Applying Graph Explanation to Operator Fusion

Keith G. Mills[1], Muhammad Fetrat Qharabagh[1], Weichen Qiu[1], Fred X. Han[2], Mohammad Salameh[2], Wei Lu[2], Shangling Jui[3] and Di Niu[1]

[1]Department of Electrical and Computer Engineering, University of Alberta [2]Huawei Technologies Canada [3]Huawei Kirin Solution

1. Introduction

Deep Neural Networks have become an indispensable tool when applying ML techniques to solve real-world problems.

Layer Fusion methods such as Line Buffer Depth-First (LBDF) and Buffer Requirement Reduction (BRR) execution allow for faster DNN inference on the limited on-chip buffers of accelerators.

On-chip buffer has a limited memory size, forming a hard constraint. Partition the DNN into *fusion groups* for inference.

In a *valid* partitioning, each fusion group fits in the buffer. A *good* partitioning is valid and requires low DRAM transaction cost.

What if we encounter invalid fusion groups? Split or discard plan. The former is preferable, but a non-trivial challenge.

We propose to use Graph Explanation Techniques (GET) for LF:
1. Treat LF partitioning as a recursive optimization problem.
2. Determine buffer validity through binary classification.

Incorporate our scheme with several search algorithms to demonstrate how it can find fusion groups with lower DRAM cost.

2. Methodology

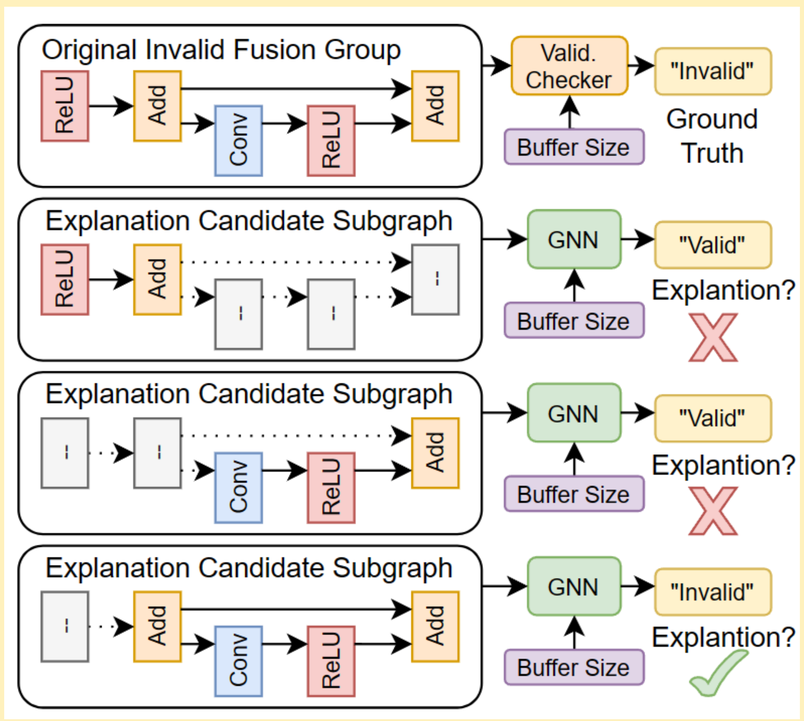
Recursive splitting of invalid fusion groups:

DNNs form computational graph structures where nodes are operations (e.g., Conv, ReLU) and edges guide the forward-pass.

Partition plan is a set of fusion groups – or disjoint DNN subgraphs that execute at once according to an efficient scheme (e.g., LBDF).

Three scenarios when an invalid fusion group is split in two:
1. Both new fusion groups are now valid. Desirable end point.
2. One group is valid, but the other is invalid: Greedy selection.
3. Both are invalid. Treat each group separately and split again.

