JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

Compacting Binary Neural Networks by Sparse Kernel Selection

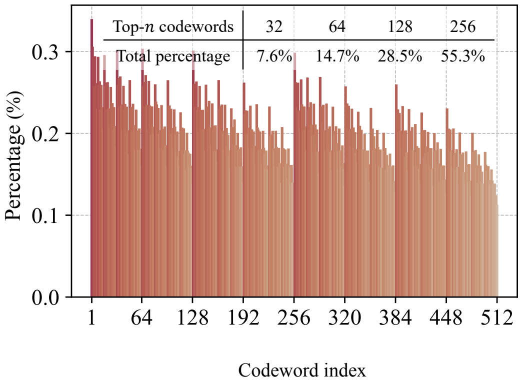
Yikai Wang[1], Wenbing Huang[2], Yinpeng Dong[1,3], Fuchun Sun[1], Anbang Yao[4]

[1]BNRist Center, State Key Lab on Intelligent Technology and Systems,
Department of Computer Science and Technology, Tsinghua University
[2]Gaoling School of Artificial Intelligence, Renmin University of China [3]RealAI [4]Intel Labs China

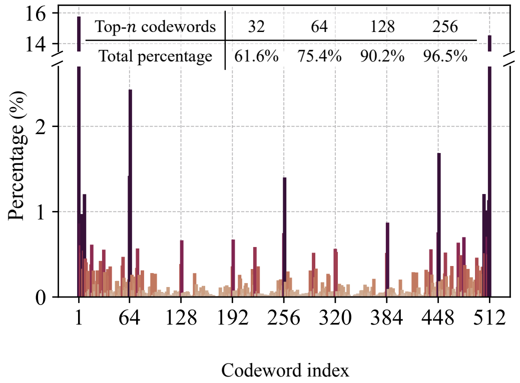
Primary Contact: Yikai Wang (yikaiw@outlook.com)

Compacting Binary Neural Networks by Sparse Kernel Selection

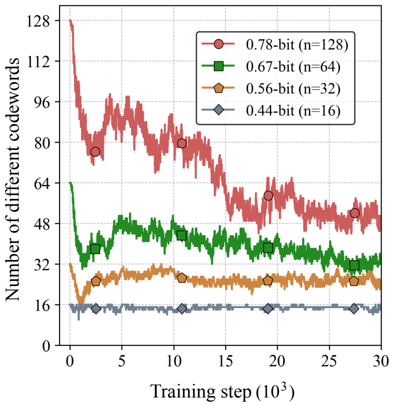
- We introduce how to compact and accelerate BNN further by Sparse Kernel Selection, abbreviated as **Sparks**.
- Our work is build based on a previously revealed phenomenon (by SNN[1]) that the 3×3 binary kernels in successful BNNs are nearly power-law distributed, **their values being mostly clustered into a small portion of codewords**. See the difference between Figure (a) and (b).
- In SNN, we observe that the sub-codebook is easy to degenerate during training (see Figure (c)), since codewords tend to be repetitive when being updated independently.
- While in our Sparks (Figure (d)), the diversity of codewords preserves by **selection-based learning**.



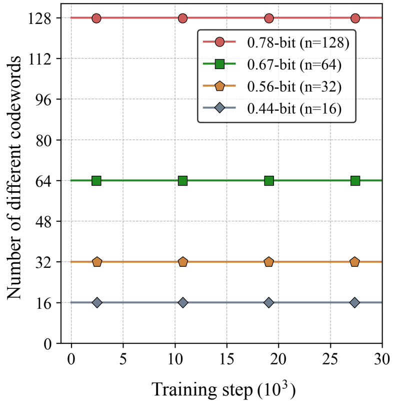
(a) Codebook constructed with sub-vectors (multi-channel codewords)



(b) Codebook constructed with kernels (single-channel codewords) **(ours)**



(c) Product quantization-based optimization for binary codewords



(d) Selection-based optimization for binary codewords **(ours)**

[1] Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks. ICCV 2021.

