

Non-Uniform Interpolation in Integrated Gradients for Low-Latency Explainable-AI

Ashwin Bhat, Arijit Raychowdhury

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

ashwinbhat@gatech.edu, arijit.raychowdhury@ece.gatech.edu

Abstract—There has been a surge in Explainable-AI (XAI) methods that provide insights into the workings of Deep Neural Network (DNN) models. Integrated Gradients (IG) is a popular XAI algorithm that attributes relevance scores to input features commensurate with their contribution to the model’s output. However, it requires multiple forward & backward passes through the model. Thus, compared to a single forward-pass inference, there is a significant computational overhead to generate the explanation which hinders real-time XAI. This work addresses the aforementioned issue by accelerating IG with a hardware-aware algorithm optimization. We propose a novel non-uniform interpolation scheme to compute the IG attribution scores which replaces the baseline uniform interpolation. Our algorithm significantly reduces the total interpolation steps required without adversely impacting convergence. Experiments on the ImageNet dataset using a pre-trained InceptionV3 model demonstrate $2.6\text{-}3.6\times$ performance speedup on GPU systems for iso-convergence. This includes the minimal $0.2\text{-}3.2\%$ latency overhead introduced by the pre-processing stage of computing the non-uniform interpolation step-sizes.

Index Terms—Explainable AI (XAI), Deep Neural Networks (DNN), Hardware-Aware Algorithm Design, GPU systems

I. INTRODUCTION

There has been a massive growth in the field of Machine Learning (ML) and Artificial Intelligence (AI). However, the black-box nature of DNN models has hindered its ubiquitous utilization [1]. Explainable-AI (XAI) provides insights into the workings of these models to enable adoption in safety-critical tasks [2] which require transparency and interpretability [3]. Within the field of XAI, feature attribution methods generate an explanation by scoring input features proportional to their contribution to the network’s output [4]. For image based applications, these relevance scores are visualized as a heatmap [5]. These post-hoc techniques can be applied to existing pre-trained models [6]. Integrated Gradients (IG), a feature attribution algorithm, has become popular thanks to its ease of implementation, axiomatic theoretical underpinnings, and applicability to any differentiable model [7].

IG accumulates gradients along a straight-line interpolation path between a baseline and the input. A baseline is indicative of missingness or lack of input [8]. For example, a black image is a commonly used baseline for vision tasks. IG requires multiple forward (inference) and backward (gradient backpropagation) passes through the model for each input. Thus, there is a large computational overhead in generating

the explanation ($50\text{-}1000\times$ slower) compared to just evaluating the model’s output which required a single forward pass. As noted in this XAI deployment study [9], it is necessary to reduce this overhead to enable real-time low-latency XAI. It is vital to overcome the technical limitations of computing explanations quickly in domains like smart healthcare [10], medicine [11], finance [12], and hardware security [13] where IG is being utilized.

Several optimizations over baseline IG have been proposed to improve the quality of the generated heatmaps. [8] proposes averaging the attributions obtained by using several different baselines. Google’s XRAI [14] segments the input into several regions and applies IG on each segment before stitching the results together. Captum, a PyTorch based XAI library developed by Meta [15], uses Noise Tunnel which averages the IG attributions over several noisy copies of the original input [16]. Despite using baseline IG multiple times in their pipeline, none of these algorithms attempt to reduce its computational overhead. Thus, they stand to gain significant performance benefits from an IG implementation optimized for low-latency on the underlying hardware platform.

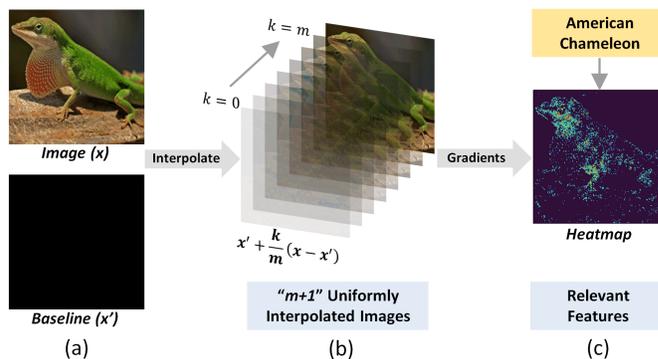


Fig. 1. Overview of Feature Attribution using the Integrated Gradients (IG) algorithm (a) Inputs to the algorithm: Image (x) and Baseline (x') (b) Interpolated images along straight line path between Baseline ($k = 0$) and Image ($k = m$) (c) Visualization of accumulated gradients to highlight input features relevant for classifying image as American Chameleon (target class)

