

# Saliency Clouds: Visual Analysis of Point Cloud-oriented Deep Neural Networks in DeepRL for Particle Physics

Raju Ningappa Mulawade<sup>1</sup> , Christoph Garth<sup>2</sup> , and Alexander Wiebel<sup>1</sup>  on behalf of the *Bergen pCT Collaboration*

<sup>1</sup>Hochschule Worms University of Applied Sciences, Germany, <sup>2</sup>TU Kaiserslautern, Germany

## Abstract

We develop and describe saliency clouds, that is, visualization methods employing explainable AI methods to analyze and interpret deep reinforcement learning (DeepRL) agents working on point cloud-based data. The agent in our application case is tasked to track particles in high energy physics and is still under development. The point clouds contain properties of particle hits on layers of a detector as the input to reconstruct the trajectories of the particles. Through visualization of the influence of different points, their possible connections in an implicit graph, and other features on the decisions of the policy network of the DeepRL agent, we aim to explain the decision making of the agent in tracking particles and thus support its development. In particular, we adapt gradient-based saliency mapping methods to work on these point clouds. We show how the properties of the methods, which were developed for image data, translate to the structurally different point cloud data. Finally, we present visual representations of saliency clouds supporting visual analysis and interpretation of the RL agent's policy network.

## CCS Concepts

• *Human-centered computing* → *Visualization techniques*; • *Computing methodologies* → *Neural networks*;

## 1. Introduction

In recent years, artificial intelligence (AI) has seen an exponential increase in its integration and deployment in a wide variety of sectors. This increase in the application of AI has led to the increased demand for explainable AI (XAI) algorithms that can explain or at least help to explore [ACJ19,ZLZ\*20] these complex AI models to help developers in debugging and development and also to increase the user's trust in these models. Explainable reinforcement learning (XRL) is a subfield of XAI which focuses on explaining RL agents, e.g. the deep neural networks in the RL agent (DeepRL). Image data has been one of the most common datatypes for the research work in XAI and XRL. However, the use of 3D data has seen an increased use due to the ubiquitousness of sensors yielding point-clouds in applications such as autonomous driving and unmanned aerial vehicles (UAVs). The point-cloud data considered in this paper stems from yet another application which is a detector (Figure 1) in particle/medical physics and the RL agent in our application case is tasked to reconstruct particle trajectories from the hits in this detector. The agent is still under development. Our work is aimed at supporting the development and debugging of this and other agents working on similar data. Our main contributions are:

- Visualization of saliency attributions in 2D (aggregated) and 3D to analyze the *policy* output of the deep RL agent.
- Adaptation of saliency methods to the agent's point cloud based neural network (NN), in particular we define
  - suitable baselines for integrated gradients [STY17] and

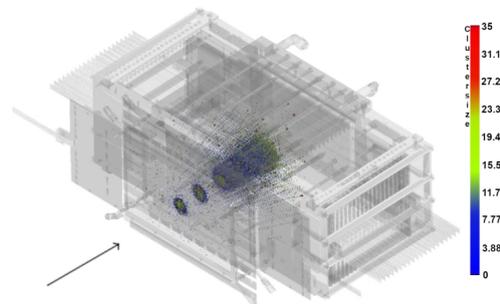


Figure 1: Illustration of detector with hits of 10k simulated protons entering the 43 detector layers. The arrow indicates the proton beam direction. Hits in the detector layers are color coded by the number of detector pixels firing at the same time (cluster size).

- suitable noise for features of different range for Smooth-Grad [STK\*17]

## 2. Related Work

A significant amount of XAI literature deals with the analysis of supervised learning algorithms, one of the types of ML algorithms. Gradients [SVZ13], Guided Backpropagation [SDBR14], Deconvolution [ZF14], integrated gradi-

ents [STY17], SmoothGrad [STK\*17], Class Activation Mapping (CAM) [ZKL\*16], Gradient-weighted Class Activation Mapping (Grad-CAM) [SCD\*17] are some of the significant works in the last decade towards the explanation of ML models. However, the research work in XAI field with a focus on point clouds as the input has been very limited or focused on recognizing objects described by the point clouds. One work of the latter category by Zheng et al. [ZCY\*19] proposes a gradient-based saliency method that generates saliency attributions for a point/set of points based on the change in prediction of the network when the point/set of points is excluded. Qi et al. [QYSG17] developed a point cloud classification which is self-explanatory in the sense that its learned features can be visualized by highly activating example point clouds.

