Ricardo Henao, and Lawrence Carin. 2021. Affinitention Nets: Kernel Perspective on Attention Architectures for Set Classification with Applications to Medical Text and Images. In *ACM CHIL '21: ACM Conference on Health, Inference, and Learning, April 08–10, 2021*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3450439.3451856>

1 INTRODUCTION

In many machine learning applications, data such as text and images can be represented by sets of multiple instances. The question we address here is how to learn representations of sets via attention mechanisms for set classification, where the goal is to predict a single set-level label. The examples we consider are (i) the prediction of thyroid cancer from pathology whole slide images (WSI) that, due to their large size, are divided into sets of multiple image regions (patches); and (ii) patient text message triage, where each message is represented by a set of multiple tokens (words).

In the medical-imaging literature, set classification is often referred to as multiple-instance learning (MIL) [17, 21, 33]. In MIL, the set is considered a bag of multiple instances, and the classical assumption is that a single positive instance in a bag triggers a positive bag-level label *independently* of all other instances. Classical MIL algorithms include average-pooling, *noisy-or* [48] and *noisy-and* [21]. Due to the independence assumption, these methods only loosely consider complex relations between instances in the bag. Other set classification and MIL approaches that study the dependence between instances include, for example, [17], which uses an attention mechanism to weigh the importance of the instances.

A typical assumption in set classification is that there is no natural or predefined order between instances of the set. Accordingly, the learned representation of a set should be invariant to permutations, *i.e.*, the order of its instances. For example, in the case of pathology WSIs, the spatial locations of diagnostic image regions in the WSIs are irrelevant to diagnosis. Therefore, the prediction of malignancy from the set of image regions should be invariant to their order in the set. In text data, however, such an order does exist between words of a message or a sentence. Indeed, widely used natural language processing (NLP) methods such as Text-CNN [18] and Long Short-Term Memory (LSTM) [45, 49] networks are based on learning such positional relations within the messages. Yet, state-of-the-art methods such as Bidirectional Encoder Representations from Transformers (BERT) [7, 34, 40, 44], encode location information into token (word) embeddings, which can be understood as a preprocessing step. Then, these methods use attention to model complex relations between the token embeddings, providing representations that are invariant to the location of tokens in the set. This makes transformer-based architectures particularly suitable for set classification tasks. Indeed, the transformer was adapted for set classification by Lee *et al.* in [25].

The main disadvantage of transformer-based attention methods is their complex architecture comprising multiple attention heads and layers, thus resulting in specifications with hundreds of millions of parameters. This hampers their use especially for medical tasks where the number of labeled training examples (sets) is limited. Moreover, these architectures are largely based on heuristics and little effort has been put into understanding the representations learned by these methods. A step toward the design of principled attention architectures has been considered in [36], in which an architecture motivated by the message-passing algorithm was proposed. Their analysis reveals a connection between the attention mechanism and affinities between the instances in the set. However, their analysis is limited to single-head, single-layer architectures.

In this paper, we propose a kernel-based representation learning framework for set classification. Our framework elucidates a connection between representations of instances of the set obtained from attention-based architectures and affinity kernels constructed between the instances. We show that learning the product and the sum of multiple parametric affinity kernels is equivalent to learning multi-layer and multi-head attention architectures, respectively. Our analysis reveals a trade-off between learning affinity kernels that seemingly capture complex relations between instances and obtaining over-smoothed representations. With this trade-off in mind, we devise the “affinitention” (affinity-attention) architecture, and show that it leads to a permutation-invariant set-level representation. The affinitention architecture is equivalent to BERT (without the fully connected layers that follow the attention heads), but enjoys theoretical connections to kernel learning. We apply affinitention nets to patient message triage, Set-CIFAR10 dataset, as well as to thyroid malignancy prediction from biopsy images. On all tasks, affinitention nets achieve competitive performance compared to heuristic architectures with complex attention mechanisms and outperform other competing baselines.

2 PROBLEM FORMULATION

Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ be a matrix consisting of a set of N vectors (instances) of dimension d , and let $x_i \in \mathbb{R}^d$ be the i -th row of $\mathbf{X}$, *i.e.*, the i -th instance. In the context of text data, $\mathbf{X}$ represents a set of token embeddings of a sentence or message. In pathology slides, $\mathbf{X}$ represents a set of patches taken from a slide, and x_i is the feature vector for the i -th patch, *e.g.*, obtained from a convolutional neural network [29]. Our goal is to learn a representation $f_\theta(\mathbf{X})$ such that $f_\theta(\cdot) : \mathbb{R}^{N \times d} \rightarrow \mathbb{R}^{N \times d'}$, where θ are learnable parameters and d' is the dimension of the representation. In turn, we use the representation $f_\theta(\mathbf{X})$ to predict the label Y for each set $\mathbf{X}$. For compactness, we denote $f_\theta(x_i)$ as the representation of instance x_i even though the proposed representation will depend on all other instances in the set, akin to kernel machines.

We use the representation $f_\theta(\mathbf{X})$ to learn a mapping $\Phi(\mathbf{X}) \rightarrow Y$, where Y is the set-level label and $\Phi(\mathbf{X})$ is given by:

$$\Phi(\mathbf{X}) = h \left(\sum_{x_i \in \mathbf{X}} g(f_\theta(x_i)) \right), \quad (1)$$

where both $g(\cdot)$ and $h(\cdot)$ are (parametric) layers, *e.g.*, linear or multi layered perceptron (MLP), and with the appropriate activation function used for classification.

The representation $f_\theta(\cdot)$ should be permutation-invariant so that the mapping $\Phi(\mathbf{X})$ is invariant to the order of the instances (due to the sum operator) [47]. We propose the affinitention architecture, which provides a representation consistent with $f_\theta(\cdot)$ by implicitly learning affinities between the instances.

3 KERNEL-BASED REPRESENTATION LEARNING ON SETS

We consider the following representation learning framework with kernel $\mathbf{K}_\theta \in \mathbb{R}^{N \times N}$:

$$\mathbf{K}_\theta(i, j) \triangleq k(f_\theta(x_i), f_\theta(x_j)), \quad (2)$$

where $k(\cdot, \cdot)$ is a predefined kernel function, which in our case is the dot product unless otherwise noted. The kernel matrix $\mathbf{K}_\theta$ has two important properties: i) it is a valid kernel, i.e., its elements $\mathbf{K}_\theta(i, j)$ can be written in the form $\langle \Psi(x_i), \Psi(x_j) \rangle_{\mathcal{H}}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is an inner product in a reproducing kernel Hilbert space $\mathcal{H}$, and $\Psi(x_i)$ is some mapping $\Psi(\cdot) : \mathbb{R}^d \rightarrow \mathcal{H}$; and ii) any proper inner product $\langle \Psi(x_i), \Psi(x_j) \rangle_{\mathcal{H}}$ can be represented in the form of (2). When $k(\cdot, \cdot)$ is a dot product, the proof is trivial using $f_\theta \equiv \Psi(\cdot)$. The proof and conditions for the general $k(\cdot, \cdot)$ can be found in [38]. These two properties allow learning the parameters θ using a loss function defined over the representation $f_\theta(\cdot)$ or the kernel matrix $\mathbf{K}_\theta$.