All the currently known implementations of IG use uniform interpolation between baseline and input to approximate the continuous IG integral [Eq. 1] as Riemann-sum [Eq. 2]. In this work, we propose a non-uniform interpolation scheme to reduce the discretization steps required while maintaining

similar convergence accuracy. We first identify regions in the IG path with higher information content. By having smaller step-size in these regions and larger step-size outside of them, the overall compute overhead of IG is reduced.

In summary, this work makes the following contributions:

- To the best of our knowledge, our work is the first of its kind to design and employ a non-uniform interpolation scheme along the IG path to compute the discrete Riemann-sum approximation of the IG integral.
- Based on the experimental observations, we propose and justify the use of change in classification probability along the IG path as an information content metric to determine the non-uniform discretization step-size.
- We demonstrate the performance improvement achieved on GPU systems and quantify the latency overhead introduced by our algorithm compared to the baseline IG which uses a uniform interpolation scheme.

II. INTEGRATED GRADIENTS (IG)

The Integrated Gradients (IG) algorithm is a feature attribution method [Fig. 1]. Formally, for a given input (x) and a model function (f), a feature attribution method assigns a relevance score ($\phi_i(x, f)$) to the i^{th} input feature. The score is a measure of that feature’s contribution to the model’s output.

The simplest way to assign a relevance score is to evaluate the gradient of the model’s output with respect to the input feature. A large gradient value implies that small changes in the feature value produce a large change in the model’s output, thereby, indicating higher relevance. However, gradients are a local explanation method that can suffer from saturation effects. Path attribution methods (PAM) overcome this issue by accumulating gradients along a path between a baseline and the actual input [17]. These are mathematically motivated and satisfy desirable properties such as completeness and sensitivity. IG is a subset of PAM which considers a straight-line path between the baseline (x') and the input (x) as shown in Eq. 1. The commonly used baselines for computer vision applications include black, white, or random noise images [8]. They represent the notion of missingness or lack of any input.

$$\phi_i(f, x, x') = (x - x') \times \int_0^1 \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha \quad (1)$$

In Eq. 1, f is the model function and α is the interpolation constant along the straight-line path. In practice, to evaluate IG attributions, the continuous integral is approximated as a Riemann sum with $m + 1$ uniform steps as shown in Eq.2.

$$\phi_i(f, x, x') = (x - x') \times \frac{1}{m} \sum_{k=0}^m \frac{\partial f(x' + \frac{k}{m}(x - x'))}{\partial x_i} \quad (2)$$

The number of steps (m) typically ranges from 200 to 1000 [18]. Its value is chosen based on the convergence metric (δ) which is defined (Eq. 3) using the completeness property [17] satisfied by the continuous integral formulation of IG.

$$\delta = \left| \sum_i \phi_i(f, x, x') - [f(x) - f(x')] \right| \quad (3)$$

From Eq. 2, we observe that for every input, the IG algorithm creates several interpolated versions [Fig. 1(b)]. It then computes the gradient of the model’s output with respect to input features for each one. This step requires a forward and a backward pass through the model. These gradients are then aggregated to assign the overall attribution score [Fig. 1(c)].

III. METHODOLOGY

Background. The run-time latency of IG depends on the number of interpolation steps [Fig. 2(a)]. More steps yield better convergence δ [Fig. 2(b)] at the cost of higher latency.

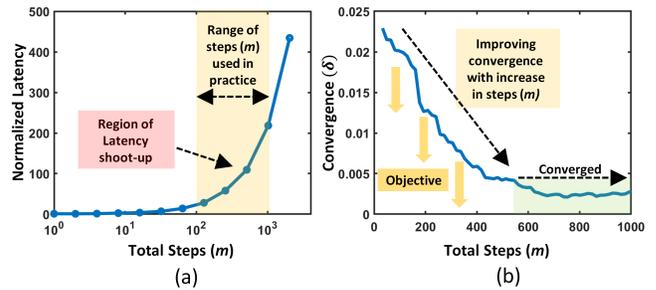


Fig. 2. (a) Latency increases with increase in number of interpolation steps [values normalized relative to the latency for $m=1$] (b) Decreasing convergence δ with increasing steps implies better convergence at reduced interpolation step-size. Thus, there is a latency penalty for better convergence.