How to Intelligently Split Fusion Groups with GNNs/GETs:

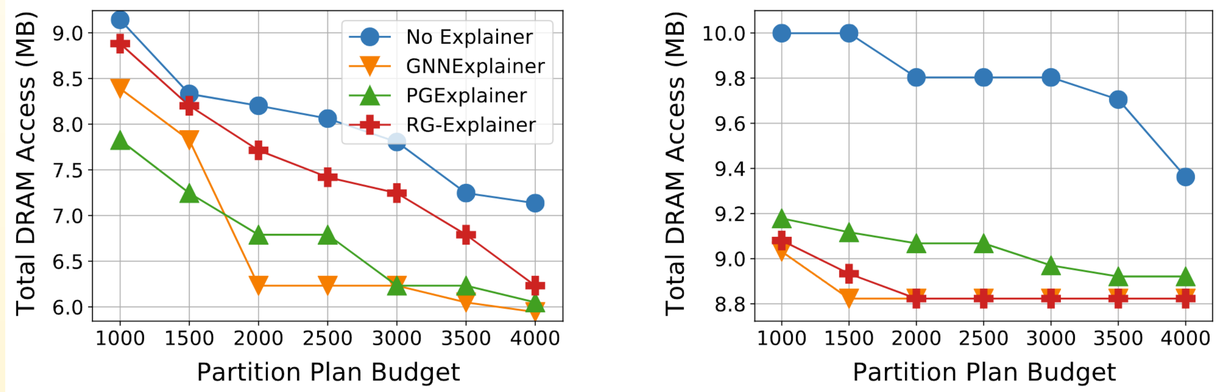


GNN predicts validity given buffer size.

Finds an explanation: Subgraph that maximizes Mutual Information between itself and fusion group.

Each edge in the explanation is a potential solution for splitting the fusion group.

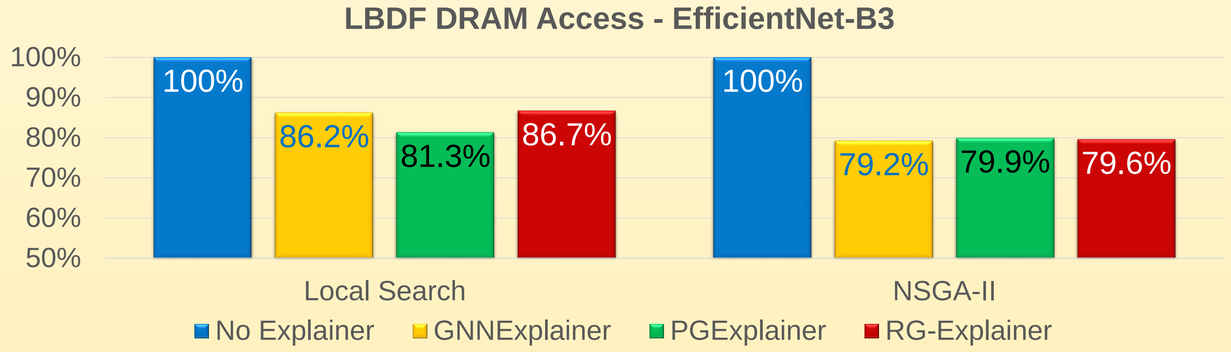
3. Results



(a) SqueezeNet on BRR using LS

(b) MBv2 on LBDF using LS

Search algorithms: NSGA-II and Local Search (LS).
3 Solver GETs: GNNExplainer, PGExplainer or RG-Explainer.



4. Conclusion

We approach the problem of Layer Fusion (LF) optimization by applying Graph Explanation Techniques (GET) to improve search.

We pair GETs with a recursive partitioning method to split invalid fusion groups of a LF partition plan in a cost-conscious manner to minimize DRAM access given a specified on-chip buffer size.

We consider modern and classical DNN designs such as EfficientNets, MobileNets, and ResNets in the LBDF and BRR layer fusion scenarios by pairing our method with off-the-shelf search algorithms like Local Search and NSGA-II.

Experimental results show that our proposed scheme is effective at minimizing DRAM cost, e.g., we find low DRAM access plans faster and can reduce DRAM access by over 20% on EfficientNet-B3.

UNIVERSITY OF ALBERTA

HUAWEI HISILICON