



King's Research Portal

Document Version
Peer reviewed version

[Link to publication record in King's Research Portal](#)

Citation for published version (APA):

Joutard, S. R. D., Dorent, R. P. R., Isaac, A., Ourselin, S., Vercauteren, T., & Modat, M. (in press). Permutohedral Attention Module for Efficient Non-Local Neural Networks. In Medical Image Computing and Computer Assisted Intervention – MICCAI 2019

Citing this paper

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

General rights

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Research Portal

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Permutohedral Attention Module for Efficient Non-Local Neural Networks

Samuel Joutard, Reuben Dorent, Amanda Isaac, Sebastien Ourselin,
Tom Vercauteren, Marc Modat

School of Biomedical Engineering & Imaging Sciences, King's College London

Abstract. Medical image processing tasks such as segmentation often require capturing non-local information. As organs, bones, and tissues share common characteristics such as intensity, shape, and texture, the contextual information plays a critical role in correctly labeling them. Segmentation and labeling is now typically done with convolutional neural networks (CNNs) but the context of the CNN is limited by the receptive field which itself is limited by memory requirements and other properties. In this paper, we propose a new attention module, that we call Permutohedral Attention Module (PAM), to efficiently capture non-local characteristics of the image. The proposed method is both memory and computationally efficient. We provide a GPU implementation of this module suitable for 3D medical imaging problems. We demonstrate the efficiency and scalability of our module with the challenging task of vertebrae segmentation and labeling where context plays a crucial role because of the very similar appearance of different vertebrae.

Keywords: Non-local neural networks · Attention module · Permutohedral Lattice · Vertebrae Segmentation

1 Introduction

Convolutional neural networks (CNNs) have become one of the most effective tools for many medical image processing tasks such as segmentation. However, working with medical images has its own idiosyncratic challenges. The organs, tissues or bones can have very similar characteristics, such as intensity, texture, or shape. As a consequence, the differentiating aspects of each individual structure come from the context and the position of the item of interest in the larger surroundings. However, naively extracting non-local characteristics of a region requires much more computation and memory than focusing on its local characteristics. This currently makes using non-local context highly non-trivial in medical imaging. Hence, an efficient approach to exploit non-local characteristics in deep learning could transform several medical imaging pipelines.

The notion of contextual information is intimately related to the concept of receptive field in deep learning. The receptive field of an output variable corresponds to the region in the input influencing its value. Recent studies on receptive field in CNNs [9] have proven that the receptive field size is sub-linear

in the number of convolutional layers. In order to improve the receptive fields of a CNN, two main solutions have been adopted: down-sampling layers and dilated convolutions [13]. Use of down-sampling layers efficiently increases the receptive field size but decreases the resolution of the information. Hence, it is not suitable for very granular segmentation in which case dilated convolutions are often preferred [8]. Both of these solutions result in a fixed receptive field, which means that all contextual information in the receptive field will be taken into account whether it is relevant or not. Attention modules have been used to prune irrelevant information in medical imaging [11,14]. Yet, these tools remain suboptimal as they do not allow to capture large scale context. However, the extended self-attention formulation of [12] offers a solution to dynamically adapt the individual receptive field of each output variable to only make use of relevant non-local information. Despite its attractive properties, this formulation of self-attention has not yet been applied to medical images partly because its computational requirements scale as $O(N^2)$ (N is the number of voxels).

In this paper, we propose a new self-attention module called Permutohedral Attention Module (PAM), which makes use of the efficient algorithm approximation of the Permutohedral Lattice [1]. We adapted the algorithm of [1], originally designed to perform denoising, into a trainable module able to capture and process contextual information. Our module, similarly to the original non-local self-attention mechanism formulation, dynamically adapts the receptive field of each variable in a learned way while being, in contrast to [12], applicable to medical images as it has low memory requirements, computationally scaling as $O(N)$. We evaluate our module on the challenging task of vertebrae segmentation. Vertebrae segmentation aims to label each individual vertebra and is used in practice as an initial step of various pipelines such as modality fusion, spine surgery planning and surgical guidance. As consecutive vertebrae have very similar local appearance, non-local information is compelling to identify them.