Objective. The goal of this work is to reduce the compute overhead of IG for generating explanations. The typically used range of values for the number of steps lies beyond the knee-point of the latency v/s step-count graph [Fig. 2(a)]. Thus, for lower latency, the number of interpolation steps must be decreased without compromising convergence [Fig. 2(b)].

Observation. Along the interpolation path, close to the baseline (small values of α), the actual input image is unrecognizable from its interpolated version. However, after a certain α , the input can be identified. Beyond this threshold, as we move towards the input along the IG path, the brightness of the interpolated image increases. Intuitively, for a human observer, the change in classification confidence is not uniform along the IG path as seen in Fig. 3(a).

We test this intuition on the model. The classification probability of the model changes sharply as we increase the interpolation constant α [Fig. 3(b)] along the IG path. At $\alpha = 0.25$, the classification probability (0.83) is $>90\%$ of its final value (0.89) for the input image ($\alpha = 1$). Thus, the model’s confidence about the prediction is built over a small interval along the IG path with minimal change outside of it.

The IG algorithm accumulates the gradient of the classification probability with respect to input features for each interpolated image. Thus, the change in classification probability can be used as a metric for information content along the IG path.

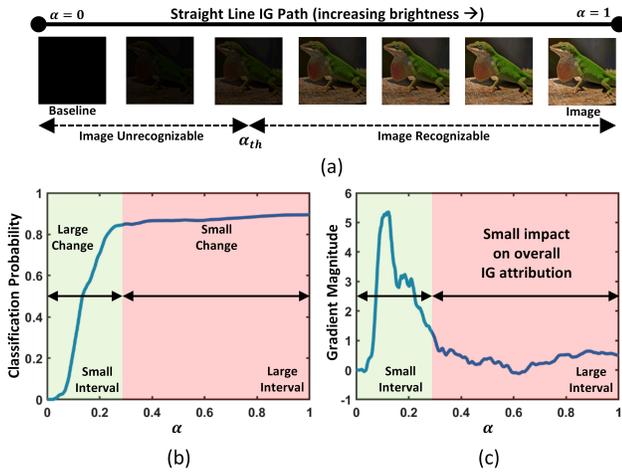


Fig. 3. Variation along the IG path of (a) Distinguishability of the image from its interpolated copies (b) Classification probability of the model (c) Contribution to the convergence term based on the relative gradient magnitude. Thus, a small region along the IG path contains most of the information.

In regions of large change, the gradient values are also larger [Fig. 3(c)] and contribute more to the overall IG attribution.

Proposed Method. We propose a non-uniform interpolation scheme to replace the baseline uniform interpolation [Fig. 4(a)] along the IG path. Specifically, the IG path is divided into multiple intervals and uniform interpolation is performed within each interval using a different step-size. Based on the earlier observation [Fig. 3], we can use a smaller step size in the regions where there is a large change in the classification probability. Outside of such regions, a larger discretization step size can be used. Overall, the total steps are non-uniformly distributed along the IG path with a bias towards regions with higher information content.

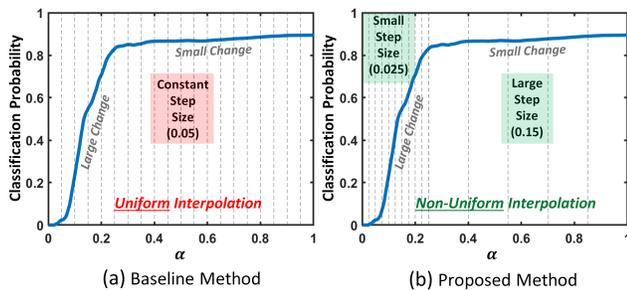


Fig. 4. (a) Uniform interpolation with constant step size for the entire IG path (b) Non-Uniform Interpolation with small uniform step size in the region of large change and large uniform step size in the region of small change.