XRL, on the other hand, has not seen such an extensive research work in the same period of time. The current XRL methods can be grouped into two categories: *Intrinsic methods* use ML models with transparent architecture which are self-explanatory such as linear regression and decision tree models. It also includes representation learning methods wherein a model learns to extract abstract features which explain the decision making process of the RL agent [HJTvdS20] [ZRS\*18] [RHT\*19] [RHT\*18] [ZWL\*20]. However, these methods lack clear explanations aimed at end users [HCDR21]. *Post-Hoc explainability* methods analyze the ML model after the training process. These methods are model agnostic, meaning the architecture of the ML model remains unaltered. For example, Greydanus, Samuel, et al. [GKDF18] presents a perturbation-based saliency method for analyzing the RL model trained to play Atari games. One of the main drawbacks of the perturbation-based saliency methods is the need for multiple passes to generate a saliency map corresponding to one input. Gradient-based saliency methods, on the other hand, use the backpropagation process to compute the derivative of the output value with respect to the input values of the RL agent [CWF\*20]. The XRL research work involving point clouds as the input data is very limited.

In this paper, we extend the gradient-based saliency methods to the analysis of RL agents with point clouds as the input. We implement ("vanilla") gradients [SVZ13], integrated gradients [STY17] and SmoothGrad [STK\*17] methods to compute saliency attributions and visually analyze the policy output of the RL agent.

### 3. Application Case: RL Particle Tracking System

The point-cloud data considered in this paper describes particle hits in a detector for proton computed tomography (pCT) [Joh17]. To develop a particle detector and the techniques essential for creating the pCT system, the *Bergen pCT Collaboration* [ABB\*20] has been established. The detector being developed is a digital tracking calorimeter (DTC) which is a multilayer structure made of several detector/absorber layers. Protons emitted during the imaging process are captured throughout their path in the DTC in each layer by multiple strips of ALPIDE silicon sensors [Mag16], providing both spatial information of the hits as well as the amount of deposited energy through the activated pixel cluster size. Due to frequent interactions throughout the detector, significantly influenced by the absorber plates, the trajectories of the particles tend to deviate from a straight line path, making the reconstruction of proton traces challenging.

In this work, we use the above-mentioned data with the density of 100 tracks per frame generated from the simulations performed during the development phase of the detector. The acquired data has the form of a layered point cloud (figure 1) and is used in the form of a graph implicitly given by the spatial neighborhood in the layers (figure 2). Nodes  $V$  in the graph represent hits in the detector layers and edges  $E$  represent possible parts of a reconstructed particle trajectory. The input data to the RL agent consists of features describing a one-step history of the tracking process over the last two reconstructed layers and features of the possible hits of the particles in the following layer.

**Model** RL models, in general, are value based or policy based or model based. A policy defines the action that a learning agent takes at a given state (in our case, the selection of the desired hit in the following layer when tracking a hit in the current layer), whereas the value of a state is defined as the total amount of reward that the learning agent can expect to accumulate over the future, starting from the current state. In this work, we use actor-critic model of RL which is a combination of policy and value based approaches.

## 4. Methods

In this section, we describe the adaptations we propose for the saliency methods working on point cloud data to reconstruct tracks of particles and which we employ for the visual analysis of the RL agent. The saliency methods are Integrated Gradients [STY17] and SmoothGrad [STK\*17].