In Section 2, we first define self-attention and how it has been used, we then introduce the PAM. In section 3 we first highlight the capability of our module to capture and process contextual information without requiring a deep architecture. Then, we demonstrate its capability to improve state of the art segmentation architectures for vertebrae segmentation.

2 Methods

The self-attention mechanism Self-attention used in deep learning frameworks can be defined as follows: consider a standard deep learning framework where the input $\mathbf{x} = (x_1, \dots, x_N)$ is processed first by a section of the network we call descriptor network \mathbf{v}_ψ , and then by the rest of the network we call prediction network g_θ (ψ and θ are the respective parameter sets). The model predicts \mathbf{y} so that:

$$\mathbf{y} = g_\theta(\mathbf{v}_\psi(\mathbf{x})) \quad (1)$$

We define $A_\phi(\cdot)$ the self-attention mechanism parameterized by ϕ which combines the non-local input descriptors in a learned way. For all input \mathbf{x} , $A_\phi(\mathbf{x})$ is

a $N \times N$ self-attention matrix where the coefficient $A_\phi(\mathbf{x})_{i,j}$ characterizes the attention of x_i towards x_j . Our framework including an attention mechanism predicts \mathbf{y}_{att} :

$$\mathbf{y}_{att} = g_\theta(A_\phi(\mathbf{x}) \cdot \mathbf{v}_\psi(\mathbf{x})) \quad (2)$$

where \cdot represents the matrix multiplication operator. This formulation has two principal strengths; it can increase the receptive field of each output variable up to the whole input, and it can modulate the receptive field of each output variable with respect to the input characteristics. To our knowledge, attention modules in deep learning either compute the entire self-attention matrix on a low dimensional input or use a local attention mechanism that can be seen as a strong approximation of the non-local self-attention formulation. Specifically in the medical imaging context, previous works [11,14,10] implicitly used a simplification of (2) with a diagonal self-attention matrix. This solution can be applied to large images since it scales linearly with the number of voxels but does not help to capture contextual information.

Different implementations of the non-local self-attention matrix are listed in [12]. These can be unified as follows:

$$A_\phi(\mathbf{x}) = [\gamma(\phi_1(x_i)^T \cdot \phi_2(x_j))]_{1 \leq i, j \leq N} \quad (3)$$

where $\phi = (\phi_1, \phi_2)$ is a pair of embedding functions (possibly identities) and γ is typically either identity, exponential or ReLU. Hence, these approaches are impractical to apply to 3D images because the number of interactions to be computed scales as $O(N^2)$.

Permutohedral Attention Module The proposed PAM relies on a slightly different formulation of the self-attention matrix to align more closely with the formulation of the non-local means filtering algorithm [3] used in the denoising literature. When applied to the set of feature-descriptor pairs $(f_i, v_i)_{i \leq n}$ (where n is the number of variables described), non-local mean gives the set of filtered descriptors:

$$\forall 1 \leq i \leq n, \quad v'_i = \sum_{j=1}^n \exp(-\|f_i - f_j\|_2) v_j \quad (4)$$

Hence:

$$A_\phi(\mathbf{x}) = [\exp(-\|f_\phi(x_i) - f_\phi(x_j)\|_2)]_{i, j \leq N}$$

is the corresponding attention formulation with f_ϕ a feature extractor network (ϕ its parameter set).