Algorithm. The proposed algorithm comprises two stages. The *first stage* takes the number of intervals (n_{int}) as a parametric input to divide the IG path into n_{int} equal pieces. The classification probability of the interpolated images at the interval boundaries is evaluated to calculate its normalized change in each interval $[\Delta f(x_{int})]$. The total number of steps (m) is then distributed across each interval proportional to

the square root of the change $[\sqrt{|\Delta f(x_{int})|}]$. We observed that linear dependence ($m_{int} \propto \Delta$) allotted negligible discretization steps to regions with small change. Hence, we use $m_{int} \propto \sqrt{\Delta}$ to attenuate the bias towards intervals of large change. In the *second stage*, we perform uniform IG within each interval with the respective step count. The IG attributions of different intervals are then summed up to determine the overall IG attribution. The proposed algorithm is, therefore, *uniform-in-intervals* but *non-uniform overall* along IG path. Fig. 4(b) illustrates this for $n_{int} = 4$.

IV. RESULTS

Experimental Setup. The proposed method can directly replace the baseline IG algorithm and be applied to any differentiable model. To demonstrate its efficacy, we test it on the ImageNet dataset [19] using a pre-trained InceptionV3 model [20]. We consider our *baseline* to be the existing IG implementation that employs uniform interpolation. We compare it against our proposed *non-uniform interpolation algorithm* and vary the number of intervals (n_{int}) parameter.

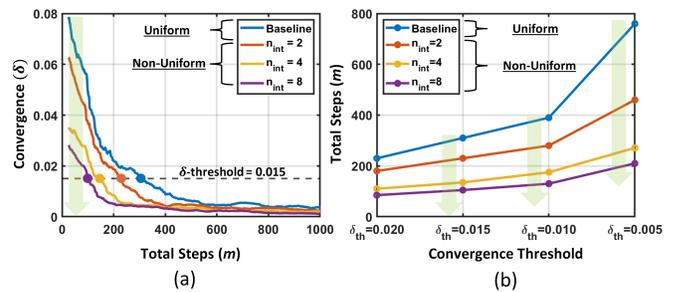


Fig. 5. For different IG interpolation schemes, the variation of (a) Convergence delta (δ) as we increase the total interpolation steps (m) (b) Total interpolation steps required for reaching the convergence threshold (δ_{th}).

Convergence. In Fig. 5(a), we demonstrate the effect of the total number of steps on the convergence δ for different interpolation schemes. For any given number of steps (m), our proposed algorithm achieves better convergence δ compared to baseline. Thus, for iso-convergence, the proposed algorithm is able to reduce the total steps needed.

In practice, the total interpolation steps for IG are determined by fixing a threshold tolerance for the convergence. For example, in Fig. 5(a), this threshold (δ_{th}) is set to 0.015. The total number of steps is then chosen such that the convergence δ lies below the threshold [Fig. 5(a)]. We vary δ_{th} and determine the number of steps required for different interpolation schemes [Fig. 5(b)] to meet the convergence criterion. For all δ_{th} values, our proposed algorithm requires fewer steps and outperforms the baseline. Increasing the number of intervals further reduces the steps required for convergence. The benefits are more pronounced at smaller δ_{th} values. For $\delta_{th} = 0.02$, we observe a $2.7\times$ reduction while for $\delta_{th} = 0.005$, we observe a $3.6\times$ reduction in the total steps required for convergence.

We further observe that increasing the number of intervals (n_{int}) up to a certain point reduces δ . Further increasing n_{int} causes δ to increase since certain intervals are allotted negligible discretization steps which negatively impact convergence. Consequently, it increases the number of steps required to meet δ_{th} . We observe that $n_{int} > 8$ manifests this issue.

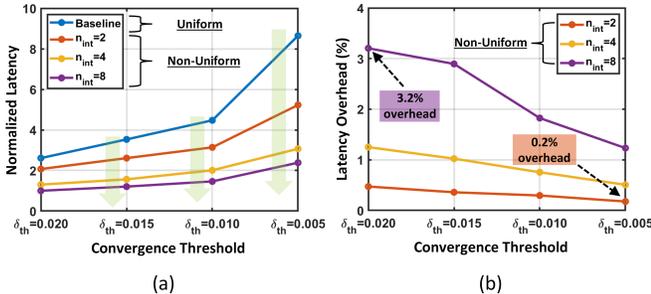


Fig. 6. For each IG interpolation scheme, the variation of (a) normalized latency to meet the convergence threshold (δ_{th}) (b) latency overhead (%) of total latency) of the first stage of the non-uniform interpolation algorithm.

