Poster Abstract: Xpi: Real-Time Progressive Inference Serving with Explainable AI in Edge-Cloud Systems

Changyao Lin
Harbin Institute of Technology
Harbin, China
lincy@stu.hit.edu.cn

Zhenming Chen
China Construction Steel Structure Engineering Corp., LTD
Shenzhen, China
chenzm@cscec.com

Jie Liu
Harbin Institute of Technology
Shenzhen, China
jieliu@hit.edu.cn

***Abstract*—The constrained computing and memory resources at the edge pose challenges for satisfying different service-level objectives (SLOs) of deep learning inference requests. In this paper, we propose a novel edge-cloud progressive inference framework Xpi, which integrates explainable AI technique to facilitate early-exit, and learning-based online execution control to satisfy different SLOs and optimize edge resource overheads. We implement Xpi on an edge-cloud platform, and conduct partial experiments on two datasets. Xpi outperforms several advanced edge-cloud progressive inference frameworks in terms of accuracy and deadline satisfaction rate.**

***Index Terms*—edge computing, progressive inference, explainable AI, reinforcement learning.**

I. Introduction

With the advancement of edge computing and hardware capabilities, an increasing number of intelligent applications are being deployed at the edge to provide users with more real-time services. These applications generate diverse inference tasks (data) that usually vary in difficulty, resource overhead, and service-level objective (SLO, such as real-time responsiveness and accuracy). Different deep learning models need to process these tasks promptly.

However, the constrained computing and memory resources at the edge pose significant challenges in satisfying the inference demands. To enhance the quality of service (QoS) and satisfy different SLOs, we propose a novel edge-cloud progressive inference framework Xpi. The framework combines eXplainable AI (XAI) technique [1] with early-exit strategy [2], constructing an efficient multi-branch network. We shift the rationale of early-exit network construction from fixed to agile and data-centric. Our goal in network design is to progressively extract the most critical features for inference, ensuring high accuracy even at early-exit points.

Furthermore, since the diversity of SLOs introduces various scheduling opportunities, Xpi incorporates a reinforcement learning (RL)-based online scheduling mechanism that dynamically adapts to SLOs and runtime resource fluctuations. By periodically updating the early-exit thresholds for all branch during execution, Xpi enables tasks with different SLOs to exit at appropriate points. Additionally, by dynamically determining the partition point of the model, some of the computational workload is offloaded to cloud. The Xpi framework can efficiently compress the intermediate features that are less critical, reducing edge-cloud transmission overhead while maintaining end-to-end accuracy. Such edge-cloud collaborative inference not only addresses deployment challenges for large early-exit models at resource-constrained edge devices, but also optimizes edge resource utilization and protects privacy.

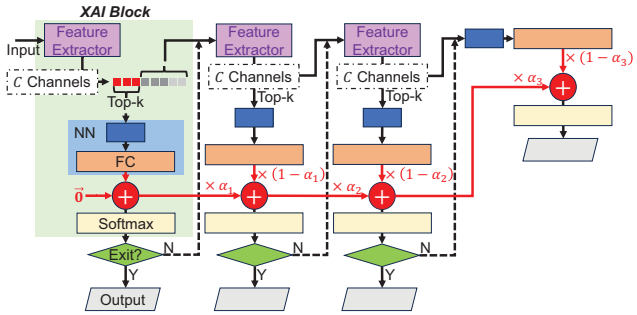


Fig. 1. Xpi network architecture. An example of stacking three XAI blocks.

II. System Overview

Fig. 1 shows our progressive inference framework Xpi. The basic idea of our design is to incorporate the knowledge about the heterogeneity of different input data into training, so as to migrate the computation required to extract important features from online inference to offline training. More specifically, we interpret this heterogeneity as the importance or contribution of different data features for neural network (NN) inference, and use XAI tool [1] to explicitly evaluate the importance during training. Then, at the inference stage, the most important features can be given to the early branches for inference, making the early results very accurate. The later branches can further infer on the less important features, and add up the early results with a certain weight, so that the results of previous branches can be reused. Ultimately, the inference accuracy can satisfy the SLO threshold earlier, that is, exit as early as possible to minimize the inference latency.

To ensure that such importance distribution is skewed on different data, Xpi intentionally manipulates the importance of data features by non-linear transformation in the high-

979-8-3503-6201-5/24/$31.00 ©2024 IEEE
DOI 10.1109/IPSN61024.2024.00037

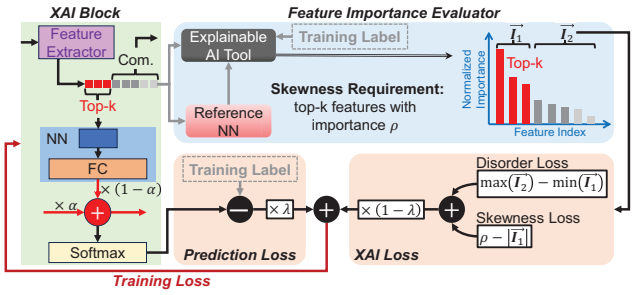


Fig. 2. Offline training for each XAI block.