Compacting Binary Neural Networks by Sparse Kernel Selection

($K = 3$ for 3×3 binary kernels)

Property 1 *We denote* $\mathbb{B} = \{-1, +1\}^{K\times K}$ *as the codebook of binary kernels. For each* $\boldsymbol{w} \in \mathbb{R}^{K\times K}$, *the binary kernel* $\hat{\boldsymbol{w}}$ *can be derived by a grouping process:*

$$\hat{\boldsymbol{w}} = \mathrm{sign}(\boldsymbol{w}) = \arg\min_{\boldsymbol{u}\in\mathbb{B}} \|\boldsymbol{u} - \boldsymbol{w}\|_2. \tag{1}$$

We compact BNNs by recasting the grouping as $\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{u}\in\mathbb{U}} \|\boldsymbol{u} - \boldsymbol{w}\|_2,\ s.t.\ \mathbb{U} \subseteq \mathbb{B}.$

Matrix representation, where $\boldsymbol{P}$ is a permutation matrix and $\boldsymbol{V}$ is fixed as a certain initial selection,

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{u}\in\mathbb{U}} \|\boldsymbol{u} - \boldsymbol{w}\|_2,\ s.t.\ \boldsymbol{U} = \boldsymbol{B}\boldsymbol{P}\boldsymbol{V}, \boldsymbol{P} \in \mathbb{P}_N,$$

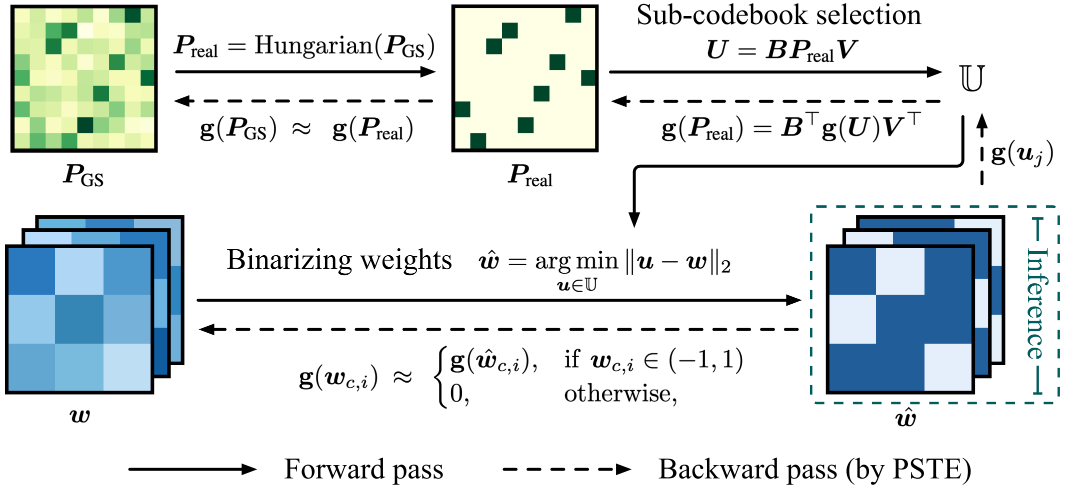
We learn the permutation matrix $\boldsymbol{P}$ by Gumbel-Sinkhorn, denoted as $\boldsymbol{P}_{\mathrm{GS}}$.