This paper focuses on the affinitention architecture, which we obtain by analyzing $\mathbf{K}_\theta(\cdot, \cdot)$ in (2) and its corresponding representation $f_\theta(\cdot)$. The latter is used for learning purposes by defining a loss function over $f_\theta(\cdot)$, rather than over $\mathbf{K}_\theta(\cdot, \cdot)$. Learning with the kernel representation is still possible, in which case we refer the reader to [38] for further details.

The basic building block of the affinitention net is given by the following kernel:

$$\mathbf{K}_\theta^{\text{base}} = \mathbf{W} \triangleq \mathbf{X}\mathbf{V}\mathbf{V}^T\mathbf{X}^T, \quad (3)$$

where $\mathbf{V} \in \mathbb{R}^{d \times d'}$ is a learnable weight matrix, i.e., θ , and $\mathbf{V}^T$ is its transpose. The corresponding representation $f_\theta^{\text{base}}(\mathbf{X}) = \mathbf{X}\mathbf{V}$ is simply a linear mapping.

Affinitention nets are obtained from the combination of multiple such kernels, into a composite kernel, using the product and the sum operators. Specifically, the proposed kernel is given by:

$$\mathbf{K}_\theta = \sum_{l=1}^L \prod_{m=0}^{2M} \mathbf{W}_{l,m} \triangleq \sum_{l=1}^L \prod_{m=0}^{2M} \mathbf{X}\mathbf{V}_{l,m}\mathbf{V}_{l,m}^T\mathbf{X}^T, \quad (4)$$

where $\mathbf{V}_{l,m} \in \mathbb{R}^{d \times d'}$ are learnable parameter matrices defined similarly to $\mathbf{V}$. We further set $\mathbf{W}_{l,m} \equiv \mathbf{W}_{l,2M-m}$ to guarantee the symmetry of $\mathbf{K}_\theta$. Below we address the analysis of (4) and the derivation of the corresponding *affinitention representation* $f_\theta(\cdot)$, which as we will show has a form similar to a multi-head and multi-layer attention architecture. Further, we address the selection of L , M and other free parameters of the kernels $\mathbf{W}_{l,m}$.

3.1 The product of kernels

We first analyze the product of kernels assuming $L = 1$ in (4). In the special case, where $M = 1$, the corresponding affinitention representation, denoted by $f_{\theta,1}(\mathbf{X})$, in which $l = 1$, is given by:

$$f_{\theta,1}(\mathbf{X}) = \mathbf{W}_{1,0}\mathbf{X}\mathbf{V}_{1,1}, \quad (5)$$

where the obtained representation is no longer linear with respect to $\mathbf{X}$. Note that $\mathbf{W}_{1,0} \triangleq \mathbf{X}\mathbf{V}_{1,0}\mathbf{V}_{1,0}^T\mathbf{X}^T$ functions as an affinity kernel provided it is a function of $\mathbf{X}$. As a result, the representation of each instance x_i is a function of all other instances.

In the case $M > 1$, the representation is given by:

$$f_{\theta,1}^{\text{prod}}(\mathbf{X}) = \left(\prod_{m=0}^{M-1} \mathbf{W}_{1,m} \right) \mathbf{X}\mathbf{V}_{1,M}, \quad (6)$$

where the product $\prod_{m=0}^{M-1} \mathbf{W}_{1,m}$ allows the capture of increasingly complex relations between the instances. Proposition 1, shows that a multilayered version of (5) is in fact also obtained from the product of kernels:

PROPOSITION 1. *The kernel corresponding to a composition of multiple layers of (5) has the form of the product of kernels in (4), for $L = 1$.*

Let $\square^{(s)}$ denote the s th layer; the proof for the case of two layers ($s = 2$) is given by explicitly writing the output of the second layer, $\mathbf{X}^{(2)}$, similarly to (5):

$$\mathbf{X}^{(2)} = \mathbf{W}_0^{(1)}\mathbf{X}^{(1)}\mathbf{V}_1^{(1)} \equiv \mathbf{X}^{(1)}\mathbf{V}_0^{(1)}\mathbf{V}_0^{(1)T}\mathbf{X}^{(1)T}\mathbf{X}^{(1)}\mathbf{V}_1^{(1)}, \quad (7)$$

where for compactness we omit the subscript ($l = 1$). By plugging $\mathbf{X}^{(1)} = \mathbf{X}\mathbf{V}_0^{(0)}\mathbf{V}_0^{(0)T}\mathbf{X}^T\mathbf{X}\mathbf{V}_1^{(0)}$, where $\mathbf{X} \equiv \mathbf{X}^0$, into (7), we get:

$$\begin{aligned} \mathbf{X}^{(2)} = & \mathbf{X}\mathbf{V}_0^{(0)}\mathbf{V}_0^{(0)T}\mathbf{X}^T. \\ & \mathbf{X}\mathbf{V}_1^{(0)}\mathbf{V}_0^{(1)}\mathbf{V}_0^{(1)T}\mathbf{V}_1^{(0)T}\mathbf{X}^T. \\ & \mathbf{X}\mathbf{V}_0^{(0)}\mathbf{V}_0^{(0)T}\mathbf{X}^T. \\ & \mathbf{X}\mathbf{V}_0^{(0)}\mathbf{V}_0^{(1)T}\mathbf{X}^T. \\ & \mathbf{X}\mathbf{V}_1^{(0)}\mathbf{V}_1^{(1)}. \end{aligned} \quad (8)$$

Accordingly, the overall learned kernel $\mathbf{X}^{(2)}\mathbf{X}^{(2)T}$ has the form of the product of kernels as in (4) with $L = 1$. This result implies that a representation obtained from the product of kernels with $M = 4$ has the representative power of an architecture composed of two layers of the form in (5). Due to this equivalence, we control the expressiveness of the model using the parameter M rather than using multiple layers as in the transformer. The proof of the general case with the composition of S layers is carried out by induction, and is provided in the Appendix.

3.2 The sum of kernels

We proceed with the analysis of the role of the sum operator in (4), considering the general form:

$$\mathbf{K}_\theta^{\text{sum}} = \sum_{l=1}^L f_{\theta,l}(\mathbf{X})(f_{\theta,l}(\mathbf{X}))^T, \quad (9)$$

where $f_{\theta,l}(\mathbf{X})$ is defined as in (6). Proposition 2 shows that the sum of kernels is equivalent to a multi-head architecture.

PROPOSITION 2. *The representation $f_\theta^{\text{sum}}(\mathbf{X})$ corresponding to the kernel in (9) is given by:*

$$f_\theta^{\text{sum}}(\mathbf{X}) = f_{\theta,1}(\mathbf{X}) || f_{\theta,2}(\mathbf{X}) || \dots || f_{\theta,L}(\mathbf{X}). \quad (10)$$

where $||$ is the concatenation operator.