### 4.1. Gradients as Saliency Measure

Gradient-based class saliency mapping has been proposed [SVZ13] for analyzing convolutional neural networks tasked for image classification. The saliency map is generated by computing the gradient of a particular class of interest with respect to the input image. The number of elements in the saliency map is equal to the number of pixels in the input image and each element indicates how much a minor change in pixel value can affect the output class value.

In this work, we compute the gradients of the *policy* output of the RL agent with respect to the input of the RL agent representing the features of the possible next nodes in the following layer. From here on, we will refer to the features of the node to be tracked as *observation features* and the features of the possible next nodes in the following layer as *action features* (see figure 3). A part of these features are computed based on the node in the previous layer. One of the major issues associated with using vanilla gradients to visually analyze NNs is that the generated saliency maps are usually noisy. The noise in the gradients could be the result of meaningless local variations in partial derivatives [STK\*17]. This fluctuation makes gradients non-reliable as saliency maps. The class acti-

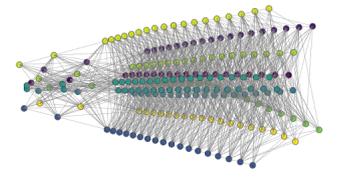


Figure 2: Graph representation of the particle data containing 10 tracks. Particles are moving from left to right and hits of the same particle have the same color.

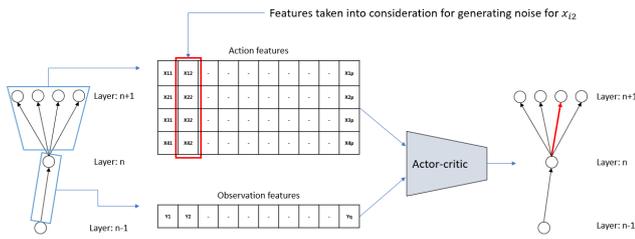


Figure 3: Noise generation process for the inputs of RL agent.

vation function  $S_c$  is also not continuously differentiable due to the presence of common activation functions such as Rectified Linear Units (ReLU) in the neural networks.

## 4.2. SmoothGrad

The SmoothGrad [STK\*17] method tries to address these issues by introducing noise into the input  $x$  and computing the average of the saliency maps produced for a given number  $n$  of noisy inputs. Given an input  $x$ , the predicted class  $c_p \in C$  returned by the network is determined by finding the argument  $c \in C$  of the maximum value in the class activation function  $S_c(x)$ , i.e.  $c_p = \operatorname{argmax}_{c \in C} S_c(x)$ . The gradients of class  $c_p$  with respect to input  $x$  are noted as  $M_c(x)$  like  $M_c(x) = \partial S_c(x) / \partial x$ . The SmoothGrad saliency map  $M_{SG}$  for input  $x$  and output class  $c$  can then be formulated as:

$$M_{SG} = \frac{1}{n} \sum_1^n (M_c(x + N(0, \sigma^2))) \quad (1)$$

Here,  $N(0, \sigma^2)$  represents the Gaussian noise with given standard deviation  $\sigma$  that is added to the input  $x$ , and  $n$  represents the number of noisy samples of  $x$  generated.

### 4.2.1. Necessary Adaption

For the analysis of policy network of the RL agent, we generate noisy samples of inputs by adding Gaussian noise. For this point cloud data, the range  $\delta$  and the  $\sigma$  parameters of the Gaussian noise are based on the variation in the input values. Unlike for image data where each pixel value varies in the range of 0-255 (or 0-1 depending on the normalization), the input variables of the RL agent's policy network vary differently when compared to each other. In other words, the ranges of the values of input variables of each node are independent of each other. Therefore, the range parameter  $\delta$  is computed for each input variable (see figure 3). We consider the *action features* for computing the mean and range of each variable and add Gaussian noise ( $\sigma = \delta \cdot 2\%$ ) to the variables. Figure 4 shows the effect of adding noise on the saliency maps generated. We observe that the noise level upto 10% produces a smoother saliency cloud. Saliency maps with noise level of 2% are visualized in figure 5. The number of samples to average over is set to  $n = 100$  which produces smoother gradients for the chosen noise level.

