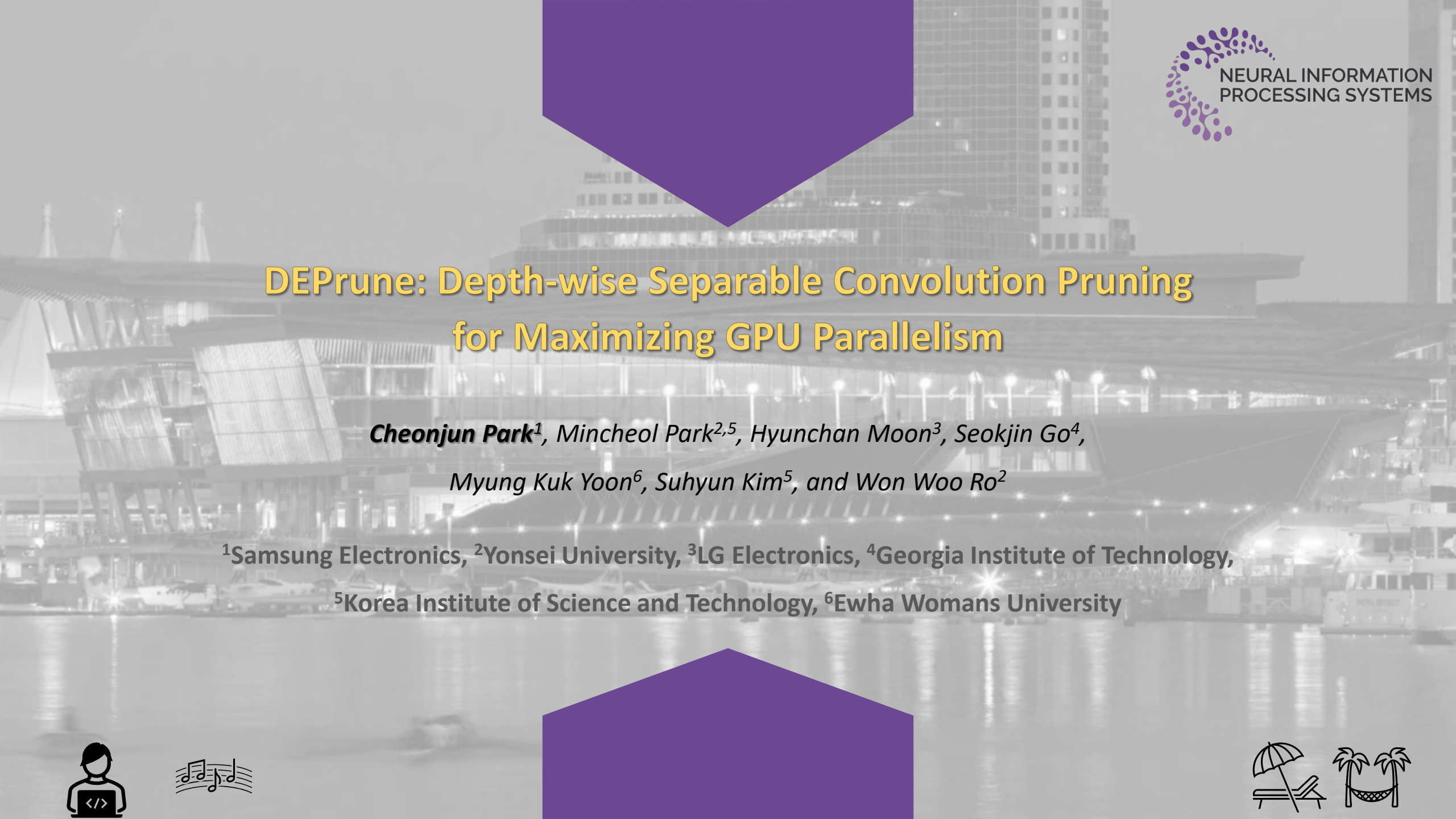

NEURAL INFORMATION PROCESSING SYSTEMS

DEPrune: Depth-wise Separable Convolution Pruning for Maximizing GPU Parallelism

Cheonjun Park[1], *Mincheol Park*[2,5], *Hyunchan Moon*[3], *Seokjin Go*[4],

Myung Kuk Yoon[6], *Suhyun Kim*[5], *and Won Woo Ro*[2]