Avoiding a brute-force computation of (4), we adapted the Permutohedral Lattice approximation algorithm [1] to estimate the self-attention module output $\mathbf{v}'_{\phi, \psi}(\mathbf{x}) = A_\phi(\mathbf{x}) \cdot \mathbf{v}_\psi(\mathbf{x})$ in $O(N)$ against $O(N^2)$ for the original non-local neural network formulations listed in [12]. Learning the parameter sets ϕ and ψ is achieved through back-propagation. Hence, the PAM can be integrated in a deep learning framework to compute self-attention for high dimensional inputs (cf. Section 3 for concrete architectures examples). The PAM approximates

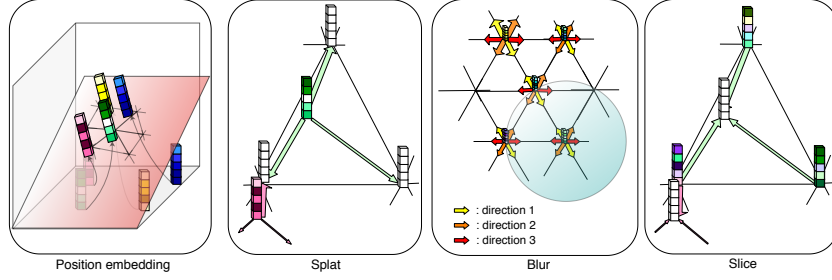


Fig. 1. The features lying in \mathbb{R}^f are embedded in a hyperplane of \mathbb{R}^{f+1} to position each variable. This hyperplane is partitioned in simplices by a mesh called the Permutohedral Lattice. The Splat phase describes the vertices of the Permutohedral Lattice based on the neighbouring variables. The Blur step applies a Gaussian blur along each direction consecutively. Finally, the Slice step re-projects the filtered descriptors from the vertices to the variables.

the proposed attention mechanism in 4 steps: embedding of the features into the Permutohedral Lattice higher dimensional space, Splat, Blur and Slice, as illustrated in Fig. 1. Each of these steps scales linearly in N .

The advantage of this approximation algorithm against other possibilities [2,5] is that the gradients with respect to the input feature vectors $\mathbf{f}_\phi(\mathbf{x})$ and the descriptor vectors $\mathbf{v}_\psi(\mathbf{x})$ can be expressed using the four steps composing the forward pass and be fully parallelized. Omitting the dependencies in \mathbf{x} , ϕ , and ψ , we can express the forward pass as:

$$\mathbf{v}' = Sl(\mathcal{B}(\mathcal{S}(\mathcal{E}(\mathbf{f}), \mathbf{v})), \mathcal{E}(\mathbf{f})) \quad (5)$$

where \mathcal{E} is the embedding operator, \mathcal{S} is the Splat operator, \mathcal{B} is the Blur operator and Sl is the Slice operator. With the same notations, the backward pass can be expressed as:

$$\frac{\partial L}{\partial \mathbf{v}} = \frac{\partial L}{\partial \mathbf{v}'} \circ \frac{\partial \mathbf{v}'}{\partial \mathbf{v}} = Sl(\tilde{\mathcal{B}}(\mathcal{S}(\mathcal{E}(\mathbf{f}), \frac{\partial L}{\partial \mathbf{v}'})), \mathcal{E}(\mathbf{f})) \quad (6)$$

$$\frac{\partial L}{\partial f_i} = \frac{\partial L}{\partial \mathbf{v}'} \circ \frac{\partial \mathbf{v}'}{\partial f_i} = E^T \cdot \left[\tilde{\mathcal{B}}\mathcal{S}\mathcal{E}(\mathbf{f}, \frac{\partial L}{\partial \mathbf{v}'})_{\sigma_i^j} \cdot v_i^T + \mathcal{B}\mathcal{S}\mathcal{E}(\mathbf{f}, \mathbf{v})_{\sigma_i^j} \cdot \left(\frac{\partial L}{\partial \mathbf{v}'_i} \right)^T \right]_{j \leq (f+1)} \quad (7)$$

where L is the loss and $\forall a, b \mathcal{B}\mathcal{S}\mathcal{E}(a, b) = \mathcal{B}(\mathcal{S}(\mathcal{E}(a), b))$ (similarly with $\tilde{\mathcal{B}}\mathcal{S}\mathcal{E}$). $\tilde{\mathcal{B}}$ is the Gaussian blurring operator where the Gaussian blur is applied in the reverse order in terms of direction of the Lattice. E is the position embedding matrix and σ_i is permutation computed during \mathcal{E} .