## 4.3. Integrated Gradients

As other post-hoc saliency mapping methods, the integrated gradients [STY17] method tries to explain the influence of input values

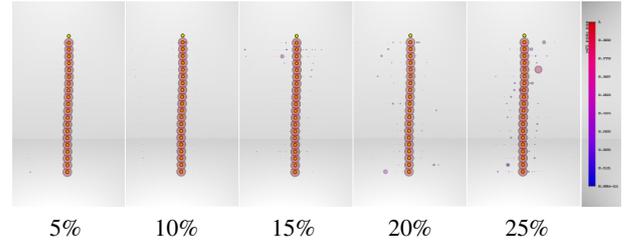


Figure 4: Effect of adding noise (as % of input values as shown above) to the input on the saliency maps generated by SmoothGrad.

Noise %	Track 1		Track 2	
	< 0.1	< 0.2	< 0.1	< 0.2
2	1	1	1	1
5	0.83	0.94	0.8	1
10	0.39	0.44	0.1	0.4
15	0	0.11	0	0

Table 1: Effect of noise on the output (with baseline as the input) when used as a baseline for integrated gradients method.

over the output value of the network. Here, we consider the network as a function  $F(x)$  that maps some input  $x$  to  $[0,1]$ . Further a *baseline* input  $x'$ , i.e. an input satisfying  $F(x') \approx 0$  is considered. Gradients of  $F(x)$  are computed at inputs interpolated (*straight line path*) between the baseline  $x'$  and the current input  $x^c$ . Finally, these gradients are cumulated to obtain integrated gradients. The baseline  $x'$  is central to this process and can be quite different for different types of input data. For image data usually, a black image satisfies the above mentioned condition for a baseline.

### 4.3.1. Necessary Adaption

In this work, we use Gaussian noise generated based on the input data as the baseline for the *action features*. Figure 3 shows the methodology of generating this noise. The noise  $x'$  we use as baseline has  $\sigma$  of 2% of the range of the input values and satisfies the condition  $F(x') \approx 0$  for every tracking step. Table 1 shows the fraction of layers in two example tracks that output values less than a threshold (0.1 and 0.2 in this case) for baselines with varying noise levels. Use of higher magnitude of noise results in  $F(x')$  being closer to zero for a part of the tracking steps but fails to satisfy this condition for every tracking step. The number of inputs generated along the straight line path is set to 200 based on how well it approximates the integral.

## 5. Results and Discussion

In this section, we look into the visualization of the saliency attributions computed using gradients, Integrated Gradients and SmoothGrad methods. We focus on the saliency attributions corresponding to the seven features which have a physical meaning in the real world. These seven features represent the energy deposition of the proton hit, Cartesian coordinates in the x and y axis, one-hot encoded z coordinate and the polar coordinates  $(r, \theta, \phi)$  of the proton hit compared to the previous hit.