Latency. Since our method is applicable to any differentiable model and the performance benefit is not specific to a hardware architecture, we report normalized latency values. The latency is normalized relative to the algorithm configuration which yields the smallest run-time latency. In our experiments, we measure the run-time latency for running IG on InceptionV3 model with batch-size of 16 on a NVIDIA TITAN Xp GPU using the PyTorch benchmark profiler. The profiler supports CUDA (excludes the overhead of thread synchronization), performs an initial warm-up, and averages over multiple runs to determine accurate execution latency.

The normalized latency [Fig. 6(a)] depends on the number of interpolation steps (m) which in turn depends on the convergence threshold (δ_{th}). We make two observations. *First*, the latency increases as we decrease δ_{th} because of an increase in m . However, the relative increase in latency as we reduce δ_{th} from 0.02 to 0.005 is higher for baseline ($3.3\times$) when compared to non-uniform interpolation (2.3 - $2.5\times$). *Second*, in terms of performance, our proposed non-uniform interpolation scheme outperforms the baseline across all δ_{th} values. The latency reduces as we increase n_{int} yielding higher performance benefits. For $\delta_{th} = 0.02$, we achieve a $2.6\times$ and for $\delta_{th} = 0.005$, we achieve a $3.6\times$ latency reduction when compared to the baseline.

Overhead. Determining the number of steps for each interval is the overhead of our proposed scheme. The memory overhead is minimal since we only store the classification probability of $n_{int} + 1$ interpolated images to determine the step-sizes. The latency overhead is measured as a fraction of the total latency of running the IG algorithm [Fig. 6(b)]. The overhead varies between 0.2-3.2% of the total latency.

We observe that the absolute value of the latency overhead depends only on the parameter n_{int} . This is because we run the inference pass through the network $n_{int} + 1$ times to determine the classification probability change in each

interval and distribute the total steps commensurately. Thus, the absolute overhead increases as we increase n_{int} . However, the relative value of the overhead depends on both n_{int} and δ_{th} [Fig. 6(b)] because these parameters affect the total steps required and hence the overall latency of non-uniform IG.

V. DISCUSSION & RELATED WORK

With XAI research being in its infancy, there is a paucity of work that focuses on improving performance (low-latency, high throughput, real-time) either through (a) specialized hardware architectures or (b) algorithmic optimizations with a hardware-aware design approach (this work).

Pan et al. [21] accelerates model-distillation based explanation on TPU or systolic-array hardware substrate. This method is unsuitable for low-latency explanation generation since it requires training a new model which locally mimics the input-output behavior of the black-box model for each input. In this work, we focus on a post-hoc explanation method that is directly applicable to any off-the-shelf differentiable model. Bhat et al. [22] accelerates gradient based heatmap visualization on FPGA platform. However, the implemented feature attribution algorithms suffer from local saturation effects. In this work, we focus on IG which overcomes this issue via path-attribution. Although we demonstrate our results on GPU systems, our algorithm is agnostic to the underlying hardware.

Sotoudeh et al. [18] proposes exact computation of the IG integral but achieves smaller performance gains (upto $1.7\times$) compared to this work (upto $3.6\times$). Kapishnikov et al. [23] avoids high-loss regions by updating a subset of features with low gradient magnitude at each point along the path. However, the next step is dynamically determined which limits the performance on GPUs as batch-size is restricted to 1. In our work, we design a static processing stage to pre-determine the discretization step size and leverage batching on GPU systems. Rahman et al. [24] modifies the path by performing a local gradient ascent around each uniform interpolation point in the IG path. This magnifies the compute overhead of generating the explanation. Although both methods modify the baseline IG path to improve the quality of attribution heatmaps, they require more steps thereby incurring a performance overhead. Our work modifies the IG path to improve performance while maintaining iso-convergence with baseline IG.