3 Experiments

Data We evaluate the impact of PAM for non-local neural networks for the task of simultaneous segmentation and labeling of vertebrae. We performed our

experiment on the CSI 2014 workshop challenge data¹, which consists of 20 CT images. We used all 20 CT images in our framework using a 5-fold cross validation for evaluation. We resampled the data to obtain (1mm, 1mm, 3mm) voxels.

Implementation details We implemented the PAM as well as all our pipelines using Pytorch. We optimized our networks with ADAM on $160 \times 160 \times 96$ patches with a fixed learning rate of 0.001 and a batch size of 1. We used the Dice loss as loss function. Our implementation is publicly available².

Models As a preliminary experiment, we consider a specific 6-layer fully convolutional network (referred to as FCN). We design 2 baselines for this shallow setting. FCN is a plain fully convolutional network with a first $(3 \times 3 \times 3)$ convolution with 18 output channels followed by 4 $(3 \times 3 \times 3)$ embedding convolutions with 18 output channels each and a prediction $(1 \times 1 \times 1)$ convolutional layer. Dil.FCN is a similar architecture where we replace each embedding convolution by a dilated block. A dilated block corresponds to 3 $(3 \times 3 \times 3)$ convolutions in parallel with 6 output channels each. Of these 3 convolutions, two have dilated filters (dilatation factor of 2 and 4 respectively). The outputs of a dilated block are then concatenated before the next block. Then, we incorporate in each baseline the PAM (networks are respectively called FCN+PAM and Dil.FCN+PAM) and compare the results of those 4 configurations.

Figure 2 represents the Dil.FCN+PAM architecture. In this figure, we observe that, once we obtain the features $\mathbf{f} = (f_1, \dots, f_N)$ and descriptors $\mathbf{v} = (v_1, \dots, v_N)$ to compute attention, we split each feature and descriptor vector in two. Hence we obtain two sets of feature-descriptor pairs $(f_i^0, v_i^0)_{i \leq N}$ and $(f_i^1, v_i^1)_{i \leq N}$ on which we apply the PAM independently. There are two main advantages to doing so. First, it allows us to further reduce computation time and memory footprint. Second, it generates a per-group-of-channel attention map which makes the model more flexible (as a unique attention matrix for all descriptor channels is a particular case of two attention matrix, one for each group of channel). The reason for not splitting the feature-descriptor pairs set into more subsets is because we want a trade-off between the advantages described above and the preservation of relevant features to compute attention.

Then, we consider a 3D U-Net [4] which is one of the most popular architectures for segmentation [6]. We refer to our 3D U-Net simply as U-Net. We incorporate the PAM into our U-Net as shown in Fig. 3 and demonstrate that the PAM can also improve architectures which have large receptive fields (we call this network U-PAM-Net). As shown in Fig. 3, we incorporate the PAM at the half-resolution level. Hence, we compute attention for $(2 \times 2 \times 2)$ voxel regions which, in our experiments, led to similar results as computing attention at the voxel level while decreasing computation time and making convergence faster.