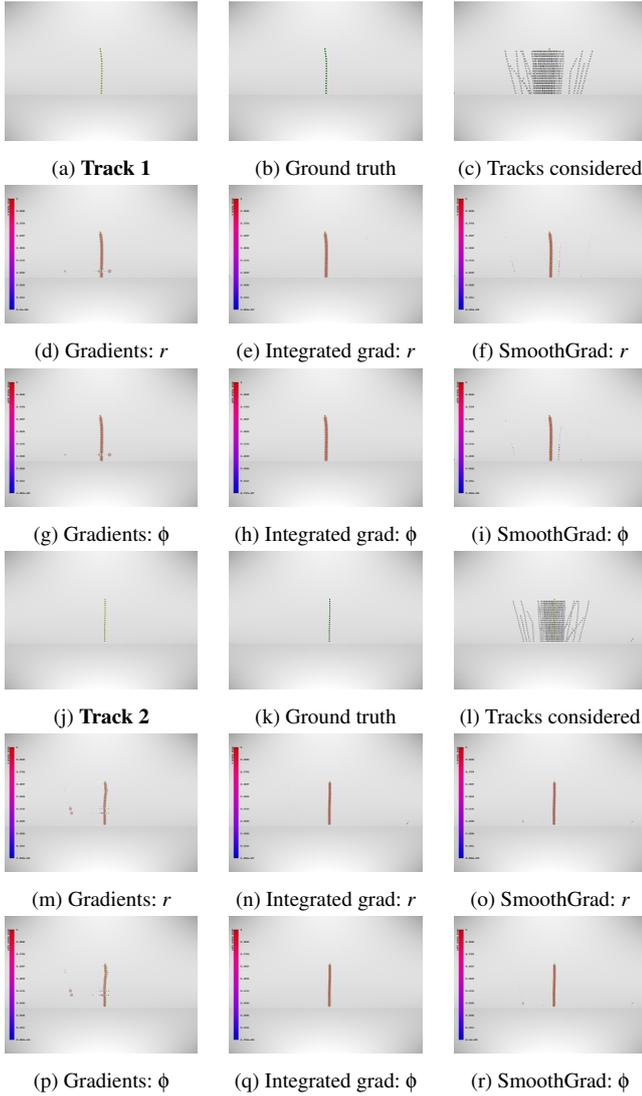


Figure 5: Visualization of the saliency attributions corresponding to the *action features*  $r$  and  $\phi$  for two reconstructed tracks (Track 1 and Track 2). The size and the color of translucent blobs indicate how important the node feature is for the *policy* output.

Figure 5 shows the normalized saliency attributions for the input features  $r$  and  $\phi$  of *action features* for two exemplary tracks. The saliency attributions of all nodes in a layer are normalized to (0,1) per layer. We focus on the saliency attributions of  $r$  and  $\phi$  since these two features are considered the most important features for tracking a particle and therefore, help us comparing the chosen methods with the gradients. Figures 5a-5i and 5j-5r show the reconstruction details and the saliency attributions corresponding to the  $r$  and  $\phi$  features of *action features* for tracks 1 and 2 respectively.

A closer inspection of the gradients computed for input features  $r$  and  $\phi$  for both the tracks (5d, 5g, 5m, 5p) shows that the gradients have higher activation for the neighboring nodes in a few layers indicating the presence of fluctuations. Figures 5f, 5i, 5o, 5r show

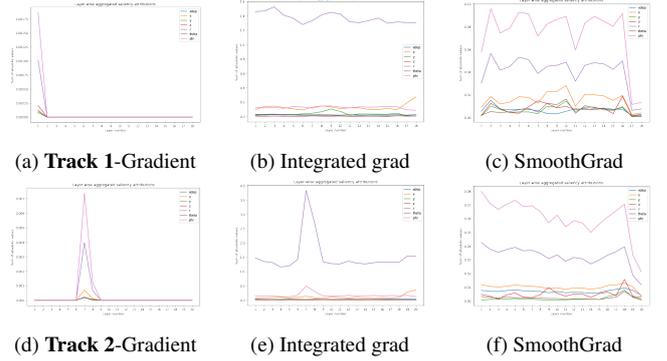


Figure 6: Visualization of the aggregated saliency attributions corresponding to the seven *action features* of the reconstructed tracks.

that the SmoothGrad method, by adding noise, addresses these fluctuations in the gradients quite effectively. It reinforces the idea of sharpening the gradient-based saliency maps by averaging the maps generated from multiple noisy copies of the input. The saliency attributions computed using the Integrated gradients are shown in 5e, 5h, 5n, 5q. These saliency maps demonstrate the importance of input features  $r$  and  $\phi$  of the selected node in the following layer for every tracking step along the two tracks visualized.