[1]Samsung Electronics, [2]Yonsei University, [3]LG Electronics, [4]Georgia Institute of Technology,

[5]Korea Institute of Science and Technology, [6]Ewha Womans University

Background

- **Depth-wise Convolution on GPUs**
  - Depth-wise Separable convolution (**DSConv**) is widely used in many fields.
  - In GPUs, depth-wise convolution (**DW-conv**) is rearranged in a channel-by-channel format and performs multi-GEMV.
  - To process **DW-conv** more efficiently, **diagonal-wise refactorization (DR)** is needed.
  - With **DR**, it transforms into a large GEMM that GPUs can handle efficiently.
  - When processing large GEMMs, GPUs use **tiling** (**grouping**) to perform multi-GEMM operations.

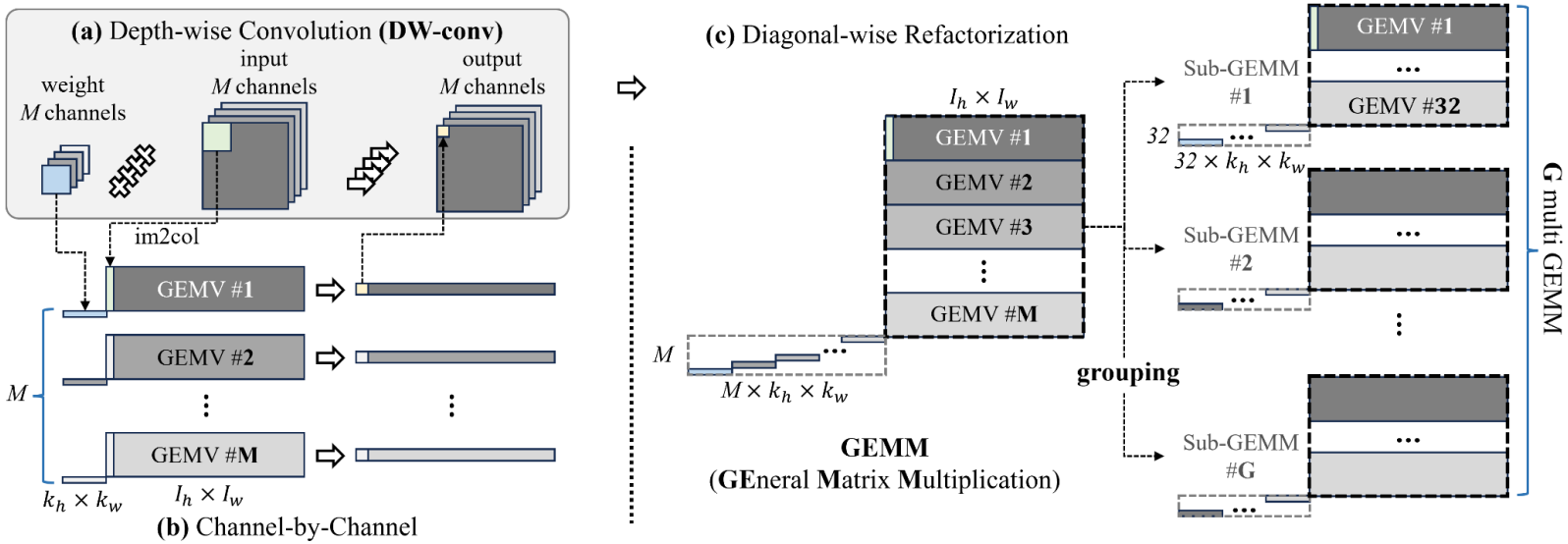

Fig.1 (a) DW-conv is rearranged to multi GEMV through **(b)** Channel-by-Channel on GPU execution. **(c)** Diagonal-wise Refactorization (DR) rearranges DW-conv to multi-sub-GEMM. It is effective to group DR with 32 channels to form sub-GEMM, which is proportional to GPU tile size