¹ <http://spineweb.digitalimaginggroup.ca/>

² https://github.com/SamuelJoutard/Permutohedral_attention_module

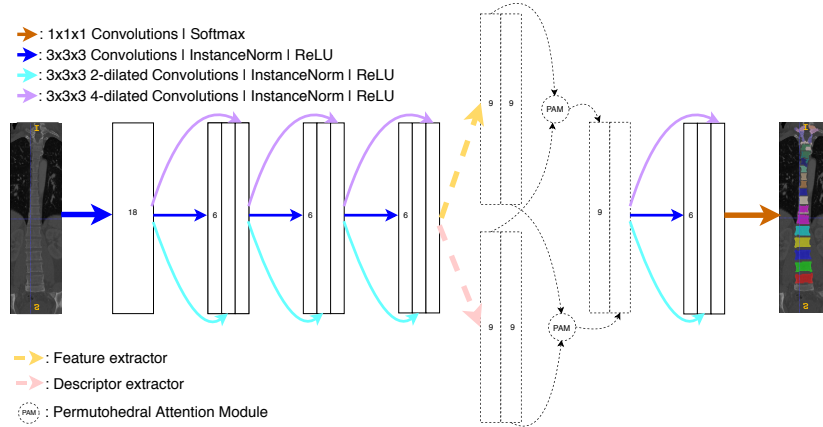


Fig. 2. Dil.FCN+PAM, a shallow architecture including dilated convolutions and PAM. The feature extractor and descriptor extractor are $(1 \times 1 \times 1)$ convolutions. The feature extractor incorporates a mesh of spatial coordinates before applying its convolution, and is followed by a Leaky-ReLU activation function. The number correspond at each stage corresponds to the number of channels. We call the combination of the dashed elements Permutohedral block.

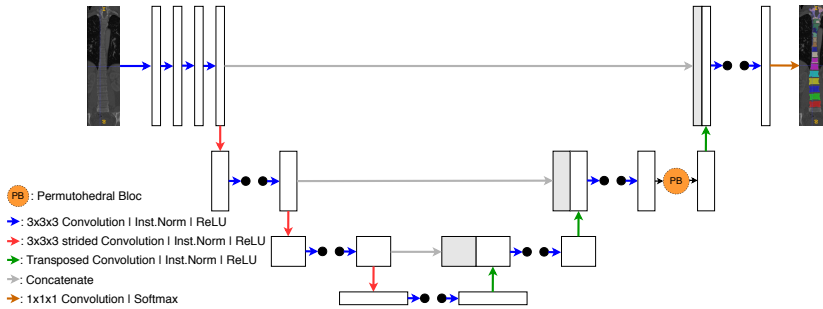


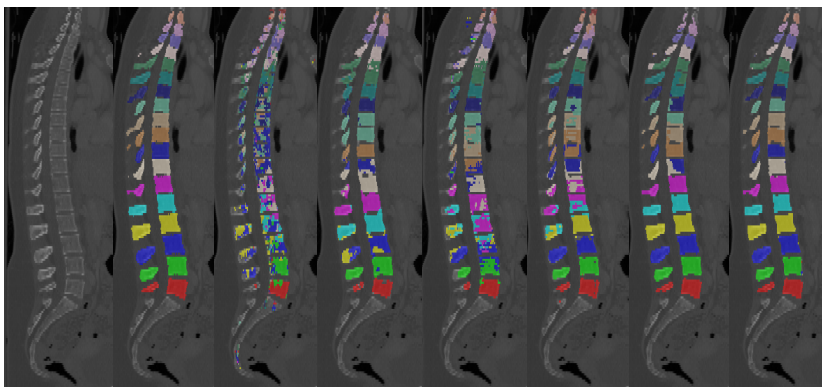
Fig. 3. U-PAM-Net. We make use of the Permutohedral block defined in Fig. 2.

As the PAM introduces a small number of extra parameters, we compensate with additional channels in the first convolution on the architectures without the PAM so that the corresponding networks have either as many as or more degrees of freedom than networks with the PAM integrated.

Results We measure the performance of the different architectures with the Dice scores. Table 1 shows that the PAM improves performance for all the architectures it was incorporated into. In addition, we highlight that the shallow network Dil.FCN+PAM performs almost as well as the much deeper network 3D U-Net. Indeed, the dilated convolutions manage to describe the voxels using contextual information while the PAM uses those meaningful features to com-

Table 1. Mean(std) Dice score (%) of the different networks tested

Network	FCN	FCN+PAM	Dil.FCN	Dil.FCN+PAM	U-Net	U-PAM-Net
Full	28(3)	54 (14)	49(7)	70 (9)	72(9)	81 (9)
Cervical	45(30)	53 (37)	60 (27)	56(33)	25(38)	55 (39)
Thoracic	21(3)	46 (14)	42(10)	67 (12)	67(13)	80 (12)
Lumbar	29(4)	69 (19)	58(8)	79 (12)	93 (2)	91(5)