PROOF. Recalling the definition $\mathbf{K}_\theta \triangleq f_\theta(\mathbf{X})(f_\theta(\mathbf{X}))^T$ and by plugging $f_\theta^{\text{sum}}(\mathbf{X})$, we obtain (9) by the matrix-product rule. $\square$

According to (10), the dimension of the representation of instance x_i is $d' \cdot L$. For convenience, we set $d' \cdot L \equiv d$ to keep the original dimension of the instances. In our experiments we use $L = 8$ unless otherwise stated, which is also a common practice in the transformer.

Propositions 1 and 2 imply that the parameters L and M control the expressiveness of the model. Moreover, while the kernels in (4) have a pair of the product operator followed by the sum operator, it is possible to concatenate multiple such pairs to further increase the expressiveness of the model.

3.3 The link to diffusion and information fusion

To obtain a more interpretable architecture, we extend our framework using a different version of the base kernel in (3). First, we use a point-wise exponential as the predefined kernel $k(\cdot, \cdot)$:

$$U(i, j) \triangleq \exp \{W(i, j)\}. \quad (11)$$

From Schur's product theorem, which states that the Hadamard product of a positive semi-definite (PSD) matrix is PSD, it follows that U is also PSD, and thus is a valid kernel. We then normalize the rows of U , which leads to the final version of affinintention representation $f_\theta(\cdot)$, given by replacing $f_{\theta,l}(\cdot)$ in (6) with an updated version of $f_{\theta,l}^{\text{prod}}(\cdot)$:

$$f_{\theta,l}^{\text{prod}}(X) \triangleq \prod_{m=0}^{M-1} D_{l,m}^{-1} U_{l,m} X V_{l,m} \equiv \prod_{m=0}^{M-1} \bar{U}_{l,m} X V_{l,m}, \quad (12)$$

where D is the degree matrix of U . For $M = 1$, $\bar{U}$ is a row-normalized affinity kernel, *i.e.*, a diffusion operator [5], which is applied to X , forming an attention mechanism so that the representation of x_i is a convex combination of all the instances in the set. In the general case of $M \geq 1$, the operator $\prod_{m=0}^{M-1} \bar{U}_{l,m}$ is also row normalized, and is also a diffusion operator [24].

The product and the sum between affinity kernels have been widely studied in the context of multi-view data fusion [6, 11, 24, 28, 32, 41]. In these works, the kernels represent affinities between data points (instances) in different views, and their fusion is based on the product or the sum operators. While in these studies the kernels are predefined and view-specific, here they are data-driven, so their combination learns to fuse the data into a consolidated representation.

The information fusion perspective, as well as Propositions 1 and 2, imply that the combination of the product and the sum operators allows for learning complex relations between the instances. Yet, overly complex architectures can lead to over-smoothed kernels and representations. Specifically, the product kernel is associated with a diffusion (smoothing) operator in (12). Similarly, the (i, j) -th entry of K_θ^{sum} is given by averaging (smoothing) over L learned affinities between instances x_i and x_j . With attention to this trade-off between the complexity of the learned affinities and oversmoothing, we select in our experiments $M \leq 2$ in (4).

3.4 Permutation equivariance

PROPOSITION 3. *The representation of affinintention nets $f_\theta(X)$, given by (10) and (12), is permutation equivariant.*

The proof is provided in the Appendix.

3.5 Implementation:

For the classification of image data, we obtain the vector embedding of the i th instance, x_i , by feeding the corresponding image (patch) into a convolutional neural network comprising several convolutional layers followed by a fully connected layer (more details are provided in the Appendix). Then, we use the following loss function:

$$\sum_i l(y_i, g(f_\theta(x_i))), \quad (13)$$

where $l(\cdot, \cdot)$ is the binary cross entropy (BCE) loss used in our experiments, and $g(\cdot)$ is a neural network with fully connected and residual layers [16] (details are provided in the Appendix). We note that this loss function assumes the availability of instance-level labels y_i , while in set-classification only a single, set-level label is available. Therefore, we propose to propagate the set-level label to each instance. Recalling that Y is the label of X , we assume that $\forall x_i \in X : y_i = Y$. This strategy introduces label noise, which was explored in detail in [9]. The prediction of the label Y is given according by $\sum_{x_i \in X} g(f_\theta(x_i))$ such that $h(\cdot)$ in (1) is the identity function. We also note that it is possible to optimize the parameters θ using a loss function defined directly over the kernel K_θ , as in [38], however, we consider it out of the scope of this paper. Full details of the implementation of affinintention nets is provided in the Appendix.

For the classification of text data, we divide each text sequence into multiple tokens and compute token embeddings (summed with positional embeddings) as in [40]. Thus, each text sequence is represented by a set X of multiple token embeddings, *i.e.*, multiple instances. Then, affinintention nets are trained and evaluated in the same way as for the image data. In addition, we consider another variant similar to [7], where we replace the transformer encoder with the affinintention representation. Specifically, we add the token CLS at the beginning of the sequence and predict the label Y by $h(f_\theta(x_{\text{cls}}))$, where x_{cls} is the embedding of the CLS token, and $h(\cdot)$ is MLP. This approach can be formulated as in (1) by selecting $g(\cdot)$ as the indicator function, *i.e.*, 1 if $x_i \equiv x_{\text{cls}}$, and 0 otherwise.

4 KERNEL PERSPECTIVE ON THE LITERATURE

Affinintention generalizes multiple set-classification and MIL methods. In its simplest form, when the kernel is given by (3) and the weights V are set to the identity matrix, we get the identity representation $f_\theta(X) \equiv X$, so there is no dependence between the instances. In this case, affinintention nets degenerate into average pooling MIL, where the averaging is done with the sum operator in (1).

A single-layer, single-head affinintention net obtained by (5) with $L = 1$ can further degenerate into attention-based weighting similar to the one presented in [17] by zeroing all off-diagonal weights of $W_{1,0}$. In this case, we get a self-attention mechanism since the (unnormalized) weight of each instance is the function only of the instance itself as the attention mechanism does not exploit cross-instance affinities.

The representation in (12) with $M = 1$ is similar to a single-head single-layer transformer encoder (BERT) [25]. Further, according to Propositions 1 and 2, the affinintention architecture is equivalent to BERT without the fully connected layers that follow each attention layer. Also, in BERT, the weight matrix V of the base kernel is replaced by two learnable matrices termed “key” and “query”. This breaks the symmetry in (11) so that the attention mechanism does not have the interpretation of a diffusion operator. The transformer is constructed by stacking multiple multi-head layers, each followed by multiple fully-connected and normalization layers. Many studies [3, 4, 20, 23, 25] proposed variants of the transformer focusing on improving computational efficiency, by reducing the number of

learnable parameters by, for example, factorizing the learnable parameter matrices. These methods, however, modify only the attention heads, which are then plugged into the architecture of the transformer. The transformer and its variants are widely used in many NLP applications [7, 34, 40, 44], and are becoming prevalent for the analysis of set-structured data [25, 36], and image- and video-data [2, 3, 15, 30, 35, 42, 46]. In contrast to the transformer, which is based on heuristics, the representation of affinitention net is obtained from the kernel-learning framework, and it has the representative power of a multi-head multi-layer architecture, as shown in Propositions 1 and 2.