Forward pass

$$\boldsymbol{P}_{\mathrm{real}} = \mathrm{Hungarian}(\boldsymbol{P}_{\mathrm{GS}}),$$
$$\boldsymbol{U} = \boldsymbol{B}\boldsymbol{P}_{\mathrm{real}}\boldsymbol{V},$$
$$\hat{\boldsymbol{w}}_c = \arg\min_{\boldsymbol{u}\in\mathbb{U}} \|\boldsymbol{u} - \boldsymbol{w}_c\|_2,$$

Backward pass

$$\mathbf{g}(\boldsymbol{w}_{c,i}) \approx \begin{cases} \mathbf{g}(\hat{\boldsymbol{w}}_{c,i}), & \text{if } \boldsymbol{w}_{c,i} \in (-1,1), \\ 0, & \text{otherwise}, \end{cases}$$
$$\mathbf{g}(\boldsymbol{u}_j) = \sum_{c=1}^{C_{\mathrm{in}}\times C_{\mathrm{out}}} \mathbf{g}(\hat{\boldsymbol{w}}_c)\cdot \mathbb{I}_{\boldsymbol{u}_j = \arg\min_{\boldsymbol{u}\in\mathbb{U}}\|\boldsymbol{u}-\boldsymbol{w}_c\|_2},$$
$$\mathbf{g}(\boldsymbol{P}_{\mathrm{real}}) = \boldsymbol{B}^\top \mathbf{g}(\boldsymbol{U})\boldsymbol{V}^\top,$$
$$\mathbf{g}(\boldsymbol{P}_{\mathrm{GS}}) \approx \mathbf{g}(\boldsymbol{P}_{\mathrm{real}}),$$ (our **PSTE**, will be introduced)



JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Compacting Binary Neural Networks by Sparse Kernel Selection

How Gumbel-Sinkhorn in our setting works?

Given a matrix $\boldsymbol{X} \in \mathbb{R}^{N\times N}(N = |\mathbb{B}|)$, the Sinkhorn operator over $\mathcal{S}(\boldsymbol{X})$ is proceeded as follow,

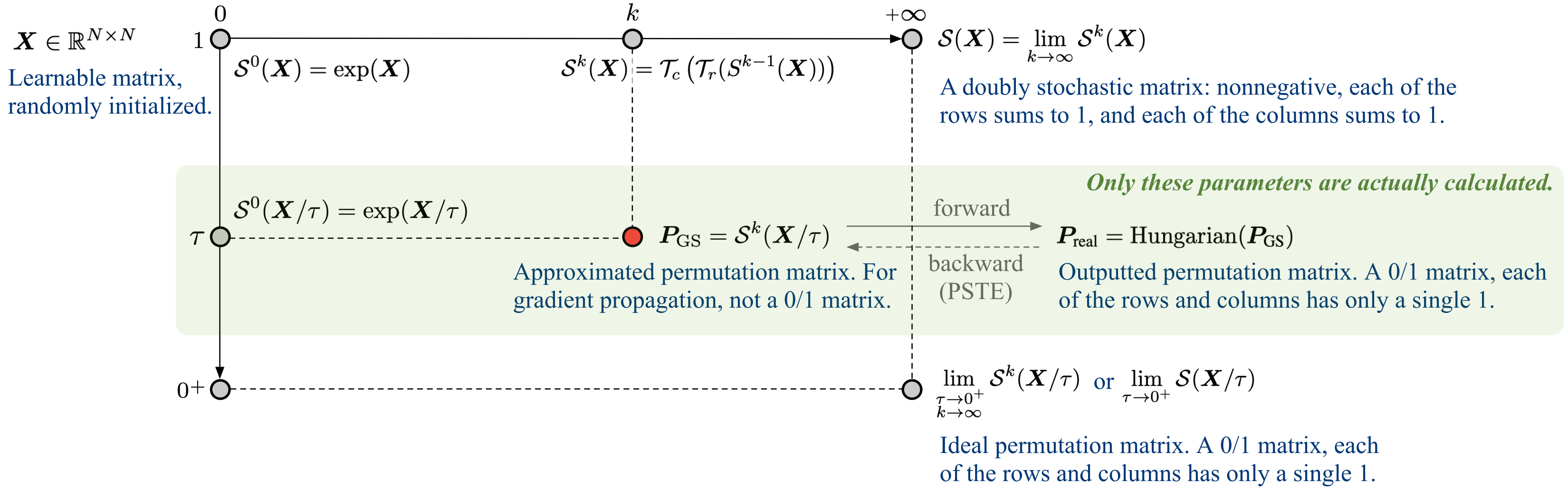
$$\mathcal{S}^0(\boldsymbol{X}) = \exp(\boldsymbol{X}), \tag{5}$$