**Fig. 4.** Example of segmentation obtained by our different networks. In the corresponding order: Input slice, ground truth, FCN, FCN+PAM, Dil.FCN, Dil.FCN+PAM, U-Net, U-PAM-Net.

pute voxels interactions. Table 1 also illustrates the limitation of down-sampling layers pointed earlier as U-Net performs poorly on cervical vertebrae which appear very small in our images. U-PAM-Net manages to reach higher accuracy performances than [7], which makes use of a task-specific framework especially tuned to "count" the vertebrae from spine segmentation. While [7] report an accuracy of 81%, our proposed framework obtained 89% using the same evaluation metric and on the same dataset. It should be noted that the training frameworks in terms of test-train split were different for both approaches. Figure 4 shows a representative example of the results we observed.

4 Discussion

In this work, we propose the Permutohedral Attention Module, a computationally efficient attention module to be applied in 3D deep learning framework. The PAM can be incorporated in any CNN architecture. We demonstrated its ability to efficiently handle non-local information in the context of vertebrae segmentation and presented its potential to reduce networks size in specific tasks. Future

work will notably include the investigation of asymmetric attention matrix for feature filtering and the integration of the PAM formulation in path training.

Acknowledgement We thank E. Molteni, C. Sudre, B. Murray, K. Georgiadis, Z. Eaton-Rosen, M. Ebner for their useful comments. This work is supported by the Wellcome/EPSRC Centre for Medical Engineering [WT 203148/Z/16/Z]. TV is supported by a Medtronic / RAEng Research Chair [RCSRF1819/7/34].

References

1. Adams, A., Baek, J., Davis, M.A.: Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum* (2010)
2. Adams, A., Gelfand, N., Dolson, J., Levoy, M.: Gaussian KD-trees for fast high-dimensional filtering. *ACM Trans. Graph.* **28**(3), 21:1–21:12 (Jul 2009)
3. Buades, A., Coll, B.: A non-local algorithm for image denoising. In: *CVPR*. pp. 60–65 (2005)
4. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In: *MICCAI* (2016)
5. Chen, J., Paris, S., Durand, F.: Real-time edge-aware image processing with the bilateral grid. In: *ACM SIGGRAPH 2007 Papers. SIGGRAPH '07* (2007)
6. Isensee, F., Petersen, J., Klein, A., Zimmerer, D., Jaeger, P.F., Kohl, S., Wasserthal, J., Koehler, G., Norajitra, T., Wirkert, S.J., Maier-Hein, K.H.: nnU-Net: Self-adapting framework for u-net-based medical image segmentation. *arXiv preprint arXiv:1809.10486* (2018)
7. Lessmann, N., van Ginneken, B., de Jong, P.A., Isgum, I.: Iterative fully convolutional neural networks for automatic vertebra segmentation and identification. *Medical Image Analysis* **53**, 142–155 (2019)
8. Li, W., Wang, G., Fidon, L., Ourselin, S., Cardoso, M.J., Vercauteren, T.: On the compactness, efficiency, and representation of 3D convolutional networks: Brain parcellation as a pretext task. In: *IPMI* (2017)
9. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. In: *Advances in Neural Information Processing Systems 29*. pp. 4898–4906 (2016)
10. Roy, A.G., Navab, N., Wachinger, C.: Recalibrating fully convolutional networks with spatial and channel "squeeze and excitation" blocks. *IEEE transactions on medical imaging* **38**(2), 540–549 (2019)
11. Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., Rueckert, D.: Attention gated networks: Learning to leverage salient regions in medical images. *Medical Image Analysis* **53**, 197 – 207 (2019)
12. Wang, X., Girshick, R.B., Gupta, A., He, K.: Non-local neural networks. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 7794–7803 (2018)
13. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: *International Conference on Learning Representations (ICLR)* (2016)
14. Zhang, Z., Xie, Y., Xing, F., McGough, M., Yang, L.: MDNet: A semantically and visually interpretable medical image diagnosis network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 3549–3557 (2017)