Closely related to this paper is [42], where Wang *et al.* proposed a single-head attention architecture based on a (row-normalized) kernel for image and video classification. Shi, *et al.* [36] extended their work by using multiple attention layers as in the transformer, except they shared the attention weights among the multiple layers. Their analysis further revealed the relationship of the attention mechanism to diffusion. However, the analysis in [36] is limited to a single-head, single-layer architecture, and stacking multiple such layers, each followed by an MLP, provides a complex and uninterpretable architecture. Affinitention nets generalize [36, 42] by revealing the relationship of the multi-head multi-layer architectures to diffusion. In addition, the combination of the sum and the product of multiple kernels increases the expressiveness of affinitention nets in a principled way, without the need for heuristics such as adding an MLP after each layer.

Closely related to our work are also graph convolutional networks (GCNs) [1, 14, 19, 27, 43]. These consider data represented via a set of nodes connected by edges. In our context, consider a graph associated with the set X , where the nodes are the instances and the edges are given by a predefined affinity kernel $A \in \mathbb{R}^{N \times N}$. By using $M = 1$ in (12) and replacing U with A , we obtain a single-layer GCN. Our study justifies the use of multi-layer GCNs, widely used for graph data. In [38], a kernel-based representation learning framework was investigated for graph node classification. While [38] focused on the learning process, our study addresses the design of the attention architecture. Indeed, [38] uses an *ad hoc* variant of GCN, which does not naturally arise from their framework.

5 EXPERIMENTS

5.1 Baselines

Text classification is a fundamental task in NLP, thus there are many popular text classifiers. We compare affinitention nets with strong baselines like SVM [13, 39], textCNN [18], Bi-LSTM attention [45, 49], BERT [7], and BioBERT [26]. The SVM classifier takes TF-IDF features of each document as input, and we implement it with a ℓ_2 penalty. For the Bi-LSTM attention model, we add an attention layer on top of the Bi-directional LSTM architecture. BERT was pre-trained on open-sourced datasets, and BioBERT was further pre-trained on a large external biomedical corpora.

For the classification of sets of images, the baseline approaches to which we compare the affinitention net include the following multiple-instance-learning methods: *average-pooling*, *noisy-or* [48] and *noisy-and* [21] functions. We also compare with Ilse *et al.* [17], who propose an attention mechanism to weight the importance

on 5/26/16 at 3:15 pm i had a tia that caused speech slurring and lopsided smile (right side affected) . it lasted about 5 min and was followed by a brief loss of vision in my lower right eye . both of these have happened before , but not for months . no noticeable accompanying signs of afib . this is just for documentation . thanks .

i have been feeling really tired and sleeping a lot and getting dizzy

hello . i noticed that the results of the ecg that was done on august 29th were abnormal (non specific t wave abnormality) . is this something new that i need to be concerned about and what causes this test to come back abnormal ? thank you , jenelle terry

(a) Transformer.

on 5/26/16 at 3:15 pm i had a tia that caused speech slurring and lopsided smile (right side affected) . it lasted about 5 min and was followed by a brief loss of vision in my lower right eye . both of these have happened before , but not for months . no noticeable accompanying signs of afib . this is just for documentation . thanks .

i have been feeling really tired and sleeping a lot and getting dizzy

hello . i noticed that the results of the ecg that was done on august 29th were abnormal (non specific t wave abnormality) . is this something new that i need to be concerned about and what causes this test to come back abnormal ? thank you , jenelle terry

(b) Affinitention Net.

Figure 1: Visualization of affinities (attention weights) between the CLS token and the other tokens in the set, taken from the first layer. Red color indicates higher affinities. All examples are selected from the “Medium” or “Urgent” classes in the test set.

of instances based on [47] (termed *AttenWeight* in the tables). Additionally, a comparison is made to the (set) transformer encoder (BERT), where we use the original architecture of BERT [7] rather than the architecture in [25], since the latter focuses on computational efficiency, which is not a bottleneck in our experiments. We also evaluate the method in [36] (*DMPs* in the tables). Further, we consider a single-instance-learning (SIL) approach, where the set-level labels are propagated to the instance level [10].

5.2 Patient message triage

In this experiment, we demonstrate the use of affinitention nets for low resource text sequence classification. We utilize patient message urgency data, which consists of web portal messages generated from 10/2014 to 08/2018 by adult patients (> 18 years old) of a large academic medical center. A detailed description of the dataset can be found in [37]. The messages include free, unstructured plain text sent by patients to their healthcare team. Responses and messages sent from the clinician or health system to the patients were excluded from the analysis. Portal messages were manually labeled by experienced sub-specialty (cardiology) clinicians into three levels of priority: non-urgent, medium, and urgent. Non-urgent labels include notes of appreciation (e.g., thank-you messages). The Medium urgency class contains messages that could be reasonably responded to in 1-3 days. Urgent messages are those requiring an immediate phone call to the patient by the clinician.

The dataset comprises a total number of 1,756 messages, including 631 non-urgent, 955 medium, and 170 urgent messages as is summarized in Table 1. The table also includes a typical example

Table 1: Typical examples of patient messages to providers grouped by urgency.

Label	Count	Typical Example
Non-urgent	631	That would be awesome... thank you.
Medium	955	Dr. [name]. All seems well now. I am at home resting. My wife and I have a trip planned to Maryland this week beginning on Wednesday. We can fly, drive or stay home if I should not travel. Are there any reasons that I should not fly.
Urgent	170	I have continued having chest pain shortness of breath since waking. Please tell me what to do. I have tried in hailers am going to try nebulizers. I just feel extremely tight in my chest.

Figure 2: Set-Cifar10 dataset: accuracy vs PPI**Table 2: Patient message triage - classification of urgency of messages.**

	Macro-F1
SVM	0.748 ± 0.007
Text CNN	0.754 ± 0.020
Bi-LSTM Attention	0.761 ± 0.016
BERT	0.761 ± 0.021
BioBERT	0.764 ± 0.010
Proposed CLS ($M = 1$)	0.775 ± 0.019
Proposed CLS ($M = 2$)	0.779 ± 0.016
Proposed ($M = 1$)	0.784 ± 0.014
Proposed ($M = 2$)	0.764 ± 0.013

message for each class. An urgent message, for example, could be that a patient reports chest pain. In our experiments, the dataset is split into 80% training set ($\sim 1.4K$) and 20% test set ($\sim 0.35K$).

Table 3: Patient message triage. Comparison of the number of parameters (excluding embeddings) and inference time per sample in seconds.

	#Params	Infer. Time
TextCNN	1.5 M	0.15
Bi-LSTM Att.	1.0 M	0.55
BERT	86.5 M	8.89
Proposed ($M = 1$)	1.3 M	0.51
Proposed ($M = 2$)	1.9 M	0.76

Table 2 presents the performance of the different methods on the message triage dataset. The Macro-F1 score is employed to evaluate the performance of classifiers, because the dataset is imbalanced. BioBERT, which is pretrained on a large external dataset of medical text, provides the best performance among the baselines. Affinattention net outperforms BioBERT (and the other baselines) even though no external data was used for its training. Note that the variant of the proposed approach that uses the CLS token for classification

(*Proposed CLS* in the table), with $M = 2$ in (4) performs better than the variant with $M = 1$. This supports the improved representative power of the product of kernels in (4). The variant of affinitation nets that is based on averaging the predictions from each one of the tokens via $\sum_{x_i \in X} g(f_\theta(x_i))$ as for the image data (and that does not use the CLS token) provides the best performances.