$$\mathcal{S}^k(\boldsymbol{X}) = \mathcal{T}_c\left(\mathcal{T}_r(\mathcal{S}^{k-1}(\boldsymbol{X}))\right), \tag{6}$$

$$\mathcal{S}(\boldsymbol{X}) = \lim_{k\to\infty} \mathcal{S}^k(\boldsymbol{X}), \tag{7}$$

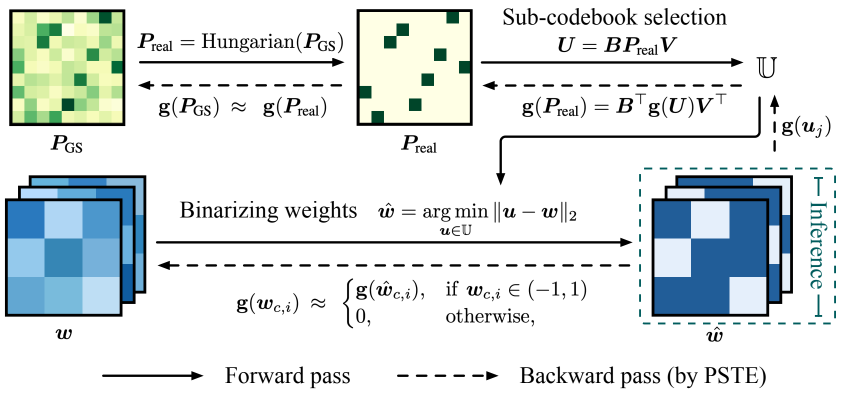
where $\mathcal{T}_r(\boldsymbol{X}) = \boldsymbol{X} \oslash (\boldsymbol{X}\mathbf{1}_N\mathbf{1}_N^\top)$ and $\mathcal{T}_c(\boldsymbol{X}) = \boldsymbol{X} \oslash (\mathbf{1}_N\mathbf{1}_N^\top\boldsymbol{X})$ are the row-wise and column-wise normalization operators, and $\oslash$ denotes the element-wise division. For stability purpose, both normalization operators are calculated in the log domain in practice. The work by [41] proved that $\mathcal{S}(\boldsymbol{X})$ belongs to the Birkhoff polytope—the set of doubly stochastic matrices.

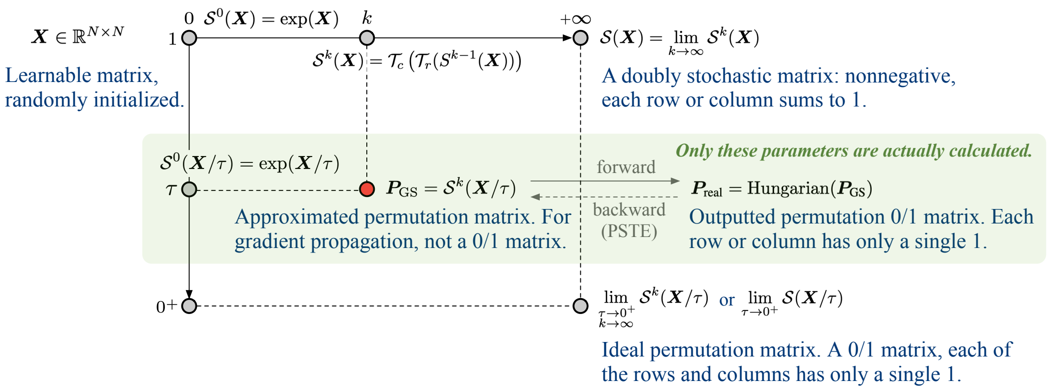
By substituting the Gumbel-Sinkhorn matrix, we characterize the sub-codebook selection as $\boldsymbol{U} = \boldsymbol{B}\mathcal{S}^k((\boldsymbol{X}+\epsilon)/\tau)\boldsymbol{V}$,



JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Compacting Binary Neural Networks by Sparse Kernel Selection





PSTE: Approximate the gradient of the Gumbel-Sinkhorn matrix $\boldsymbol{P}_{\mathrm{GS}}$ with $\boldsymbol{P}_{\mathrm{real}}$. We have the following theorem to guarantee the convergence for sufficiently large k and small τ.

Lemma 1 *For sufficiently large k and small τ, we define the entropy of a doubly-stochastic matrix $\boldsymbol{P}$ as $h(\boldsymbol{P}) = -\sum_{i,j} P_{i,j} \log P_{i,j}$, and denote the rate of convergence for the Sinkhorn operator as r $(0 < r < 1)$[3]. There exists a convergence series s_τ ($s_\tau \to 0$ when $\tau \to 0^+$) that satisfies*

$$\|\boldsymbol{P}_{\mathrm{real}} - \boldsymbol{P}_{\mathrm{GS}}\|_2^2 = \mathcal{O}\big(s_\tau^2 + r^{2k}\big). \tag{18}$$

Theorem 1 *Assume that the training objective f w.r.t. $\boldsymbol{P}_{\mathrm{GS}}$ is L-smooth, and the stochastic gradient of $\boldsymbol{P}_{\mathrm{real}}$ is bounded by $\mathbb{E}\|\mathbf{g}(\boldsymbol{P}_{\mathrm{real}})\|_2^2 \leq \sigma^2$. Denote the rate of convergence for the Sinkhorn operator as r $(0 < r < 1)$ and the stationary point as $\boldsymbol{P}_{\mathrm{GS}}^\star$. Let the learning rate of PSTE be $\eta = \frac{c}{\sqrt{T}}$ with $c = \sqrt{\frac{f(\boldsymbol{P}_{\mathrm{GS}}^0) - f(\boldsymbol{P}_{\mathrm{GS}}^\star)}{L\sigma^2}}$. For a uniformly chosen $\boldsymbol{u}$ from the iterates $\{\boldsymbol{P}_{\mathrm{real}}^0, \cdots, \boldsymbol{P}_{\mathrm{real}}^T\}$, concretely $\boldsymbol{u} = \boldsymbol{P}_{\mathrm{real}}^t$ with the probability $p_t = \frac{1}{T+1}$, it holds in expectation over the stochasticity and the selection of $\boldsymbol{u}$:*

$$\mathbb{E}\|\nabla f(\boldsymbol{u})\|_2^2 = \mathcal{O}\left(\sigma\sqrt{\frac{f(\boldsymbol{P}_{\mathrm{GS}}^0) - f(\boldsymbol{P}_{\mathrm{GS}}^\star)}{T/L}} + L^2\big(s_\tau^2 + r^{2k}\big)\right). \tag{19}$$

Compacting Binary Neural Networks by Sparse Kernel Selection

- Comparisons of top-1 and top-5 accuracies with state-of-the-art methods on ImageNet based on ResNet-18.