Motivation

- **Motivation 1: Channel pruning on DW-conv has a large pruning unit size problem**
  - **DW-conv** generates a multi-GEMV format for each channel, on GPUs.
  - **DW-conv** can also achieve structured data format, by evaluating the significance of each GEMV and eliminating an unnecessary weight vector of GEMV.
  - Eliminating a single channel from **DW-conv** can greatly diminish its representation power.
- **Motivation 2: Hardware-unfriendly problem of weight pruning without DR**
  - (**w/o DR**) Weight pruning does not result in practical speedup from pruning.
  - Since this is smaller than the GPU's tile size (**32**), there is almost no change in inference time since GEMV underutilizes processing units of GPU.

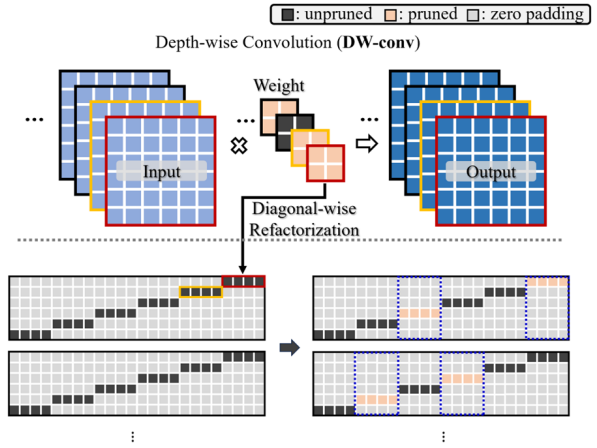

Fig.2 Depth-wise convolution on GPUs

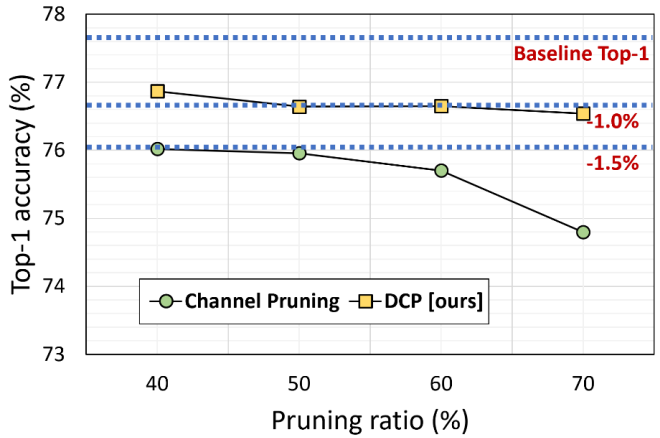

Fig.3 Comparison of accuracy drop between DCP and channel pruning on EfficientNet-B0 using ImageNet.

Proposed Method - DEPrune

- To address the above two issues, We propose **Depth-wise Convolution Pruning (DEPrune - DCP).**
- We discover that **weight pruning** after **DR** can even achieve a structured sparsity, making large GEMM on **maximize GPU parallelism.**
  - First, we take the weight matrix rearranged in the form of GEMM by **DR**.
  - Second, we sort the unpruned values in ascending order and select the threshold value that corresponds to the target pruning ratio.
  - Last, for each unpruned value, if it is smaller than the threshold, we change it to zero.

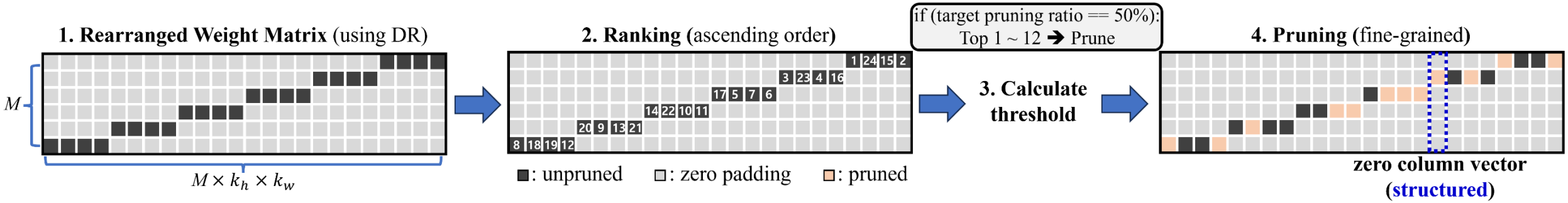

Fig.4 Process of Depth-wise Convolution Pruning (**DEPrune**).