In Table 3, we further present a comparison between the affinitation nets and BERT in terms of the number of parameters and the training time. The number of parameters reported in this table excludes the token and positional embedding matrices. The affinitation nets have a much smaller number of parameters (45 times) than BERT, and its training time per iteration is much shorter (10 times) than BERT. The experiment demonstrates the advantage of the light architecture of affinitation nets when only a small number of training examples is available, which is often the case in medical applications.

Figure 1 shows visualizations of affinities (attention weights) between the CLS token and the other tokens in the set, learned both by BERT and affinitation nets. In these examples, affinitation nets better capture key words corresponding to urgency of message. For instance, affinitation nets have more clear focus on words like “right side affected”, “dizzy” and “abnormal”, that indicate the necessity of medical treatment for the patients.

5.3 Set-CIFAR10

We simulate a set-classification setting using the CIFAR10 dataset [22], which originally comprises natural images from 10 categories. We consider the images as instances grouped into sets. As in MIL, a single positive instance in a set triggers a positive set-level label. Specifically, if an instance belongs to one of 5 arbitrarily chosen categories, the corresponding set is considered positive. CIFAR10 data allow us to control the structure of sets in terms of the prevalence of positive instances (PPI), thus simulating multiple scenarios of set-classification. For example, in the case of thyroid malignancy prediction from biopsy slides, there are cases (slides) that are clearly malignant. In these cases, there is a consistency in malignancy characteristics across the instances (image regions), *i.e.*, high PPI. On the other hand, there are indeterminate cases with intermediate PPI: the slides include a mix of benign instances and instances with various atypical characteristics, potentially indicating malignancy.

We separately evaluate the performance of affinitation nets for different PPI values. We construct multiple datasets, each with a different average PPI. Each dataset comprises 1000 sets, with 100 instances each. We sample the instances from CIFAR10 uniformly at random, with an equal probability of negative and positive sets. Given the average PPI value, we allow a slight variation of the PPI in each set, so that it varies in the range of 0.8 – 1.2 of the average PPI. We train each algorithm for 30 epochs and repeat the experiment 10 times for each average PPI value. The performance of the different algorithms is presented in Fig. 2.

Figure 2 shows that the classification accuracy of all methods increases monotonically with PPI. High PPI facilitates the classification since fewer negative examples appear in positive sets. For high PPI values, the SIL model, in which each instance is classified independently, achieves almost perfect classification accuracy. However, in the more challenging case of lower PPI values, both affinitation

Figure 3: Set-Cifar10 dataset. Visualization of learned affinities between instance in an example set with PPI 0.3. Positive instances are colored with yellow and negative with blue. Left: 2D diffusion maps. Right: corresponding instances. Top row: diffusion maps obtained from the raw data. Middle row: diffusion maps of X . Bottom row: diffusion maps of affinitation representation $f_\theta(X)$.

nets and BERT achieve significantly higher accuracy scores. This implies the advantage of learning affinities between the instances. The improved performance of the variant of affinitation nets with the product of $M = 2$ kernels demonstrates the improved representative power of the product of kernels in (4). We also note that there is no quantifiable advantage to the more complex architecture of BERT.

To get insights into the learned affinities between the instances, we further visualize the learned kernel K_θ in (4) that is associated by our analysis to the affinitation architecture. Among the wide range of available spectral analysis and low-dimensional representation tools, we use diffusion maps [5], which manifest a low-dimensional embedding of the instances obtained via the eigenvalue decomposition of a row-normalized version of K_θ . In diffusion maps, instances with high affinities between them are mapped close to each other in the low dimensional domain. In Fig. 3 we present a two-dimensional visualization of the affinitation representation $f_\theta(X)$, as well as those of the original (convolutional) feature representation X and the raw images.

The visualization of both the raw images (top row in the figure) and of the convolutional features (middle row in the figure) shows no separation between the positive and negative instances. The last row of the figure demonstrates the separation obtained by the affinitation representation between the positive and the negative instances of the set. This highlights the ability of affinitation nets to implicitly model (hidden) interactions between instances,

Figure 4: Examples of follicular groups with TBS categories 2 – 6 (left to right).

which are potentially associated with patterns appearing in positive (and negative) instances. We emphasize that the learning process is performed solely using set-level labels and without any prior information on separate instances.

5.4 Thyroid malignancy prediction

We consider thyroid malignancy prediction from *pre-operative* fine needle aspiration biopsy (FNAB) slides, scanned at $\times 10$ magnification. Our dataset comprises 692 and 288 whole slide images (WSIs) used for training and testing, respectively. Each slide is associated with a diagnostic score according to The Bethesda System (TBS). TBS 2 and 6 are assigned to benign and malignant slides, respectively. TBS 3,4,5 represent indeterminate diagnoses with an increased risk of malignancy. TBS 1, which is assigned to inadequate slides, is out of the scope of this paper. Our dataset also includes *post-operative* pathology adjudications (benign/malignant), which are unambiguous and are used as the ground truth of malignancy in our experiments.

Most indicative for malignancy are various features of thyroid (follicular) groups of cells such as their architecture, size and color. As a preprocessing stage, we apply a neural network [10, 12] to identify patches (instances) containing groups of follicular cells, separating them from background regions (that mostly contain blood cells). Following [10], each slide is represented by a set of 1000 patches in the training, which correspond to the instances with the highest predicted values, *i.e.*, the most informative instances. For the test set, we use sets with 100 patches, as we did not find an advantage using larger sets. Examples of patches corresponding to slides with different TBS categories are presented in Figure 4. Each patch is represented by a 16-dimensional vector obtained from a VGG11-based neural network (details are provided in the Appendix). Due to the small dimension of these vectors, in this experiment we use $L = 4$ attention heads in (4).

We train the affinitention nets to predict the slide-level ground-truth malignancy label using the loss in (13). Simultaneously, we predict the TBS category using an ordinal regression framework, with the following loss function [8, 31]:

$$\sum_{j=2}^5 \sum_i l(z_i^j, g(f_\theta(x_i)) - \beta_j), \quad (14)$$

where β_j are learnable parameters and $z_i^j = 1$ if the TBS category is higher than j , and zero otherwise. The total loss function is given by the sum of (13) (binary classification for malignancy) and (14)

(ordinal regression for TBS). We examined empirically that this strategy leads to improved performance relative to only using (13).