In figure 6, the sum of the magnitude of the saliency attributions corresponding to the seven features of each node in the *action features* for Track 1 (6a, 6b and 6c) and Track 2 (6d, 6e and 6f) for every layer are shown. The vanilla gradients (6a and 6d) show high attention for  $r$  and  $\phi$  in one or two of the layers but remain low and unclear in the remaining layers. The saliency attributions computed using the Integrated gradients and SmoothGrad methods clearly indicate that the RL agent pays high attention to the features  $r$  and  $\phi$  compared to the other features with  $r$  having the highest attention followed by  $\phi$  in integrated gradients method (see figures 6b and 6e) and feature  $\phi$  having the highest attention of the RL agent followed by feature  $r$  in SmoothGrad method (see figures 6c and 6f). These plots indicate that the RL agent is paying higher attention to the features  $r$  and  $\phi$  that are considered to be the most important features among the seven features considered for the analysis.

## 6. Conclusion

In this paper, we proposed saliency clouds, a set of visualization methods employing explainable AI methods to analyze and interpret deep reinforcement learning (DeepRL) agents working on point cloud-based data. We extended the SmoothGrad and integrated gradients methods to generate saliency maps for specific point clouds and visualized the results in 3D and, in aggregated way, in 2D graphs. The aggregated view provides insight into the relative relevance of different features, while the 3D view shows where a single feature is influencing the result the most. We believe that these techniques will be used as a tool to analyze the decision making process of neural networks working on point cloud data.

**Acknowledgments** This work was supported by the German federal state Rhineland-Palatinate (Forschungskolleg SIVERT) and by the Research Council of Norway (Norges forskningsråd) and the University of Bergen, grant number 250858. We use OpenWalnut for the 3D visualizations. Special thanks go to T. Kortus and R. Keidel for helpful discussions and comments on an early draft of this paper and for providing the RL agent and model.