Proposed Enhance Method - BWT

- **Enhanced Method: DEPrune-B**
  - **Motivation: Imbalance overhead problem of DEPrune**
    - ✓ GPUs allocate operations of a certain size to streaming multiprocessors (SMs) for massive parallel processing.
    - ✓ Therefore, DW-conv's multiple sub-GEMMs are also assigned to SMs, respectively.
    - ✓ However, when applying **DEPrune** on **DW-conv**, the pruning ratio of sub-GEMMs may differ, given the varying importance of weights between sub-GEMMs.
  - **Method: Balanced Workload Tuning (BWT)**
    - ✓ To address the workload imbalance issue of **DEPrune**, we propose a **BWT** that takes into account the operation structure of **DW-conv**.
    - ✓ Every sub-GEMM achieves the same target pruning ratio.

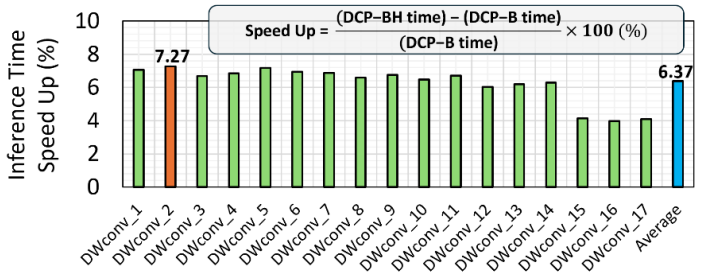

Fig.5 Measurement of speed increase by layer due to HSR. The orange bar is the max speedup layer. DW-conv PR is 71%.

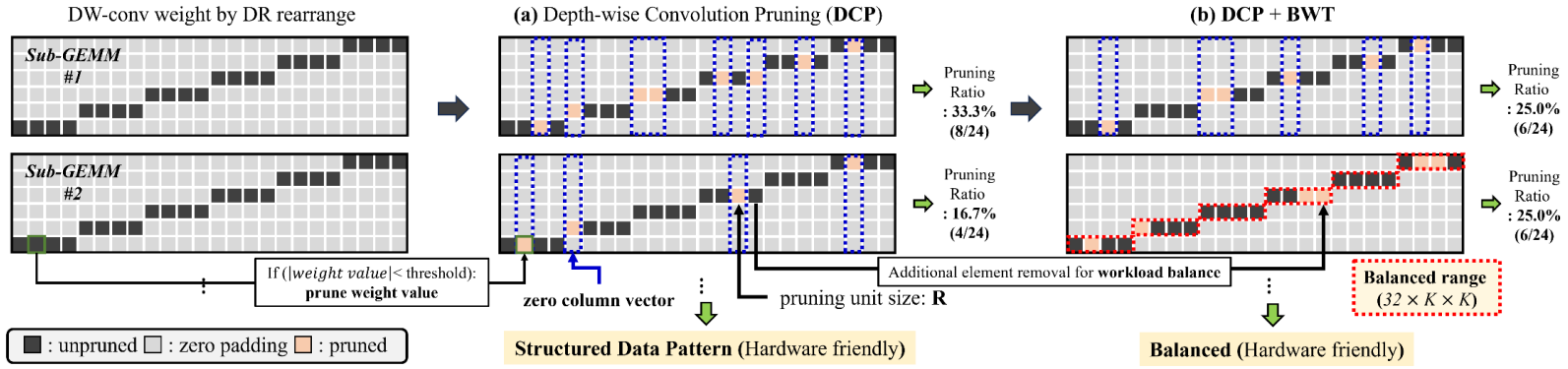

Fig.6 Overview of **DEPrune** and Balanced Workload Tuning (**BWT**). **(a) DEPrune** is an element-wise pruning method, that can create a structured data pattern. **(b)** BWT equalizes the pruning ratio of all sub-GEMMs.