In addition to the competing baseline, we compare affinitention nets to three cytopathologists (Cyto 1-3 in the plots), who reviewed the slides and assigned TBS categories, as well as the cytopathologist on the electronic medical record (EMR in the plots). The performance of the cytopathologists is evaluated by associating the TBS categories to increasing probabilities of malignancy. To be able to provide competitive performance compared to the cytopathologists, we observed in our experiments that the use of ensembles significantly improves the performance of the different methods. Specifically, for each of the methods, we train an ensemble of 5 models. We use a cross-validation procedure where we split the training data into 5 folds, and each model is trained with a different fold used for validation and the rest 4 folds for training. In this procedure, all models use the same set of hyper parameters. We use the validation fold to evaluate the performance of a given model after each training epoch and select the set of parameters providing the best performance on the validation data. In the test phase, we evaluate the performance of each ensemble. The experiment is repeated 5 times to calculate the average and standard deviation of the performance metrics. We note that due to the small number of training sets, we observed relatively large variation in the performance of the different methods on the different folds of the training data and especially of the models that use attention mechanisms. The use of ensembles allowed us to stabilize their performance, as well as to achieve performance comparable to the human level.

Table 4 summarizes the performance of the different methods in terms of area under the curve (AUC) and average precision (AP). We also present in the performance of the different methods over a subset of 118 indeterminate cases with the diagnostic scores TBS 3 to 5. Table 4 shows that affinitention nets with $M = 1$ perform better than a single-layer BERT. The improvement may be due to the smaller number of parameters in each attention head of the affinitention nets, V , compared to the two learnable matrices “key” and “query” in BERT. Table 4 also shows that the attention-based methods such as affinitention nets, DMPS, and BERT perform favorably compared to methods that assume no dependence between instances such as SIL. Yet, the improvement is more significant in the evaluation over the subset of the indeterminate cases. In clearly malignant cases (TBS 6), there is consistency between the instances such that malignant findings appear in most instances. Similarly, in benign slides (TBS 2) the instances do not contain findings that

Table 4: Thyroid malignancy prediction. Area under the curve (AUC) and average precision (AP) scores over the entire test set and over a subset of indeterminate cases with the diagnostic scores TBS 3 to 5.

	AUC	AP	AUC(TBS3-5)	AP(TBS3-5)
EMR	0.915	0.778	0.677	0.385
Cyto1	0.885	0.759	0.779	0.528
Cyto2	0.835	0.684	0.698	0.461
Cyto3	0.814	0.678	0.636	0.413
Nosiy Or	0.884±0.018	0.748±0.037	0.755±0.034	0.478±0.036
NoisyAnd	0.905±0.006	0.791±0.01	0.793±0.009	0.521±0.018
SIL	0.904±0.004	0.786±0.005	0.782±0.007	0.496±0.007
AttenWeights	0.765±0.032	0.574±0.053	0.617±0.035	0.341±0.031
DMPS	0.907±0.007	0.813±0.012	0.795±0.010	0.533±0.010
BERT	0.903±0.008	0.799±0.019	0.792±0.026	0.519±0.037
BERT (1 layer)	0.902±0.006	0.798±0.016	0.788±0.014	0.514±0.018
Proposed MIL	0.911±0.003	0.812±0.006	0.808±0.006	0.537±0.010
Proposed ($M = 1$)	0.910±0.006	0.818±0.009	0.811±0.009	0.560±0.015
Proposed ($M = 2$)	0.913±0.005	0.810±0.008	0.809±0.013	0.546±0.022

are suspicious for malignancy. In such cases, good predictions are obtained separately from each one of the instances. In contrast, the indeterminate cases (TBS 3,4,5) include a mix of benign groups and groups with various atypical characteristics potentially indicating malignancy. In this case, the attention-based methods provide improved performance. This suggests the benefit of modeling the complex interactions between instances in these complex cases. The improved AUC and AP scores of affinitention nets, compared to BERT, implies no advantage to the more complex architecture of BERT.

CONCLUSIONS

We have presented a kernel-based representation learning framework for set-classification, which motivates the use of affinity-attention architectures, by associating their multiple layers and heads to the product and the sum of learned kernels. We showed the connection of our framework to information fusion such that each learned kernel represents affinities between instances of the set in different views. This interpretation revealed a trade-off between learning complex affinities between instances and avoiding over-smoothed kernels. From this framework, we have devised affinitention net, a simplified affinity-attention architecture. We demonstrated that on medical text and image datasets the affinitention net provides competitive performance compared to state-of-the-art heuristic attention architectures.

ACKNOWLEDGMENTS

This research was supported in part by NIH/NIBIB R01-EB025020, the Simons Math+X Investigators Award (400837) and the Rhodes Information Initiative at Duke.

Hongteng Xu joined in this work when he was affiliated with Duke University. He was supported in part by Beijing Outstanding

Young Scientist Program (NO. BJJWZYJH012019100020098) and NSFC No. 61832017.

REFERENCES

- [1] Sami Abu-El-Hajja, Amol Kapoor, Bryan Perozzi, and Joonseok Lee. 2020. N-gcn: Multi-scale graph convolution for semi-supervised node classification. In *uncertainty in artificial intelligence*. PMLR, 841–851.
- [2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-End Object Detection with Transformers. *arXiv:2005.12872 [cs.CV]*
- [3] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509* (2019).
- [4] Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. 2020. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794* (2020).
- [5] R. R. Coifman and S. Lafon. 2006. Diffusion maps. *Applied and computational harmonic analysis* 21, 1 (2006), 5–30.
- [6] V. R. De Sa, P. W. Gallagher, J. M. Lewis, and V. L. Malave. 2010. Multi-view kernel construction. *Machine learning* 79, 1-2 (2010), 47–71.
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [8] M. Dorado-Moreno, P. A. Gutiérrez, and C. Hervás-Martínez. 2012. Ordinal classification using hybrid artificial neural networks with projection and kernel basis functions. In *International Conference on Hybrid Artificial Intelligence Systems*. Springer, 319–330.
- [9] David Dov, Shahar Z Kovalsky, Serge Assaad, Jonathan Cohen, Danielle Elliott Range, Avani A Pendse, Ricardo Henao, and Lawrence Carin. 2021. Weakly supervised instance learning for thyroid malignancy prediction from whole slide cytopathology images. *Medical Image Analysis* 67 (2021), 101814.
- [10] David Dov, Shahar Z Kovalsky, Jonathan Cohen, Danielle Elliott Range, Ricardo Henao, and Lawrence Carin. 2019. Thyroid cancer malignancy prediction from whole slide cytopathology images. In *Machine Learning for Healthcare Conference*. PMLR, 553–570.
- [11] David Dov, Ronen Talmon, and Israel Cohen. 2016. Kernel-based sensor fusion with application to audio-visual voice activity detection. *IEEE Transactions on Signal Processing* 64, 24 (2016), 6406–6416.
- [12] Danielle Elliott Range, David Dov, Shahar Z Kovalsky, Ricardo Henao, Lawrence Carin, and Jonathan Cohen. 2020. Application of a machine learning algorithm to predict malignancy in thyroid cytopathology. *Cancer cytopathology* 128, 4 (2020), 287–295.
- [13] Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. 2000. Regularization networks and support vector machines. *Advances in computational mathematics* 13, 1 (2000), 1.