VI. CONCLUSION

In this paper, a novel Non-Uniform Interpolation scheme for computing the Integrated Gradients attribution is presented. Our methodology is motivated by the observations we made on the baseline uniform interpolation. Regions of high-information content along the straight-line IG path are identified using the change in classification probability. Utilizing classification probability as an information metric is justified based on its correlation with gradient magnitude and its contribution to the overall IG attribution. The proposed algorithm distributes the steps among intervals with a bias towards ones with higher information content. Compared to the baseline, our algorithm meets iso-convergence thresholds

with fewer total steps. We quantify the performance benefit on GPU systems using pre-trained models on the ImageNet dataset. Our experiments show that we can achieve a speed-up of **2.6-3.6**× at a very low latency overhead of **0.2-3.2**%. In summary, our hardware-aware algorithm design enables low-latency real-time explainable AI.

VII. ACKNOWLEDGEMENT

This work was supported by Semiconductor Research Corporation (SRC) AI Hardware (AIHW) Task 2969.001 titled EXPERT: EXplainable-AI through Efficient hardware design in EmeRging Technologies.

REFERENCES

- [1] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.
- [2] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.
- [3] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, "Explainable machine learning for scientific insights and discoveries," *Ieee Access*, vol. 8, pp. 42 200–42 216, 2020.
- [4] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pp. 1–16, 11 2017. [Online]. Available: <http://arxiv.org/abs/1711.06104>
- [5] G. Ras, N. Xie, M. van Gerven, and D. Doran, "Explainable deep learning: A field guide for the uninitiated," *Journal of Artificial Intelligence Research*, vol. 73, pp. 329–397, 2022.
- [6] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K. R. Müller, "Explaining deep neural networks and beyond: A review of methods and applications," *Proceedings of the IEEE*, vol. 109, pp. 247–278, 2021.
- [7] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.
- [8] P. Sturmfels, S. Lundberg, and S.-I. Lee, "Visualizing the impact of feature attribution baselines," *Distill*, 2020, <https://distill.pub/2020/attribution-baselines>.
- [9] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. Moura, and P. Eckersley, "Explainable machine learning in deployment," in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 648–657.
- [10] L. Ibrahim, M. Mesinovic, K.-W. Yang, and M. A. Eid, "Explainable prediction of acute myocardial infarction using machine learning and shapley values," *IEEE Access*, vol. 8, pp. 210 410–210 417, 2020.
- [11] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Toward medical xai," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 11, pp. 4793–4813, 2020.
- [12] B. Hadji Misheva, A. Hirska, J. Osterrieder, O. Kulkarni, and S. Fung Lin, "Explainable ai in credit risk management," *Credit Risk Management (March 1, 2021)*, 2021.
- [13] A. Golder, A. Bhat, and A. Raychowdhury, "Exploration into the explainability of neural network models for power side-channel analysis," in *Proceedings of the Great Lakes Symposium on VLSI 2022*, 2022, pp. 59–64.
- [14] A. Kapishnikov, T. Bolukbasi, F. Viegas, and M. Terry, "Xrai: Better attributions through regions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
- [15] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan *et al.*, "Captum: A unified and generic model interpretability library for pytorch," *arXiv preprint arXiv:2009.07896*, 2020.
- [16] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *arXiv preprint arXiv:1706.03825*, 2017.
- [17] D. D. Lundstrom, T. Huang, and M. Razaviyayn, "A rigorous study of integrated gradients method and extensions to internal neuron attributions," in *International Conference on Machine Learning*. PMLR, 2022, pp. 14 485–14 508.
- [18] M. Sotoudeh and A. V. Thakur, "Computing linear restrictions of neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [21] Z. Pan and P. Mishra, "Hardware acceleration of explainable machine learning," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 1127–1130.
- [22] A. Bhat, A. S. Assoa, and A. Raychowdhury, "Gradient backpropagation based feature attribution to enable explainable-ai on the edge," *arXiv preprint arXiv:2210.10922*, 2022.
- [23] A. Kapishnikov, S. Venugopalan, B. Avci, B. Wedin, M. Terry, and T. Bolukbasi, "Guided integrated gradients: An adaptive path method for removing noise," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5050–5058.
- [24] M. M. Rahman, N. Lewis, and S. Plis, "Geometrically guided saliency maps," in *ICLR 2022 Workshop on PAIR'2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022. [Online]. Available: <https://openreview.net/forum?id=rtleCBr8W-5>