Proposed Method (3)

- **Enhanced Method: DEPrune-BH**
  - **Motivation: Unaligned problem**
    - ✓ to maximize parallelism, GPUs divide GEMM operations into small tiles.
    - ✓ In general, the size of the tile depends on the hardware specification of GPUs, but it is usually a multiple of **32**.
  - **Method: Hardware-aware Sparsity Recalibration (HSR)**
    - ✓ We propose **HSR** to solve the unaligned memory access problem and enhance **DEPrune** -B.
      - **1st step** : We pre-prune **DEPrune**-B to DW-conv.
      - **2nd step** : We measure two essential factors (alpha and epsilon) within the **DEPrune**-B model.
      - **3rd step :** The beta values of all layers are ranked by comparing them with each other.
      - **4th step :** The layer with the beta value of the top 50% is additionally removed as much as it overflows.

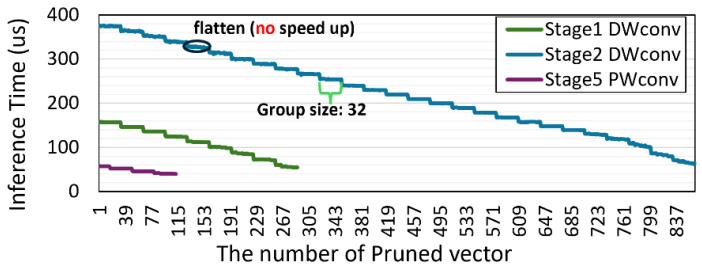

Fig.7 Measurement of DW-conv inference time of EfficientNet-B0 on ImageNet. Inference time decreases for each increase of 32 or more pruned vectors. GPU tile size is 32.

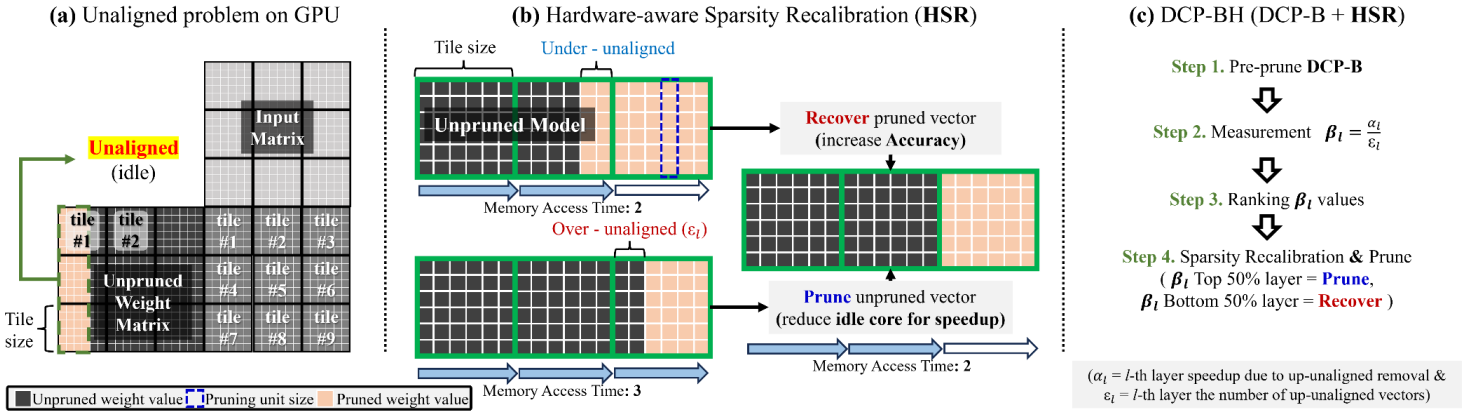

Fig.8 (a) Problem of unaligned pruning ratio. **(b)** Concept of HSR. **(c)** Process of DEPrune-BH.