- [14] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur. 2017. Protein interface prediction using graph convolutional networks. In *Advances in neural information processing systems*. 6530–6539.
- [15] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. 2019. Video Action Transformer Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [16] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
- [17] Maximilian Ilse, Jakob M Tomczak, and Max Welling. 2018. Attention-based deep multiple instance learning. *arXiv preprint arXiv:1802.04712 (ICML18)* (2018).
- [18] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. 1746–1751. <http://aclweb.org/anthology/D/D14/D14-1181.pdf>
- [19] T. N. Kipf and M. Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [20] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451* (2020).
- [21] O. Z. Kraus, J. L. Ba, and B. J. Frey. 2016. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics* 32, 12 (2016), i52–i59.
- [22] A. Krizhevsky, G. Hinton, et al. 2009. *Learning multiple layers of features from tiny images*. Technical Report. Citeseer.
- [23] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).
- [24] Roy R Lederman and Ronen Talmon. 2018. Learning the geometry of common latent variables using alternating-diffusion. *Applied and Computational Harmonic Analysis* 44, 3 (2018), 509–536.
- [25] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R Kosiorek, Seungjin Choi, and Yee Whye Teh. 2018. Set transformer: A framework for attention-based permutation-invariant neural networks. *arXiv preprint arXiv:1810.00825* (2018).
- [26] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.
- [27] Renjie Liao, Zhizhen Zhao, Raquel Urtasun, and Richard S Zemel. 2019. Lanczosnet: Multi-scale deep graph convolutional networks. *arXiv preprint arXiv:1901.01484* (2019).
- [28] Ofir Lindenbaum, Arie Yeredor, Moshe Salhov, and Amir Averbuch. 2020. Multi-view diffusion maps. *Information Fusion* 55 (2020), 127–149.
- [29] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AWM Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. 2017. A survey on deep learning in medical image analysis. *Medical image analysis* 42 (2017), 60–88.
- [30] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. 2020. Object-centric learning with slot attention. *arXiv preprint arXiv:2006.15055* (2020).
- [31] P. McCullagh. 1980. Regression models for ordinal data. *Journal of the royal statistical society. Series B (Methodological)* (1980), 109–142.
- [32] Tomer Michaeli, Weiran Wang, and Karen Livescu. 2016. Nonparametric canonical correlation analysis. In *International Conference on Machine Learning*. 1967–1976.
- [33] G. Quellec, G. Cazuguel, B. Cochener, and M. Lamard. 2017. Multiple-instance learning for medical image and video analysis. *IEEE reviews in biomedical engineering* 10 (2017), 213–234.
- [34] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- [35] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. 2020. Sketchformer: Transformer-Based Representation for Sketched Structure. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [36] Y. Shi, J/ Oliva, and M. Niethammer. 2019. Deep Message Passing on Sets. *arXiv preprint arXiv:1909.09877* (2019).
- [37] Shijing Si, Rui Wang, Jedrek Wosik, Hao Zhang, David Dov, Guoyin Wang, and Lawrence Carin. 2020. Students Need More Attention: BERT-based Attention Model for Small Data with Application to Automatic Patient Message Triage (*Proceedings of Machine Learning Research*, Vol. 126), Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens (Eds.), PMLR, Virtual, 436–456. <http://proceedings.mlr.press/v126/si20a.html>
- [38] Y. Tian, L. Zhao, X. Peng, and D. Metaxas. 2019. Rethinking kernel methods for node representation learning on graphs. In *Advances in Neural Information Processing Systems*. 11681–11692.
- [39] Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research* 2, Nov (2001), 45–66.
- [40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [41] B. Wang, J. Jiang, W. Wang, Z. H. Zhou, and Z. Tu. 2012. Unsupervised metric fusion by cross diffusion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. IEEE, 2997–3004.
- [42] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7794–7803.
- [43] Z. Wu, Shirui Pan, Fengwen C., G. Long, C. Zhang, and P. S. Yu. 2019. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596* (2019).
- [44] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., 5753–5763. <https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf>
- [45] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*. 1480–1489.
- [46] Junbo Yin, Jianbing Shen, Chenye Guan, Dingfu Zhou, and Ruigang Yang. 2020. LiDAR-Based Online 3D Video Object Detection With Graph-Based Message Passing and Spatiotemporal Transformer Attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [47] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. 2017. Deep sets. In *Advances in Neural Information Processing Systems*. 3391–3401.
- [48] C. Zhang, J. C. Platt, and P. Viola. 2006. Multiple instance boosting for object detection. In *Advances in neural information processing systems (NIPS)*. 1417–1424.
- [49] Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Attention-based LSTM Network for Cross-Lingual Sentiment Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 247–256. <https://doi.org/10.18653/v1/D16-1024>

APPENDIX

Proof of Proposition 1: Let $\square^{(s)}$ denote the sth layer; the output of the second layer, $\mathbf{X}^{(2)}$, is given, similarly to (5), by:

$$\mathbf{X}^{(2)} = \mathbf{W}_0^{(1)} \mathbf{X}^{(1)} \mathbf{V}_1^{(1)} \equiv \mathbf{X}^{(1)} \mathbf{V}_0^{(1)} \mathbf{V}_0^{(1)T} \mathbf{X}^{(1)T} \mathbf{V}_1^{(1)}, \quad (15)$$

where for compactness we omit the subscript ($l = 1$). By plugging $\mathbf{X}^{(1)} = \mathbf{X} \mathbf{V}_0^{(0)} \mathbf{V}_0^{(0)T} \mathbf{X}^T \mathbf{X} \mathbf{V}_1^{(0)}$, where $\mathbf{X} \equiv \mathbf{X}^0$, into (15), we get:

$$\begin{aligned} \mathbf{X}^{(2)} = & \mathbf{X} \mathbf{V}_0^{(0)} \mathbf{V}_0^{(0)T} \mathbf{X}^T \cdot \\ & \mathbf{X} \mathbf{V}_1^{(0)} \mathbf{V}_0^{(0)T} \mathbf{V}_0^{(1)T} \mathbf{V}_1^{(0)T} \mathbf{X}^T \cdot \\ & \mathbf{X} \mathbf{V}_0^{(0)} \mathbf{V}_0^{(0)T} \mathbf{X}^T \cdot \\ & \mathbf{X} \mathbf{V}_0^{(0)} \mathbf{V}_0^{(0)T} \mathbf{X}^T \cdot \\ & \mathbf{X} \mathbf{V}_1^{(0)} \mathbf{V}_1^{(1)} \end{aligned} \quad (16)$$

Accordingly, the overall learned kernel $\mathbf{X}^{(2)} \mathbf{X}^{(2)T}$ has the form of the product of kernels as in (4) with $L = 1$.