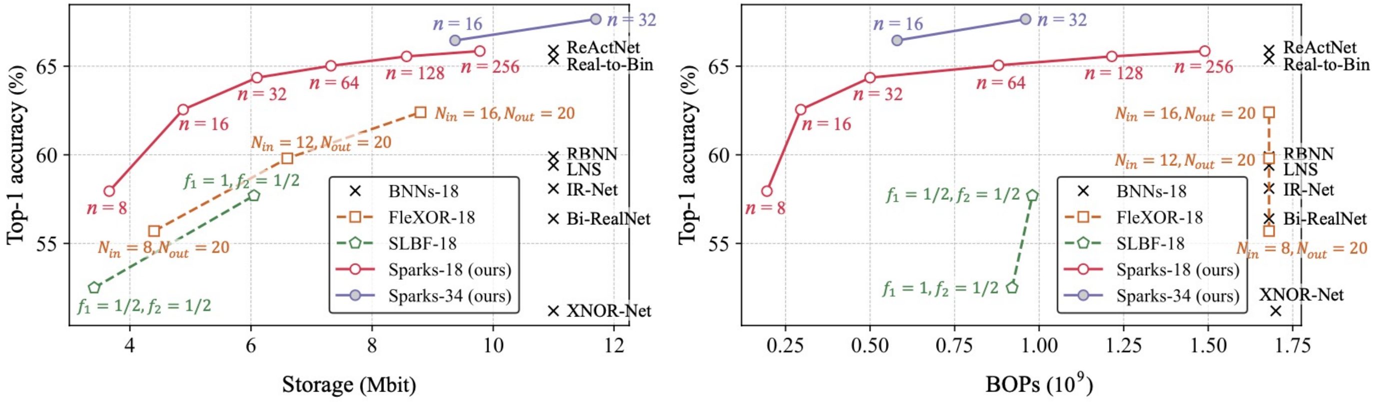
Method	Bit-width (W/A)	Accuracy (%) Top-1	Accuracy (%) Top-5	Storage (Mbit)	BOPs ($\times 10^9$)
Full-precision	32/32	69.6	89.2	351.5	107.2 (1×)
BNN [16]	1/1	42.2	69.2	11.0 (32×)	1.70 (63×)
XNOR-Net [37]	1/1	51.2	73.2	11.0 (32×)	1.70 (63×)
Bi-RealNet [31]	1/1	56.4	79.5	11.0 (32×)	1.68 (64×)
IR-Net [36]	1/1	58.1	80.0	11.0 (32×)	1.68 (64×)
LNS [10]	1/1	59.4	81.7	11.0 (32×)	1.68 (64×)
RBNN [26]	1/1	59.9	81.9	11.0 (32×)	1.68 (64×)
Ensemble-BNN [52]	(1/1)×6	61.0	-	65.9 (5.3×)	10.6 (10×)
ABC-Net [28]	$(1/1)\times 5^2$	65.0	85.9	274.5 (1.3×)	42.5 (2.5×)
Real-to-Bin [33]	1/1	65.4	86.2	11.0 (32×)	1.68 (64×)
ReActNet [32]	1/1	65.9	86.4	11.0 (32×)	1.68 (64×)
SLBF [24]	0.55/1	57.7	80.2	6.05 (58×)	0.92 (117×)
SLBF [24]	0.31/1	52.5	76.1	3.41 (103×)	0.98 (110×)
FleXOR [25]	0.80/1	62.4	83.0	8.80 (40×)	1.68 (64×)
FleXOR [25]	0.60/1	59.8	81.9	6.60 (53×)	1.68 (64×)
Sparks (ours)	0.78/1	65.5	86.2	8.57 **(41×)**	1.22 **(88×)**
Sparks (ours)	0.67/1	65.0	86.0	7.32 **(48×)**	0.88 **(122×)**
Sparks (ours)	0.56/1	64.3	85.6	6.10 **(58×)**	0.50 **(214×)**

- Results when extending our Sparks to wider or deeper models.