Experiment

- **On MobileNet-V2**, **DEPrune** reduces approximately 26.7% more FLOPs compared to RLAL, while exhibiting a 0.2% smaller accuracy drop.
- **On EfficientNet-B0**, while other methods prune around 30% of DW-conv, our method prunes 84.7% with only a 0.8% accuracy drop.
- **On MobileNet-V3-Small** and **MobileNet-V3-Large**, **DEPrune** achieves inference times 3.3 times and 1.92 times faster than GFS and FPGM, respectively, with accuracy drops of 1.1% and 0.6% less, respectively.

Method	Pruning Ratio		Pruned FLOPs	Top-1 Accuracy			Speed Up		Time
	DW-conv	PW-conv		Baseline	Pruned	diff.	DW-conv	Total	(*us*)
MobileNet-V2	-	-	-	71.9%	-	-	1.00x	1.00x	2306
CafeNet-R [41]	37.1%	37.1%		73.7%	68.2%	-5.5%	1.44x	1.46x	1581
AMC [17]	-	-	30.0%	71.8%	70.8%	-1.0%	-	-	-
CC [27]	-	-	28.3%	71.9%	70.9%	-1.0%	-	-	-
MetaPruning [30]	-	-	30.7%	72.0%	71.2%	-0.8%	-	-	-
Random-Pruning [26]	-	-	29.1%	71.9%	70.9%	-1.0%	-	-	-
ATO [48]	-	-	30.1%	71.9%	72.0%	+0.1%	-	-	-
RLAL [11]	-	-	29.4%	71.8%	71.3%	-0.5%	-	-	-
GFS [50]	42.8%	42.8%	-	72.0%	68.8%	-3.2%	1.58x	1.60x	1448
GFS [50]	37.1%	37.1%	-	72.0%	69.7%	-2.3%	1.44x	1.46x	1581
CafeNet-R [41]	22.8%	22.8%	-	73.7%	71.9%	-1.8%	1.22x	1.23x	1871
CafeNet-E [41]	14.2%	14.2%	-	73.7%	72.4%	-1.3%	1.15x	1.16x	1992
AMC [17]	17.1%	17.1%	-	72.0%	70.8%	-1.2%	1.17x	1.20x	1971
GFS [50]	22.8%	22.8%	-	72.0%	71.2%	-0.8%	1.22x	1.23x	1871
CafeNet-R [41]	14.2%	14.2%	-	73.7%	73.3%	-0.4%	1.15x	1.16x	1992
DEPrune-BH [ours]	**77.9%**	**52.7%**	56.1%	71.9%	71.6%	**-0.3%**	**3.52x**	**2.48x**	930
DEPrune-BH [ours]	**75.1%**	**64.8%**	66.2%	71.9%	71.0%	**-0.9%**	**3.11x**	**2.70x**	853
EfficientNet-B0	-	-	-	77.6%	-	-	1.00x	1.00x	6650
CafeNet-R [41]	30.2%	30.2%	-	76.4%	74.5%	-1.9%	1.41x	1.37x	4848
CafeNet-E [41]	26.4%	26.4%	-	76.4%	74.6%	-1.8%	1.34x	1.30x	5085
DEPrune-BH [ours]	**84.7%**	**62.0%**	-	77.6%	76.8%	**-0.8%**	**6.15x**	**3.74x**	1775
MobileNet-V3-Small	-	-	-	67.7%	-	-	1.00x	1.00x	1857
GFS [50]	20.0%	20.0%	-	67.5%	65.8%	-1.7%	1.24x	1.23x	1499
DEPrune-BH [ours]	**82.1%**	**70.0%**	-	67.7%	67.1%	**-0.6%**	**5.29x**	**4.12x**	450
MobileNet-V3-Large	-	-	-	74.0%	-	-	1.00x	1.00x	4892
FPGM [18]	33.0%	33.0%	-	74.0%	73.1%	-0.9%	1.48x	1.47x	3945
DEPrune-BH [ours]	**77.0%**	**43.0%**	-	74.0%	73.7%	**-0.3%**	**4.13x**	**2.83x**	1187

Table.1 Comparison of inference time (us) with **DEPrune-BH** and recent structured pruning on ImageNet. diff. means the top-1 accuracy difference rate compared to baseline.

NEURAL INFORMATION PROCESSING SYSTEMS

Thank you

Contact

cheonjun.park@yonsei.ac.kr