We prove the general case with the composition of S layers by induction, where the base case is proven above. The representation obtained by the composition of S layers is given by:

$$\mathbf{X}^{(S)} = \mathbf{W}_0^{(S-1)} \mathbf{X}^{(S-1)} \mathbf{V}_1^{(S-1)}. \quad (17)$$

By induction assumption, $\mathbf{X}^{(S-1)} (\mathbf{X}^{(S-1)})^T$ has the form of a product of kernels, so that $\mathbf{X}^{(S-1)}$ has the form $\prod_{m=0}^{M-1} \mathbf{W}_m \mathbf{X} \mathbf{V}_m$, with some matrices $\{\mathbf{W}_m\}, \mathbf{V}_m$. By plugin $\mathbf{X}^{(S-1)}$ into the last equation we obtain:

$$\mathbf{X}^{(S)} = \mathbf{W}_0^{(S-1)} \prod_{m=0}^{M-1} \mathbf{W}_m \mathbf{X} \mathbf{V}_m \mathbf{V}_1^{(S-1)}. \quad (18)$$

Therefore:

$$\mathbf{X}^{(S)}(\mathbf{X}^{(S)})^T = \mathbf{W}_0^{(S-1)} \cdot \prod_{m=0}^{M-1} \mathbf{W}_m. \quad (19)$$

$$\mathbf{XV}_M \mathbf{V}_1^{(S-1)} (\mathbf{V}_1^{(S-1)})^T \mathbf{V}_M^T \mathbf{X}^T. \quad (20)$$

$$\prod_{m=M-1}^0 \mathbf{W}_m \cdot \mathbf{W}_0^{(S-1)}, \quad (21)$$

which has the form of the product of kernels, similarly to (4) with $L = 1$.

Proof of Proposition 3: Suppose we have a permutation $\sigma(\cdot) : \{1, \dots, N\} \rightarrow \{1, \dots, N\}$, which can be expressed as a permutation matrix $\mathbf{P} \in \mathbb{R}^{N \times N}$:

$$\begin{cases} \mathbf{P}(i, j) = 1, \text{ if } i = \sigma(j) \\ 0, \text{ otherwise} \end{cases} \quad (22)$$

Any permutation matrix $\mathbf{P}$ is invertible, with $\mathbf{P}^{-1} = \mathbf{P}^T$, and $\mathbf{PX}$ corresponds to a permutation of the rows of $\mathbf{X}$. We define permutation equivariance for a map $f(\cdot)$ as $f(\mathbf{PX}) = \mathbf{P}f(\mathbf{X})$. We start by showing that the map $f_{\theta, l}(\mathbf{X})$, (WLOG, assume $l = 1$) in (5) is permutation equivariant.

$$f_{\theta, 1}(\mathbf{PX}) = (\mathbf{PX})\mathbf{V}_0\mathbf{V}_0^T(\mathbf{PX})^T(\mathbf{PX})\mathbf{V}_1 \quad (23)$$

$$= \mathbf{PXV}_0\mathbf{V}_0^T\mathbf{X}^T(\mathbf{P}^T\mathbf{P})\mathbf{XV}_1 = \mathbf{P}f_{\theta, 1}(\mathbf{X}), \quad (24)$$

where we omitted the subscripts ($l = 1$) from $\mathbf{W}$ and $\mathbf{V}$. Therefore, $f_{\theta, 1}(\mathbf{X})$ is permutation equivariant.

We proceed showing, by induction, the permutation equivariance (of the unnormalized version) of $f_{\theta, 1}^{\text{prod}}(\mathbf{X}) \triangleq \prod_{m=0}^{M-1} \mathbf{W}_m \mathbf{XV}_M$, where here also we omitted the subscript ($l = 1$).

For $M = 1$, $f_{\theta, 1}^{\text{prod}}(\mathbf{X}) \equiv f_{\theta, 1}(\mathbf{X})$, which is permutation equivariant as shown above. For $M \geq 1$, we have:

$$f_{\theta, 1}^{\text{prod}}(\mathbf{PX}) = \prod_{m=0}^{M-1} (\mathbf{PX})\mathbf{V}_m\mathbf{V}_m^T(\mathbf{PX})^T(\mathbf{PX})\mathbf{V}_M \quad (25)$$

$$= (\mathbf{PX})\mathbf{V}_0\mathbf{V}_0^T(\mathbf{PX})^T \prod_{m=1}^{M-1} (\mathbf{PX})\mathbf{V}_m\mathbf{V}_m^T(\mathbf{PX})^T(\mathbf{PX})\mathbf{V}_M. \quad (26)$$

by induction assumption, the product of $M - 1$ kernels is permutation equivariant, so that the last equation equals to:

$$(\mathbf{PX})\mathbf{V}_0\mathbf{V}_0^T(\mathbf{PX})^T \mathbf{P} \prod_{m=1}^{M-1} \mathbf{XV}_m\mathbf{V}_m^T\mathbf{X}^T\mathbf{XV}_M \quad (27)$$

$$= \mathbf{PXV}_0\mathbf{V}_0^T\mathbf{X}^T\mathbf{P}^T\mathbf{P} \prod_{m=1}^{M-1} \mathbf{XV}_m\mathbf{V}_m^T\mathbf{X}^T\mathbf{XV}_M \quad (28)$$

$$= \mathbf{PXV}_0\mathbf{V}_0^T\mathbf{X}^T \prod_{m=1}^{M-1} \mathbf{XV}_m\mathbf{V}_m^T\mathbf{X}^T\mathbf{XV}_M \quad (29)$$

$$= \mathbf{P} \prod_{m=0}^{M-1} \mathbf{XV}_m\mathbf{V}_m^T\mathbf{X}^T\mathbf{XV}_M \quad (30)$$

$$= \mathbf{P}f_{\theta, 1}^{\text{prod}}(\mathbf{X}). \quad (31)$$

Therefore, $f_{\theta, l}^{\text{prod}}$ is permutation equivariant $\forall M \in \mathbb{N}$, by induction.

The proof for the row normalized case, where $\mathbf{W}$ is replaced with $\bar{\mathbf{U}}$ is similar. The concatenation of L such representations, motivated by the sum of kernels, provides a permutation equivariant representation as well since the concatenation does not change the order of the instances. Therefore, the representation $f_{\theta}(\mathbf{X})$ of affinitention nets, given by (10) and (12) is permutation equivariant.

Table 5: Affinitention architecture, trained with SGD, lr=1e-3, momentum=0.99, weight decay=1e-7. (Top) Pathology data. $L = 4$. (Bottom) Set-Cifar10. $L = 8$.

Layer/architecture	Output size
VGG11 feature extractor	4096
Linear+BatchNorm1d+ReLU	16
Affinitention	16
SkipConnection	16
LayerNorm+Linear+ReLU+dropout	16
LayerNorm+dropout	16
SkipConnection	16
BatchNorm1d+ReLU+Linear	1

Layer/architecture	Output size
Conv2d+ReLU+BatchNorm2d+MaxPool2d	$64 \times 16 \times 16$
Conv2d+ReLU+BatchNorm2d+MaxPool2d	$128 \times 8 \times 8$
Conv2d+ReLU+BatchNorm2d+MaxPool2d	$128 \times 4 \times 4$
Linear+BatchNorm1d+ReLU	512
Affinitention	16
SkipConnection	16
LayerNorm+Linear+ReLU	16
SkipConnection	16
Linear+ReLU+BatchNorm1d	16
Linear	1

Table 6: Affinitention architecture for patient text message triage data, trained with adam, lr=2e-5.

Layer/architecture	Output size
Embedding Layer	128×768
Affinitention	128×76
BatchNorm1d+ReLU+Linear	3