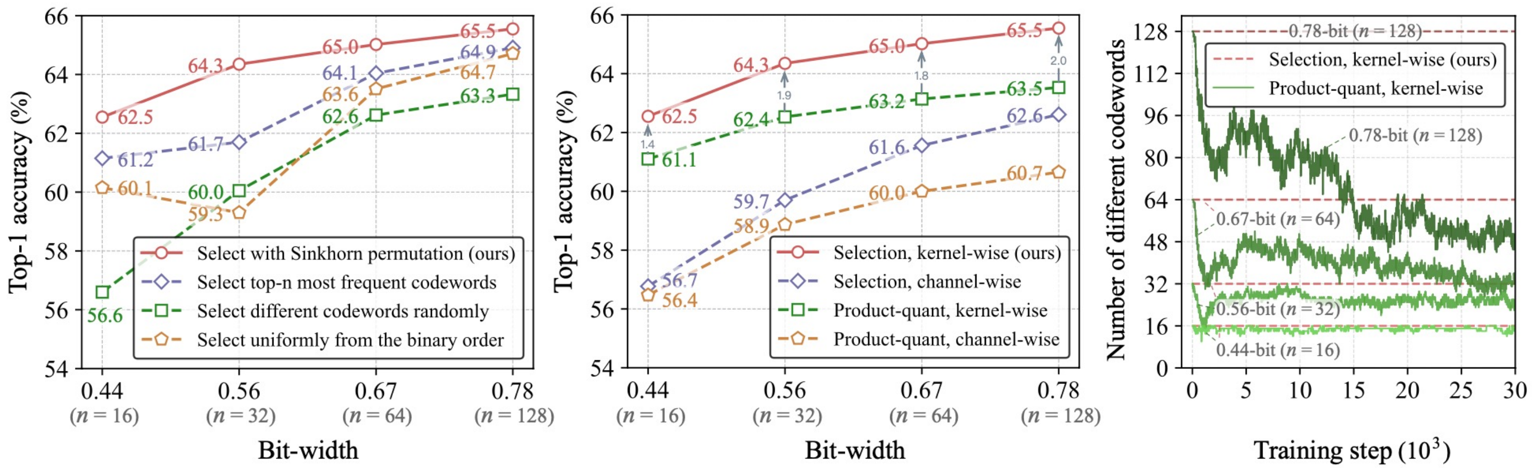
Method	Backbone	Bit-width (W/A)	Accuracy (%) Top-1	Accuracy (%) Top-5	Storage (Mbit)	BOPs ($\times 10^9$)
ReActNet [32]	ResNet-18	1/1	65.9	86.4	11.0	1.68
Sparks-wide	ResNet-18 (+ABC-Net [28])	(0.56/1)×3	**66.7**	**86.9**	18.3	**1.50**
Sparks-deep	ResNet-34	0.56/1	**67.6**	**87.5**	11.7	**0.96**
Sparks-deep	ResNet-34	0.44/1	**66.4**	**86.7**	**9.4**	**0.58**

Compacting Binary Neural Networks by Sparse Kernel Selection

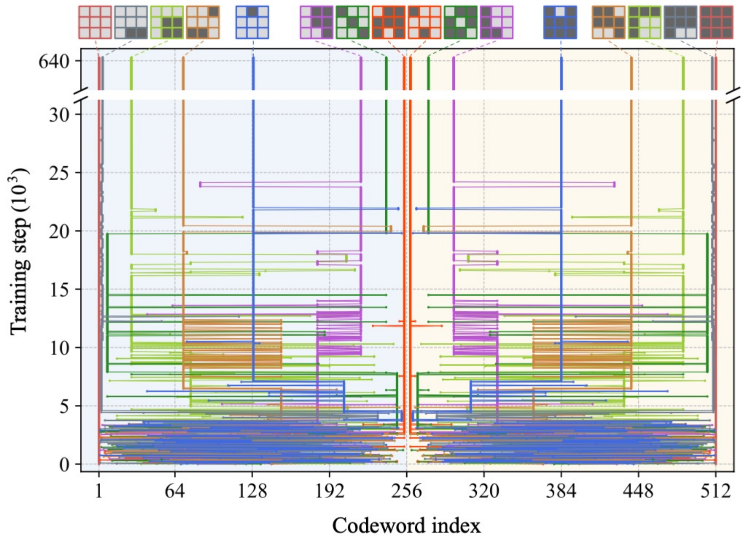
- Trade-off between performance and complexity on ImageNet,



- Ablation studies on ImageNet with ResNet-18,



- Codewords selection during training,



Thanks