## References

- [ABB\*20] ALME J., BARNAFÖLDI G. G., BARTHEL R., BORSHCHOV V., BODOVA T., VAN DEN BRINK A., BRONS S., CHAAR M., EIKELAND V., FEOFILOV G., ET AL.: A high-granularity digital tracking calorimeter optimized for proton CT. *Frontiers in Physics* (2020), 460. doi:10.3389/fphy.2020.568243. 2
- [ACJ19] ATREY A., CLARY K., JENSEN D.: Exploratory not explanatory: Counterfactual analysis of saliency maps for deep reinforcement learning. *arXiv preprint arXiv:1912.05743* (2019). doi:10.48550/arXiv.1912.05743. 1
- [CWF\*20] CHEN X., WANG Z., FAN Y., JIN B., MARDZIEL P., JOE-WONG C., DATTA A.: Towards behavior-level explanation for deep reinforcement learning. *arXiv e-prints* (2020), arXiv:2009. doi:10.48550/arXiv.2009.08507. 2
- [GKDF18] GREYDANUS S., KOUL A., DODGE J., FERN A.: Visualizing and understanding Atari agents. In *International Conference on Machine Learning* (2018), PMLR, pp. 1792–1801. doi:10.48550/arXiv.1711.00138. 2
- [HCDR21] HEUILLET A., COUTHOUIS F., DÍAZ-RODRÍGUEZ N.: Explainability in deep reinforcement learning. *Knowledge-Based Systems* 214 (2021), 106685. doi:10.1016/j.knosys.2020.106685. 2
- [HJTvdS20] HÜYÜK A., JARRETT D., TEKIN C., VAN DER SCHAAAR M.: Explaining by imitating: Understanding decisions by interpretable policy learning. In *Int. Confe. on Learning Representations* (2020). 2
- [Joh17] JOHNSON R. P.: Review of medical radiography and tomography with proton beams. *Reports on Progress in Physics* 81, 1 (Nov. 2017), 016701. doi:10.1088/1361-6633/aa8b1d. 2
- [Mag16] MAGER M.: ALPIDE, the monolithic active pixel sensor for the alice its upgrade. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 824 (2016), 434–438. Frontier Detectors for Frontier Physics: Proceedings of the 13th Pisa Meeting on Advanced Detectors. doi:10.1016/j.nima.2015.09.057. 2
- [QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems* (2017), et al. I. G., (Ed.), vol. 30, Curran Associates, Inc. doi:10.48550/arXiv.1706.02413. 2
- [RHT\*18] RAFFIN A., HILL A., TRAORÉ R., LESORT T., DÍAZ-RODRÍGUEZ N., FILLIAT D.: S-RL toolbox: Environments, datasets and evaluation metrics for state representation learning. *Preprint arXiv:1809.09369* (2018). doi:10.48550/arXiv.1809.09369. 2
- [RHT\*19] RAFFIN A., HILL A., TRAORÉ R., LESORT T., DÍAZ-RODRÍGUEZ N., FILLIAT D.: Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics. *Preprint arXiv:1901.08651* (2019). doi:10.48550/arXiv.1901.08651. 2
- [SCD\*17] SELVARAJU R. R., COGSWELL M., DAS A., VEDANTAM R., PARIKH D., BATRA D.: Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 618–626. doi:10.1109/ICCV.2017.74. 2
- [SDBR14] SPRINGENBERG J. T., DOSOVITSKIY A., BROX T., RIEDMILLER M.: Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806* (2014). doi:10.48550/arXiv.1412.6806. 1
- [STK\*17] SMILKOV D., THORAT N., KIM B., VIÉGAS F., WATTENBERG M.: SmoothGrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825* (2017). doi:10.48550/arXiv.1706.03825. 1, 2, 3
- [STY17] SUNDARARAJAN M., TALY A., YAN Q.: Axiomatic attribution for deep networks. In *International Conference on Machine Learning* (2017), PMLR, pp. 3319–3328. doi:10.48550/arXiv.1703.01365. 1, 2, 3
- [SVZ13] SIMONYAN K., VEDALDI A., ZISSERMAN A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013). doi:10.48550/arXiv.1312.6034. 1, 2
- [ZCY\*19] ZHENG T., CHEN C., YUAN J., LI B., REN K.: Pointcloud saliency maps. In *2019 IEEE/CVF Intern. Conf. on Computer Vision (ICCV)* (Los Alamitos, CA, USA, nov 2019), IEEE Computer Society, pp. 1598–1606. doi:10.1109/ICCV.2019.00168. 2
- [ZF14] ZEILER M. D., FERGUS R.: Visualizing and understanding convolutional networks. In *European conf. on comp. vis.* (2014), Springer, pp. 818–833. doi:10.1007/978-3-319-10590-1\_53. 1
- [ZKL\*16] ZHOU B., KHOSLA A., LAPEDRIZA A., OLIVA A., TORRALBA A.: Learning deep features for discriminative localization. In *Proceedings of the IEEE CVPR* (2016), pp. 2921–2929. doi:10.1109/CVPR.2016.319. 2
- [ZLZ\*20] ZHANG R., LIU B., ZHU Y., GUO S., HAYHOE M. M., BALLARD D. H., STONE P.: Human versus machine attention in deep reinforcement learning tasks. *CoRR abs/2010.15942* (2020). doi:10.48550/arXiv.2010.15942. 1
- [ZRS\*18] ZAMBALDI V., RAPOSO D., SANTORO A., BAPST V., LI Y., BABUSCHKIN I., TUYLS K., REICHERT D., LILLICRAP T., LOCKHART E., ET AL.: Relational deep reinforcement learning. *arXiv preprint arXiv:1806.01830* (2018). doi:10.48550/arXiv.1806.01830. 2
- [ZWL\*20] ZHANG R., WALSH C., LIU Z., GUAN L., MULLER K., WHRITNER J., ZHANG L., HAYHOE M., BALLARD D.: Atari-HEAD: Atari human eye-tracking and demonstration dataset. *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (04 2020), 6811–6820. doi:10.1609/aaai.v34i04.6161. 2