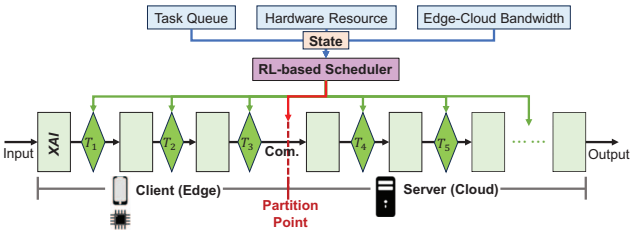


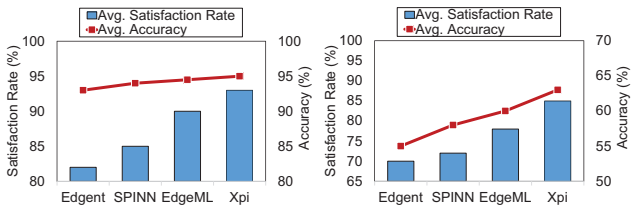
Fig. 3. Online scheduling for each task.

dimensional feature space. In other words, we enforce that after transformation, only the features in the first k channels contribute significantly to the branch NN inference, and these important features will be extracted and inferred earlier. As shown in Fig. 2, we achieve the skewness manipulation by a highly lightweight CNN-based feature extractor, and jointly train the feature extractor and NN in each branch to ensure accuracy. Since only a few important features are extracted for each branch, the NN on the branch can be lightweight and easier to deploy at the edge. We train each XAI block separately from front to back, using the method in [3].

Fig. 3 shows the online scheduling stage of Xpi. Since online model execution control brings a huge scheduling space, we adopt the RL-based runtime scheduling method [4], which decides the early-exit thresholds and edge-cloud partition point according to the task's SLO, hardware resources, and edge-cloud bandwidth, so that the task exits from the appropriate early-exit point, and the edge resource overheads can be minimized. For each exit point, the softmax and entropy will be calculated according to the method in [2], to measure the confidence of the classifier at that exit point for the output. If the confidence does not exceed the threshold (i.e., the result of that branch is very likely to be correct), the execution will be terminated, and the output result of that branch will be returned as the model output. If the model can be executed to the partition point, the less important features will be compressed [5], [6] and offloaded to cloud, which can reduce the transmission overhead and ensure accuracy.

III. Preliminary Evaluation

We conduct a comparison between Xpi and three edge-cloud progressive inference frameworks: Edgent [7], SPINN [8], and EdgeML [4]. The evaluation is performed on an edge-cloud platform comprising an edge device NVIDIA Xavier



(a) On CIFAR-10. (b) On ImageNet-200.

Fig. 4. Comparison of Xpi and other edge-cloud progressive inference frameworks on different datasets.

NX and a cloud server equipped with NVIDIA GeForce RTX 3080. Xpi is configured with 12 branches, and adopts the pre-trained EfficientNetV2 [9] as the reference NN. We test the frameworks on two computer vision datasets of varying difficulty: CIFAR-10 [10] and ImageNet-200 [11]. The task deadlines are randomly set between 100ms and 500ms. As depicted in Fig. 4, Xpi outperforms the other frameworks in terms of average accuracy and deadline satisfaction rate.

IV. Conclusion and Future Work

This paper proposes a novel edge-cloud progressive inference framework Xpi, which incorporates explainable AI technique to facilitate early-exit, and learning-based online execution control to satisfy different SLOs and optimize edge resource overheads. Xpi is leading in accuracy and deadline satisfaction rate. In the future, we will continue to conduct detailed evaluation of Xpi in various aspects and further refine the technical details.

References

[1] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.

[2] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Branchynet: Fast inference via early exiting from deep neural networks," in *2016 23rd international conference on pattern recognition (ICPR)*. IEEE, 2016, pp. 2464–2469.

[3] K. Huang and W. Gao, "Real-time neural network inference on extremely weak devices: agile offloading with explainable ai," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 200–213.

[4] Z. Zhao, K. Wang, N. Ling, and G. Xing, "Edgeml: An automl framework for real-time deep learning on the edge," in *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, 2021, pp. 133–144.

[5] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," *Advances in neural information processing systems*, vol. 30, 2017.

[6] H. Dheemanth, "Lzw data compression," *American Journal of Engineering Research*, vol. 3, no. 2, pp. 22–26, 2014.

[7] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge ai: On-demand accelerating deep neural network inference via edge computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 447–457, 2019.

[8] S. Laskaridis, S. I. Venieris, M. Almeida, I. Leontiadis, and N. D. Lane, "Spinn: synergistic progressive inference of neural networks over device and cloud," in *Proceedings of the 26th annual international conference on mobile computing and networking*, 2020, pp. 1–15.

[9] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*. PMLR, 2021, pp. 10 096–10 106.

[10] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[11] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.