## Members of the Bergen pCT Collaboration

Max Ahle<sup>a</sup>, Johan Alme<sup>b</sup>, Gergely Gábor Barnaföldi<sup>c</sup>, Tea Bodova<sup>b</sup>, Vyacheslav Borshchov<sup>d</sup>, Anthony van den Brink<sup>e</sup>, Mamdouh Chaar<sup>b</sup>, Viljar Eikeland<sup>b</sup>, Gregory Feofilov<sup>f</sup>, Christoph Garth<sup>g</sup>, Nicolas R. Gauger<sup>a</sup>, Georgi Genov<sup>b</sup>, Ola Grøttvik<sup>b</sup>, Håvard Helstrup<sup>b</sup>, Sergey Igolkin<sup>f</sup>, Ralf Keidel<sup>h</sup>, Chinorat Kobdaj<sup>i</sup>, Tobias Kortus<sup>h</sup>, Viktor Leonhardt<sup>g</sup>, Shruti Mehendale<sup>b</sup>, Raju Ningappa Mulawade<sup>i</sup>, Odd Harald Odland<sup>k</sup>, George O’Neill<sup>b</sup>, Gábor Papp<sup>l</sup>, Thomas Peitzmann<sup>e</sup>, Helge Egil Seime Pettersen<sup>k</sup>, Pierluigi Piersimoni<sup>b,m</sup>, Maksym Protosenko<sup>d</sup>, Max Rauch<sup>b</sup>, Attiq Ur Rehman<sup>b</sup>, Matthias Richter<sup>h</sup>, Dieter Röhrich<sup>b</sup>, Joshua Santana<sup>i</sup>, Alexander Schilling<sup>i</sup>, Joao Seco<sup>o,p</sup>, Aron Songmoolnak<sup>b,j</sup>, Jarle Rambo Sölve<sup>q</sup>, Ganesh Tambave<sup>h</sup>, Ihor Tymchuk<sup>d</sup>, Kjetil Ullaland<sup>b</sup>, Monika Varga-Kofarago<sup>c</sup>, Lennart Volz<sup>r,s</sup>, Boris Wagner<sup>b</sup>, Steffen Wendzel<sup>l</sup>, Alexander Wiebel<sup>l</sup>, RenZheng Xiao<sup>b,t</sup>, Shiming Yang<sup>b</sup>, Hiroki Yokoyama<sup>e</sup>, Sebastian Zillien<sup>i</sup>

a) Chair for Scientific Computing, TU Kaiserslautern, 67663 Kaiserslautern, Germany b) Department of Physics and Technology, University of Bergen, 5007 Bergen, Norway; c) Wigner Research Centre for Physics, Budapest, Hungary; d) Research and Production Enterprise “LTU” (RPELTU), Kharkiv, Ukraine; e) Institute for Subatomic Physics, Utrecht University/Nikhef, Utrecht, Netherlands; f) St. Petersburg University, St. Petersburg, Russia; g) Scientific Visualization Lab, TU Kaiserslautern, 67663 Kaiserslautern, Germany; h) Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, 5020 Bergen, Norway; i) Center for Technology and Transfer (ZTT), University of Applied Sciences Worms, Worms, Germany; j) Institute of Science, Suranaree University of Technology, Nakhon Ratchasima, Thailand; k) Department of Oncology and Medical Physics, Haukeland University Hospital, 5021 Bergen, Norway; l) Institute for Physics, Eötvös Loránd University, 1/A Pázmány P. Sétány, H-1117 Budapest, Hungary; m) UniCamillus – Saint Camillus International University of Health Sciences, Rome, Italy; n) Department of Physics, University of Oslo, 0371 Oslo, Norway; o) Department of Biomedical Physics in Radiation Oncology, DKFZ—German Cancer Research Center, Heidelberg, Germany; p) Department of Physics and Astronomy, Heidelberg University, Heidelberg, Germany; q) Department of Diagnostic Physics, Division of Radiology and Nuclear Medicine, Oslo University Hospital, Oslo, Norway; r) Biophysics, GSI Helmholtz Center for Heavy Ion Research GmbH, Darmstadt, Germany; s) Department of Medical Physics and Biomedical Engineering, University College London, London, UK; t) College of Mechanical & Power Engineering, China Three Gorges University, Yichang, People’s Republic of China